

**A Web Application to Improve Emotional Awareness
in High-Functioning Autistics**

by

Temitope O. Sonuyi

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

[June 2006]
May 26, 2006

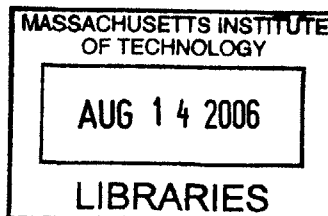
Copyright 2006 Temitope O. Sonuyi. All rights reserved.

The author hereby grants to M.I.T. permission to reproduce and
distribute publicly paper and electronic copies of this thesis
and to grant others the right to do so.

Author _____
Department of Electrical Engineering and Computer Science
May 17, 2006

Certified
by _____
Henry Lieberman
Thesis Supervisor

Accepted
by _____
Arthur C. Smith
Professor, Graduate Officer, Undergraduate Officer



BARKER

A Web Application to Improve Emotional Awareness
in High-Functioning Autistics

by
Temitope O. Sonuyi

Submitted to the
Department of Electrical Engineering and Computer Science

May 26, 2006

In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

The web application built here is based on the idea of presenting scenarios to users, using text, and having the users choose likely emotions that match the scenarios. Taken for granted by most neurotypical people, high-functioning autistics are often lacking in this area of social-skill development. This idea of emotion to scenario matching is accomplished using a series of different games that take different approaches to exercise these skills. The application relies on the two main Artificial Intelligence (AI) approaches. The first AI approach is classical, relying on computer-based algorithms developed by others to judge text and put out the correct affect or emotion. The other part of the AI relies on users of the system contributing via regular usage or explicit correction to train the system in a type of feedback loop.

Thesis Supervisor: Henry Lieberman
Title: Research Scientist, MIT Media Laboratory

Thanks, Acknowledgements and Dedications

*Adupe, Ese gaani Baba mi. O titan.
Mom, Dad, Tolu, Modupe, Gbemi, Tosin for being there for me, always.*

*Henry Lieberman, Hugo Liu, Paulina Modlitba, Edward Shen, Alea Teeters, Alki
Delichatsios*

Simon Baron-Cohen and his group, Rana El Kaliouby, Ofer Golan

Dedicated to Push Singh

Table of Contents

INTRODUCTION: PROBLEM AND IMPETUS FOR SOLUTION	5
MINDBLINDNESS, WHAT IS IT?	5
<i>An academic psychological view on mindblindness.....</i>	<i>5</i>
<i>A personal view on mindblindness.....</i>	<i>6</i>
<i>Consequences of mindblindness.....</i>	<i>7</i>
<i>Dealing with mindblindness.....</i>	<i>8</i>
A MOTIVATION FOR WORK.....	9
HISTORY AND BACKGROUND OF RELATED WORK	9
THE EMERGENCE OF SOFTWARE TO ASSIST THE MINDBLIND	9
THE APPLICATION: DETAILS AND DESIGN, HOW IT WORKS.....	11
OVERVIEW: THE IDEA BEHIND WHAT IT DOES AND WHY IT IS DONE THIS WAY	11
ARTIFICIAL INTELLIGENCE: THE HEART OF THE APPLICATION	13
<i>Getting a PAD reading from an event.....</i>	<i>14</i>
<i>Going from a PAD to an Ekman 6 emotion.....</i>	<i>14</i>
<i>Gathering, Judging and Storing events in the Database.....</i>	<i>16</i>
THE BASE ARCHITECTURE.....	18
THE BASE UI.....	21
GAME 1.....	25
<i>Overview</i>	<i>25</i>
<i>Design Details.....</i>	<i>29</i>
GAME 2.....	38
<i>Overview</i>	<i>38</i>
<i>Design Details.....</i>	<i>41</i>
GAME 3.....	47
<i>Overview</i>	<i>47</i>
<i>Design Details.....</i>	<i>50</i>
FINDINGS.....	57
ANALYZING USAGE STATISTICS	57
FEEDBACK FORM RESULTS.....	58
CONCLUSIONS AND FUTURE WORK.....	59
CONCLUSIONS	59
FUTURE WORK	61
FINAL WORD.....	62
BIBLIOGRAPHY.....	63
APPENDIX OF CODE.....	64

Introduction: Problem and Impetus for Solution

Mindblindness, what is it?

Simply put, mindblindness is the inability for someone to perceive or deduce emotions.

Mindblindness can more specifically be described as an inability to understand what another person might be feeling or thinking emotionally. Far from an “either-or” diagnosis, mindblindness has varying degrees of severity and is a core trait of those affected by autism.

Coined by Professor Simon Baron-Cohen, the term “mindblind” has become the definitive way to categorize this psychological phenomenon. Mindblindness is generally found amongst high-functioning autistics and people with Asperger's syndrome. Both conditions belong under the Autism Spectrum Disorder, or ASD, category. The phenomenon of mindblindness is not only well documented in academic literature like Baron-Cohen's book, but also has been well described from the viewpoint of those afflicted. The autobiographical descriptions range from harmless confusion to terrifying anxiety when describing mindblindness and living with it. A relatively new discovery compared to other psychological phenomena, there is not a wealth of research or information on how to combat the issue. Recent methods of treating it rely on fairly low-tech and "manual" techniques. This thesis looks at the very recent steps that have been taken to make computer software a useful tool in this realm of dealing with mindblindness, and to push the current boundaries within it.

An academic psychological view on mindblindness

Baron-Cohen uses the following example when describing mindblindness¹:

¹ The Sally and Anne example was designed by Simon Baron-Cohen and Uta Frith

"Sally has a marble. She puts her marble into the box, and then she goes outside. Anne comes in, takes the marble out of the box, and puts it in her basket. When Sally comes back, where will she look for the marble?"

By about the age of 4, most children who are presented with this scenario know the answer, this includes children with Down's syndrome and other developmental problems. But some children do not know the answer. They do not understand that what they know and what Sally knows are two separate things. They do not understand that Sally has her own thoughts about where the marble is. These children expect Sally go to the basket, because they know the marble is there. These children cannot understand how she doesn't know where the marble is. These same children are the ones likely to be diagnosed with some form of autism or Asperger's Syndrome. (Frith 158-161)

As shown in the example, the "Sally Anne Test", autistic children are slower in understanding the point of view of others. In his book, Baron-Cohen shows how autistic people have a hard time realizing that other minds exist, separate from themselves. This is as compared to the ability of those considered as "normal", or neurotypical, in the majority of society. It is also noted that mindblindness is very autism-specific, as even mentally disabled children are able to recognize the existence of minds outside of their own. A lot of these findings are based on the development of eye contact and attention patterns during the developmental process.

A personal view on mindblindness

"Now I know that it's simply that I have to **think** in order to **feel**. Emotions remain undifferentiated and un-understood until I can process them cognitively." –Hubert Cross, Adult High-Functioning Autistic (Cross)

These words of a high-functioning autistic man, offer a glimpse into a world where people like him often walk around confused and blind to the emotional mind-states that "normal" people take

for granted. Hubert describes his life of mindblindness as something he was never cognizant about as a child. He likens his life experience to the example of a physically blind person who was born blind but never told that everyone else around him could see, leaving him to fend for himself and have a skewed view of how "reality" was for everyone else. Having eventually learned about mindblindness and some valuable techniques of dealing with it, Hubert was amazed at just how widespread the affliction was and how little was being done to combat it when the solution was simple as making mindblind people aware. Spurred by a grippingly odd upbringing due to his mindblindness, and sometimes embarrassing eventfulness Hubert's biography about his experiences from an autistic point of view, give great insight to neurotypical people into what life is like when you are mindblind.

Consequences of mindblindness

Whether one is coming from the academic or personal view of autism, both views lead to the conclusion that untreated mindblindness can be critically debilitating. A common consequence of mindblindness in children is a lack of healthy socially interaction in settings like school. Often times mindblind people may unwittingly annoy others around them with behaviors that are considered unacceptable by society. There have been stories of mindblind children relieving themselves of waste in areas that are entirely off limits for that type of activity, leading to extreme embarrassment and often group alienation. Often times incognizant mindblind people will do things like stare at someone for a prolonged period of time without realizing the uncomfortableness that someone might feel. This also leads to alienation or cold-behavior by neurotypical peers. Consequences of mindblindness have even reached the realm of unintentionally harming other people or property because of that lack of understanding.

Dealing with mindblindness

While the effects change with the varying environments and degrees of mindblindness, solutions to helping this issue are relatively straightforward in that they all revolve around recognition. By simply recognizing that they are mindblind, those affected can begin to develop skills and techniques to overcome the impediments mindblindness causes. It should also be noted that it is best if treatment of mindblindness is started at a young age so that it is a part of the critical mental developmental cycle.

One common method of fighting mindblindness, especially in children, is the reading of stories (Edelson). These stories that are read to or read by the child with mindblindness are special in the fact that they are explicitly emotion based. The stories usually describe common scenarios that someone is likely to experience and explicitly states and reflects on the emotions of the characters in the story. So in essence multiple mental states of multiple different people are constantly highlighted. The hope is that the mindblind reader becomes accustomed to the idea that other minds exist and have emotional state. It also helps the child to understand what events or situations lead to different emotional states.

While helpful coping methods are used when mindblindness is diagnosed, many mindblind people who never learn about their condition simply have to learn “the hard way” about their surroundings. Some have been able to figure out how to not only survive, but often take advantage of their mindblindness, while others have lead miserable lives until their "eyes" were opened (Cross). These different experiences are due to the varying degrees of mindblindness and the different environments that those affected with it grow up in. In the worst case mindblind people may never learn how to cope with their condition and go through their entire lives confused and unaware forever.

Very recently, software has begun to be explored as a means of teaching mindblind people how to see. While there are not many at this point, the general idea of these applications is similar to that of the story reading idea, but more interactive. Because the learning is not restricted to a book, software tools can generate many different scenarios and give users a chance to interact through more descriptive story telling (audio, video), or through playing games in a user interface. There is a lot of room for growth in this area of mindblind assistance and that is where this thesis project aims to take root.

A motivation for work

The idea that people see the world so differently from the neurotypical majority is a great motivation for interest in the topic; but it is the negative consequences that arise due to untreated cases and a general lack of understanding by neurotypical people, that drive the motivations for this thesis. This thesis further explores and builds a software-based computational approach to try and contribute to helping people with mindblindness.

History and background of related work

The emergence of software to assist the mindblind

Software applications are just beginning to be turned to as an option for helping the mindblind. Because of the inherent versatility of software, it provides a host of different ways to teach people about emotions. One of the most important traits of these new ways of teaching is interactivity. Compared to a story book, a software application can provide a user interface that can be used to play games and display video/audio/text scenarios. If the saying "a picture is worth a thousand words" has any merit, then the medium of video alone offers so much more description and a more concrete real world connection for users to understand. It is because of these benefits that software is an interesting avenue of aide that continues to be expanded and explored.

The status quo in the existing realm of software applications has two main pillars. Firstly, most of the scenarios/events and emotions in current software are static and human-compiled. The term “static” means that the emotions that are chosen to be associated with certain events are “hard coded” or permanent. These associations are usually chosen manually (by human decision). The answers are what they are and will stay that way forever. The term “human-compiled” refers to the way in which the software must be made, relying on human judgment of different events and emotions and coming up with an answer. Although this allows for more specificity and higher rates of correctness, this idea of static human-compiled answers makes this process of software creation long, hard and costly. The second pillar of the status quo is rooted in the desktop installation. The term “desktop installation” refers to the required installation from a CD-ROM or an executable file to gain access to use the software. The desktop install idea has a few issues. One problem is accessibility, where people with different operating systems from the intended software are out of luck. Another limit of this style of software deployment is coverage. Because people need to order, wait and then install the software, the spread of its effectiveness is somewhat slowed. All that being said this approach has proved useful and affective in terms of quality and results. The point of this thesis project is not to analyze or critique the status quo of this type of software but rather propose a different, and hopefully better, style of software applications.

While there are a few software applications for mindblindness in existence, one of the more popular and most extensive examples today is Mind Reading: the interactive guide to emotions, which was created in part by Professor Simon Baron-Cohen (Baron-Cohen). This piece of software uses video, audio, image and text expressions of emotions in its repertoire. With over 412 different emotions organized into 24 groups, this software is one of the most expansive pieces out there today. Basically the software goes through many different scenarios, using its different medium, and tries to explain the association of them with emotions. It should be noted

that this thesis project actually borrows on the data in this software (with appropriate permission) to seed the corpus of scenarios/events being used.

The general consensus from study and testing has been that applications like Mind Reading are very useful and helpful in accomplishing the goal of helping the mindblind understand external emotions better. This sets the stage and justification for further use of software in this realm and is encouragement for this thesis project.

The Application: Details and Design, How it Works

Overview: the idea behind what it does and why it is done this way

This thesis project aims to usher in a new status quo based on web-based software applications for mindblindness aide. In its simplest form, the application is a series of different games that are playable through a user interface (UI). These games require the users to analyze and understand how different events are associated with different emotions in order to score points. The main pillars of this project rely on Artificial Intelligence (AI), a web-based UI, and community augmented error-correction.

The project relies heavily on the idea of using AI to deduce many of its answers. In this case that means associating which emotions go with which events/scenarios. This idea of using AI comes with some tradeoffs. By using a modular AI algorithm that can take in text describing an event and return an emotion, we make the software creation process much faster and cheaper since generation of answers can be dynamic. At the same time we may lose correctness/quality in our answers or associations because at this point in time the level of AI is not the same as that of human deduction. Furthermore the use of AI limits the granularity of emotions that are able to be deduced. For example, in the mind-reading program done by Professor Simon Baron-Cohen there

were 412 different emotions used, but in our own system we work with a set of Ekman's 6 basic emotions due to AI interpretation limits.

While working with a set of 6 basic emotions may seem very limiting at first glance, there are two ideas that are exploited to rebut this. First, the AI in the system allows for multiple mappings of the Ekman 6 emotions to each event. This way more complex emotions can be implicitly defined by combinations of the simpler emotions used (i.e. disgust might equate to surprise + anger). While this is not an absolute solution, it helps to alleviate the issue of using such simple emotions to describe a wide array of events that may need more complex emotions to be associated with them. Furthermore the original intent of this project was that it would be geared towards younger ASD users who are in the earlier stages of developing their emotion-event association skills. This set of users will have less of a need for explicitly denoted complex emotions tied to events. By the time these users do move towards a need for more complex defined events, the use of this system in their earlier event-emotion association development will hopefully have imparted/equipped them with skills that will carry over into their self-assessment processes.

To address these issues of higher error rates and a smaller set of emotions due to AI techniques, the system harnesses another part of the software. Because this is a web-based UI there are new advantages that can be harnessed to help prevent or correct errors made by the AI. There are multiple affordances for "reporting" errors through the User Interface which is available to all users. If a user notices an emotion that has been incorrectly assigned to an event, or an event that is missing one of many emotions that should belong to it, they can input that data which is then fed back to the system. This data is noted and used to correct the system when those events and/or emotions resurface in the future. This means that as the system gets used more, the fidelity of the

answers should get better and better. This is something made possible entirely by the web-based platform paradigm.

Besides allowing augmentation of the AI in the system, the web based application brings a host of other benefits not seen in the status quo of software in this realm. Deployment, software-upgrading, cross-platform accessibility and compatibility become problems with very simple solutions now that the application is web-based. With no need to install software on every user's system we open up the channels of usage to a larger audience more quickly. Furthermore the UI of the system allows users to input customized events/scenarios that they wish to know the associated emotion for. These user-generated events can be reused by the system to further build the corpus of events/scenarios that already exist.

Along with these main characteristics of the project, there are also some smaller details which enhance the user experience. Since the project is game-based we track the high scores of users and allow them to log-in and keep track of their best scores to date. This allows them to compete against themselves and gauge whether or not they are improving as they use the system. This data can also be used as a metric for how well the system is doing in accomplishing its goal.

Furthermore the games in this project are designed to be fun and interactive in pushing the learning experience.

Artificial Intelligence: the heart of the application

The AI of this system involves associating events with emotions and storing these relationships for retrieval and amendment.

Getting a PAD reading from an event

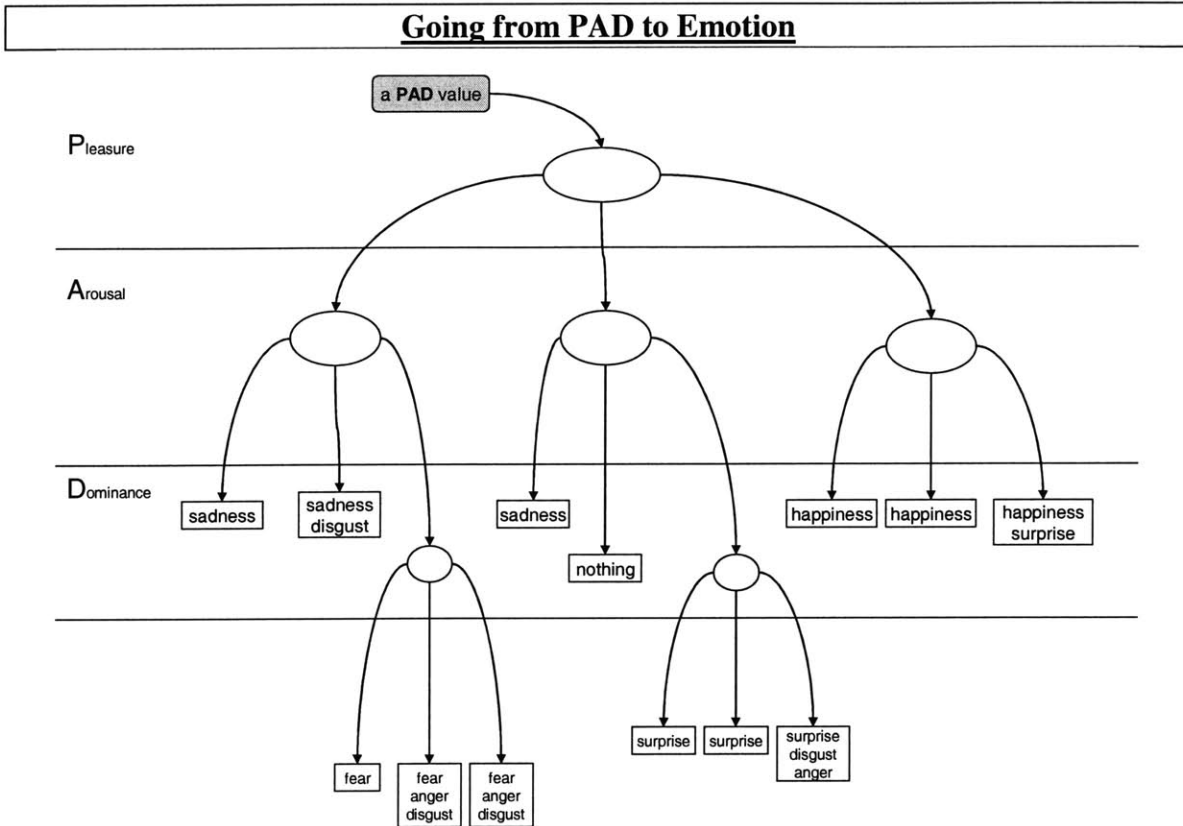
To go from a sentence to an emotion this system relies on MIT Media Lab based artificial intelligence. The specific module used is called Emotus Ponens. The module is based on the OpenMind Common Sense project started at the Media Lab. The module works by taking a string of text as input, in this case a sentence describing a scenario or event, and it outputs a list of values called a PAD rating (Liu 125-132). PAD stands for Pleasure-Arousal-Dominance and is used to describe affect in terms of numerical values (Mehrabian 121, 339-361). The PAD values output by the Emotus Ponens module range from -1 to 1 for each part of the PAD (pleasure, arousal, dominance). In this case a score of 1 would mean a high indication of that particular part of the PAD and a -1 would reflect the opposite of a particular part of the PAD. So if the following values were returned by Emotus Ponens, [1,0,0], it would mean that the text that was originally put in was highly associated with pleasure. If the 1 value was changed to -1 then the PAD rating would indicate a high amount of displeasure. One can see how a combination of these values could reduce down to various different emotional states, which is what Emotus Ponens aims to accomplish.

A key part of the AI involved in the Emotus Ponens module is "spreading activation-energy". This involves associating emotions not only with keywords in a sentence, but also with words in proximity to keywords (Lieberman 7). This idea helps to account for the issue of context, which can highly change the meanings associated with different words.

Going from a PAD to an Ekman 6 emotion

While Emotus Ponens helps in reducing a sentence to a PAD value, there is still the very crucial step of reducing this PAD value to actual emotions that can be represented in words. For example

a PAD value of [1,0,0] may be reduced to the emotional word "happy". In any case there is the need for an algorithm that can take in a PAD value and put out emotion-denoting words.



The algorithm used to go from a PAD value to an emotion word uses a tree-traversing approach. This algorithm works by taking a PAD value list as input at the single top node. The algorithm was designed so that each level of the tree is concerned with only one part of the PAD value input. Going from top to bottom, the first level of the tree is concerned with the Pleasure part of the PAD, the second level of the tree is concerned with the Arousal part of the PAD and the third level of the tree is concerned with the Dominance part of the PAD. Each level of the tree has at least one node in it. At each node on each level of the tree a decision must be made to decide which path to follow. This decision is based on part of the PAD value pertinent at that level and two threshold values set for the particular node where the decision is being made. The threshold values act as separators for each of the three possible paths to be traversed if the part of the PAD

value being examined is less than the left threshold value, then the left path is traversed, if it is greater than the right threshold value then the right path is traversed, otherwise the middle path is traversed. At the end of each edge coming from a node there is either a set of emotion words or another node that can be traversed. At the leaves of the tree, there only exist emotion words. This necessarily ensures that the PAD of every sentence gets associated with some type of textual description. The threshold values used to make decisions at each node were observationally decided but can easily be changed at any time. The same also applies for the choice and placement of emotion words in the tree. The details of current threshold values and emotion words used at the time of writing of this thesis are available in the appendix where the source code for the algorithm is given².

Gathering, Judging and Storing events in the Database

Even with a way to convert sentences to emotions there is still the need to gather a corpus of sentences appropriate for this project. After gaining a corpus of sentences, they must be ran through the algorithm and associated with emotions. Then these relationships must be stored for later retrieval and possible amendment.

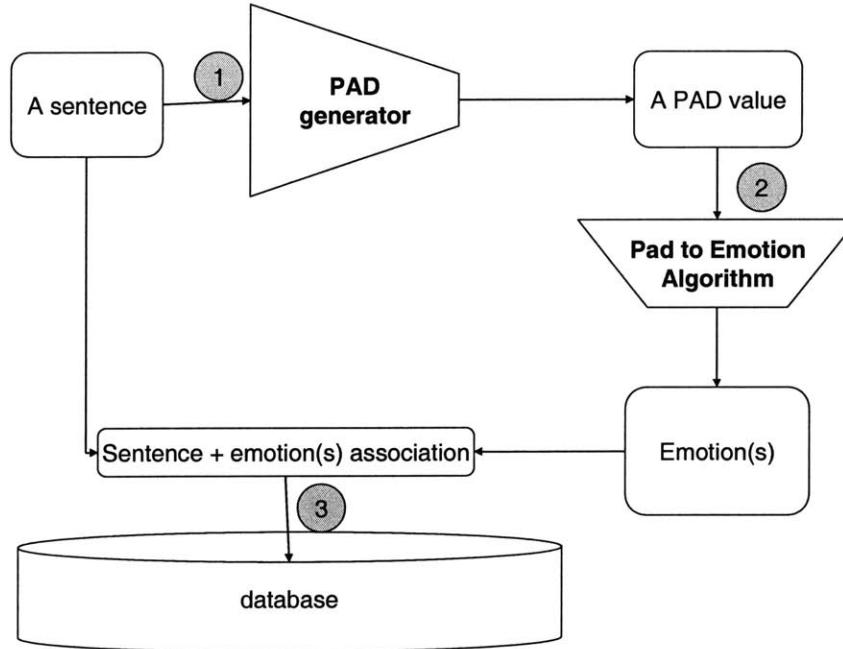
The corpus of events for this project was taken, with permission, from Professor Baron-Cohen's Mind Reading software³. These sentences were created to describe events with emotional significance so they are a very good candidate to seed our database of events. This same corpus of sentences was used to condition Emotus Ponens to key-in on particular keywords that show up with high frequency in the corpus (Liu).

² See appendix code for **PadToEmotionAlgo.py** and **EmotionDecisionNode.py**

³ See reference to Baron-Cohen DVD material in the bibliography section

Each sentence in the corpus was run through the Emotus Ponens AI module and the PAD-to-emotion algorithm in order to be associated with emotions. A database was kept to store this information. Each sentence was uniquely identified and has the ability to have one or more emotions associated with it. This ability to read and modify multiple emotions per event is important because it relates back to the idea of using community intelligence to augment the AI. After the algorithm judges a sentence we may only have one emotion associated with that sentence. Over time if users report that other emotions should also be associated with that sentence, the data can be amended to reflect that and return this amended information the subsequent times when the association between the sentence and its emotion(s) is requested. The storage design also appropriates the ability for each sentence to be denoted as "bad", due to grammar or offensiveness. This denotation is also created from community use in the UI portion of the system. Lastly we also allowed the ability for users to create their own sentences in the UI of the system. These user-created sentences can be stored in the database and add to the corpus of sentences available. The system tracks user-generated sentence addition to the corpus by flagging them with a creation date.

Associating a Sentence with an Emotion



Events and Emotions in Database

sentence	emotion	flag_type	human_corrected_emotions	human_supplied
John got a gift.	happy	Missed_emotion	Surprise-3;disgust-1	1/1/2006
Sarah read a book.	Nothing	Null	Null	Null

Sentence: The unique sentence that defines this row.

Emotion: The computer-generated emotion(s) associated with the sentence.

Flag_type: The type of correction(s), if any, made for this sentence by the community of users.

Human_corrected_emotions: A count of how many times any emotion has been suggested for this sentence.

Human_supplied: The date of creation for a sentence if the sentence was user-supplied.

The Base Architecture

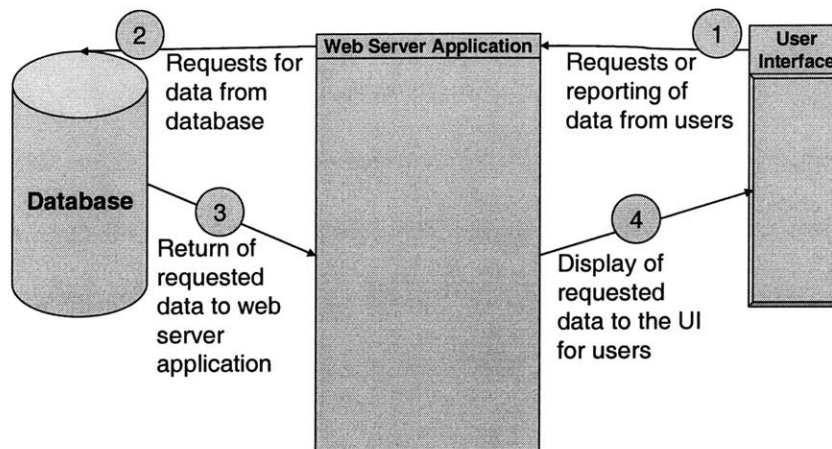
The architecture of this project is built around the idea of presenting a user interface(UI) for interactive game play. While the process-flow and component details may slightly differ for each

type of game presented in the UI, there is a lot of commonality on a basic level. The components of this basic architecture exist and interact as described here.

First we must define the general parts of the architecture.

- One important part of the system is the database. The database holds all information necessary to run the system which includes sentence-emotion relationships, user IDs, and high scores.
- Another component of the system is the web application or server. The server provides the pages necessary for the UI portion of the game and also serves as the communication channel between the database and the UI. All information between the user and the database is passed along and sometimes modified by the web server. To do its job of relaying information, the web server contains other subcomponents, or modules, which can be used as needed, depending on the game and the scenario.
- The UI component of the system is abstract at the architectural level and described in more detail in the next section. Nonetheless it is important because it is the interface provided to the user that allows interaction with the system. The UI allows the user to play each game and send reports or other data back to the system. This is also where data is displayed to the user that is not only pertinent to the game they are playing but also to their personal account.

Components of Architecture and Process Flow



The following general actions occur in relation to the components of the system:

Logging in - User's can login at the UI. This sends a request to the web server to ask the database for the high score information for a particular user, along with any other information pertaining to them

Getting sentence-emotion relationships - When a game begins or ends there is the need to get information from the database about the relationship between a set of sentences and emotions. This is initiated at the UI and sent to the db via the web server. The web server then sends back the relevant information to the UI and it is displayed to the user according to the rules of the game being played.

Displaying sentence-emotion in the UI - each game has its own rules for when and how the sentence-emotion relationships are shown, but in all cases when the game is finished they are all revealed in a section of the UI. This display of the correct emotion-sentence relationships sets the stage for community feedback to the system.

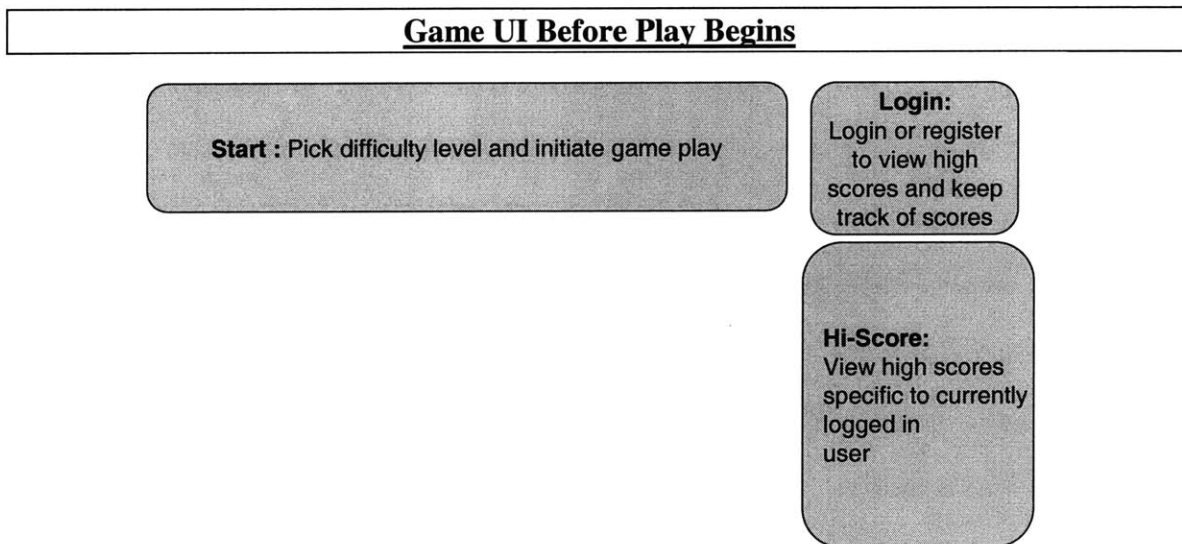
Reporting scores - As expected, at the end of a game scoring occurs. When a score has been calculated for a user the score is sent to the web server if a new high score has been reached for that particular game. This score is stored back in the database via the web server. The database

also increments how many times a new high score has been achieved so that performance can be gauged.

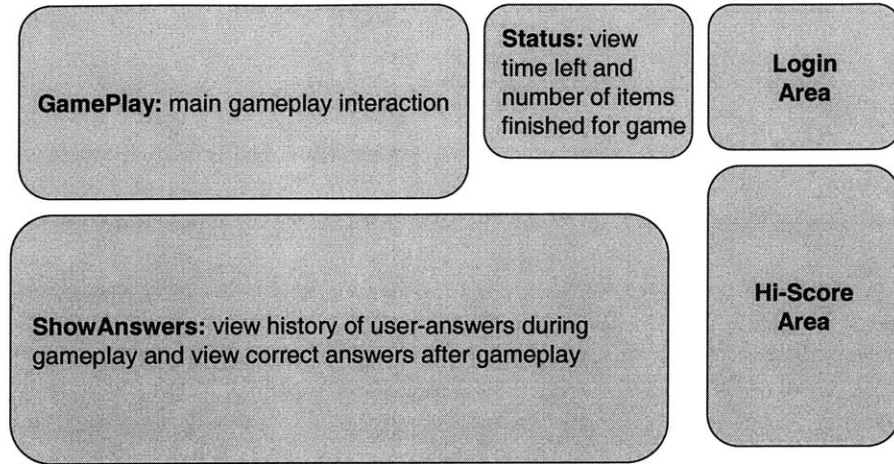
Reporting errors - At the end of the game there are UI affordances for reporting errors in sentence quality and sentence-emotion relationships. When all the AI-module-based relationships are displayed in the UI, users may report that the sentence is faulty, or choose a correct emotion that is they feel should be associated with the sentence.

The Base UI

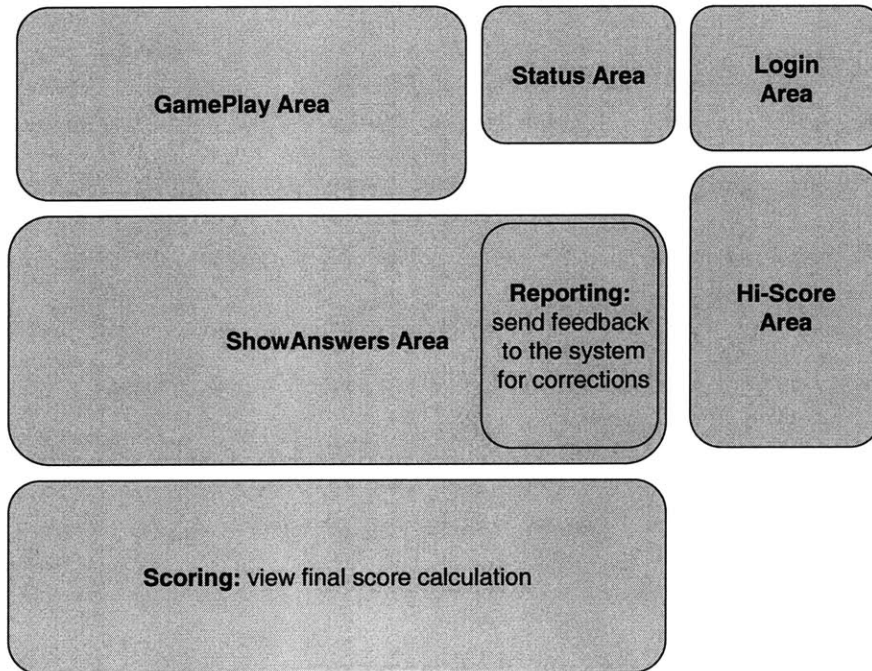
While the UI is a component of the system architecture, it is also a very complex entity unto itself. The UI for each game is slightly different per the requirements of the game, but they all have similar basic areas. These areas and their function are defined as follows on the general level.



Game UI During Play



Game UI After Play



All areas of the UI allow for input or display of useful data for the user.

The first area is called the Start Area. The Start Area is where a user first chooses what level of difficulty they would like to play the game at. This usually ranges from easy to hard. A user also initiates the actual game play from this area as well. This area is not visible after game play has started.

Another area of the UI is the Login area. This area is pretty straightforward and easy to use. This is where a user may register for the first time on the system so that they can begin to track their progress. Once a user has registered one time, they can use this area to login to the system.

The Hi-Score area is adjacent and highly related to the Login area. Once a user has logged in, the Score area display's the user's current high score for each of the games. Whenever a new high score is reached by a user, this area automatically updates to reflect it.

The GamePlay area of the UI is where the main interaction takes place between the user and the system. Once game play has been initiated from the Start area, this area is where a user actually input's their choices and answers for the game to score. In every game currently in the project a user may not only choose an answer in this area, but also "backup" and re-answer as long as time permits, which will be explained in the next area, the Status area.

The Status area is the part of the UI that the user can check to see how many tasks he or she has left to complete in a game and how much time they have left to do it. The status area always display's how many tasks are left and how many have been done. The area also displays a

countdown timer that is activated as soon as game play is initiated. By constantly checking the Status area while working in the GamePlay area a user can manage their tasks efficiently.

The ShowAnswers area is displayed during game play when a user is completing tasks and after the game has finished (time has run out or all tasks have been completed). When this area is displayed during game play it shows the history of answers a user has selected. This allows a user to review their answers and make corrections by backing up before game play has ended. Once game play is ended this area displays not only the answers chosen by the user, but also the answers determined by the system as correct. The answers are then compared for scoring in the Scoring area.

The Reporting area is actually a sub area of the ShowAnswers area. In this section of the UI a user may choose to report one of two things to the system. In the case of some games a user may choose to report that the sentence being used in a certain answer was not sensical or grammatically correct by marking it a bad sentence. In all games a user may report that the emotion associated with a certain sentence is incomplete by suggesting a different emotion that should be associated with the sentence. In both cases the report is sent back to the database via the web server and processed accordingly.

The Scoring area of the UI is displayed at the end of game play and simply shows the calculation and final score for a user's performance in a game. In this area a user may review their score and decide to play again or quit the game.

Game 1

Overview

In Game 1 users are presented with a series of scenarios or events. Users must choose which emotions they think best describe the scenarios. This must be accomplished in a limited amount of time and with a variable amount of scenarios, both based on the chosen difficulty level of the game. The scenarios are displayed to the user one at a time, so users must choose an emotion to match the sentence being presented before moving on to the next sentence. The user should analyze each scenario presented before him or her so that they grasp what emotion is most likely to be associated with it. The hope is that this analyzation will cause a user to logically deduce why an emotion should be associated with a certain scenario, thus becoming more skilled in the parallel ability to analyze real world scenarios in a similar manner.

Noting the direction of this cognitive process, the game is designed to specifically stress the skill of going from a scenario or event to an emotion. While this is only one way of making event-emotion associations it is very important and designed to mimic the most basic and frequently encountered real life situation of having to make an event-emotion association.

This game is deemed moderately difficult because the number of emotion choices for each sentence, six, is neither extremely low nor high, requiring moderate analysis and time to make each decision.

UI Before Play Begins

ASD Games

Home | Game1 | Game2 | Game3

Game 1

[more instructions](#)

Choose A Difficulty for the game:

- Easy [10 events, 55sec]
- Medium [20 events, 80sec]
- Hard [30 events, 115sec]

Don't Use a Timer

start

Event Sentence:

Emotion Choices:

Time Remaining
:XX

Done X To Go X

Answers:

Sentence	Your Answer	Our Answer	Report This
----------	-------------	------------	-------------

**If you see a mistake *REPORT IT* by clicking on the buttons at the end of the rows. [Learn More](#)

Score:

Log In

username:

password:

login

[new user's click here!](#)

Your High Scores

Game1

Log In

Game2

Log In

Game3

Log In

UI During Play

ASD Games

Home Game1 Game2 Game3

Game 1
[more instructions](#)

Event Sentence:

Julie is congratulatory when a colleague announces he and his wife are expecting a baby

Emotion Choices:

surprise

happiness

sadness

disgust

fear

anger

nothing

Time Remaining
:17

Done
6

To Go
4

backup

Log In

username:

password:

[new user's click here!](#)

Your High Scores

Game1

Game2

Game3

Answers:

Sentence	Your Answer	Our Answer	Report This
Peter feels tearful as he says goodbye to his girlfriend.	surprise		
Sandra is happy sitting in her garden looking at the flowers.	happiness		
Kyle believes the woman's story about her experiences.	sadness		
Rachel feels soppy when she buys a cute teddy bear for her baby cousin and then gets one for herself.	disgust		
Drew feels pleasure when his girlfriend massages his neck.	sadness		
When Peter tells his girlfriend that he is a Superhero, he is joking.	happiness		

****If you see a mistake "REPORT IT" by clicking on the buttons at the end of the rows. [Learn More](#)**

Score:

UI After Play

ASD Games

Home Game1 Game2 Game3

Game 1
more instructions

Event Sentence:

Emotion Choices:

Time Remaining

:0

Done
6

To Go
4

Log In

username:

password:

new user's click here!

Your High Scores

Game1

Game2

Game3

Answers:

Sentence	Your Answer	Our Answer	Report This
Peter feels tearful as he says goodbye to his girlfriend.	surprise	surprise disgust anger	<input type="button" value="report it!"/>
Sandra is happy sitting in her garden looking at the flowers.	happiness	nothing happiness	<input type="button" value="report it!"/>
Kyle believes the woman's story about her experiences.	sadness	nothing	<input type="button" value="report it!"/>
Rachel feels sorry when she buys a cute teddy bear for her baby cousin and then gets one for herself.	disgust	surprise	<input type="button" value="report it!"/>
Drew feels pleasure when his girlfriend massages his neck.	sadness	surprise disgust anger	<input type="button" value="report it!"/>
When Peter tells his girlfriend that he is a Superhero, he is joking.	happiness	surprise disgust anger	<input type="button" value="report it!"/>

****If you see a mistake "REPORT IT" by clicking on the buttons at the end of the rows. [Learn More](#)**

Score:

#Correct: (2)

Game-type factor: (0.5)

Time bonus: (0)

(#Correct * Game-type factor) + TimeBonus = Total

You received a score of **[1]** points!

Design Details

Following is an expansive scenario describing the usage of this game.

1. Log in

Like all other games in the system, a user can begin by logging in or registering.

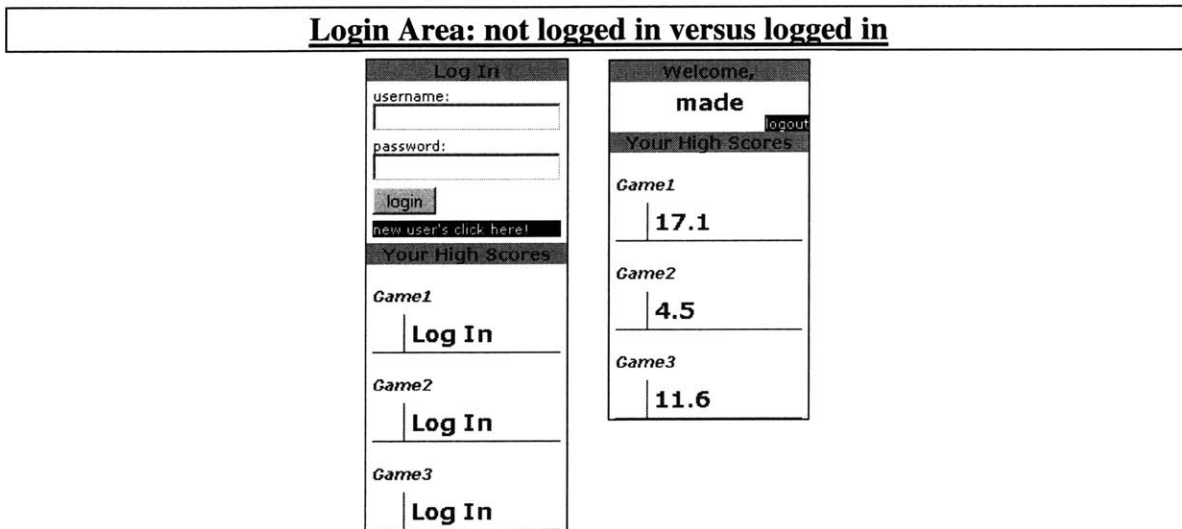
In the case of registering, a user need only provide a username, email address and password to be verified. This information is checked by client side code for errors in formatting or invalid character usage in all fields. Next this information is sent to the web server where it is further checked before being sent to the database for commitment. If any errors occur a message is sent back to the client and the applicable error is displayed in the UI. Errors can range from invalid formatted emails, to pre-existing usernames or emails in the database. If the check for verifying the sent information is ok, the username is sent back and displayed in the UI.

For logging in, only the username and password need to be supplied. When the user clicks the login button this information is sent to the web server and checked against the database. If the username and password are associated in the database, then the high score data for that user is sent back to the UI.

Once registration or login is verified by the web server a cookie is placed on the user's machine that contains the high score data for the user. This cookie also contains the username for a user. This is done so that upon returning to the site before the cookie expires, re-logging in will not be required. At this time the high score for each game is read from the placed cookie and displayed in the HiScore area. If a user does not login before using the system they are not prohibited from

play, but the high score area is filled with a message saying they are not logged in. This passively alerts the user to the option of logging in or registering without hindering usability.⁴

This part of the game scenario was designed to be extremely easy to accomplish. Thus it is done using asynchronous server calls. At the time of this writing security was not a major issue while time was, so issues about the security of asynchronous server calls were overlooked.

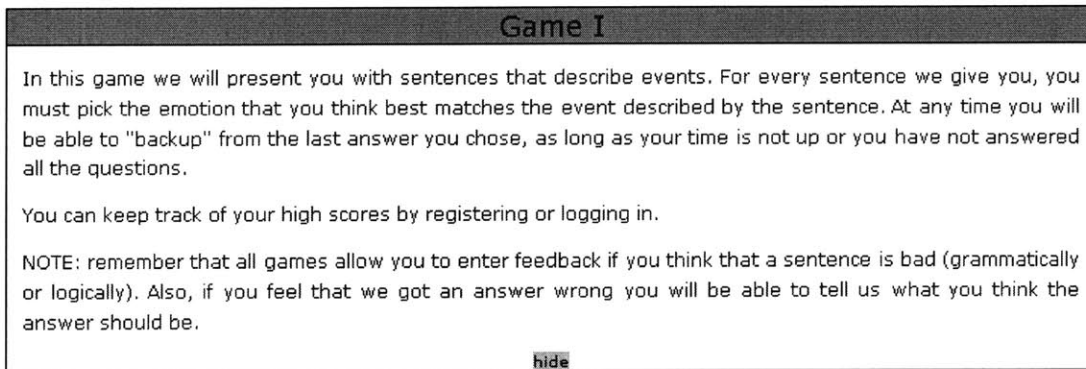


2. Read more instructions about the game, or not

A user has the ability to read more instructions in case they were linked directly to the game section of the application, or if they just want more information. This section can be expanded or collapsed to give the user more real estate during game play. At the time of this writing the instruction section reminds the user about the ability to backup when answering questions, the objective of the game and how to report errors found in the game.

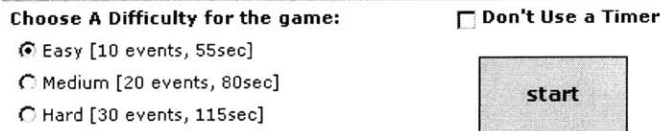
⁴ See appendix code for **Master.js:150-276**, **ASDGameServer.py:17-41**, **WebAppToDBMethods.py:24-92**

Hidden versus Unhidden Instructions



3. Pick a difficulty level

Difficulty Level and Game Initiation UI



At this point in the scenario the user needs to choose what level of game play they wish to engage in. This game has low, medium and high difficulty levels displayed to the user for selection. Each level is distinguished by the number of sentences and the amount of time that will be allotted for game play.

Note: While the actual number of sentences and time allotted differs from game to game, the description is the same, in that the ratio of the number of sentences to the amount of allotted time increases slightly, while the actual numbers themselves increase substantially as you go from easy to hard. The values used are arbitrary and can easily be changed.

4. Start game⁵

Now a user is ready to start game play, and may click the start button to do so. At this time the number of sentences to use, as determined by the difficulty level selected, is requested from the database via the web server. The data retrieved is a corresponding number of sentence-emotion pairs and is sent back to the web server and ultimately the client side code of the browser. Once this information is back to the client side, one of the sentences is presented to the user in the GamePlay area, along with a set of emotions that the user must choose from. The remaining events, not currently being shown are kept on client side variables along with their emotion associations. At this time the timer is also set to the allotted time associated with this level of difficulty and a countdown is started. Similarly, the number of sentences to match for this game and the number of sentences already matched, zero at this point, are presented in the Status area

5. GamePlay area usage⁶:

During game play the user now must choose what emotion they choose to associate with the sentence that is currently being displayed. Every time the user chooses an emotion, the client-side code stores that user-chosen event-emotion pair and also displays that association in the table of the ShowAnswers area. This data is placed in only the first two 2 of the 4 columns of the table, while the other columns remain blank. The first column of the table will hold the sentence judged, while the second column will hold the emotion that the user chose. Concurrently a new event is displayed in the "Event Sentence" box of the GamePlay area. At this time the Status area is also updated to show how many associations have been completed and how many are left to go, along with the current amount of time left to finish the game. Each time the user makes an association this process will repeat, unless there is no more time left or there are no more sentences left to judge. At any time before one of these two events occurs, signaling the end of the game, a user

⁵See appendix code for **Game1.js:125-342,ASDGameServer.py:48-69, WebAppToDBMethods.py:125-188**

⁶ See appendix code for **Game1.js:342-420**

may choose to backup and change any association that was made earlier, using the backup button in the GamePlay area. Whenever the backup button is invoked the Status area updates accordingly to reflect the new amount of completed and uncompleted associations.

GamePlay Area During Play

ShowAnswers Area During Play

Answers:

Sentence	Your Answer	Our Answer	Report This
Peter feels tearful as he says goodbye to his girlfriend.	surprise		
Sandra is happy sitting in her garden looking at the flowers.	happiness		
Kyle believes the woman's story about her experiences.	sadness		
Rachel feels sappy when she buys a cute teddy bear for her baby cousin and then gets one for herself.	disgust		
Drew feels pleasure when his girlfriend massages his neck.	sadness		
When Peter tells his girlfriend that he is a Superhero, he is joking.	happiness		

6. See answers displayed at the end of the game. View which answers were correct and incorrect.⁷

⁷ See appendix code for **Game1.js:453-533**

Once time has expired or all the available associations have been made, the Gameplay area is disabled for usage. At this point all data in that area and the Status area are frozen. At this time all the system-generated, or "correct", event-emotion associations are displayed in the third column of the table in the ShowAnswers area. The third column will hold the emotion(s) that the system feels are associated with the sentence in the first column of each row. If a user answer differs from a system answer, the system answer is highlighted in red, while a coinciding user answer will cause the system answer to be highlighted in green. This offers an easy indication of which answers the user got correct and incorrect. The user may look over their answers versus the system answers to reiterate or correct their analysis of the association.

ShowAnswers Area After Play			
Answers:			
Sentence	Your Answer	Our Answer	Report This
Peter feels tearful as he says goodbye to his girlfriend.	surprise	surprise;disgust;anger	report it!
Sandra is happy sitting in her garden looking at the flowers.	happiness	nothing;happiness	report it!
Kyle believes the woman's story about her experiences.	sadness	nothing	report it!
Rachel feels sappy when she buys a cute teddy bear for her baby cousin and then gets one for herself.	disgust	surprise	report it!
Drew feels pleasure when his girlfriend massages his neck.	sadness	surprise;disgust;anger	report it!
When Peter tells his girlfriend that he is a Superhero, he is joking.	happiness	surprise;disgust;anger	report it!

7. Report errors if applicable⁸

When the full answers are shown at the end of a game, a user may also review the ShowAnswers area to determine and report if there was a mistake made on the part of the system. In the last column of the table containing the event-emotion associations, there are buttons that allow the user to report a bad sentence or a bad answer. If a bad sentence is reported, information detailing which sentence is bad is sent to the database via the web server and recorded. If the user reports a bad answer, meaning that the emotion(s) picked by the system is/are incomplete, they can choose

⁸ See code appendix for **Game1.js:534-624**, **ASDGameServer.py:88-94**, **WebAppToDBMethods:266-399**

what they feel the correct emotion should be. This suggested emotion, along with the pertinent sentence, is then sent to the database via the web server.

Note that the sent information about errors remains persistent in the database for future use.

While the specifics of how the information is used can be tweaked, at the time of this writing a sentence that is reported as bad is wholly stricken from use. Similarly when an emotion is chosen to be augmented for a certain sentence it is associated and tallied such that each user-reported emotion for that emotion-event pairing is remembered. Whenever an event-emotion association is requested from the system, the tallies for user-suggested emotions are all counted and compared to a threshold value, currently set to two. If the tally for any emotion suggested for that sentence is over the threshold then that emotion is selected as being associated with the event (in addition to any other emotions that are currently associated with the event). Incentive and reminder to report errors is given by text and a link at the bottom of the table in the ShowAnswers section.

Learn More About Error Reporting		
ness	surprise;disgust;anger	report it!
iness	surprise;disgust;anger	report it!

end of the rows. **Learn More**

After finishing a game, you can click on the "report it!" button at the end of each row in the answers table. When you report something such as a bad sentence, or an incorrect emotion answer, we use this data to make the system aware and actually remember these reports. This improves future use of the system by making it more accurate.

Error Reporting Sequence

Answers:

Sentence	Your Answer	Our Answer	Report This
Peter feels tearful as he says goodbye to his girlfriend.	surprise	surprise;disgust;anger	report it!
Sandra is happy sitting in her garden looking at the flowers.	happiness	nothing;happiness	report it! bad answer? bad sentence?
Kyle believes the woman's story about her experiences.	sadness	nothing	report it!

Answers:

Sentence	Your Answer	Our Answer	Report This
Peter feels tearful as he says goodbye to his girlfriend.	surprise	surprise;disgust;anger	report it!
Sandra is happy sitting in her garden looking at the flowers.	happiness	nothing;happiness	report it! bad answer? If you think that our answer is wrong or missing an emotion, let us know which emotion you think should go with this sentence. <input type="text" value="choose an emotion"/> bad sentence <input type="text" value="choose an emotion"/>
Kyle believes the woman's story about her experiences.	sadness	nothing	report it
Rachel feels soppy when she buys a cute teddy bear for her baby cousin and then gets one for herself.	disgust	surprise	report it

Answers:

Sentence	Your Answer	Our Answer	Report This
Peter feels tearful as he says goodbye to his girlfriend.	surprise	surprise;disgust;anger	report it!
Sandra is happy sitting in her garden looking at the flowers.	happiness	nothing;happiness	Reported!
Kyle believes the woman's story about her experiences.	sadness	nothing	report it!

8. Check score and see calculation

Calculating and showing the final score is similar in all games. The only difference from game to game is the values used to calculate the score.

At the end of game play a user can see what their final score was and how it was calculated.

Things that factor into the score at the time of writing this thesis are the number of correct answers, the amount of time left over, the TimeWeight and the GameWeight. The number correct is simply the number of associations that the user got correct as assumed by the system. The time left over is simply the amount of time left over. The TimeWeight variable is different for each

game and is chosen based on the overall difficulty of the type of game being played. For this game the TimeWeight is 0.05. The GameWeight also differs per type of game and for this game the GameWeight is 0.5. The total score for a game is calculated by performing the following sum of products. The number correct times the GameWeight, plus, the time left over times the TimeWeight. As illustrated, parts of the formula are shown along with the final calculated score.

Scoring Area After Play
Score: #Correct: (2) Game-type factor: (0.5) Time bonus: (0) $(\#Correct * Game\text{-}type\ factor) + TimeBonus = Total$ You received a score of [1] points!
play again
back to home

9. Play again or quit

After checking ones score a user can now choose to play again or go back to the home page.

Game 2

Overview

In Game 2, users are presented with a series of sentences, and one static emotion. Users must decide whether or not the current sentence being displayed exhibits an emotion that matches the static emotion. The static emotion can be chosen randomly or explicitly by the user before they begin, and they simply have to click yes or no to indicate a match or not. This goal must be accomplished in a limited amount of time and with a variable amount of sentences, both based on the chosen difficulty level of the game. The sentences are displayed to the user one at a time, so users must indicate a match or no match before moving on to the next sentence to judge. In his or her mind, the user should construct a solid meaning of what they think is the meaning of the static emotion being used during game play. They should then analyze each sentence presented before him or her and determine whether it matches that idea they have regarding the emotion. The hope is that this analysis will cause a high functioning autistic user to determine why an event should or shouldn't be associated with a certain emotion. This should increase their skill to approach emotion association from another angle.

Noting the direction of the cognitive process, this game is designed to specifically stress the skill of going from a known emotion and deducing if an event should be associated with it. While this is again only one way of making the event-emotion association it is useful in the case where someone has a limited, but firm set of emotional knowledge. This game was designed to increase the flexibility that a high functioning autistic user would have in solving or logically deducing event-emotion associations.

This game is deemed relatively easier than the others because there are only two possible options when answering, reducing the analysis and time required to make a decision.

UI Before Play Begins

ASD Games

Home Game1 Game2 Game3

Game II

[more instructions](#)

Choose A Difficulty for the game:

Don't Use a Timer

Easy [10 events, 60sec]

Medium [20 events, 110sec]

Hard [30 events, 165sec]

To start, pick an emotion or let us randomly pick an emotion for you:

-OR-

Choose whether or not each event matches the following emotion:

Event Sentence:

A Match?

Time Remaining

:XX

Done

X

To Go

X

Answers:

Sentence

Your Answer

Our Answer

Report This

**if you see a mistake "REPORT IT" by clicking on the buttons at the end of the rows. [Learn More](#)

Score:

Log In

username:

password:

[new user's click here!](#)

Your Hi-Scores

Game1

Game2

Game3

UI During Play

ASD Games

Home Game1 Game2 Game3

Game II

[more instructions](#)

Choose whether or not each event matches the following emotion:

sadness

Event Sentence:

Julie is feeling playful and tickles her husband

A Match?

yes

no

backup

Time Remaining
:50

Done 6 To Go 4

Log In

username:

password:

[new user's click here!](#)

Your Hi-Scores

Game1

Game2

Game3

Answers:

Sentence	Your Answer	Our Answer	Report This
When Kim completes the difficult computer game she feels jubilant.	yes		
Kyle is calculating when he mentions that his friends are going to the game on Saturday. He knows his wife will suggest they go along.	no		
Kyle feels moody on dark winter days.	no		
Sandra feels sappy about rabbits because they are such gentle creatures.	yes		
Ben feels frustrated when he has to wait 45 minutes for a bus.	no		
Peter feels attacked when the lady criticises his carpentry.	yes		

****If you see a mistake "REPORT IT" by clicking on the buttons at the end of the rows. [Learn More](#)**

Score:

UI After Play

ASD Games

Home Game1 Game2 Game3

Game II

[more instructions](#)

Welcome, **made** [logout](#)

Your HI-Scores

Game1	17.1
Game2	4.5
Game3	11.6

Choose whether or not each event matches the following emotion:

fear

Event Sentence:

A Match?

Time Remaining: 0
Done: 7 To Go: 3

Answers:

Sentence	Your Answer	Our Answer	Report This
Carol is disapproving of her husband's plan to buy a Rolls Royce. She thinks expensive cars are a waste of money.	no	no-->(surprise/disgust/anger)	report it!
Tim is enjoying himself while playing music with his friends.	no	no-->(happiness/surprise)	report it!
Ali is condemning of the war. He thinks it shouldn't happen.	yes	yes-->(fear/anger/disgust)	report it!
Kim is agonising over whether to find the man guilty or not guilty when she does jury duty.	no	yes-->(fear)	report it!
Tom is miffed when he forgets to take his library books back and has to pay a fine.	yes	no-->(nothing)	report it!
Sally feels praised by her teammates when they congratulate her for scoring the winning goal	no	no-->(nothing)	report it!
Rachel feels pestered by the office junior when he won't stop asking her for a date.	yes	no-->(nothing)	report it!

****If you see a mistake "REPORT IT" by clicking on the buttons at the end of the rows. [Learn More](#)**

Score:

#Correct: (4)
Game-type factor: (0.2)
Time bonus: (0)
(#Correct * Game-type factor) + TimeBonus = Total

You received a score of [0.8] points!

[play again](#)

[back to home](#)

Design Details

Following is an expansive scenario describing usage of this game.

1. Log in

The details for logging in, registering, and displaying high scores is the same for each game. See the description for this part of Game1 for more details.

2. *Read more instructions about game, or not*

The ability to read more instructions is the same for each game in the system. See the description for this part of Game1 for more details.

3. *Pick a difficulty level*

In this part of the scenario the only differences from game to game are the actual values denoted for each level of game play. See the description for this part of Game1 for more details.

Difficulty Level and Game Initiation UI

Choose A Difficulty for the game: **Don't Use a Timer**

Easy [10 events, 60sec]

Medium [20 events, 110sec]

Hard [30 events, 165sec]

To start, pick an emotion or let us randomly pick an emotion for you:

-OR-

4. *Pick an emotion to work with, or let the system choose randomly.*

A user now needs to decide what emotion they would like to work with for the game. The user may either pick one of the six Ekman emotions from a dropdown, or opt to have the system pick and emotion randomly. This variable is stored and remembered on the client side when the decision is made and automatically starts the game.

5. *Start game*⁹

When a user completes the preceding step of choosing an emotion to work with, the game play automatically starts. At this time the number of sentences to use, as determined by the difficulty level selected, is requested from the database via the web server. The data retrieved is a

⁹ See appendix code for **Game2.js:104-331**, **ASDGameServer.py:48-69**, **WebAppToDBMethods.py:125-188**

corresponding number of sentence-emotion pairs and is sent back to the web server and ultimately the client side code of the browser. Once this information is back to the client side, the UI is ready for play. The UI in the GamePlay area will show:

- One of the sentences retrieved. (The remaining sentences, not currently being shown are kept on client side variables along with their emotion associations.)
- Two buttons that can be clicked, yes or no.
- The static emotion that has been selected for matching during game play.

At this time the timer in the Status area is set to the allotted time associated with this level of game play and a countdown is started. Also, the number of events that need to be judged for this game and the number that have been judged, zero so far, is shown in the Status Area.

6. GamePlay area usage¹⁰

Now that game play has ensued the user must determine if the current sentence matches the emotion displayed or not. Every time the user chooses yes or no the client side code stores that answer and the sentence being judged. The answer and the sentence are also displayed in the table of the ShowAnswers area. This data is placed in only the first 2 of the 4 columns of the table, while the other columns remain blank. The first column of the table will hold the sentence judged, while the second column will hold an answer of yes or no, which the user chose. At this same time a new event is displayed in the "Event Sentence" box of the GamePlay area. Concurrently, the Status area is updated to show how many sentences have been judged and how many are left to judge, along with the current amount of time left to finish the game. Each time the user makes a yes or no decision this process will repeat, unless there is no more time left or there are no more sentences left to judge. At any time before one of these two events occur, signaling the end of the game, a user may choose to backup and change any decision that was made earlier, using the

¹⁰ See appendix code for **Game2.js:332-425**

backup button in the GamePlay area. Whenever the backup button is invoked the Status area updates accordingly to reflect the new amount of completed and uncompleted decisions.

GamePlay Area During Play

ShowAnswers Area During Play

Answers:

Sentence	Your Answer	Our Answer	Report This
When Kim completes the difficult computer game she feels jubilant.	yes		
Kyle is calculating when he mentions that his friends are going to the game on Saturday. He knows his wife will suggest they go along.	no		
Kyle feels moody on dark winter days.	no		
Sandra feels soppy about rabbits because they are such gentle creatures.	yes		
Ben feels frustrated when he has to wait 45 minutes for a bus.	no		
Peter feels attacked when the lady criticises his carpentry.	yes		

7. See answers displayed at the end of the game. View which answers were correct and incorrect

11

Once time has expired or all the available sentences have been judged by the user, the GamePlay area is disabled for usage. At this point all data in that area and the Status area are frozen. In addition, all the system-generated, or "correct", judgments for each sentence are displayed in the third column of the table in the ShowAnswers area. The third column will not only hold the text of yes or no, but also the text for the emotion(s) that do/does match the corresponding sentence. If a user judgment differs from the system judgment, the system judgment in the third column is

¹¹ See appendix code for **Game2.js:457-563**

highlighted in red, while if a user answer coincides with the system answer, the system answer is highlighted in green. This offers an easy indication of which answers the user got correct and incorrect. The user may look over their answers versus the system answers to reiterate or correct their judgment of the sentences emotional association.

Show Answers Area After Play

Answers:

Sentence	Your Answer	Our Answer	Report This
Carol is disapproving of her husband's plan to buy a Rolls Royce. She thinks expensive cars are a waste of money.	no	no-->(surprise;disgust;anger)	report it!
Tim is enjoying himself while playing music with his friends.	no	no-->(happiness;surprise)	report it!
Ali is condemning of the war. He thinks it shouldn't happen.	yes	yes-->(fear;anger;disgust)	report it!
Kim ia agonising over whether to find the man guilty or not guilty when she does jury duty.	no	yes-->(fear)	report it!
Tom is miffed when he forgets to take his library books back and has to pay a fine.	yes	no-->(nothing)	report it!
Sally feels praised by her teammates when they congratulate her for scoring the winning goal	no	no-->(nothing)	report it!
Rachel feels pestered by the office junior when he won't stop asking her for a date.	yes	no-->(nothing)	report it!

8. Report errors if applicable

The process of reporting errors found in Game 2 is similar to that of Game 1. Users may report a bad sentence, meaning a sentence they were presented with was grammatically incorrect or nonsensical, or users may report a "bad answer" meaning that we have not included an emotion that should be associated with the sentence. All errors reported are kept and reused in the system. See the description of this scenario in the Game 1 section for more information.

9. Check score and see calculation

This scenario of calculating and showing the final score is similar in all games. The only difference from game to game is the values used to calculate the score.

At the end of game play a user can see what their final score was and how it was calculated here. Things that factor into the score at the time of writing this thesis are the number of correct

answers, the amount of time left over, the TimeWeight and the GameWeight. The number correct is simply the number of associations that the user got correct as assumed by the system. The time left over is simply the amount of time left over. The TimeWeight variable is different for each game and is chosen based on the overall difficulty of the type of game being played. For this game the TimeWeight is 0.02. The GameWeight also differs per type of game and for this game the GameWeight is 0.2. The total score for a game is calculated by performing the following sum of products. The number correct times the GameWeight, plus, the time left over times the TimeWeight. As illustrated, parts of the formula are shown along with the final calculated score.

Scoring Area After Play
Score: #Correct: (4) Game-type factor: (0.2) Time bonus: (0) $(\#Correct * Game\text{-}type\ factor) + TimeBonus = Total$ You received a score of [0.8] points!
play again
back to home

10. Play again or quit

After checking ones score a user can now choose to play again or go back to the home page.

Game 3

Overview

In Game 3, the paradigm of game design shifts a bit from that of games 1 and 2. Here users are given one static emotion to work with and must enter in their own sentences to try and match that emotion. This type of game requires a lot more interaction and thinking on the part of the user since they have to generate an event in their mind and then construct a sentence to convey the event, all while making sure that the event matches a particular emotion. It is also important that the user uses words in the sentence that are descriptive enough to definitively link it to the emotion. The user may not use the word and in some cases the root-word of the emotion they are trying to match, forcing them to make associative links between words and thus events. The static emotion that is being matched can be chosen randomly or explicitly by the user before they begin, just like in Game 2. The number of sentences that a user must input and the amount of time they have to do so, are both dictated by the chosen difficulty level of the game.

This game was designed so that the user must construct a sentence describing an event that they feel definitively matches the emotion they are working with. The hope is that this cognitive process will cause a high functioning autistic user to go beyond simply recognizing given emotions and events, and begin to create events that can be associated with emotions. Doing well in this game necessarily shows a mastery of the goals of games 1 and 2 and stresses another component in social interaction skills. This should not only help a user interpret and get by in social interaction, but begin to create and shape social interaction according to their will, just like neurotypical people, or even better.

In addition to the user benefits in this game, there is a flip side that can provide benefit for the system. In this case user generated sentences can be added to the corpus of sentences in the

database increasing its size. More events in the database can be more helpful so it is an option that is available and can be explored and evaluated as more users play this Game.

This game is deemed relatively harder than the others because of the creative component required on top of the matching component to complete the goal.

UI Before Play Begins

ASD Games

Home | Game1 | Game2 | Game3

Game III

[more instructions](#)

Choose A Difficulty for the game: Don't Use a Timer

Easy [5 events, 85sec]

Medium [10 events, 160sec]

Hard [15 events, 230sec]

To start, pick an emotion or let us randomly pick an emotion for you:

choose an emotion ▾ -OR- [use random emotion](#)

Enter an event matching the following emotion:

Time Remaining
:XX
Done X To Go X

Answers:

Sentence	Your Answer	Our Answer	Report This
<i>**If you see a mistake "REPORT IT" by clicking on the buttons at the end of the rows. Learn More</i>			
Score:			

Log In

username:

password:

new user's click here!

Your Hi-Scores

Game1
[Log In](#)

Game2
[Log In](#)

Game3
[Log In](#)

UI During Play

ASD Games

Home | Game1 | Game2 | Game3

Game III

[more instructions](#)

Enter an event matching the following emotion:

sadness

Time Remaining
:35

Done 3 To Go 2

One day I get

Log In

username:

password:

[new user's click here!](#)

Your Hi-Scores

Game1

Game2

Game3

Answers:

Your Sentence	Emotion To Match	A Match for 'sadness'?	Report This
I am alone int he dark	sadness		
Someone steals my favorite toy.	sadness		
I move away from my best friend.	sadness		

****If you see a mistake "REPORT IT" by clicking on the buttons at the end of the rows. [Learn More](#)**

Score:

UI After Play

ASD Games

Home Game1 Game2 Game3

Game III

[more instructions](#)

Enter an event matching the following emotion:

sadness

Time Remaining: 0

Done: 3 To Go: 2

Answers:

Your Sentence	Emotion To Match	A Match for 'sadness'?	Report This
I move away from my best friend.	sadness	no --> (happiness:surprise)	report it!
Someone steals my favorite toy.	sadness	no --> (happiness:surprise)	report it! bad answer?
I am alone in the dark	sadness	no --> (fear:anger:disgust)	report it!

****If you see a mistake "REPORT IT" by clicking on the buttons at the end of the rows. [Learn More](#)**

Score:

#Correct: (0)
Game-type factor: (2)
Time bonus: (0)
(#Correct * Game-type factor) + TimeBonus = Total

You received a score of [0] points!

[play again](#)

[back to home](#)

Log In

username:

password:

[login](#)

[new user's click here!](#)

Your Hi-Scores

Game1 [Log In](#)

Game2 [Log In](#)

Game3 [Log In](#)

Design Details

Following is an expansive scenario describing usage of this game.

1. Log in

The details for logging in, registering, and displaying high scores is the same for each game. See the description for this part of Game1 for more details.

2. Read more instructions for the game, or not

The ability to read more instructions is the same for each game in the system. See the description for this part of Game1 for more details.

3. Pick a difficulty level

In this part of the scenario the only differences from game to game are the actual values denoted for each level of game play. See the description for this part of Game1 for more details.

Difficulty and Game Initiation UI

Choose A Difficulty for the game: Don't Use a Timer

Easy [5 events, 85sec]

Medium [10 events, 160sec]

Hard [15 events, 230sec]

To start, pick an emotion or let us randomly pick an emotion for you:

-OR-

4. Pick an emotion to work with, or let the system choose one randomly.

A user now needs to decide what emotion they would like to work with for the game. The user may either pick one of the six Ekman emotions from a dropdown, or opt to have the system pick and emotion randomly. This variable is stored and remembered on the client side when the decision is made and automatically starts the game.

5. Start game¹²

When a user completes the preceding step of choosing an emotion to work with, the game play automatically starts. At this time the number of sentences to use, as determined by the difficulty level selected, is recorded, and the following are showing in the GamePlay area:

- The static emotion that has been selected for matching during game play.

¹² See appendix code for **Game3.js:113-334**, **ASDGameServer.py:48-69**, **WebAppToDBMethods.py:125-188**

- An editable text field that can be typed into
- An 'ok' button to submit the text in the given text field

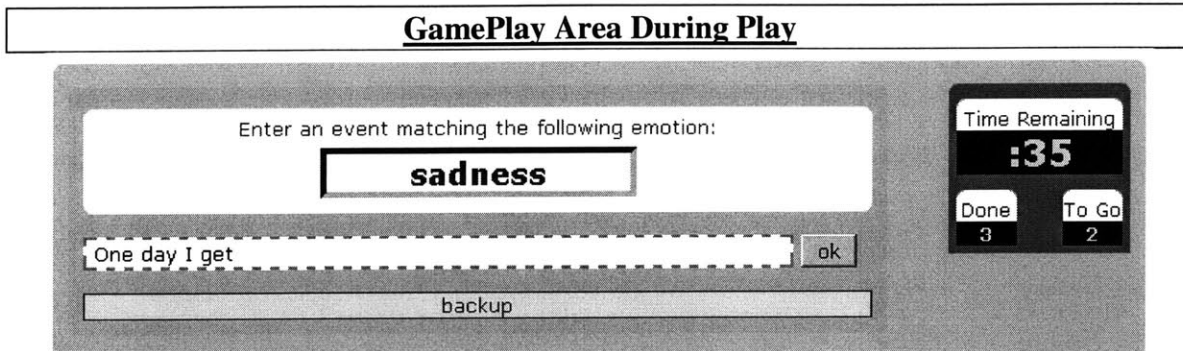
At this time the timer in the Status area is set to the allotted time associated with this level of game play and a countdown is started. Also, the number of sentences that need to be entered and that have been entered, zero so far, are shown in the Status Area. At this point no information about game play has been sent to the web server.

6. *GamePlay area usage*¹³

Now that game play has ensued the user must create an event that descriptively describes the emotion being used for this game. The user simply types in a sentence to the text field and hits enter or the Ok button to submit it. Whenever the user-created sentence is submitted, it is stored in client side code and also displayed in the table of the ShowAnswers area, along with the Emotion being matched. This data is placed in only the first 2 of the 4 columns of the table, while the other columns remain blank. The first column of the table will hold the sentence entered by the user, while the second column will hold the static emotion being matched against. At this same time a text field of the GamePlay area goes blank and takes focus so that typing of a new sentence can begin immediately. Concurrently, the Status area is updated to show how many sentences have been entered by the user and how many more need to be entered to complete the game. The Status area also displays the current amount of time left to finish the game. Each time the user enters a sentence this process will repeat, unless there is no more time left or there are no more sentences to enter. At any time before one of these two events occur, signaling the end of the game, a user may choose to backup and change any sentence that was entered earlier, using the backup button in the GamePlay area. Whenever the backup button is invoked, the last entered sentence is placed back in the text field and highlighted. Invoking the backup button also causes

¹³ See appendix code for **Game3.js**: 342-457

the Status area to update accordingly, reflecting the new amount of entered and un-entered sentences.



ShowAnswers Area During Play

Answers:

Your Sentence	Emotion To Match	A Match for 'sadness'?	Report This
I am alone int he dark	sadness		
Someone steals my favorite toy.	sadness		
I move away from my best friend.	sadness		

7. See answers displayed at the end of the game. View which answers were correct and incorrect. Once time has expired or the total number of requested sentences has been entered by the user, the GamePlay area is disabled for usage. At this point all data in that area and the Status area are frozen.

At this point the correct answers, as judged by the computer-AI, need to be displayed. In order to do this the client side code sends over all the user-generated sentences to the web server for judgment by the AI module. The AI module runs each sentence through the algorithm described earlier and returns the emotion associated with each of them. If the sentence already exists in the database, the system gets the emotions denoted in the database for that sentence; this means it will include user-suggested emotion corrections if they exist. At this point the user-generated sentences can also be entered into the database if they do not already exist. Once all sentences

have their correct emotional associations determined they are all sent back to the client side code.

14

With a list of the correct emotional associations for each of the user-entered sentences, the client side code can compare these to the emotion that was supposed to be matched in the game. For each sentence entered, if the associated emotion returned by the web server matches that of the static emotion for the game, it is deemed correct, otherwise it is deemed incorrect. These results are recorded and placed into the third column of the table in the ShowAnswers area. The third column shows whether or not the sentence in that row was a match for the given emotion and then shows which emotions do in fact go with that sentence, regardless of correctness. Incorrectly matched sentences cause the third column of their row to be highlighted in red, while matching ones cause the third column of the row to be highlighted in green. This offers an easy indication of which answers the user got correct and incorrect.¹⁵

At this point the user may look over their answers versus the system answers to reiterate or correct their judgment of the sentences that they entered and their emotional associations.

ShowAnswers Area After Play			
Answers:			
Your Sentence	Emotion To Match	A Match for 'sadness'?	Report This
I move away from my best friend.	sadness	no --> (happiness;surprise)	report it!
Someone steals my favorite toy.	sadness	no --> (happiness;surprise)	report it! bad answer?
I am alone in the dark	sadness	no --> (fear;anger;disgust)	report it!

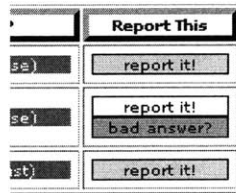
8. Report errors if applicable

¹⁴ See appendix code for **Game3.js: 508-694**, **ASDGameServer.py:68-88**, **WebAppToDBMethods.py:189-230**

¹⁵ See appendix code for **Game3.js:540-626**

The process of reporting errors found in Game 3 is similar to that of games 1 and 2, save for the lost option to report a bad sentence. Because the sentence is user-generated, the only system error plausible is that of a bad answer. Thus there is only affordance for a user to report a "bad answer" meaning that we have not included an emotion that should be associated with the sentence. As usual, all errors reported are kept and reused in the system. See the description of this scenario in the Game 1 section for more information.

Reporting Area After Play



9. Check score and see calculation

This scenario of calculating and showing the final score is similar in all games. The only difference from game to game is the values used to calculate the score.

At the end of game play a user can see what their final score was and how it was calculated here. Things that factor into the score at the time of writing this thesis are the number of correct answers, the amount of time left over, the TimeWeight and the GameWeight. The number correct is simply the number of associations that the user got correct as assumed by the system. The time left over is simply the amount of time left over. The TimeWeight variable is different for each game and is chosen based on the overall difficult of the type of game being played. For this game the TimeWeight is 0.2. The GameWeight also differs per type of game and for this game the GameWeight is 2.0. The total score for a game is calculated by performing the following sum of products. The number correct times the GameWeight, plus, the time left over times the TimeWeight. As illustrated, parts of the formula are shown along with the final calculated score.

Scoring Area After Play

Score:

#Correct: (0)

Game-type factor: (2)

Time bonus: (0)

(#Correct * Game-type factor) + TimeBonus = Total

You received a score of [0] points!

play again

back to home

10. Play again or quit

After checking ones score a user can now choose to play again or go back to the home page.

Findings

After building the application as described in the prior sections, it was put up for public consumption. At the time of writing this application was hosted at <http://asdgames.dyndns.org>. The application was built to the specification described above and many users were recruited to try it out. Links to the application were placed on sites that were resource centers for people with forms of ASD. Users with and without autism both used the application over about a 3 week period. By analyzing data about usage statistics, and collecting feedback surveys from a random sampling of users, the following findings arose. Overall, these findings summarily lead to a number of conclusions in the next section.

Analyzing Usage Statistics

Using a free stat counter on the site, the general usage statistics of people visiting the site were able to be captured. Over the three week period 473 unique users saw and used the application, with 47 of these visitors returning for later use. Altogether users ended up generating about 3400 page loads as well.

Six people signed up as registered members for the site. Registration was a prerequisite for the system to track details about individual usage of the application, thus only after signing up was activity recorded for these registered members. Of the six registered members, five of them completed game 1 after registering, five of them played game 2 after registering, and 3 of them played game 3 after registering. Two of the six users were able to improve their high scores for game 1 once.

Usage Statistics Summary:

- 473 unique users
- 47 returning visitors
- 3400+ page loads
- 6 registered users
- 2 score improvements

Feedback Form Results

One out of every three users who completed at least one game at the site was prompted with a feedback form. The forms had 4 multiple choice questions and one section for comments. Below is some of the response data gathered.

Statistics on the responses for the four multiple choice questions:

Q1) “Does this and/or the other games seem helpful to you in any way?”:

- *Yes(14 users)*
- *No(7 users)*

Q2) “Would you consider returning to this site again to play games?”:

- *Yes(15 users)*
- *No(6 users)*

Q3) “Do you think regular game play could improve your event-emotion association skills?”:

- *Yes(14 users)*
- *No(7 users)*

Q4) “What has been your overall user experience for this site so far?”:

- *Good(7 users)*
- *Mediocre(10 users)*
- *Bad(4 users)*

These are some comments that gave substantial insight on how the application was faring in public usage.

Positive Reactions:

- “I wish i could have more time to learn.... it is too quick for me, ... [...] thank you for developing these, i think they may be able to help over time.... maybe something with pictures or diagrams would help too :)”

Mixed Reactions:

- “I don't think most of these sentences go with the set of emotions here. This set comes from a limited selection of cross-cultural facial expressions, and poorly fits these sentences. The phrase "nothing" could be changed to "none of these" to help, and even better would be to allow people to enter a more appropriate emotional state label. If multiple people enter the same one, it could help the system learn.”
- “I'm thinking that it can be difficult with regards to time allowed. Some of the words are advanced vocabulary ie: maudlin for someone to immediately associate. Sometimes, there are emotions not listed like "guilt" or

"embarrassment" which are not quite anger/sadness or fear that seem unavailable in a pinch for answering."

- "I find my answers do not match up with the "Correct" answers. I am a "neuro-typical" person, and I believe I can understand and read emotions as well as the average person. This site could be useful if it gave the right answers."
- "limited by the scope/presentation of the game. It is not in real time or real circumstances, it is somewhat removed from reality therefore. Perhaps better for children rather than adults though. Would need to ask children their own perspectives of the game to get a better understanding - perhaps should ask the age of recipient at the start to gauge results better. "

Negative Reactions:

- "Your answers don't make sense. Why would Julie feel surprise when asking a boy what to eat? Why would Heather feel surprised while calculating the good and bad points of moving out of town? Why would Mark be angry or disgusted about winning an award for his landscape design, when it specifically says that he is overjoyed?"

Being targeted at high-functioning autistics, it was important that the findings and feedback results pertained to that specific group of people. While feedback forms did not explicitly ask about each user's degree of ASD or if they even had it, there are a number of factors that suggest that the findings here are largely from the ASD community. For one, all of the links to the application were posted at or distributed via ASD resource sites. Usage statistics confirmed that the traffic to the site was indeed coming from these ASD resource sites. Furthermore many personal correspondences between the author and the users, who claimed to have forms of ASD, took place during and after the 3 week trial period. Many users cited their own ASD condition and the helpfulness, or lack thereof, of the project for them specifically. The usage statistics and feedback form data exceptionally mirror the response received from personal correspondence with ASD-afflicted users of the project. These factors strengthen the suggestion that the findings presented are pertinent to high functioning autistics, and accurately describe their responses.

Conclusions and Future Work

Conclusions

Using data from user feedback forms and usage statistics, a number of conclusions became apparent.

User feedback forms revealed substantially mixed feelings about the current state of the application. Some users found the current application far too lacking in coherent matching of events and emotions, while others found it helpful to have such an application available and accessible. This is not acceptable enough to proclaim success in terms of overall user satisfaction.

User feedback forms indicated a majority of positive responses about the overall user experience. This leads to the conclusion that the user interface was adequate in terms of instructions, responsiveness, and general look&feel. The forms also indicated a majority of positive responses concerning the future usefulness of the application. Many users felt that an application like this could indeed be useful in helping people improve their event-emotion awareness. They also expressed a willingness to return to the site for further use. This leads to the conclusion that even if the current state of the application is not sufficiently robust, the overall paradigm is headed in the right direction.

With only six users choosing to register over 3 weeks, it seemed that users did not express a high level of interest in using the application as registered members. While unclear, this could have been due to a lack of confidence in the usefulness of the application at its current state. This also could have been a result of the general lack of incentives offered to the user for signing up. Since the users were still able to play the games without registering, the benefits of keeping historical track of high scores would not be readily apparent until prolonged usage (greater than the 3 week period used).

Of the small amount of registered users, only two users made improvements to existing high scores for their games. While this number does not reflect score improvement before registration, nor score improvement by unregistered users, it is still a relatively small number and can indicate a few different things. It could mean that there was simply light usage by registered users, or that these users simply were not able to improve early and often over the time period. In either case it can be said that this low number is related to the low number of registered members on the system, and the relatively short trial period. That being said, a drastic increase in these numbers should not be expected by simply increasing the trial period. Such improvements are not expected until future work is done to improve the event-emotion matching quality of the application.

Future Work

There are a number of things indicated by the findings that suggest a lot of benefit can come from future work on certain areas of the application.

Future work on this application should start with better improvements on the non-community influenced level of event-emotion matching. The current computer-generated event-emotion matching techniques used in the system are simply not robust enough to handle denotation of human emotion on a wide range of sentences. The types of improvement needed for these type of techniques to work correctly are not a reality yet, but when it does become feasible, or if alternate techniques are implemented, they will greatly improve the application as a whole.

Another avenue for future work involves first expanding the range of emotions used to denote sentences, far beyond the basic Ekman six. Secondly it involves removing the current computer-generated event-emotion matching technique altogether and relying more on static event-emotion

association and the community-generated intelligence ideas expressed earlier. This work would involve having a human-associated set of events and emotions put into the system before usage. A community-feedback based AI would come into play when correcting incorrectly associated sentences, or in adding new sentences to the corpus. This approach drastically reduces the complexity of the AI core of the system, but is a good holdover fix until better computer-generated event-emotion association techniques can be developed to correctly associate a wide range of emotions with a wide range of sentences.

The purpose of this future work would be to make the overall user satisfaction much higher than it is in the current incarnation of the application. These improvements would bring the application closer to its full potential, which was aptly recognized by many users.

Final word

At the beginning of this thesis a major question was posed that drove the work: Do the implications of a web-based application actually affect this arena of autism-related software, and if so is it for the better?

When this thesis project was set out upon the goals of the design were to improve accessibility and upgrade-deployment in autism-related software by creating a web-based application. It would have been great to also make an AI breakthrough along the way by using untested techniques as well, but that quickly became an obvious side-goal. I believe it has been proved that a web-based application definitely affects the arena of autism-related software, and for the better, simply because the goals setup at the inception of the thesis were accomplished. With a broader range of accessibility and easier deployment, the cost associated with the previous status quo for this software has fallen, and the reach of the software has expanded. The event-emotion association

techniques used have proved insufficient so we may now venture down other paths to accomplish that goal.

In the end, the current state of the application was not good enough in terms of computer-generated event-emotion association to be deemed a success (as evidenced by user feedback surveys); yet by providing a web-based application for autism-related software, it is pointing the work in this field in the right direction, and should serve as a good reference point for future work in this area.

Bibliography

Baron-Cohen, S. *Mindblindness: An essay on autism and theory of mind*. The MIT Press, Cambridge MA, 1997.

Baron-Cohen, Simon. *Mind Reading: The Interactive Guide to Emotions*. CD-ROM. London, England: Jessica Kingsley Publishers, 2004.

Cross, Hubert. *Asperger's Syndrome and Making Sense*. (2002). 13 May. 2006
<http://www.rogermeyer.com/Hubert_Cross/Asperger's_Syndrome_and_Making_Sense_July_2002.htm>.

Edelson, Meredyth Goldberg. "Social Stories." Center for the Study of Autism. (1995). Bradley Dept. of Psychology, Willamette University, Salem, Oregon. 13 May. 2006
<<http://www.autism.org/stories.html>>.

Liu, H, Lieberman, H, and Selker, T. (2003). A Model of Textual Affect Sensing using Real-World Knowledge. Proceedings of the 2003 International Conference on Intelligent User Interfaces, IUI 2003, January 12-15, 2003, Miami, FL, USA. ACM 2003, ISBN 1-58113-586-6, pp. 125-132

Lieberman, Henry, and Liu, Hugo. Adaptive linking between text and photos using common sense reasoning. Conference on Adaptive Hypermedia and Adaptive Web Systems. LNCS 2347, Springer. pp. 6-7

Mehrabian, A. (1995). Framework for a comprehensive description and measurement of emotional states. *Genetic, Social, and General Psychology Monographs*, 121, 339-361.

Mehrabian, A. (1996). Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology: Developmental, Learning, Personality, Social*, 14, 261-292.

Uta Frith: *Autism. Explaining the Enigma*. Oxford (Blackwell), 1989, pages 158-161.

Appendix of Code

PadToEmotionAlgo.py

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51

```
#####  
# this is the overall architecture that is being built for this algo along wiht a method  
to run it  
# head - 0.05, 0.25: node, node, node  
  
# L - 0.05, 0.1: sadness, node(LM), node(LR)  
# LM - 0.05, 0.05: sadness;disgust, sadness;disgust, sadness;disgust  
# LR - 0.05, 0.05: fear, fear;anger;disgust, fear;anger;disgust  
  
# M - 0.1, 0.05: nothing, node(MM), node(MR)ad  
# MM - 0.05, 0.05: nothing, nothing, nothing  
# MR - 0.05, 0.05: surprise, surprise, surprise;disgust;anger  
  
# R - 0.1, 0.05: happiness, happiness, happiness;surprise  
#####  
  
import EmotionDecisionNode as EDN  
  
LR = EDN.EmotionDecisionNode(['fear', 'fear;anger;disgust', 'fear;anger;disgust'], 0.05,  
0.05)  
#LM = EDN.EmotionDecisionNode(['sadness;disgust', 'sadness;disgust', 'sadness;disgust'],  
0.05, 0.05)  
L = EDN.EmotionDecisionNode(['sadness', 'sadness;disgust', LR], 0.05, 0.1)  
  
MR = EDN.EmotionDecisionNode(['surprise', 'surprise', 'surprise;disgust;anger'], 0.05,  
0.05)  
#MM = EDN.EmotionDecisionNode(['nothing', 'nothing', 'nothing'], 0.05, 0.05)  
M = EDN.EmotionDecisionNode(['nothing', 'nothing', MR], 0.1, 0.05)  
  
R = EDN.EmotionDecisionNode(['happiness', 'happiness', 'happiness;surprise'], 0.1, 0.05)  
  
head = EDN.EmotionDecisionNode([L, M, R], 0.05, 0.25)  
  
#test  
#print head.findEmotion([0.0,-0.14,0.3])  
  
import emotuslite  
el = emotuslite.emotuslite()  
  
def textToEmotion(txt):  
    if(txt[-1] == '.'):dapad = el.appraise_document(txt[:-1])  
    else:dapad = el.appraise_document(txt)  
  
    if(len(dapad) != 3):dapad = [0,0,0]  
    # print 'DP1: ',dapad, head.findEmotion(dapad)  
    return head.findEmotion(dapad)
```

+EmotionDecisionNode.py

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42

```
class EmotionDecisionNode:
    def __init__(self, lmr, tl, tr):
        self.lmr = lmr
        self.thresh_right = tr
        self.thresh_left = tl

    def findEmotion(self, pad):
        #should try-catch to make sure list has element
        #check for left node first
        #print self.thresh, pad, self.lmr
        #print pad[0]
        if(pad[0] < -self.thresh_left):
            a = self.lmr[0]
            if(isinstance(a, EmotionDecisionNode)):
                #print "recursing negative"
                #make sure it has second member of the list here try catch or
                throw error
                return a.findEmotion(pad[1:])
            else:
                return a
        #check for right node
        elif(pad[0] > self.thresh_right):
            b = self.lmr[2]
            if(isinstance(b, EmotionDecisionNode)):
                #print "recursing positive"
                #make sure it has second member of the list here try catch or
                throw error
                return b.findEmotion(pad[1:])
            else:
                return b
        #check for middle node
        else:
            c = self.lmr[1]
            if(isinstance(c, EmotionDecisionNode)):
                #print "recursing middle"
                #make sure it has second member of the list here try catch or
                throw error
                return c.findEmotion(pad[1:])
            else:
                return c
```

+Master.js

```

function $( ) {
    var elements = new Array();
    for (var i = 0; i < arguments.length; i++) {
        var element = arguments[i];
        if (typeof element == 'string')
            element = document.getElementById(element);
        if (arguments.length == 1)
            return element;
        elements.push(element);
    }
    return elements;
}
//ADD EVENT stuff from top JS functions
function addEvent( obj, type, fn ) {
    if (obj.addEventListener) {
        obj.addEventListener( type, fn, false );
        EventCache.add(obj, type, fn);
    }
    else if (obj.attachEvent) {
        obj["e"+type+fn] = fn;
        obj[type+fn] = function() { obj["e"+type+fn]( window.event ); }
        obj.attachEvent( "on"+type, obj[type+fn] );
        EventCache.add(obj, type, fn);
    }
    else {
        obj["on"+type] = obj["e"+type+fn];
    }
}

var EventCache = function(){
    var listEvents = [];
    return {
        listEvents : listEvents,
        add : function(node, sEventName, fHandler){
            listEvents.push(arguments);
        },
        flush : function(){
            var i, item;
            for(i = listEvents.length - 1; i >= 0; i = i - 1){
                item = listEvents[i];
                if(item[0].removeEventListener){
                    item[0].removeEventListener(item[1], item[2],
item[3]);
                }
                if(item[1].substring(0, 2) != "on"){
                    item[1] = "on" + item[1];
                }
                if(item[0].detachEvent){
                    item[0].detachEvent(item[1], item[2]);
                }
                item[0][item[1]] = null;
            }
        }
    };
}();
addEvent(window, 'unload', EventCache.flush);
//END ADD EVENT stuff

Array.prototype.inArray = function (value) {
    var i;
    for (i=0; i < this.length; i++) {
        if (this[i] === value) {
            return true;
        }
    }
    return false;
};

function togIns(mol){
    if(mol == 'more'){
        //hide more element
        $('tihide').style.display = '';
        $('timore').style.display = 'none';
        /*$('timore').style.visibility = 'hidden';
        $('tihide').style.visibility = 'visible';*/
        //put display of mainheadercontent to empty
        $('mainHeaderContent').style.display = '';
    }else{
        //hide hide element
        $('tihide').style.display = 'none';
    }
}

```

```

    $('timore').style.display = '';
    /*$("tihide").style.visibility = 'hidden';
    $("timore").style.visibility = 'visible';*/
    //put display ofmainheadercotent to none
    $('mainHeaderContent').style.display = 'none';
}

function runMail(){
    var a = $('daMail');
    var b = a.innerHTML;
    b = b.replace('G9', '@');
    b = b.replace('G8', '.');
    b = b.replace('G9', '@');
    b = b.replace('G8', '.');
    b = b.replace('G7', 'mailto');
    a.innerHTML = b;
}

//cookie stuff
function getCookie( name ) {
    var start = document.cookie.indexOf( name + "=" );
    var len = start + name.length + 1;
    if ( ( !start ) && ( name != document.cookie.substring( 0, name.length ) ) ) {
        return null;
    }
    if ( start == -1 ) return null;
    var end = document.cookie.indexOf( ";", len );
    if ( end == -1 ) end = document.cookie.length;
    return unescape( document.cookie.substring( len, end ) );
}

function setCookie( name, value, expires, path, domain, secure ) {
    var today = new Date();
    today.setTime( today.getTime() );
    if ( expires ) {
        expires = expires * 1000 * 60 * 60 * 24;
    }
    var expires_date = new Date( today.getTime() + (expires) );
    document.cookie = name+"="+escape( value ) +
        ( ( expires ) ? ";expires="+expires_date.toGMTString() : "" ) +
//expires.toGMTString()
        ( ( path ) ? ";path=" + path : "" ) +
        ( ( domain ) ? ";domain=" + domain : "" ) +
        ( ( secure ) ? ";secure" : "" );
    //alert(document.cookie + "cookie area");
}

function deleteCookie( name, path, domain ) {
    if ( getCookie( name ) ) document.cookie = name + "=" +
        ( ( path ) ? ";path=" + path : "" ) +
        ( ( domain ) ? ";domain=" + domain : "" ) +
        ";expires=Thu, 01-Jan-1970 00:00:01 GMT";
}

//create AJAX obj//
function createRequestObject() {
    var ro;
    var browser = navigator.appName;
    //netscape.security.PrivilegeManager.enablePrivilege('UniversalBrowserRead');
    if(browser == "Microsoft Internet Explorer"){
        ro = new ActiveXObject("Microsoft.XMLHTTP");
    }else{
        ro = new XMLHttpRequest();
    }
    return ro;
}
var http = createRequestObject();

// email address
// must be in user@host format
var objPatEmail = /^[a-zA-Z0-9]+([a-zA-Z0-9_-])*@[a-zA-Z0-9_-]+([a-zA-Z0-9_-]+)+/;
//register function call that then auto login
function regUsr(){
    //call python function via ajax to see if anything good comes back
    var uname = $('loginUname').value;
    var pw = $('loginPw').value;
    var email = $('regEmail').value;
    if(!objPatEmail.test(email)) {
        $('LoginBoxWarningText').innerHTML = 'Please enter a valid email, not: '+email;
        return;
    }
}

```

```

63     }
64     if(uname.length <1){
65         $('#LoginBoxWarningText').innerHTML = "Username must be greater than 0
66 characters";
67         return;
68     }
69     if(pw.length <1){
70         $('#LoginBoxWarningText').innerHTML = "password must be greater than 0
71 characters";
72         return;
73     }
74
75     $('#UserStatusAreaProgress').style.display = "";
76     http.open("get", "/regUsr?e="+email+"&u="+uname+"&p="+pw, "true");
77     http.onreadystatechange = hdlRegUsr;
78     http.send(null);
79
80 }
81 function hdlRegUsr(){
82     if(http.readyState == 4){
83         $('#UserStatusAreaProgress').style.display = "none";
84         var data = http.responseText;
85         //if response text !=contain any bad codes
86         var a = data.split('[]');
87         if(a[0] == '1'){
88             //del all Cookies about to set
89             deleteCookie( 'usr', null, null);
90             deleteCookie( 'G1HS', null, null);
91             deleteCookie( 'G2HS', null, null);
92             deleteCookie( 'G3HS', null, null);
93             //setCookie to asdgUsr, with username in response and hiscores in
94 response(split) [hiscores are all null, or just not set?]
95             setCookie( 'usr', a[1], 14, null, null, null );
96             setCookie( 'G1HS', -1, 14, null, null, null );
97             setCookie( 'G2HS', -1, 14, null, null, null );
98             setCookie( 'G3HS', -1, 14, null, null, null );
99
100             //make sure cookies got set, usr length > 0
101             var c = getCookie('usr');
102             if(c){
103                 //stop displaying th whoel login area
104                 $('#LoginBox').style.display = 'none';
105                 //fill and show the welcom name
106                 $('#UserNameDisplayBox').style.display = '';
107                 $('#UserNameDisplayText').innerHTML = c;
108                 //fill the HiScoresBox values
109                 refreshUsr();
110             }else{
111                 //else show "need cookies to register and login
112                 $('#LoginBoxWarningText').innerHTML = 'You need to have
113 cookies enabled in your browser in order to register or logon';
114             }
115             }else{
116                 //else tell msg saying bad register username/pw already exist, or email is
117 taken, or bad username chars
118                 $('#LoginBoxWarningText').innerHTML= a[1];
119             }
120         }
121     }
122 //login function call (ajax then setCookie)
123 function logUsr(){
124     //call python function via ajax to see if anything good comes back
125     var uname = $('#loginUname').value;
126     var pw = $('#loginPw').value;
127     $('#UserStatusAreaProgress').style.display = "";
128     http.open("get", "/logUsr?u="+uname+"&p="+pw, "true");
129     http.onreadystatechange = hdlLogUsr;
130     http.send(null);
131
132 }
133 function hdlLogUsr(){
134     if(http.readyState == 4){
135         $('#UserStatusAreaProgress').style.display = "none";
136         var data = http.responseText;
137         //alert(data);
138         var a = data.split('[]');
139         //if response text !=contain any bad coes
140         if(a[0] == '1'){
141             var b = a[1].split(';');
142             //del all Cookies about to set
143             deleteCookie( 'usr', null, null);

```

```

244         deleteCookie( 'G1HS', null, null);
245         deleteCookie( 'G2HS', null, null);
246         deleteCookie( 'G3HS', null, null);
247         //setCookie to asdgUsr, with usrname in response and hiscores in
248
249 response(split)
250
251         setCookie( 'usr', b[0], 14, null, null, null );
252         setCookie( 'G1HS', b[1], 14, null, null, null );
253         setCookie( 'G2HS', b[2], 14, null, null, null );
254         setCookie( 'G3HS', b[3], 14, null, null, null );
255
256         //make sure cookies got set, if usr length > 0
257         var c = getCookie('usr');
258         if(c){
259             //stop displaying th whoel login area
260             $('LoginBox').style.display = 'none';
261             //fill and show the welcom name
262             $('UserNameDisplayBox').style.display = '';
263             $('UserNameDisplayText').innerHTML = c;
264             //fill the HiScoresBox values
265             refreshUsr();
266         }else{
267             //else show "need cookies to register and login
268             $('LoginBoxWarningText').innerHTML = 'You need to have
269 cookies enabled in your browser in order to register or logon';
270         }
271         }else{
272             //else tell msg saying bad login bad username/password combo, forgot
273             //password, oh well make a new account
274             $('LoginBoxWarningText').innerHTML= a[1];
275         }
276     }
277 }
278
279 //logout function call (deleteCookie)
280 function delLogUsr(){
281     //delete all cookies named
282     deleteCookie( 'usr', null, null);
283     deleteCookie( 'G1HS', null, null);
284     deleteCookie( 'G2HS', null, null);
285     deleteCookie( 'G3HS', null, null);
286
287     //set inner html for boxes to Log In
288     refreshUsr();
289     location.reload();
290 }
291
292 function refreshUsr(){
293     //if usr vars/cookies exist reset their expiry by getting them by name and
294     //resetting them by name and new expiry date (if date is not today)
295     var a = getCookie('usr');
296     //alert(a);
297     var agone = getCookie('G1HS');
298     var agtwo = getCookie('G2HS');
299     var agthree = getCookie('G3HS');
300     //simply renews the lifetime of all cookies present
301     if(a){deleteCookie( 'usr', null, null); setCookie( 'usr', a, 14, null, null, null
302 );}
303     if(agon){deleteCookie( 'G1HS', null, null); setCookie( 'G1HS', agone, 14, null,
304 null, null );}
305     if(agtwo){deleteCookie( 'G2HS', null, null); setCookie( 'G2HS', agtwo, 14, null,
306 null, null );}
307     if(agthree){deleteCookie( 'G3HS', null, null); setCookie( 'G3HS', agthree, 14,
308 null, null, null );}
309
310     if(a){
311         //also update username and scores display now using ? syntax or normal
312         people talk
313         if(agon && agone > -1){$('Game1HiScore').innerHTML =
314 agone;}else{($('Game1HiScore').innerHTML = 'None';}
315         if(agtwo && agtwo > -1){$('Game2HiScore').innerHTML =
316 agtwo;}else{($('Game2HiScore').innerHTML = 'None';}
317         if(agthree && agthree > -1){$('Game3HiScore').innerHTML =
318 agthree;}else{($('Game3HiScore').innerHTML = 'None';}
319         //hide login area if cookies
320         $('LoginBox').style.display = 'none';
321         //show welcome area if cookies
322         $('UserNameDisplayBox').style.display = '';
323         $('UserNameDisplayText').innerHTML = a;
324     }

```

```

325     }else{
326         //alert('cant find a so maybe no cookie set');
327         // they simply arenot logge din
328         //show login area
329         showLoginUI();
330         //make sure "no score is set to score areas", shoudl automatically be so
331 or not...as people rely on me to clear up mistakes or not
332         $('#Game1HiScore').innerHTML = 'Log In';
333         $('#Game2HiScore').innerHTML = 'Log In';
334         $('#Game3HiScore').innerHTML = 'Log In';
335     }
336 }
337
338
339
340 addEvent(window, 'load', refreshUsr);
341 //regUsr('test2@temi.com', 'tintop', 'pass');
342 //logUsr('tintop', 'pass');
343 //alert(getCookie('GlHS'));
344 function showRegUI(){
345     $('#LoginBox').style.display = '';
346     $('#loginTitle').style.display='none';
347     $('#registerTitle').style.display='';
348     $('#registerEmail').style.display='';
349     $('#loginButton').style.display='none';
350     $('#registerButton').style.display='';
351     $('#cancelRegisterButton').style.display='';
352     $('#registerGateway').style.display='none';
353     $('#LoginBoxWarningText').innerHTML = '';
354     $('#UserNameDisplayBox').style.display = 'none';
355     $('#UserNameDisplayText').innerHTML = '';
356 }
357 function showLoginUI(){
358     $('#LoginBox').style.display = '';
359     $('#loginTitle').style.display='';
360     $('#registerTitle').style.display='none';
361     $('#registerEmail').style.display='none';
362     $('#loginButton').style.display='';
363     $('#registerButton').style.display='none';
364     $('#cancelRegisterButton').style.display='none';
365     $('#registerGateway').style.display='';
366     $('#LoginBoxWarningText').innerHTML = '';
367     $('#UserNameDisplayBox').style.display = 'none';
368     $('#UserNameDisplayText').innerHTML = '';
369 }

```

+Game1.js


```

1  /* Misc utility functions not related to arch */
2  function show_eval(){
3      var eval_show = Math.round(Math.random()*2);
4      if( (eval_show == 1) && !(getCookie('feedback_form')) ){
5          $('eval_frame_div').style.display='';
6          $('eval_frame').src = '../getEvalPage';
7      }
8  }
9  /* END Misc utility functions not related to arch */
10 function createRequestObject() {
11     var ro;
12     var browser = navigator.appName;
13     //netscape.security.PrivilegeManager.enablePrivilege('UniversalBrowserRead');
14     if(browser == "Microsoft Internet Explorer"){
15         ro = new ActiveXObject("Microsoft.XMLHTTP");
16     }else{
17         ro = new XMLHttpRequest();
18     }
19     return ro;
20 }
21 var http = createRequestObject();
22
23 /*
24 function toStart(){
25     setUsrVars();
26
27 }
28 //check for cookie and if so get and set username and GameHiScore variable[at end of game
29 must set this score variable if necessary and send it off for DB storage
30 /*var username;
31 var HiScore = -2;
32 function setUsrVars(){
33     //if usr vars/cookies exist reset their expiry by getting them by name and
34     resetting them by name and new expiry date (if date is not today)
35     var a = getCookie('usr');
36     var oldScore = getCookie('GlHS');
37     //if get cookie asdgUsr [meaning they logged in recently]
38     if(a){
39         //display in approp alement boxalso store in a var
40         username = a;
41         //alert(username);
42         if(oldScore){
43             //if HiScore value is int then display it in apporpiate element box
44             and store in a var
45             HiScore = oldScore;
46             //alert(HiScore);
47         }
48     }
49     //else do nothing [should start with "Log In" in both spots
50 }
51 //start the page as loads
52 //toStart();
53 */
54
55 function tryLogScore(newScore){
56     var oldScore = getCookie('GlHS');
57     //if newScoreis greater than HiScore variable AND username var !=null
58     then send ajax message to log in new HighScore for user
59     if(newScore > oldScore){
60         //alert(newScore + "im in");
61         //del old cookie info, it will mess stuff up if you dont
62         deleteCookie( 'GlHS', null, null);
63         //set new info in Cookie, just updating this part or this cookie
64         setCookie( 'GlHS', newScore, 14, null, null, null);
65         var username = getCookie('usr');
66         //alert(username);
67         if(username){//only send if the cookie is there, meaning they
68         logged in or registered or have been here lately
69         //setUsrVars();
70         $('GameHiScore').innerHTML = newScore;
71         http.open("get",
72         "/tryLogScore?u="+username+"&g=1&hs="+newScore, "true");
73         http.onreadystatechange = hdlTryLogScore;
74         http.send(null);
75     }
76 }
77     //else mada [and dont even handle it;]
78 }
79
80 function hdlTryLogScore(){
81     //literally do nothing
82 }

```

```

var GameOver;
//things that need to be called from html and referred to in other fucntions
var timer_div;//referred to in starttimer function
var currentEventBox; //referred to in placeCurrEvt funtciton
var MTD; //referred ti i startGame
/*end misc vars to be named*/
//remember to append an END "</table>" tag at the end of this variable AND to depend when
modifying the table itself to make it work
var TblHTMLNeedEndTag = "<table border=1px><th>Sentence</th> <th>Your Answer</th><th><div
id=PCATH style='visibility:visible;''>Our Answer</div></th> <th><div id=OTH
style='visibility:visible;''>Report This</div></th>";

//perhaps these should be listed by a number/code
var EmoChoiceList = "\
<ul> \
  <li><a href='javascript:void(0)' class=choiceInput id=0
onClick='recAns(this);this.blur();return false;''>surprise</a></li>\
  <li><a href='javascript:void(0)' class=choiceInput id=1
onClick='recAns(this);this.blur();return false;''>happiness</a></li>\
  <li><a href='javascript:void(0)' class=choiceInput id=2
onClick='recAns(this);this.blur();return false;''>sadness</a></li>\
  <li><a href='javascript:void(0)' class=choiceInput id=4
onClick='recAns(this);this.blur();return false;''>disgust</a></li>\
  <li><a href='javascript:void(0)' class=choiceInput id=5
onClick='recAns(this);this.blur();return false;''>fear</a></li>\
  <li><a href='javascript:void(0)' class=choiceInput id=6
onClick='recAns(this);this.blur();return false;''>anger</a></li>\
  <li><a href='javascript:void(0)' class=choiceInput id=7
onClick='recAns(this);this.blur();return false;''>nothing</a></li>\
</ul> \
";

var GameEventLength;
var GameTimeLength;
var IgnoreInputs;
var GameWeight = 0.5;
var TimeWeight = 0.05;
var use_time = 1;

function startGame(){
//set the board/UI to look as it should at the start of the game
//set game time and event lengths based off of what kind of difficulty given in start
game
GameEventLength = 10;
GameTimeLength = 55;

if($('#use_time_cb').checked){
  use_time = null;
}

//disable view of start button since cant be used till gaem stopped now
//document.getElementById('startbutton').style.visibility = 'hidden';
//document.getElementById('StartArea').style.visibility = 'hidden';
document.getElementById('StartArea').style.display = 'none';
//a variable that will hold the index number of the selected radio button
var theone
var dagdradz = $('egd', 'mgd', 'hgd');
for (var i=0;i<dagdradz.length;i++){
  if (dagdradz[i].checked==true){
    theone=i;
    break; //exist for loop, as target acquired.
  }
}
switch (dagdradz[theone].value){
  case "Easy":
    GameEventLength = 10;
    GameTimeLength = 55;
    break;

  case "Medium":
    GameEventLength = 20;
    GameTimeLength = 80;
    break;
}

```

```

63     case "Hard":
64         GameEventLength = 30;
65         GameTimeLength = 115;
66         break;
67
68     default :
69         GameEventLength = 10;
70         GameTimeLength = 45;
71 }
72
73 //clear all variable sto null that need to be for new game
74
75 //variables need to be setup
76 //things that need to be displayed after the game is over, this list is dynamic as
77 answers are given, or it is at least not inclusive
78 /*PCAnswerTblHdr = document.getElementById('PCATH');
79 OptionsTblHdr = document.getElementById('OTH');
80 PlayAgainBtn = document.getElementById('playagainbutton');
81 QuitBtn = document.getElementById('quitbutton');*/
82 //things that need to be called from html and referred to in other fucntions
83 timer_div = document.getElementById('showtimeBody'); //referebd to in starttimer
84
85 function
86     currentEventBox = document.getElementById('currenteventBody'); //referred to in
87 placeCurrEvt funtciton
88
89
90 //get event-emotion from DB(or X event-emotionsss at once) and display 1 in currentEvent
91 box:
92 reqXEvtEmo(GameEventLength);
93
94 //+get and display number of elements to go in left/togo status boxes:
95 updateEvtStatii(0, GameEventLength);
96
97 //start timer
98 if(use_time){
99     startTimer(GameTimeLength);
100 }
101 // set the main table...only done in start game
102 MTD = document.getElementById('maintabledivBody');
103 MTD.innerHTML = TblHTMLNeedEndTag + "</table>";
104 //set choice list to visible
105 var ep = document.getElementById('currentemotionoptionsBody');
106 ep.innerHTML = EmoChoiceList;
107 /*//set the event field to visible
108 var es = document.getElementById('currentevent');
109 es.innerHTML = EvtCurrent;
110 */
111
112 }
113
114 function reqXEvtEmo(x){
115 //+send request for x events-emo pairs
116     document.getElementById('GameArea').style.visibility = 'hidden';
117     document.getElementById('loadScreen').style.visibility = 'visible';
118     http.open("get", "/getEvEmPairs?num="+x, "true");
119     http.onreadystatechange = hdlIncEvtEmo;
120     http.send(null);
121 //disable (invisible) abilityt o pick using emotio options (set grand variable) ORnot,
122 depending on
123 //document.getElementById('currentemotionoptionsBody').display='none';
124 //must specify what to do when request comesback
125 //var temp = "I get up[s]happiness[b]I go to sleep[s]no emotion[b]I get shot[s]scared";
126 //hdlIncEvtEmo();
127
128 }
129
130
131 var events = new Array();
132 var pc_ans = new Array();
133 var usr_ans = new Array();
134 var reports = new Array();
135 //var global_count = 0;
136
137 //+handle event-emotion return from DB (ajax comeback)
138 function hdlIncEvtEmo(){
139     if(http.readyState == 4){
140         var data = http.responseText;
141         // from evt-emo gotten back, place it and store it along with its answer
142         in a twin array OR a holder variable of currentRightEmotion
143         var tmpB = data.split('[[[]]');
144         for(var i=0; i<tmpB.length; i++){

```

```

244         var tmpS = tmpB[i].split('[]');
245         var a =tmpS[0];
246         var b = tmpS[1];
247         events.push(a);
248         pc_ans.push(b);
249         //usr_ans
250         //reports
251     };
252     //place current event
253     placeCurrEvt(events[0]);
254     //enable (visible) inputs for emotion options OR not since all comes back
255     once
256         document.getElementById('currentemotionoptionsBody').display='';
257         document.getElementById('GameArea').style.visibility = 'visible';
258         document.getElementById('loadScreen').style.visibility = 'hidden';
259     }
260 }
261
262 //+given a sentence, it will display it in the current event box
263 function placeCurrEvt(d){
264 //set the inner HTML of the current Eventbox to the event
265 //es.innerHTML = events[global_count];
266 //alert(currentEventBox);
267 currentEventBox.innerHTML = d;
268 //global_count++;
269 }
270
271
272
273
274 var togonum;
275 var donenum;
276
277 //display number of lements to go/lef tin status boxes
278 function updateEvtStatii(dir, total){
279 /*-set innerHTML of both to reasonable values
280 -check to see if innerHTML of togo = 0 do something (call show answer method)..or maybe
281 not done here*/
282 //if dir = 0 then set done=0 and togo=total
283 if(dir ==0){
284     togonum = total;
285     donenum = 0;
286 }
287 //if dir = 1 then inc done and dec togo
288 if(dir ==1){
289     if(togonum != 0){
290         togonum--;
291         donenum++;
292     }
293 }
294 //if dir = -1 then dec done and in togo
295 if(dir ==-1){
296     if(donenum != GameEventLength){
297         togonum++;
298         donenum--;
299     }else{
300         //should we stop the game, or should they be allowed to still remove and
301         redo events and click a done button
302     }
303 }
304
305 var a = document.getElementById('itemstogoBody');
306 var b = document.getElementById('itemsdoneBody');
307 a.innerHTML = togonum;
308 b.innerHTML = donenum;
309 }
310
311
312
313 var timeLeft;
314 //starts the game timer. call function with X seconds that updated the timer element
315 innerHTML
316 function startTimer(x){
317 //when timer=0 AND not finished do someting (showanswers)
318 //if certain variable set STOP timer countdown
319 //if var passed in is NOT null
320 if(x){
321     // set timeLeft to value passed in
322     timeLeft = x;
323     timer_div.innerHTML = ":" + timeLeft;
324     window.setTimeout('startTimer(null)',1000)

```

```

325 }else if(timeLeft && !GameOver){
326     timeLeft = timeLeft - 1;
327     timer_div.innerHTML = ":" + timeLeft;
328     if(timeLeft == 10){
329         timer_div.style.color = 'red';
330     }
331     //if timeleft is greater than 0 then call self again
332     if(timeLeft > 0){
333         //alert(timeLeft);
334         window.setTimeout('startTimer(null)',1000)
335     }else{
336         stopGame();
337     }
338 }else{
339     //nothing game over
340 }
341 }
342
343 //var addedRows = new Array();
344 //record answer just given by user with emotion click
345 function recAns(pe){
346     //only if theyhavent already answered everything
347     if(donenum != events.length){
348         //incoming variable is value of what was just clicked, picked emotion
349         var emtxt = pe.innerHTML;
350         usr_ans.push(emtxt);
351
352         var evshown = $('currenteventBody');
353         //update any UI stuff needed to certify click (disable, darken, outline, sleep for
354         few mseconds, enable, lighten, no outline, GO OR just do it onMousePress)
355         evshown.style.background = 'black';
356         window.setTimeout("$('currenteventBody').style.background = 'white',200);
357         //get the current event text in the currevent box
358
359         var evtxt = evshown.innerHTML;
360
361         var rightOrWrong;
362         if(pc_ans[donenum].indexOf(emtxt) > -1){
363             rightOrWrong = 'right';
364         }else{
365             rightOrWrong = 'wrong';
366         }
367         //place the current event, picked emotion, correctanswer, report/flag in an array
368         and in the maintable
369         var apcans = "<div id='"+ pcans+String(donenum)+ "' class='"+rightOrWrong+"'"
370         style='display:none;'>"+pc_ans[donenum]+"</div>";
371         var areport = "<div id='"+ 'report'+String(donenum)+ "' style='display:none;'"
372         onClick='repClicked(this);this.blur();return false;'+ "<a href='javascript:void(0)'"
373         id='"+ 'reportLinkreport'+String(donenum)+ "'>report it!</a>"+ "</div>";
374         reports.push('report'+String(donenum));
375
376         var arow = "\
377         <tr> \
378         <td>"+ evtxt +"\
379         </td>\
380         <td>"+ emtxt +"\
381         </td>\
382         <td>"+ apcans +"\
383         </td>\
384         <td>"+ areport +"\
385         </td>\
386         </tr> \
387         "
388         //addedRows.push(arow);
389
390         //chop off top by substringing finding last of </th> +5
391         //keep bottom half too
392         //add chopped off top + arow + bottom
393         var a = MTD.innerHTML;
394         var bFFX = a.lastIndexOf('</th>');
395         var bIE = a.lastIndexOf('</TH>');
396         var b = Math.max(bFFX, bIE);
397
398         var top = a.substring(0, b + 10);
399         var bot = a.substring(b+10, a.length);
400         //alert(top);
401         //alert(bot);
402         MTD.innerHTML = top + arow + bot;
403         //add closed table tag
404
405         //update eventStatii to increasase done and decrease togo

```

```

406         updateEvtStatii(1, null);
407
408         //request another event (from DB or from stored amount) to be placed in current
409 event. or place it yourself by taking/popping off the stack of returned ev-emo array
410         var nextev = events[donenum];
411         placeCurrEvt(nextev);
412         document.getElementById('backupbutton').style.visibility='';
413         if(donenum == GameEventLength){
414             stopGame();
415         }
416     }
417 }
418
419 function removeLastAns(){
420 if(donenum > 0 && donenum < GameEventLength ){
421     var a = MTD.innerHTML;
422     var bFFX = a.lastIndexOf('</th>');
423     var bIE = a.lastIndexOf('</TH>');
424     var b = Math.max(bFFX, bIE);
425     var top = a.substring(0, b + 10);
426     var bot = a.substring(b+10, a.length);
427     var tmpBotFFX = bot.indexOf('</tr>');
428     var tmpBotIE = bot.indexOf('</TR>');
429     var tmpBot = Math.max(tmpBotFFX, tmpBotIE);
430     bot = bot.substring(tmpBot + 5, bot.length);
431     MTD.innerHTML = top + bot;
432     /*var sIE = MTD.innerHTML.lastIndexOf('<TR>');
433     var sFFX = MTD.innerHTML.lastIndexOf('<tr>');
434     var s = Math.max(sIE, sFFX);
435     //alert(s);
436     var f = MTD.innerHTML.lastIndexOf('</tr>');
437     MTD.innerHTML = MTD.innerHTML.substring(0, s) + '</table>';
438     */
439     //statii updated to decrease
440     updateEvtStatii(-1, null);
441     //usr_ans.pop() to take off last entry
442     usr_ans.pop();
443     reports.pop();
444     //latest added row to table to be removed
445     //addedRows.pop();
446     $('currenteventBody').style.background = 'lightgreen';
447     window.setTimeout(function(){
448         $('currenteventBody').style.background = 'white';
449     },150);
450     placeCurrEvt(events[donenum]);
451 }
452 }
453
454 function stopGame(){
455 //reveal hidden elements in maintable
456     var pcanscolhdr = document.getElementById('PCATH');
457     var rptcolhdr = document.getElementById('OTH');
458     pcanscolhdr.style.display='';
459     rptcolhdr.style.display='';
460 //end use of back button by hiding
461     document.getElementById('backupbutton').style.visibility = 'hidden';
462 //we use the length of reports so that we only reveal answers that have been placed in
463 the table, not ALL of them available
464 for(var i=0; i<usr_ans.length; i++){
465     var a = document.getElementById(reports[i]);
466     var b = document.getElementById('pcans'+ String(i));
467     a.style.display='';
468     b.style.display='';
469 }
470
471 //hide certain input elemetns (emotion list, event read)
472 document.getElementById('currentemotionoptionsBody').style.visibility = 'hidden';
473 document.getElementById('currenteventBody').style.visibility = 'hidden';
474 //document.getElementById('startbutton').style.visibility = 'hidden';
475
476 //calculate score and display it to user (possibly record it)
477 var corr=0;
478 for(var i=0; i<usr_ans.length; i++){
479     var b = usr_ans[i];
480     var a = pc_ans[i];
481     //if(a == b){
482     if(a.indexOf(b) > -1){
483         //inc correct total
484         corr++
485     }
486 }

```

```

//add img src text for star to inner HTML of ? AND hilighthe boxes
certain way/color yellow or lime or green for both boxes (like a match)
    }else{
//add other img text for red x to inner html of ? AND higlight and hilight
boxesc certain way/color/ red or black for each box so they are the same color
    }
}
//stop the clock
GameOver = 1;
var tp = (use_time) ? timeLeft : 0;
tp = Math.round(tp * TimeWeight*10)/10;
//Math.round()
//alert(corr*GameWeight);
var TS = Math.round((tp+corr*GameWeight)*10)/10;

//send off tototal score to be logged in Db
tryLogScore(TS);
//tp++;
//alert(tp);
var sb = document.getElementById('scoreboxBody');
sb.innerHTML = "\
                                <p>#Correct: (" +corr+")<br>\
                                Game-type factor: (" +GameWeight+")<br> \
                                Time bonus: (" +tp+")<br> \
                                (#Correct * Game-type factor) + TimeBonus = Total</p>\
                                <p>You received a score of <span
id=scoreBoxActualScore["+TS+"]</span> points!</p>\
                                ";

// "Your score is (" +String(corr)+" correct answers) X (" +timeLeft+" seconds left + 1) =
"+(tp *corr)+" Total Points!";

//reveal hidden buttons that should be shown now (quit,play next-round/again, my-alltime-
high-score) : specify what should happen for each button first
/*document.getElementById('playagainbutton').style.display = '';
document.getElementById('quitbutton').style.display = '';
*/
$('EndGameButtonArea').style.display = '';
show_eval();
}

//reset UI and variables, so that its like a page refresh
function playAgain(){
//can we just reload the page at this point? only if not ndoing rounds

}

function repClicked(ro){
//get the ahref "button" link and change color of it
var repLinkBox = document.getElementById("reportLink" + ro.id);
repLinkBox.style.background='white';
//ro is the element that was clicked on. get ID
//create a div using this id appended to something
var mi = "rptBox" + ro.id;
var ii = 'rptBoxA' + ro.id;
var iid = 'rptBoxAData' + ro.id;
var oii = 'rptBoxB' + ro.id;
var oiid = 'rptBoxBData' + ro.id;
var rbtxt = "<span id='"+ mi +"'" > \
                                <div><a href='#' id='"+ii+"'
onClick='rptBoxClicked(this.id);this.blur();return false;'>bad answer?</a> \
                                <span id='"+iid+"Holder' style='display:none;'>\
                                If you think that our answer is wrong or missing an
emotion, let us know which emotion you think should go with this sentence.\
                                <select id='"+iid+"' onChange='rptValSent(this.id);'> \
                                <option selected>choose an emotion</option> \
                                <option value=happiness>happiness</option> \
                                <option value=sadness>sadness</option> \
                                <option value=surprise>surprise</option> \
                                <option value=anger>anger</option> \
                                <option value=fear>fear</option> \
                                <option value=disgust>disgust</option> \
                                <option value=nothing>nothing</option> \
                                </select> \
                                </span> \
                                </div> \
                                <div><a href='#' id='"+oii+"'
onClick='rptBoxClicked(this.id);this.blur();return false;'>bad sentence?</a> \
                                <input id='"+oiid+"' type=text style='display:none;'></input> \
                                </div> \

```

```

568         "+"</span>";
569
570 //add the div to the page somehow (empty div/span)
571 var c = document.getElementById(ii);
572 //only if not already expanded
573 if(!c && ro.innerHTML.indexOf('report it!')>-1){
574     ro.innerHTML+=rbtxt;
575 }
576 //get the div object and move its position to the position fo the ro object (to make fit
577 with others and keep up, or to disappear?) [its already there]
578 }
579 function rptBoxClicked(d){
580 //take this id and see if it is a BoxA or B
581 var i = d.indexOf('rptBoxA');
582 if(i > -1){
583     // it was found so show the iid element by adding rptBoxAData.style.display='';
584     //alert('rptBoxAData'+d.substring(7, d.length));
585     document.getElementById('rptBoxAData'+d.substring(7)+'Holder').style.display='';
586 }else{
587     //wasnt found show the oiid element
588     //document.getElementById('rptBoxBData'+d.substring(7)).style.display='';
589     rptValSent(d);
590 }
591 }
592
593 function rptValSent(id){
594 //check if it is a sentence bad report OR
595 var a = id.indexOf('report');
596 var evt = events[id.substring(a+6)];
597 var i = id.indexOf('rptBoxA');
598 //check if its an emotion report
599 if(i > -1){
600     //if emotion then get value of that selected and the appropriate sentence send to
601 server/function the variables
602     var d = document.getElementById(id).value;
603     http.open("get", "/reportEvtError?evt="+evt+"&type=type1&data="+d, "true");
604     http.onreadystatechange = hdlReportVerification;
605     http.send(null);
606     alert('The following emotion('+d+') has been reported as being associated with
607 this event:('+ evt +')');
608 }else{
609     //its sentence, so send the bad sentence to the server (server should mark time/count
610 and flag)
611     http.open("get", "/reportEvtError?evt="+evt+"&type=type0", "true");
612     http.onreadystatechange = hdlReportVerification;
613     http.send(null);
614     alert('This sentence has been reported as bad: ' + evt);
615 }
616 //set innerHTML to be reported. Thank you.
617 var repBox = document.getElementById(id.substring(a));
618 repBox.innerHTML = 'Reported!';
619 }
620
621 function hdlReportVerification(){
622 var data = http.responseText;
623 }
624
625 /*/--accept flags/corrections if given for an event
626 function repEvtFlag(a, b, data){
627 //given the event (via id or event itself) = a [may have to depend part of the id to get
628 id of the actual event]OR[may have to find event match and use twin array structure]
629 //and given a type of flagging/reporting = b
630 //and given any extra data if needed depending on flag type
631 //post event and flag to the server to be timestamped/counted/record in db
632 //do nothing when it gets back, just update the UI when this function is over (like
633 reported in red comes up and it fades out? or stays there) It leaves, but then you
634 replace the innerHTML of that reportID to Reported! with no link*/
635
636

```


+Game2.js

```

1  /* Misc utility functions not related to arch */
2  function show_eval(){
3      var eval_show = Math.round(Math.random()*2);
4      if( (eval_show == 1) && !(getCookie('feedback_form')) ){
5          $('eval_frame_div').style.display='';
6          $('eval_frame').src = '../getEvalPage';
7      }
8  }
9  /* END Misc utility functions not related to arch */
10 function createRequestObject() {
11     var ro;
12     var browser = navigator.appName;
13     //netscape.security.PrivilegeManager.enablePrivilege('UniversalBrowserRead');
14     if(browser == "Microsoft Internet Explorer"){
15         ro = new ActiveXObject("Microsoft.XMLHTTP");
16     }else{
17         ro = new XMLHttpRequest();
18     }
19     return ro;
20 }
21 var http = createRequestObject();
22 /*misc variables that need to be named*/
23 //things that need to be displayed after the game is over, this list is dynamic as
24 answers are given, or it is at least not inclusive
25 //var PCAnswerTblHdr;
26 //var OptionsTblHdr;
27 //var PlayAgainBtn;
28 //var QuitBtn;
29
30 function tryLogScore(newScore){
31     var oldScore = getCookie('G2HS');
32     //if newScore is greater than HiScore variable AND username var !=null
33     then send ajax message to log in new HighScore for user
34     if(newScore > oldScore){
35         //alert(newScore + "im in");
36         //del old cookie info, it will mess stuff up if you dont
37         deleteCookie( 'G2HS', null, null);
38         //set new info in Cookie, just updating this part or this cookie
39         setCookie( 'G2HS', newScore, 14, null, null, null);
40         var username = getCookie('usr');
41         //alert(username);
42         if(username){//only send if the cookie is there, meaning they
43         logged in or registered or have been here lately
44             //setUsrVars();
45             $('Game2HiScore').innerHTML = newScore;
46             http.open("get",
47             "/tryLogScore?u="+username+"&g=2&hs="+newScore, "true");
48             http.onreadystatechange = hdlTryLogScore;
49             http.send(null);
50         }
51     }
52     //else mada [and dont even handle it;]
53 }
54 function hdlTryLogScore(){
55     //literally do nothing
56 }
57
58 var GameOver;
59 //things that need to be called from html and referred to in other fuctions
60 var timer_div;//referred to in starttimer function
61 var currentEventBox; //referred to in placeCurrEvt funtciton
62 var MTD; //referred ti i startGame
63 /*end misc vars to be named*/
64 //remember to append an END "</table>" tag at the end of this variable AND to depend when
65 modifying the table itself to make it work
66 //var TblHTMLNeedEndTag = "<table><th>Event</th> <th>Your Answer</th><th><div id=PCATH
67 style='display:none;'>Our Answer</div></th> <th><div id=OTH style='display:none;'>Report
68 This</div></th>";
69 var TblHTMLNeedEndTag = "<table border=1px><th>Sentence</th> <th>Your Answer</th><th><div
70 id=PCATH style='visibility:visible;'>Our Answer</div></th> <th><div id=OTH
71 style='visibility:visible;'>Report This</div></th>";
72
73 //perhaps these should be listed by a number/code
74 var EmoChoiceList = "\
75 <ul> \
76     <li><a href='javascript:void(0)' id=0 onClick='recAns(this);this.blur();return
77 false;'>yes</a></li>\
78     <li><a href='javascript:void(0)' id=1 onClick='recAns(this);this.blur();return
79 false;'>no</a></li>\
80 </ul> \
81 ";

```

```

var PossibleEmotions = new Array();
PossibleEmotions.push('happiness');
PossibleEmotions.push('sadness');
PossibleEmotions.push('surprise');
PossibleEmotions.push('fear');
PossibleEmotions.push('anger');
PossibleEmotions.push('disgust');
PossibleEmotions.push('nothing');

/*entry innher HTML should have constant width done in style, along with colors
var EvtCurrent = "\
    <div id='evtshowingtitle'>What emotion goes with the event below?</div> \
    <div id='evtshowing'></div> \
";*/

var GameEventLength;
var GameTimeLength;
var IgnoreInputs;
var GameWeight = 0.2;
var TimeWeight = 0.02;
var use_time = 1;

function startGame(emotouse){
//set the board/UI to look as it should at the start of the game
//set game time and event lengths based off of what kind of difficulty given in start
game
//GameEventLength = 3;
//GameTimeLength = 15;
if($('#use_time_cb').checked){
    use_time = null;
}

//if passed in an emo to use, call place emotion with it, else call it anyway

//also make emotion picking non-plausible anymore
//document.getElementById('pickemotiontype').style.visibility='hidden';
document.getElementById('StartArea').style.display='none';
//a variable that will hold the index number of the selected radio button
var theone
var dagdradz = $('egd', 'mgd', 'hgd');
for (var i=0;i<dagdradz.length;i++){
    if (dagdradz[i].checked==true){
        theone=i;
        break; //exist for loop, as target acquired.
    }
}
switch (dagdradz[theone].value){
    case "Easy":
        GameEventLength = 10;
        GameTimeLength = 60;
        break;

    case "Medium":
        GameEventLength = 20;
        GameTimeLength = 110;
        break;

    case "Hard":
        GameEventLength = 30;
        GameTimeLength = 165;
        break;

    default :
        GameEventLength = 10;
        GameTimeLength = 60;
}

placeEmotion(emotouse);

timer_div = document.getElementById('showtimeBody'); //referred to in starttimer function
currentEventBox = document.getElementById('eventtojudgeBody'); //referred to in
placeCurrEvt funtciton

//start timer
if(use_time){
    startTimer(GameTimeLength);
}

```

```

62 //get event-emotion from DB(or X event-emotions at once) and display 1 in currentEvent
63 box:
64 reqxEvtEmo (GameEventLength);
65
66 //+get and display number of elements to go in left/togo status boxes:
67 updateEvtStatii(0, GameEventLength);
68
69
70
71 // set the main table...only done in start game
72 MTD = document.getElementById('maintabledivBody');
73 MTD.innerHTML = TblHTMLNeedEndTag + "</table>";
74 //set choice list to visible
75 document.getElementById('currentyesnooptionsBody').innerHTML = EmoChoiceList;
76 /**/set the event field to visible
77 var es = document.getElementById('currentevent');
78 es.innerHTML = EvtCurrent;
79 */
80
81 }
82
83 var PickedEmotionForMatching;
84 function placeEmotion(a){
85 //if emotion passed in set that, else set random from some home variable
86 var daEmotion;
87 if(a){
88     daEmotion = a;
89 }
90 else{
91     var randnum = Math.floor(Math.random()*(PossibleEmotions.length));
92     daEmotion = PossibleEmotions[randnum];
93 }
94 PickedEmotionForMatching = daEmotion;
95 document.getElementById('emotiontomatchBody').innerHTML = daEmotion;
96 }
97
98 function reqxEvtEmo(x){
99 //+send request for x events-emo pairs
100 //this is number (random between 1-3)
101 var xtra = Math.round(2*Math.random()) + 1;
102 x = x - xtra;
103 //disable (invisible) ability to pick using emotio options (set grand variable) ORnot,
104 depending on
105 document.getElementById('GameArea').style.visibility = 'hidden';
106 document.getElementById('loadScreen').style.visibility = 'visible';
107 http.open("get",
108 "/getEvEmPairs?num="+x+"&xemt="+PickedEmotionForMatching+"&xemtnum="+xtra, "true");
109 http.onreadystatechange = hdlIncEvtEmo;
110 http.send(null);
111 //document.getElementById('currentyesnooptionsBody').display='none';
112 //must specify what to do when request comesback
113 //var temp = "I get up[s]happiness[b]I go to sleep[s]no emotion[b]I get shot[s]scared";
114 //hdlIncEvtEmo(temp);
115 }
116
117 }
118
119 var events = new Array();
120 var pc_ans = new Array();
121 var usr_ans = new Array();
122 var reports = new Array();
123 //var global_count = 0;
124
125 //+handle event-emotion return from DB (ajax comeback)
126 function hdlIncEvtEmo(){
127 // from evt-emo gotten back, place it and store it along with its answer in a twin
128 array OR a holder variable of currentRightEmotion
129 if(http.readyState == 4){
130     var data = http.responseText;
131     var tmpB = data.split('[[[]]');
132     for(var i=0; i<tmpB.length; i++){
133         var tmpS = tmpB[i].split('[]');
134         var a = tmpS[0];
135         var b = tmpS[1];
136         events.push(a);
137         pc_ans.push(b);
138         //usr_ans
139         //reports
140     };
141     //place current event
142     placeCurrEvt(events[0]);

```

```

243         //enable (visible) inputs for emotion options OR not since all comes back
244 once
245     document.getElementById('currentyesnooptionsBody').display='';
246     document.getElementById('GameArea').style.visibility = 'visible';
247     document.getElementById('loadScreen').style.visibility = 'hidden';
248 }
249 }
250
251
252 //+given a sentence, it will display it in the current event box
253 function placeCurrEvt(d){
254     //set the inner HTML of the current Eventbox to the event
255     //es.innerHTML = events[global_count];
256     //alert(currentEventBox);
257     currentEventBox.innerHTML = d;
258     //global_count++;
259 }
260
261
262
263 var togonum;
264 var donenum;
265
266 //display number of lements to go/lef tin status boxes
267 function updateEvtStatii(dir, total){
268     /*-set innerHTML of both to reasonable values
269     -check to see if innerHTML of togo = 0 do something (call show answer method)..or maybe
270     not done here*/
271     //if dir = 0 then set done=0 and togo=total
272     if(dir ==0){
273         togonum = total;
274         donenum = 0;
275     }
276     //if dir = 1 then inc done and dec togo
277     if(dir ==1){
278         if(togonum != 0){
279             togonum--;
280             donenum++;
281         }
282     }
283     //if dir = -1 then dec done and in togo
284     if(dir ==-1){
285         if(donenum != GameEventLength){
286             togonum++;
287             donenum--;
288         }else{
289             //should we stop the game, or should they be allowed to still remove and
290             redo events and click a done button
291         }
292     }
293 }
294
295 var a = document.getElementById('itemstogoBody');
296 var b = document.getElementById('itemsdoneBody');
297 a.innerHTML = togonum;
298 b.innerHTML = donenum;
299 }
300
301
302 var timeLeft;
303 //starts the game timer. call function with X seconds that updated the timer element
304 innerHTML
305 function startTimer(x){
306     //when timer=0 AND not finished do someting (showanswers)
307     //if certain variable set STOP timer countdown
308     //if var passed in is NOT null
309     if(x){
310         // set timeLeft to value passed in
311         timeLeft = x;
312         timer_div.innerHTML = ":" + timeLeft;
313         window.setTimeout('startTimer(null)',1000)
314     }else if(timeLeft && !GameOver){
315         timeLeft = timeLeft - 1;
316         timer_div.innerHTML = ":" + timeLeft;
317         if(timeLeft == 10){
318             timer_div.style.color = 'red';
319         }
320     }
321     //if timeleft is greater than 0 then call self again
322     if(timeLeft > 0){
323         //alert(timeLeft);
324         window.setTimeout('startTimer(null)',1000)

```

```

324     }else{
325         stopGame();
326     }
327 }else{
328     //nothing game over
329 }
330 }
331
332 //var addedRows = new Array();
333 //record answer just given by user with emotion click
334 function recAns(pe){
335     //only if theyhavent already answered everything
336     if(donenum != events.length){
337         //incoming variable is value of what was just clicked, picked yes or no
338
339         for match
340             var usr_yes_or_no_txt = pe.innerHTML;
341             usr_ans.push(usr_yes_or_no_txt);
342             //update any UI stuff needed to certify click (disable, darken, outline,
343             sleep for few mseconds, enable, lighten, no outline, GO OR just do it onMousePress)
344             var evtotjudge = $('eventtojudgeBody');
345
346             evtotjudge.style.background = 'black';
347             window.setTimeout("${'eventtojudgeBody'}.style.background = 'white'",200);
348             //get the current event text in the currevent box
349             var evtxt = evtotjudge.innerHTML;
350
351             var pc_yes_or_no_txt;
352             var rightOrWrong;
353
354             //determine if OUR answer is yes or no if matchemotion contained in
355             pcmotion-> yes, else no
356             if(pc_ans[donenum].indexOf(PickedEmotionForMatching) > -1){
357                 pc_yes_or_no_txt = 'yes';
358
359             }else{
360                 pc_yes_or_no_txt = 'no';
361
362             }
363
364             if(pc_yes_or_no_txt == usr_yes_or_no_txt){
365                 rightOrWrong = 'right';
366             }
367             else{
368                 rightOrWrong = 'wrong';
369             }
370             //place the current event, picked emotion, correctanswer, report/flag in
371             an array and in the maintable
372             var apcans = "<div id='"+ pcans'+String(donenum)+ " '
373             class='"+rightOrWrong+"' style='display:none;'>"+pc_yes_or_no_txt+"--
374             >("+pc_ans[donenum]+")</div>";
375             var areport = "<div id='"+ 'report'+String(donenum)+ " '
376             style='display:none;' onClick='repClicked(this);this.blur();return false;'>"+<a
377             href='javascript:void(0)' id='"+ 'reportLinkreport'+String(donenum)+ "'>report
378             it!</a>"+</div>";
379             reports.push('report'+String(donenum));
380
381             var arow = "\
382             <tr> \
383             <td>"+ evtxt +" \
384             </td>\
385             <td>"+ usr_yes_or_no_txt +" \
386             </td>\
387             <td>"+ apcans +" \
388             </td>\
389             <td>"+ areport +" \
390             </td>\
391             </tr> \
392             "
393             //addedRows.push(arow);
394             var a = MTD.innerHTML;
395             var bFFX = a.lastIndexOf('</th>');
396             var bIE = a.lastIndexOf('</TH>');
397             var b = Math.max(bFFX, bIE);
398
399             var top = a.substring(0, b + 10);
400             var bot = a.substring(b+10, a.length);
401             //alert(top);
402             //alert(bot);
403             MTD.innerHTML = top + arow + bot;
404             /*var tmp = MTD.innerHTML;
405             tmp = tmp.substring(0, tmp.length - 8); //remove </table> tag

```

```

405         tmp = tmp + arow; //add new arow
406         MTD.innerHTML = tmp + '</table>'//add closed table tag
407         */
408
409         //update eventStatii to increasase done and decrease togo
410         updateEvtStatii(1, null);
411
412         //request another event (from DB or from stored amount) to be placed in
413         current event. or place it yourself by taking/popping off the stack of returned ev-emo
414         array
415         var nextev = events[donenum];
416         placeCurrEvt(nextev);
417         document.getElementById('backupbutton').style.visibility='visible';
418         if(donenum == GameEventLength){
419             stopGame();
420         }
421     }
422 }
423 //document.getElementById('GameArea').focus();
424 }
425
426 function removeLastAns(){
427 if(donenum > 0 && donenum < GameEventLength ){
428     var a = MTD.innerHTML;
429     var bFFX = a.lastIndexOf('</th>');
430     var bIE = a.lastIndexOf('</TH>');
431     var b = Math.max(bFFX, bIE);
432     var top = a.substring(0, b + 10);
433     var bot = a.substring(b+10, a.length);
434     var tmpBotFFX = bot.indexOf('</tr>');
435     var tmpBotIE = bot.indexOf('</TR>');
436     var tmpBot = Math.max(tmpBotFFX, tmpBotIE);
437     bot = bot.substring(tmpBot + 5, bot.length);
438     MTD.innerHTML = top + bot;
439     /*var sIE = MTD.innerHTML.lastIndexOf('<TR>');
440     var sFFX = MTD.innerHTML.lastIndexOf('<tr>');
441     var s = Math.max(sIE, sFFX);
442     //var f = MTD.innerHTML.lastIndexOf('</tr>');
443     MTD.innerHTML = MTD.innerHTML.substring(0, s) + '</table>';
444     */
445     //statii updated to decrease
446     updateEvtStatii(-1, null);
447     //usr_ans.pop() to take off last entry
448     usr_ans.pop();
449     reports.pop();
450     //latest added row to table to be removed
451     //addedRows.pop();
452     $('eventtojudgeBody').style.background = 'lightgreen';
453     window.setTimeout("${'eventtojudgeBody'}.style.background = 'white'",150);
454     placeCurrEvt(events[donenum]);
455 }
456 }
457
458 function stopGame(){
459 //reveal hidden elements in maintable
460     var pcanscolhdr = document.getElementById('PCATH');
461     var rptcolhdr = document.getElementById('OTH');
462     pcanscolhdr.style.display='';
463     rptcolhdr.style.display='';
464 //end use of back button by hiding
465     document.getElementById('backupbutton').style.visibility = 'hidden';
466 //we use the length of reports so that we only reveal answers that have been placed in
467 the table, not ALL of them available
468 for(var i=0; i<usr_ans.length; i++){
469     var a = document.getElementById(reports[i]);
470     var b = document.getElementById('pcans'+ String(i));
471     a.style.display='';
472     b.style.display='';
473 }
474 }
475
476 //hide certain input elemetns (emotion list, event read)
477 document.getElementById('currentyesnooptionsBody').style.visibility = 'hidden';
478 document.getElementById('eventtojudgeBody').style.visibility = 'hidden';
479 //document.getElementById('startrandom').style.visibility = 'hidden';
480
481 //calculate score and display it to user (possibly record it)
482 var corr=0;
483 for(var i=0; i<usr_ans.length; i++){
484     var b = usr_ans[i];
485

```



```

567 var repLinkBox = document.getElementById("reportLink" + ro.id);
568 repLinkBox.style.background='white';
569 //ro is the element that was clicked on. get ID
570 //create a div using this id appended to something
571 var mi = "rptBox" + ro.id;
572 var ii = 'rptBoxA' + ro.id;
573 var iid = 'rptBoxAData' + ro.id;
574 var oii = 'rptBoxB' + ro.id;
575 var oiid = 'rptBoxBData' + ro.id;
576 var rbtxt = "<span id='"+ mi +' ' > \
577         <div><a href='#' id='"+ii+"'
578         onClick='rptBoxClicked(this.id);this.blur();return false;'>bad answer?</a> \
579         <span id='"+iid+"Holder' style='display:none;'>\
580         If you think that our answer is wrong or missing an
581 emotion, let us know which emotion you think should go with this sentence.\
582         <select id='"+iid+"' onChange='rptValSent(this.id);'> \
583         <option selected>choose an emotion</option> \
584         <option value=happiness>happiness</option> \
585         <option value=sadness>sadness</option> \
586         <option value=surprise>surprise</option> \
587         <option value=anger>anger</option> \
588         <option value=fear>fear</option> \
589         <option value=disgust>disgust</option> \
590         <option value=nothing>nothing</option> \
591         </select> \
592         </span> \
593         </div> \
594         <div><a href='#' id='"+oii+"'
595         onClick='rptBoxClicked(this.id);this.blur();return false;'>bad sentence?</a> \
596         <input id='"+oiid+"' type=text style='display:none;'></input> \
597         </div> \
598         "+</span>";
599
600 //add the div to the page somehow (empty div/span)
601 var c = document.getElementById(ii);
602 //only if not already expanded
603 if(!c && ro.innerHTML.indexOf('report it!')>-1){
604     ro.innerHTML+=rbtxt;
605 }
606 //get the div object and move its position to the position fo the ro object (to make fit
607 with others and keep up, or to disappear?) [its already there]
608 }
609 function rptBoxClicked(d){
610 //take this id and see if it is a BoxA or B
611 var i = d.indexOf('rptBoxA');
612 if(i > -1){
613     // it was found so show the iid element by adding rptBoxAData.style.display='';
614     //alert('rptBoxAData'+d.substring(7, d.length));
615     document.getElementById('rptBoxAData'+d.substring(7)+'Holder').style.display='';
616 }else{
617     //wasnt found show the oiid element
618     //document.getElementById('rptBoxBData'+d.substring(7)).style.display='';
619     rptValSent(d);
620 }
621 }
622
623 function rptValSent(id){
624 //check if it is a sentence bad report OR
625 var a = id.indexOF('report');
626 var evt = events[id.substring(a+6)];
627 var i = id.indexOF('rptBoxA');
628 //check if its an emotion report
629 if(i > -1){
630     //if emotion then get value of that selected and the appropriate sentence send to
631 server/function the variables
632     var d = document.getElementById(id).value;
633     http.open("get", "/reportEvtError?evt="+evt+"&type=type1&data="+d, "true");
634     http.onreadystatechange = hdlReportVerification;
635     http.send(null);
636     alert('The following emotion('+d+') has been reported as being associated with
637 this event:(' + evt +)');
638 }else{
639     //its sentence, so send the bad sentence to the server (server should mark time/count
640 and flag)
641     http.open("get", "/reportEvtError?evt="+evt+"&type=type0", "true");
642     http.onreadystatechange = hdlReportVerification;
643     http.send(null);
644     alert('This sentence has been reported as bad: ' + evt);
645 }
646 //set innerHTML to be reported. Thank you.
647

```

```
648 var repBox = document.getElementById(id.substring(a));
649 repBox.innerHTML = 'Reported!';
650 }
651
652 function hdlReportVerification(){
653 var data = http.responseText;
654 }
```

+Game3.js

```

1  /* Misc utility functions not related to arch */
2  function show_eval(){
3      var eval_show = Math.round(Math.random()*2);
4      if( (eval_show == 1) && !(getCookie('feedback_form')) ){
5          $('eval_frame_div').style.display='';
6          $('eval_frame').src = '../getEvalPage';
7      }
8  }
9  function getKey(e){
10     if (window.event){
11         //alert(window.event.keyCode);
12         return window.event.keyCode;
13     }
14     else if (e){
15         //alert(e.which);
16         return e.which;
17     }
18     else{
19         return null;
20     }
21 }
22 function trim(str){
23     return str.replace(/^\s*|\s*$/g, "");
24 }
25 /* END Misc utility functions not related to arch */
26 function createRequestObject() {
27     var ro;
28     var browser = navigator.appName;
29     //netscape.security.PrivilegeManager.enablePrivilege('UniversalBrowserRead');
30     if(browser == "Microsoft Internet Explorer"){
31         ro = new ActiveXObject("Microsoft.XMLHTTP");
32     }else{
33         ro = new XMLHttpRequest();
34     }
35     return ro;
36 }
37 var http = createRequestObject();
38
39 function tryLogScore(newScore){
40     var oldScore = getCookie('G3HS');
41     //if newScore is greater than HiScore variable AND username var !=null
42     then send ajax message to log in new HighScore for user
43     if(newScore > oldScore){
44         //alert(newScore + "im in");
45         //del old cookie info, it will mess stuff up if you dont
46         deleteCookie( 'G3HS', null, null);
47         //set new info in Cookie, just updating this part or this cookie
48         setCookie( 'G3HS', newScore, 14, null, null, null);
49         var username = getCookie('usr');
50         //alert(username);
51         if(username){//only send if the cookie is there, meaning they
52         logged in or registered or have been here lately
53             //setUsrVars();
54             $('Game3HiScore').innerHTML = newScore;
55             http.open("get",
56             "/tryLogScore?u="+username+"&g=3&hs="+newScore, "true");
57             http.onreadystatechange = hdlTryLogScore;
58             http.send(null);
59         }
60     }
61     //else mada [and dont even handle it;]
62 }
63 function hdlTryLogScore(){
64     //literally do nothing
65 }
66
67 var TblHTMLNeedEndTag;
68 /*misc variables that need to be named*/
69 //things that need to be displayed after the game is over, this list is dynamic as
70 answers are given, or it is at least not inclusive
71 //var PCAnswerTblHdr;
72 //var OptionsTblHdr;
73 //var PlayAgainBtn;
74 //var QuitBtn;
75
76 var GameOver;
77 //things that need to be called from html and referred to in other fuctions
78 var timer_div;//referred to in starttimer function
79 var currentEventBox; //referred to in placeCurrEvt funtciton
80 var MTD; //referred ti i startGame
81 /*end misc vars to be named*/

```

```

//perhaps these should be listed by a number/code
var EmoChoiceList = "\
<ul><li> \
    <div id=0 onClick='recAns(this);'>yes</div>\
    <div id=1 onClick='recAns(this);'>no</div>\
</li> </ul> \
";

var PossibleEmotions = new Array();
PossibleEmotions.push('happiness');
PossibleEmotions.push('sadness');
PossibleEmotions.push('surprise');
PossibleEmotions.push('fear');
PossibleEmotions.push('anger');
PossibleEmotions.push('disgust');
//PossibleEmotions.push('nothing');

/*entry innher HTML should have constant width done in style, along with colors
var EvtCurrent = "\
    <div id='evtshowingtitle'>What emotion goes with the event below?</div> \
    <div id='evtshowing'></div> \
";*/

var GameEventLength;
var GameTimeLength;
var IgnoreInputs;
var GameWeight = 2;
var TimeWeight = 0.2;
var use_time = 1;

function startGame(emotouse){
//set the board/UI to look as it should at the start of the game
//set game time and event lengths based off of what kind of difficulty given in start
game
//GameEventLength = 3;
//GameTimeLength = 15;

if($('use_time_cb').checked){
    use_time = null;
}

//if passed in an emo to use, call place emotion with it, else call it anyway
placeEmotion(emotouse);
//also make emotion picking non-plausible anymore
//document.getElementById('pickemotiontype').style.visibility = 'hidden';
document.getElementById('StartArea').style.display = 'none';
//a variable that will hold the index number of the selected radio button
var theone
var dagdradz = $('egd', 'mgd', 'hgd');
for (var i=0;i<dagdradz.length;i++){
    if (dagdradz[i].checked==true){
        theone=i;
        break; //exist for loop, as target acquired.
    }
}
switch (dagdradz[theone].value){
    case "Easy":
        GameEventLength = 5;
        GameTimeLength = 85;
        break;

    case "Medium":
        GameEventLength = 10;
        GameTimeLength = 160;
        break;

    case "Hard":
        GameEventLength = 15;
        GameTimeLength = 230;
        break;

    default :
        GameEventLength = 5;
        GameTimeLength = 85;
}

timer_div = document.getElementById('showtimeBody'); //referred to in starttimer function
currentEventBox = document.getElementById('eventtoenterinput'); //referred to in
placeCurrEvt funtciton

```

```

63 //start timer
64 if(use_time){
65     startTimer(GameTimeLength);
66 }
67
68 //get event-emotion from DB(or X event-emotions at once) and display 1 in currentEvent
69 box:
70     //reqXEvtEmo(GameEventLength);
71
72 //+get and display number of elements to go in left/togo status boxes:
73 updateEvtStatii(0, GameEventLength);
74
75
76
77
78
79 // set the main table...only done in start game
80 //remember to append an END "</table>" tag at the end of this variable AND to depend when
81 modifying the table itself to make it work
82 TblHTMLNeedEndTag = "<table border='1px'><th>Your Sentence</th> <th>Emotion To
83 Match</th><th><div id=PCATH style='visibility:visible;'>A Match for
84 '"+PickedEmotionForMatching+"?</div></th> <th><div id=OTH
85 style='visibility:visible;'>Report This</div></th>";
86 MTD = document.getElementById('maintabledivBody');
87 MTD.innerHTML = TblHTMLNeedEndTag + "</table>";
88 //set input box visible
89 document.getElementById('eventtoenterbox').style.display = '';
90 /**/set the event field to visible
91 var es = document.getElementById('currentevent');
92 es.innerHTML = EvtCurrent;
93 */
94
95 //clear or reset any client side elements needed
96 currentEventBox.disabled=0;
97 currentEventBox.value = "";
98 currentEventBox.focus(); //give it focus
99 document.getElementById('emotionselector').selectedIndex = 0;
100 }
101
102 var PickedEmotionForMatching;
103 function placeEmotion(a){
104 //if emotion passed in set that, else set random from some home variable
105 var daEmotion;
106 if(a){
107     daEmotion = a;
108 }
109 else{
110     var randnum = Math.floor(Math.random()*(PossibleEmotions.length));
111     daEmotion = PossibleEmotions[randnum];
112 }
113 PickedEmotionForMatching = daEmotion;
114 document.getElementById('emotiontomatchBody').innerHTML = daEmotion;
115 }
116 /*
117 function reqXEvtEmo(x){
118 //+send request for x events-emo pairs
119 //disable (invisible) ability to pick using emotio options (set grand variable) ORnot,
120 depending on
121     document.getElementById('currentyesnooptions').display='none';
122 //must specify what to do when request comesback
123     var temp = "I get up[s]happiness[b]I go to sleep[s]no emotion[b]I get
124 shot[s]scared";
125 hdlIncEvtEmo(temp);
126 }
127 */
128
129 var events = new Array();
130 var pc_ans = new Array();
131 var usr_ans = new Array();
132 var reports = new Array();
133 //var global_count = 0;
134
135 //+handle event-emotion return from DB (ajax comeback)
136 /**function hdlIncEvtEmo(data){
137     // from evt-emo gotten back, place it and store it along with its answer in a twin
138 array OR a holder variable of currentRightEmotion
139     var tmpB = data.split('[b]');
140     for(var i=0; i<tmpB.length; i++){
141         var tmpS = tmpB[i].split('[s]');
142         var a =tmpS[0];
143         var b = tmpS[1];

```

```

244         events.push(a);
245         pc_ans.push(b);
246         //usr_ans
247         //reports
248     };
249     //place current event
250     placeCurrEvt(events[0]);
251     //enable (visible) inputs for emotion options OR not since all comes back once
252     document.getElementById('currentyesnooptions').display='';
253 }
254 */
255
256 //+given a sentence, it will display it in the current event box
257 function placeCurrEvt(d){
258     //set the inner HTML of the current Eventbox to the event
259     //es.innerHTML = events[global_count];
260     //alert(currentEventBox);
261     currentEventBox.innerHTML = d;
262     //global_count++;
263 }
264
265
266 var togonum;
267 var donenum;
268
269 //display number of lements to go/lef tin status boxes
270 function updateEvtStatii(dir, total){
271     /*-set innerHTML of both to reasonable values
272     -check to see if innerHTML of togo = 0 do something (call show answer method)..or maybe
273     not done here*/
274     //if dir = 0 then set done=0 and togo=total
275     if(dir ==0){
276         togonum = total;
277         donenum = 0;
278     }
279     //if dir = 1 then inc done and dec togo
280     if(dir ==1){
281         if(togonum != 0){
282             togonum--;
283             donenum++;
284         }
285     }
286     //if dir = -1 then dec done and in togo
287     if(dir ==-1){
288         if(donenum != GameEventLength){
289             togonum++;
290             donenum--;
291         }else{
292             //should we stop the game, or should they be allowed to still remove and
293             redo events and click a done button
294         }
295     }
296 }
297
298 var a = document.getElementById('itemstogoBody');
299 var b = document.getElementById('itemsdoneBody');
300 a.innerHTML = togonum;
301 b.innerHTML = donenum;
302
303 }
304
305
306 var timeLeft;
307 //starts the game timer. call function with X seconds that updated the timer element
308 innerHTML
309 function startTimer(x){
310     //-when timer=0 AND not finished do someting (showanswers)
311     //if certain variable set STOP timer countdown
312     //if var passed in is NOT null
313     if(x){
314         // set timeLeft to value passed in
315         timeLeft = x;
316         timer_div.innerHTML = ":" + timeLeft;
317         window.setTimeout('startTimer(null)',1000)
318     }else if(timeLeft && !GameOver){
319         timeLeft = timeLeft - 1;
320         timer_div.innerHTML = ":" + timeLeft;
321         if(timeLeft == 10){
322             timer_div.style.color = 'red';
323         }
324         //if timeleft is greater than 0 then call self again
325         if(timeLeft > 0){

```

```

32 //alert(timeLeft);
33 window.setTimeout('startTimer(null)',1000)
34 }else{
35     stopGame();
36 }
37 }else{
38     //nothing game over
39 }
40 }
41
42 /*-record/accept answer for event input chosen by user
43     -put in a table (evt and report)
44     -put in an array (evt and report)
45 -decord accept answer
46     -pop from evt array
47     -pop from report array
48     -remove last tr from table
49
50 */
51 function recAns(pe){
52 //trim text in box and only do something if they entered something (an actual char.)
53 var inctxt = escape(trim(pe)).replace(/\%20/g, ' ');
54 if(inctxt.length > 0){
55     if(donenum != GameEventLength){
56         if(inctxt.indexOf(PickedEmotionForMatching)>-1 ||
57         usr_ans.inArray(inctxt)){//check if contains daEmotion word and if its inARRAY of usr_cns
58             var ws = $('warningSpan');
59
60             //write into span warning class "you may not use the word
61             ["+PickedEmotionForMatching+" in your sentence."
62             if(inctxt.indexOf(PickedEmotionForMatching) > -1){
63                 ws.innerHTML = "you may not use the word
64             ["+PickedEmotionForMatching+" in your sentence.";
65             }else{
66
67                 //write into span: "[you may not use the same sentence twice in
68                 matching this emotion.]"
69                 ws.innerHTML = "you may not use the same sentence twice in matching
70                 this emotion.";
71             }
72             //make span visible
73             currentEventBox.value =pe;
74             currentEventBox.select();
75             ws.style.display = '';
76             return;//exit, dont do anything if this if catches. except update ui
77         }
78         //make span invisible, just in case [display noine] if it made it this far
79         $('warningSpan').style.display = 'none';
80
81         //store incoming txt, since its an answer
82         usr_ans.push(inctxt);
83
84         //update any UI stuff needed to certify click (disable, darken, outline,
85         sleep for few mseconds, enable, lighten, no outline, GO OR just do it onMousePress)
86
87         //get the current event text in the currevent box
88         /*var evtxt = document.getElementById('eventtojudge').innerHTML;
89
90         var pc_yes_or_no_txt;
91
92         //determine if OUR answer is yes or no if matchemotion contained in
93         pcmotion-> yes, else no
94         if(pc_ans[donenum].indexOf(PickedEmotionForMatching) != -1){
95             pc_yes_or_no_txt = 'yes';
96         }else{
97             pc_yes_or_no_txt = 'no';
98         }
99     }
100 */
101     //place the current event given, correctanswerholder, report/flagholder in
102     an array and in the maintable
103     var apcans = "<div id='"+ pcans'+String(donenum)+ "'
104     style='display:none;'>"+i dunno answer yet'+"/div>";
105
106     var areport = "<div id='"+ 'report'+String(donenum)+ "'
107     style='display:none;' onClick='repClicked(this);this.blur();return false;'>"+<a
108     href='javascript:void(0)'>report it!</a>"+"/div>";
109     reports.push('report'+String(donenum));
110
111     var arow = "\
112     <tr> \
113     <td>"+ unescape(inctxt) +" \
114     </td>\

```



```

4006 <td>"+ PickedEmotionForMatching +"\  

4007 </td>\  

4008 <td>"+ apcans +"\  

4009 </td>\  

4010 <td>"+ areport +"\  

4011 </td>\  

4012 </tr> \  

4013 "  

4014 //addedRows.push(arow);  

4015 //addedRows.push(arow);  

4016 var a = MTD.innerHTML;  

4017 var bFFX = a.lastIndexOf('</th>');  

4018 var bIE = a.lastIndexOf('</TH>');  

4019 var b = Math.max(bFFX, bIE);  

4020 var top = a.substring(0, b + 10);  

4021 var bot = a.substring(b+10, a.length);  

4022 //alert(top);  

4023 //alert(bot);  

4024 MTD.innerHTML = top + arow + bot;  

4025 "  

4026 /*var tmp = MTD.innerHTML;  

4027 //alert(tmp);  

4028 //tmp = tmp.substring(0, tmp.indexOf('</table>')); //remove </table> tag  

4029 var tmpIE = tmp.substring(0, tmp.indexOf('</TABLE>'));  

4030 var tmpFFX = tmp.substring(0, tmp.indexOf('</table>'));  

4031 if(tmpIE){  

4032 tmp = tmpIE;  

4033 }else{  

4034 tmp = tmpFFX;  

4035 }  

4036 "  

4037 tmp = tmp + arow; //add new arow  

4038 MTD.innerHTML = tmp + '</table>'//add closed table tag  

4039 */  

4040 "  

4041 //update eventStatii to increasase done and decrease togo  

4042 updateEvtStatii(1, null);  

4043 "  

4044 //clear the input box to allow another add  

4045 currentEventBox.value="";  

4046 //currentEventBox.focus();  

4047 document.getElementById('backupbutton').style.visibility='visible';  

4048 if(donenum == GameEventLength){  

4049 stopGame();  

4050 }else{  

4051 currentEventBox.focus();  

4052 }  

4053 }  

4054 }  

4055 }  

4056 }  

4057 }  

4058 }  

4059 }  

4060 }  

4061 }  

4062 }  

4063 }  

4064 }  

4065 }  

4066 }  

4067 }  

4068 }  

4069 }  

4070 }  

4071 }  

4072 }  

4073 }  

4074 }  

4075 }  

4076 }  

4077 }  

4078 }  

4079 }  

4080 }  

4081 }  

4082 }  

4083 }  

4084 }  

4085 }  

4086 }  

4087 }  

4088 }  

4089 }  

4090 }  

4091 }  

4092 }  

4093 }  

4094 }  

4095 }  

4096 }  

4097 }  

4098 }  

4099 }  

4100 }  

4101 }  

4102 }  

4103 }  

4104 }  

4105 }  

4106 }  

4107 }  

4108 }  

4109 }  

4110 }  

4111 }  

4112 }  

4113 }  

4114 }  

4115 }  

4116 }  

4117 }  

4118 }  

4119 }  

4120 }  

4121 }  

4122 }  

4123 }  

4124 }  

4125 }  

4126 }  

4127 }  

4128 }  

4129 }  

4130 }  

4131 }  

4132 }  

4133 }  

4134 }  

4135 }  

4136 }  

4137 }  

4138 }  

4139 }  

4140 }  

4141 }  

4142 }  

4143 }  

4144 }  

4145 }  

4146 }  

4147 }  

4148 }  

4149 }  

4150 }  

4151 }  

4152 }  

4153 }  

4154 }  

4155 }  

4156 }  

4157 }  

4158 }  

4159 }  

4160 }  

4161 }  

4162 }  

4163 }  

4164 }  

4165 }  

4166 }  

4167 }  

4168 }  

4169 }  

4170 }  

4171 }  

4172 }  

4173 }  

4174 }  

4175 }  

4176 }  

4177 }  

4178 }  

4179 }  

4180 }  

4181 }  

4182 }  

4183 }  

4184 }  

4185 }  

4186 }  

4187 }  

4188 }  

4189 }  

4190 }  

4191 }  

4192 }  

4193 }  

4194 }  

4195 }  

4196 }  

4197 }  

4198 }  

4199 }  

4200 }  

4201 }  

4202 }  

4203 }  

4204 }  

4205 }  

4206 }  

4207 }  

4208 }  

4209 }  

4210 }  

4211 }  

4212 }  

4213 }  

4214 }  

4215 }  

4216 }  

4217 }  

4218 }  

4219 }  

4220 }  

4221 }  

4222 }  

4223 }  

4224 }  

4225 }  

4226 }  

4227 }  

4228 }  

4229 }  

4230 }  

4231 }  

4232 }  

4233 }  

4234 }  

4235 }  

4236 }  

4237 }  

4238 }  

4239 }  

4240 }  

4241 }  

4242 }  

4243 }  

4244 }  

4245 }  

4246 }  

4247 }  

4248 }  

4249 }  

4250 }  

4251 }  

4252 }  

4253 }  

4254 }  

4255 }  

4256 }  

4257 }  

4258 }  

4259 }  

4260 }  

4261 }  

4262 }  

4263 }  

4264 }  

4265 }  

4266 }  

4267 }  

4268 }  

4269 }  

4270 }  

4271 }  

4272 }  

4273 }  

4274 }  

4275 }  

4276 }  

4277 }  

4278 }  

4279 }  

4280 }  

4281 }  

4282 }  

4283 }  

4284 }  

4285 }  

4286 }  

4287 }  

4288 }  

4289 }  

4290 }  

4291 }  

4292 }  

4293 }  

4294 }  

4295 }  

4296 }  

4297 }  

4298 }  

4299 }  

4300 }  

4301 }  

4302 }  

4303 }  

4304 }  

4305 }  

4306 }  

4307 }  

4308 }  

4309 }  

4310 }  

4311 }  

4312 }  

4313 }  

4314 }  

4315 }  

4316 }  

4317 }  

4318 }  

4319 }  

4320 }  

4321 }  

4322 }  

4323 }  

4324 }  

4325 }  

4326 }  

4327 }  

4328 }  

4329 }  

4330 }  

4331 }  

4332 }  

4333 }  

4334 }  

4335 }  

4336 }  

4337 }  

4338 }  

4339 }  

4340 }  

4341 }  

4342 }  

4343 }  

4344 }  

4345 }  

4346 }  

4347 }  

4348 }  

4349 }  

4350 }  

4351 }  

4352 }  

4353 }  

4354 }  

4355 }  

4356 }  

4357 }  

4358 }  

4359 }  

4360 }  

4361 }  

4362 }  

4363 }  

4364 }  

4365 }  

4366 }  

4367 }  

4368 }  

4369 }  

4370 }  

4371 }  

4372 }  

4373 }  

4374 }  

4375 }  

4376 }  

4377 }  

4378 }  

4379 }  

4380 }  

4381 }  

4382 }  

4383 }  

4384 }  

4385 }  

4386 }  

4387 }  

4388 }  

4389 }  

4390 }  

4391 }  

4392 }  

4393 }  

4394 }  

4395 }  

4396 }  

4397 }  

4398 }  

4399 }  

4400 }  

4401 }  

4402 }  

4403 }  

4404 }  

4405 }  

4406 }  

4407 }  

4408 }  

4409 }  

4410 }  

4411 }  

4412 }  

4413 }  

4414 }  

4415 }  

4416 }  

4417 }  

4418 }  

4419 }  

4420 }  

4421 }  

4422 }  

4423 }  

4424 }  

4425 }  

4426 }  

4427 }  

4428 }  

4429 }  

4430 }  

4431 }  

4432 }  

4433 }  

4434 }  

4435 }  

4436 }  

4437 }  

4438 }  

4439 }  

4440 }  

4441 }  

4442 }  

4443 }  

4444 }  

4445 }  

4446 }  

4447 }  

4448 }  

4449 }  

4450 }  

4451 }  

4452 }  

4453 }  

4454 }  

4455 }  

4456 }  

4457 }  

4458 }  

4459 }  

4460 }  

4461 }  

4462 }  

4463 }  

4464 }  

4465 }  

4466 }  

4467 }  

4468 }  

4469 }  

4470 }  

4471 }  

4472 }  

4473 }  

4474 }  

4475 }  

4476 }  

4477 }  

4478 }  

4479 }  

4480 }  

4481 }  

4482 }  

4483 }  

4484 }  

4485 }  

4486 }  

4487 }  

4488 }  

4489 }  

4490 }  

4491 }  

4492 }  

4493 }  

4494 }  

4495 }  

4496 }  

4497 }  

4498 }  

4499 }  

4500 }  

4501 }  

4502 }  

4503 }  

4504 }  

4505 }  

4506 }  

4507 }  

4508 }  

4509 }  

4510 }  

4511 }  

4512 }  

4513 }  

4514 }  

4515 }  

4516 }  

4517 }  

4518 }  

4519 }  

4520 }  

4521 }  

4522 }  

4523 }  

4524 }  

4525 }  

4526 }  

4527 }  

4528 }  

4529 }  

4530 }  

4531 }  

4532 }  

4533 }  

4534 }  

4535 }  

4536 }  

4537 }  

4538 }  

4539 }  

4540 }  

4541 }  

4542 }  

4543 }  

4544 }  

4545 }  

4546 }  

4547 }  

4548 }  

4549 }  

4550 }  

4551 }  

4552 }  

4553 }  

4554 }  

4555 }  

4556 }  

4557 }  

4558 }  

4559 }  

4560 }  

4561 }  

4562 }  

4563 }  

4564 }  

4565 }  

4566 }  

4567 }  

4568 }  

4569 }  

4570 }  

4571 }  

4572 }  

4573 }  

4574 }  

4575 }  

4576 }  

4577 }  

4578 }  

4579 }  

4580 }  

4581 }  

4582 }  

4583 }  

4584 }  

4585 }  

4586 }  

4587 }  

4588 }  

4589 }  

4590 }  

4591 }  

4592 }  

4593 }  

4594 }  

4595 }  

4596 }  

4597 }  

4598 }  

4599 }  

4600 }  

4601 }  

4602 }  

4603 }  

4604 }  

4605 }  

4606 }  

4607 }  

4608 }  

4609 }  

4610 }  

4611 }  

4612 }  

4613 }  

4614 }  

4615 }  

4616 }  

4617 }  

4618 }  

4619 }  

4620 }  

4621 }  

4622 }  

4623 }  

4624 }  

4625 }  

4626 }  

4627 }  

4628 }  

4629 }  

4630 }  

4631 }  

4632 }  

4633 }  

4634 }  

4635 }  

4636 }  

4637 }  

4638 }  

4639 }  

4640 }  

4641 }  

4642 }  

4643 }  

4644 }  

4645 }  

4646 }  

4647 }  

4648 }  

4649 }  

4650 }  

4651 }  

4652 }  

4653 }  

4654 }  

4655 }  

4656 }  

4657 }  

4658 }  

4659 }  

4660 }  

4661 }  

4662 }  

4663 }  

4664 }  

4665 }  

4666 }  

4667 }  

4668 }  

4669 }  

4670 }  

4671 }  

4672 }  

4673 }  

4674 }  

4675 }  

4676 }  

4677 }  

4678 }  

4679 }  

4680 }  

4681 }  

4682 }  

4683 }  

4684 }  

4685 }  

4686 }  

4687 }  

4688 }  

4689 }  

4690 }  

4691 }  

4692 }  

4693 }  

4694 }  

4695 }  

4696 }  

4697 }  

4698 }  

4699 }  

4700 }  

4701 }  

4702 }  

4703 }  

4704 }  

4705 }  

4706 }  

4707 }  

4708 }  

4709 }  

4710 }  

4711 }  

4712 }  

4713 }  

4714 }  

4715 }  

4716 }  

4717 }  

4718 }  

4719 }  

4720 }  

4721 }  

4722 }  

4723 }  

4724 }  

4725 }  

4726 }  

4727 }  

4728 }  

4729 }  

4730 }  

4731 }  

4732 }  

4733 }  

4734 }  

4735 }  

4736 }  

4737 }  

4738 }  

4739 }  

4740 }  

4741 }  

4742 }  

4743 }  

4744 }  

4745 }  

4746 }  

4747 }  

4748 }  

4749 }  

4750 }  

4751 }  

4752 }  

4753 }  

4754 }  

4755 }  

4756 }  

4757 }  

4758 }  

4759 }  

4760 }  

4761 }  

4762 }  

4763 }  

4764 }  

4765 }  

4766 }  

4767 }  

4768 }  

4769 }  

4770 }  

4771 }  

4772 }  

4773 }  

4774 }  

4775 }  

4776 }  

4777 }  

4778 }  

4779 }  

4780 }  

4781 }  

4782 }  

4783 }  

4784 }  

4785 }  

4786 }  

4787 }  

4788 }  

4789 }  

4790 }  

4791 }  

4792 }  

4793 }  

4794 }  

4795 }  

4796 }  

4797 }  

4798 }  

4799 }  

4800 }  

4801 }  

4802 }  

4803 }  

4804 }  

4805 }  

4806 }  

4807 }  

4808 }  

4809 }  

4810 }  

4811 }  

4812 }  

4813 }  

4814 }  

4815 }  

4816 }  

4817 }  

4818 }  

4819 }  

4820 }  

4821 }  

4822 }  

4823 }  

4824 }  

4825 }  

4826 }  

4827 }  

4828 }  

4829 }  

4830 }  

4831 }  

4832 }  

4833 }  

4834 }  

4835 }  

4836 }  

4837 }  

4838 }  

4839 }  

4840 }  

4841 }  

4842 }  

4843 }  

4844 }  

4845 }  

4846 }  

4847 }  

4848 }  

4849 }  

4850 }  

4851 }  

4852 }  

4853 }  

4854 }  

4855 }  

4856 }  

4857 }  

4858 }  

4859 }  

4860 }  

4861 }  

4862 }  

4863 }  

4864 }  

4865 }  

4866 }  

4867 }  

4868 }  

4869 }  

4870 }  

4871 }  

4872 }  

4873 }  

4874 }  

4875 }  

4876 }  

4877 }  

4878 }  

4879 }  

4880 }  

4881 }  

4882 }  

4883 }  

4884 }  

4885 }  

4886 }  

4887 }  

4888 }  

4889 }  

4890 }  

4891 }  

4892 }  

4893 }  

4894 }  

4895 }  

4896 }  

4897 }  

4898 }  

4899 }  

4900 }  

4901 }  

4902 }  

4903 }  

4904 }  

4905 }  

4906 }  

4907 }  

4908 }  

4909 }  

4910 }  

4911 }  

4912 }  

4913 }  

4914 }  

4915 }  

4916 }  

4917 }  

4918 }  

4919 }  

4920 }  

4921 }  

4922 }  

4923 }  

4924 }  

4925 }  

4926 }  

4927 }  

4928 }  

4929 }  

4930 }  

4931 }  

4932 }  

4933 }  

4934 }  

4935 }  

4936 }  

4937 }  

4938 }  

4939 }  

4940 }  

4941 }  

4942 }  

4943 }  

4944 }  

4945 }  

4946 }  

4947 }  

4948 }  

4949 }  

4950 }  

4951 }  

4952 }  

4953 }  

4954 }  

4955 }  

4956 }  

4957 }  

4958 }  

4959 }  

4960 }  

4961 }  

4962 }  

4963 }  

4964 }  

4965 }  

4966 }  

4967 }  

4968 }  

4969 }  

4970 }  

4971 }  

4972 }  

4973 }  

4974 }  

4975 }  

4976 }  

4977 }  

4978 }  

4979 }  

4980 }  

4981 }  

4982 }  

4983 }  

4984 }  

4985 }  

4986 }  

4987 }  

4988 }  

4989 }  

4990 }  

4991 }  

4992 }  

4993 }  

4994 }  

4995 }  

4996 }  

4997 }  

4998 }  

4999 }  

5000 }  

5001 }  

5002 }  

5003 }  

5004 }  

5005 }  

5006 }  

5007 }  

5008 }  

5009 }  

5010 }  

5011 }  

5012 }  

5013 }  

5014 }  

5015 }  

5016 }  

5017 }  

5018 }  

5019 }  

5020 }  

5021 }  

5022 }  

5023 }  

5024 }  

5025 }  

5026 }  

5027 }  

5028 }  

5029 }  

5030 }  

5031 }  

5032 }  

5033 }  

5034 }  

5035 }  

5036 }  

5037 }  

5038 }  

5039 }  

5040 }  

5041 }  

5042 }  

5043 }  

5044 }  

5045 }  

5046 }  

5047 }  

5048 }  

5049 }  

5050 }  

5051 }  

5052 }  

5053 }  

5054 }  

5055 }  

5056 }  

5057 }  

5058 }  

5059 }  

5060 }  

5061 }  

5062 }  

5063 }  

5064 }  

5065 }  

5066 }  

5067 }  

5068 }  

5069 }  

5070 }  

5071 }  

5072 }  

5073 }  

5074 }  

5075 }  

5076 }  

5077 }  

5078 }  

5079 }  

5080 }  

5081 }  

5082 }  

5083 }  

5084 }  

5085 }  

5086 }  

5087 }  

5088 }  

5089 }  

5090 }  

5091 }  

5092 }  

5093 }  

5094 }  

5095 }  

5096 }  

5097 }  

5098 }  

5099 }  

5100 }  

5101 }  

5102 }  

5103 }  

5104 }  

5105 }  

5106 }  

5107 }  

5108 }  

5109 }  

5110 }  

5111 }  

5112 }  

5113 }  

5114 }  

5115 }  

5116 }  

5117 }  

5118 }  

5119 }  

5120 }  

5121 }  

5122 }  

5123 }  

5124 }  

5125 }  

5126 }  

5127 }  

5128 }  

5129 }  

5130 }  

5131 }  

5132 }  

5133 }  

5134 }  

5135 }  

5136 }  

5137 }  

5138 }  

5139 }  

5140 }  

5141 }  

5142 }  

5143 }  

5144 }  

5145 }  

5146 }  

5147 }  

5148 }  

5149 }  

5150 }  

5151 }  

5152 }  

5153 }  

5154 }  

5155 }  

5156 }  

5157 }  

5158 }  

5159 }  

5160 }  

5161 }  

5162 }  

5163 }  

5164 }  

5165 }  

5166 }  

5167 }  

5168 }  

5169 }  

5170 }  

5171 }  

5172 }  

5173 }  

5174 }  

5175 }  

5176 }  

5177 }  

5178 }  

5179 }  

5180 }  

5181 }  

5182 }  

5183 }  

5184 }  

5185 }  

5186 }  

5187 }  

5188 }  

5189 }  

5190 }  

5191 }  

5192 }  

5193 }  

5194 }  

5195 }  

5196 }  

5197 }  

5198 }  

5199 }  

5200 }  

5201 }  

5202 }  

5203 }  

5204 }  

5205 }  

5206 }  

5207 }  

5208 }  

5209 }  

5210 }  

5211 }  

5212 }  

5213 }  

5214 }  

5215 }  

5216 }  

5217 }  

5218 }  

5219 }  

5220 }  

5221 }  

5222 }  

5223 }  

5224 }  

5225 }  

5226 }  

5227 }  

5228 }  

5229 }  

5230 }  

5231 }  

5232 }  

5233 }  

5234 }  

5235 }  

5236 }  

5237 }  

5238 }  

5239 }  

5240 }  

5241 }  

5242 }  

5243 }  

5244 }  

5245 }  

5246 }  

5247 }  

5248 }  

5249 }  

5250 }  

5251 }  

5252 }  

5253 }  

5254 }  

5255 }  

5256 }  

5257 }  

5258 }  

5259 }  

5260 }  

5261 }  

5262 }  

5263 }  

5264 }  

5265 }  

5266 }  

5267 }  

5268 }  

5269 }  

5270 }  

5271 }  

5272 }  

5273 }  

5274 }  

5275 }  

5276 }  

5277 }  

5278 }  

5279 }  

5280 }  

5281 }  

5282 }  

5283 }  

5284 }  

5285 }  

5286 }  

5287 }  

5288 }  

5289 }  

5290 }  

5291 }  

5292 }  

5293 }  

5294 }  

5295 }  

5296 }  

5297 }  

5298 }  

5299 }  

5300 }  

5301 }  

5302 }  

5303 }  

5304 }  

5305 }  

5306 }  

5307 }  

5308 }  

5309 }  

5310 }  

5311 }  

5312 }  

5313 }  

5314 }  

5315 }  

5316 }  

5317 }  

5318 }  

5319 }  

5320 }  

5321 }  

5322 }  

5323 }  

5324 }  

5325 }  

5326 }  

5327 }  

5328 }  

5329 }  

5330 }  

5331 }  

5332 }  

5333 }  

5334 }  

5335 }  

5336 }  

5337 }  

5338 }  

5339 }  

5340 }  

5341 }  

5342 }  

5343 }  

5
```



```

568 //REMOVE Progress UI thing
569 document.getElementById('GameArea').style.visibility = 'visible';
570 document.getElementById('loadScreen').style.visibility = 'hidden';
571
572 var holdRows = "";
573 var corr = 0;
574 var correct_usr_ans_idnums = new Array();
575 var wrong_usr_ans_idnums = new Array();
576 for(var i=0; i < usr_ans.length; i++){
577     //find out if the pc ans matches th current emotion (yes or no)
578     var aans = pc_ans_array[i];
579     var ayn;
580     var rightOrWrong;
581     if(aans.indexOf(PickedEmotionForMatching) > -1){
582         ayn = 'yes -->' + ' ('+aans+')';
583         correct_usr_ans_idnums.push(i); //go thru this list later and
584 append nums to familiar idnames to change their style
585         rightOrWrong = 'right';
586         corr++;
587     }else{
588         ayn = 'no -->' + ' ('+aans+')';
589         rightOrWrong = 'wrong';
590         wrong_usr_ans_idnums.push(i); //go thru this list later and append
591 nums to familiar idnames to change their style
592     }
593     //place the current event given, correctanswerholder, report/flagholder in
594 an array and in the maintable
595     var apcans = "<div id='"+ "pcans"+i+ "'
596 class='"+rightOrWrong+" '"+ayn+"</div>";
597
598     var areport = "<div id='"+ 'report'+i+ "'
599 onClick='repClicked(this);this.blur();this.blur();return false;'>"+<a
600 href='javascript:void(0)' id='"+ 'reportLinkreport'+String(i)+ "'>report
601 it!</a>"+</div>";
602     //reports.push('report'+String(donenum));
603
604     var ausrans = usr_ans[i];
605     var arow = "\
606 <tr> \
607 <td>"+ unescape(ausrans) +"\
608 </td>\
609 <td>"+ PickedEmotionForMatching +"\
610 </td>\
611 <td>"+ apcans +"\
612 </td>\
613 <td>"+ areport +"\
614 </td>\
615 </tr> \
616 "
617     //addedRows.push(arow);
618
619     holdRows+= arow;
620 }
621 //reset that table :)
622 TblHTMLNeedEndTag = "<table border='1px'><th>Your Sentence</th> <th>Emotion To
623 Match</th><th><div id=PCATH>A Match for '"+PickedEmotionForMatching+"'?</div></th>
624 <th><div id=OTH>Report This</div></th>";
625 MTD.innerHTML = TblHTMLNeedEndTag + holdRows + '</table>'//add closed table tag
626
627
628 //hide certain input elemetns (emotion list, event read)
629 document.getElementById('eventtoenterbox').style.visibility = 'hidden';
630
631 //stop the clock
632 GameOver = 1;
633 var tp = (use_time) ? timeLeft : 0;
634
635 tp = Math.round(tp * TimeWeight*10)/10;
636 //Math.round()
637 //alert(corr*GameWeight);
638 var TS = Math.round((tp+corr*GameWeight)*10)/10;
639 tryLogScore(TS);
640 //tp++;
641 //alert(tp);
642 var sb = document.getElementById('scoreboxBody');
643 sb.innerHTML = "\
644
645 <p>#Correct: (" +corr+)<br>\
646 Game-type factor: (" +GameWeight+)<br> \
647 Time bonus: (" +tp+)<br> \
648 (#Correct * Game-type factor) + TimeBonus = Total</p>\

```

```

648         <p>You received a score of <span
649 id=scoreBoxActualScore>["+TS+"]</span> points!</p>\
650         ";
651
652 //reveal hidden buttons that should be shown now (quit,play next-round/again, my-alltime-
653 high-score) : specify what should happen for each button first
654 /*document.getElementById('playagainbutton').style.display = '';
655 document.getElementById('quitbutton').style.display = '';
656 */
657 $('EndGameButtonArea').style.display = '';
658 show_eval();
659 }
660
661 //reset UI and variables, so that its like a page refresh
662 function playAgain(){
663 //can we just reload the page at this point? only if not ndoing roundgs. buy client side
664 elements retian state (unless you explicitly reset them here
665 //or you can change those elements (like selectores, to be div's or ul's)
666
667
668 }
669 /*
670 //-get score and display score for this round
671 function doScoring(){
672 //+calculate a score based on time left and on number events correct
673 //+how many were correct? have answers given in stored arrays and run eventarrays thru
674 scorer and see if match
675 var total=0;
676 for(var i=0; i<usr_ans.length; i++){
677     var b = usr_ans[i];
678     var a = pc_ans[1];
679     if(a == b){
680         total++
681     }
682 }
683
684 }
685 return total;
686 //for each our answer check same stored your answer and see if it matches, if so inc a
687 global score (make sure it was zero before doing it)
688 //get current time left in the timeleft thing (or from a variable)
689 //correct * timeleft = score
690 //set score box to reflect score
691 }
692
693 */
694
695
696 function repClicked(ro){
697 /**/ro is the element that was clicked on. get ID
698 //create a div using this id appended to something
699 var mi = "rptBox" + ro.id;
700 var ii = 'rptBoxA' + ro.id;
701 var iid = 'rptBoxAData' + ro.id;
702 var oii = 'rptBoxB' + ro.id;
703 var oiid = 'rptBoxBData' + ro.id;
704 var rbtxt = "<span id='"+ mi +" ' > \
705         <div id='"+ii+"' onClick='rptBoxClicked(this.id);'>bad
706 answer?</div> \
707         <select id='"+iid+"' style='display:none;'
708 onChange='rptValSent(this.id);'> \
709         <option selected>If you are sure that our answer is wrong,
710 choose the correct emotion for this event</option> \
711         <option>happiness</option> \
712         <option>sadness</option> \
713         <option>surprise</option> \
714         <option>disgust</option> \
715         <option>fear</option> \
716         <option>anger</option> \
717         </select> \
718         <div id='"+oii+"' onClick='rptBoxClicked(this.id);'>bad
719 sentence?</div> \
720         <input id='"+oiid+"' type=text style='display:none;'></input> \
721         "+ "</span>";
722
723 */
724 //get the ahref "button" link and change color of it
725 var repLinkBox = document.getElementById("reportLink" + ro.id);
726 repLinkBox.style.background='white';
727 //ro is the element that was clicked on. get ID
728 //create a div using this id appended to something
729 var mi = "rptBox" + ro.id;

```

```

729 var ii = 'rptBoxA' + ro.id;
730 var iid = 'rptBoxAData' + ro.id;
731 var oii = 'rptBoxB' + ro.id;
732 var oiid = 'rptBoxBData' + ro.id;
733 var rbxtxt = "<span id='"+ mi +" ' > \
734         <div><a href='#' id='"+ii+"'
735 onClick='rptBoxClicked(this.id);this.blur();return false;'>bad answer?</a> \
736         <span id='"+iid+"'Holder' style='display:none;'>\
737         If you think that our answer is wrong or missing an
738 emotion, let us know which emotion you think should go with this sentence.\
739         <select id='"+iid+"' onChange='rptValSent(this.id);'> \
740         <option selected>choose an emotion</option> \
741         <option value=happiness>happiness</option> \
742         <option value=sadness>sadness</option> \
743         <option value=surprise>surprise</option> \
744         <option value=anger>anger</option> \
745         <option value=fear>fear</option> \
746         <option value=disgust>disgust</option> \
747         <option value=nothing>nothing</option> \
748         </select> \
749         </span> \
750         </div> \
751         <div><a href='#' id='"+oii+"' style='display:none;'
752 onClick='rptBoxClicked(this.id);this.blur();return false;'>bad sentence?</a> \
753         <input id='"+oiid+"' type=text style='display:none;'></input> \
754         </div> \
755         "+</span>";
756
757 //add the div to the page somehow (empty div/span)
758 var c = document.getElementById(ii);
759 //only if not already expanded
760 if(!c && ro.innerHTML.indexOf('report it!')>-1){
761     ro.innerHTML+=rbxtxt;
762 }
763 //get the div object and move its position to the position fo the ro object (to make fit
764 with others and keep up, or to disappear?) [its already there]
765 }
766 function rptBoxClicked(d){
767 //take this id and see if it is a BoxA or B
768 var i = d.indexOf('rptBoxA');
769 if(i > -1){
770     // it was found so show the iid element by adding rptBoxAData.style.display='';
771     //alert('rptBoxAData'+d.substring(7, d.length));
772     document.getElementById('rptBoxAData'+d.substring(7)+'Holder').style.display='';
773 }else{
774     //wasnt found show the oiid element
775     //document.getElementById('rptBoxBData'+d.substring(7)).style.display='';
776     rptValSent(d);
777 }
778 }
779
780 function rptValSent(id){
781 //check if it is a sentence bad report OR
782 var a = id.indexOf('report');
783 //var evt = events[id.substring(a+6)];
784 var evt = usr_ans[id.substring(a+6)];
785 var i = id.indexOf('rptBoxA');
786 //check if its an emotion report
787 if(i > -1){
788     //if emotion then get value of that selected and the appropriate sentence send to
789 server/function the variables
790     var d = document.getElementById(id).value;
791     http.open("get", "/reportEvtError?evt="+evt+"&type=type1&data="+d, "true");
792     http.onreadystatechange = hdlReportVerification;
793     http.send(null);
794     alert('The following emotion('+d+') has been reported as being associated with
795 this event:('+ evt +')');
796 }
797 }else{
798 //its sentence, so send the bad sentence to the server (server should mark time/count
799 and flag)
800     http.open("get", "/reportEvtError?evt="+evt+"&type=type0", "true");
801     http.onreadystatechange = hdlReportVerification;
802     http.send(null);
803     alert('This sentence has been reported as bad: ' + evt);
804 }
805 //set innerHTML to be reported. Thank you.
806 var repBox = document.getElementById(id.substring(a));
807 repBox.innerHTML = 'Reported!';
808 }
809

```

```
810 function hdlReportVerification(){  
811   var data = http.responseText;  
812 }
```

+ASDGameServer.py

```

1 import cherrypy as cpg
2 import WebAppToDBMethods as dbm
3 import random
4 import re
5 Emailpattern = r'\b[A-Za-z0-9._%-]+@[A-Za-z0-9._%-]+\.[A-Za-z]{2,4}\b'
6 pattobj = re.compile(Emailpattern)
7 eps = pattobj.search
8
9
10 class Root:
11     def index(self):
12         return open("../static/index.html")
13     index.exposed = True
14
15     def getEvalPage(self):
16         return open("Evaluation.html")
17     getEvalPage.exposed = True
18
19     def regUsr(self, e, u, p):
20         if(e and u and p): #they must exist
21             #make sure email is regex right (else return error)
22             if(not eps(e)):
23                 return "0[invalid email address"
24             #make sure uname is all alphanum (else return error)
25             if(not u.isalnum()):
26                 return "0[username must contain numbers and letters only"
27             #take hash of pw before its ready
28             return dbm.regNewUsr(e, u, p)
29             #return "1[test;35;25;15"
30         else:
31             return "0[must include a username email and password to register" #means one
32 of the three was null
33     regUsr.exposed = True
34
35     def logUsr(self, u, p):
36         if(u and p):
37             if(u.isalnum()):
38                 return dbm.logUsr(u, p)
39                 #return "1[test;35;25;15"
40         else:
41             return "0[must includea username and password to login[]" #one arg was null
42     logUsr.exposed = True
43
44     def tryLogScore(self, u, g, hs):
45         g = int(g)
46         #hs = int(hs)
47         dbm.logNewHiScore(u, g, hs)
48     tryLogScore.exposed = True
49
50     def getEvEmPairs(self, num=0, xemt="", xemtnum=0):
51         pairListString = ''
52         #num = num - xemtnum
53         num = int(num)
54         xemtnum = int(xemtnum)
55         regList = list()
56         xregList = list()
57         if(num > 0):
58             #print num
59             regList = dbm.getXEVEmpairs(num)
60             #print len(regList)
61         if(len(xemt)>0 and xemtnum>0):
62             xregList = dbm.getXEEventsForAnEmotion(xemt, xemtnum)
63             regList.extend(xregList)
64             random.shuffle(regList)
65             for apair in regList:
66                 pairListString = pairListString + apair[0] + '[' + apair[1] + '[]]'
67             #return pairListString.strip('::asd-bigsep:')
68             return pairListString[:pairListString.rfind('[]')]
69     getEvEmPairs.exposed = True
70
71     def getEmsForEvs(self, evtliststring=""):
72         emsback = ''
73         if(len(evtliststring) > 0):
74             evlst = evtliststring.split('[]')]
75             emlst = dbm.getEmotionsCorrespondingToEvents(evlst)
76             for em in emlst:
77                 emsback = emsback + em[1] + '[]]'
78             #return emsback.strip('::asd-bigsep:')
79             return emsback[:emsback.rfind('[]')]
80     getEmsForEvs.exposed = True
81

```


97
96
95
94
93
92
91
90
89
88
87
86
85
84
83
82
81
80
79
78
77
76
75
74
73
72
71
70
69
68
67
66
65
64
63
62
61
60
59
58
57
56
55
54
53
52
51
50
49
48
47
46
45
44
43
42
41
40
39
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1

```
def storeEvents(self, els=""):
    if(len(els) > 0):
        evlst = els.split('[[[]]')
        dbm.storeNewEvents(evlst)
    return 'okay'
storeEvents.exposed = True

def reportEvtError(self, evt="", type="", data=""):
    if(len(evt)>0 and len(type)>0):
        dbm.setFlagForEvent(evt, type, data)
    return 'okay'
reportEvtError.exposed = True

cpg.root = Root()
cpg.config.update(file = '../wvtd.conf')
cpg.server.start()
```

+WebAppToDBMethods.py

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81

```
# import necessary libraries
import PadToEmotionAlgo as pag
import MySQLdb
import time
#make 1 time connection to dB needed
Gameconn = MySQLdb.connect(host = "localhost",
                           user = "XXXX",
                           passwd = "XXXX",
                           db = "XXXX")

Gcur = Gameconn.cursor()

#global variables
emo_list = "happiness,sadness,surprise,fear,disgust,anger,nothing"
max_results_back = 40
min_emotion_correction_threshold = 0

#####
##User Login Functions
#####

#register a new user and return fail or pass, along with username
def regNewUsr(email, uname, pw):
    #query for existing email
    try:
        Gcur.execute("""
            SELECT email
            FROM asdusers
            WHERE email = %s
            """, (email,))
        rows = Gcur.fetchall()
    except:
        print "error in trying to find existing email"
    #if exists report that error
    if(len(rows) > 0):
        return "0[]this email already exists in our system, choose another"
    #query for existing username
    try:
        Gcur.execute("""
            SELECT username
            FROM asdusers
            WHERE username = %s
            """, (uname,))
        rows = Gcur.fetchall()
    except:
        print "error in trying to find existing username"
    #if exists report that error
    if(len(rows) > 0):
        return "0[]this username already exists in our system, choose another"

    else: #or by default if it gets this far
        pwhash = hash(pw)
        joindate = time.asctime()
        try:
            Gcur.execute("""
                INSERT INTO asdusers (username,email,pw,member_join_date)
                VALUES
                (%s,%s,%s,%s)""", (uname,email,pwhash,joindate))
        except: #print erro and sentence(actually all values, trying to be inputed here)
            print "\nDarn, error occured when registering new user. Details below:"
            print "\nuname, email, pwhash, joindate", uname, email, pwhash, joindate
            #make sure changes get committed before exiting
            try:
                Gameconn.commit()
            except : # print error
                print "\nDouble Dar, error occured when committing all new adds in
registration. Details below:"
                #print "\nActual Error--> "

                return "1[]" +uname

def logUsr(uname, pw):
    #get the columns for this uname and pw and game scores
    pwhash = hash(pw)
    try:
        Gcur.execute("""
            SELECT username, Game1Hi, Game2Hi, Game3Hi
            FROM asdusers
            WHERE username = %s and pw = %s
```

```

    """ , (uname, pwhash))
    rows = Gcur.fetchall()
except:
    print "error in trying to find existing username password match"
    #if len of return is greater than 0, take 1 and return good string
    if(len(rows) > 0):
        darow = rows[0]
        return "1["+darow[0]+';'+darow[1]+';'+darow[2]+';'+darow[3]
    #else, return not found error (dont exist)
    else:
        return "0[]bad username and password combination"

def logNewHiScore(uname, gamenum, score):
    #score = str(score)
    #insert/update in db for gamenum and username where score > current
    gnVar = ''
    if(gamenum == 1):
        gnVar = 'Game1Hi'
    if(gamenum == 2):
        gnVar = 'Game2Hi'
    if(gamenum == 3):
        gnVar = 'Game3Hi'

    try:
        Gcur.execute("""
        UPDATE asdusers
        SET """+gnVar+"" = """+str(score)+"" , """+gnVar+""_count = """+gnVar+""_count
+ 1
        WHERE username = %s and """+gnVar+"" < """+str(score)+""
        """, (uname,))
    except: #print erro and sentence(actually all values, trying to be inputed here)
        print "\nDarn, error ocured when reporting new hi score"
        print "\nunam, gamenum, score", uname, gamenum, score
        #make sure changes get committed before exiting
        try:
            Gameconn.commit()
        except : # print error
            print "\nDouble Dar, error ocured when committing all new adds in new Hi Score.
Details below:"
            #print "\nActual Error--> "

```

```

#####
##Game Functions
#####
# return X sentence-emotion pairs In a double-list format
#i.e. (['i get up', 'happiness;sadness;disgust'], ['i go home', 'disgust;fear'])
def getXEVEmpPairs(x):
    numtoget = min(x, max_results_back)
    rows = list()
    largeHold = list()
    try:
        Gcur.execute("""
        SELECT sentence, algo_emotion, human_corrected_emotion
        FROM eventdata
        WHERE flag_types NOT LIKE '%type0%'
        ORDER BY RAND() LIMIT %s
        """, (numtoget,))
        rows = Gcur.fetchall()
    except:
        print "error in getting XevEmpPairs, basic method of retrieval ", numtoget
    for row in rows:
        smallHold = list()
        smallHold.append(row[0])
        aggemotion = row[1]
        # if flagged over thresh # add it:should be formatted at "happiness:1;sadness:4"
        human_corrections = row[2]
        hclist = human_corrections.split(';')
        for e in hclist:
            if(e.find(':') != -1):#then there is an emotion wtha value
                apair = e.split(':')
                if(apair[1] > min_emotion_correction_threshold):
                    #add the emotion from this part to the algo_emotion part
                    aggemotion = aggemotion + ';' + apair[0]

        smallHold.append(aggemotion)
        largeHold.append(smallHold)
    return largeHold

```

```

#given an emotion and number. Return X events that match that emotion note: to get that
%% will make one % in actual string when doing string formatting

```

```

163 def getXEventsForAnEmotion(em, x):
164     # get min(MAX,x) nubme of events with this emotion as possibility
165     # -includes trying user-appended and pc-algo generated
166     # -exclude events marked as bad
167     numToGet = min(x, max_results_back)
168     em = "%"+em+"%"
169     returnEvents = list()
170     EvsAndEms = list()
171     rows = list()
172     try:
173         Gcur.execute("""
174             SELECT sentence
175             FROM eventdata
176             WHERE flag_types NOT LIKE '%type0%' AND (algo_emotion LIKE %s OR
human_corrected_emotion LIKE %s)
177             ORDER BY RAND() LIMIT %s
178             """, (em, em, numToGet))
179         rows = Gcur.fetchall()
180     except:
181         print "error in gets from the dB using (getXevtsForAnemotion)-->", em, numToGet
182     for row in rows:
183         #put into a simple list of events
184         returnEvents.append(row[0])
185     EvsAndEms = getEmotionsCorrespondingToEvents(returnEvents)
186     return EvsAndEms
187
188 # given events, get emotions and return list of those emotions(or string)
189 def getEmotionsCorrespondingToEvents(evtlist):
190     evemlistreturn = list()
191     for evt in evtlist:
192         #print evt
193         #groom the evt
194         #evt = standardizeSent(evt)
195         # run pad-algo judge for each incoming event
196         algoem = pag.textToEmotion(evt)
197         darow = None
198         try:
199             #now make query for this sentence to see if it exists in db (by hash) and if
200             #so take its user-appended emotions as well if possible and if not conflicting ONLY if
201             #flag has been reported will the user-appended even exist so we can void bringin queries
202             #back if so
203             Gcur.execute("""
204                 SELECT human_corrected_emotion
205                 FROM eventdata
206                 WHERE sentence_hash = %s AND human_corrected_emotion != ""
207                 """, (hash(evt),))
208             darow = Gcur.fetchone()
209         except:
210             print "error in getting EmotionsCorresptoevents ->", evt, hash(evt)
211         if(darow):
212             human_corrections = darow[0]
213             hclist = human_corrections.split(';')
214             for e in hclist:
215                 apair = e.split(':')
216                 ##print apair, apair[1], (int(apair[1]) >
217                 min_emotion_correction_threshold), algoem, algoem.find(apair[0])
218                 if(int(apair[1])>min_emotion_correction_threshold and
219                 algoem.find(apair[0])!=-1):
220                     #append this value to algoem
221                     algoem = algoem + ';' + apair[0]
222
223             #after chcking current algo and user-reported in db if any add the modified
224             algoem
225             evemlistreturn.append((evt, algoem))
226
227         # return corresponding list or list-item pairs
228         return evemlistreturn
229
230 # store new events into the database
231 def storeNewEvents(evtlist):
232     for evt in evtlist:
233         # -standardize sentence for entry and check sentence validity (2 words at
234         # least, no delineator(bydefaultanyway)). erase double-space?, nah
235         evt = standardizeSent(evt)
236         if(evt):
237             #run through pad-algo judger
238             #place in DB (one statement/query) multipel lnies thingie online
239             documentationf orhtis
240             #put in sentence, sentence_hash, algo-emotion, user-supplied
241             sent = evt
242
243

```

```

244     algem = pag.textToEmotion(sent)
245     senthash = hash(sent)
246     usrdate = time.asctime(time.gmtime())
247     #no duplicates cauze senthash is primary key
248     #mark user-supplied by putting datetimestamp in that column(at least date,
249     username later)
250     try:
251         Gcur.execute("""
252             INSERT IGNORE INTO eventdata
253             (sentence,algo_emotion,sentence_hash,human_supplied)
254             VALUES
255             (%s,%s,%s,%s)""" , (sent,algem,senthash,usrdate))
256     except: #print erro and sentence(actually all values, trying to be inputed
257     here)
258         print "\nDarn, error ocured when adding a new event. Details below:"
259         print "\nsent, algo_em, hash, datetime", sent, algem, senthash, usrdate
260     #make sure changes get committed before exiting
261     try:
262         Gameconn.commit()
263     except : # print error
264         print "\nDouble Dar, error ocured when committing all new adds. Details below:"
265         #print "\nActual Error--> "
266
267     # set a flag for an event given event type and data
268     def setFlagForEvent(evt, ftype, data):
269         stdsent = standardizeSent(evt)
270         if(stdsent):
271             senthash = hash(stdsent)
272             # if emotion thing, count and place
273             if(ftype == "type1" and (emo_list.find(data)>-1) ):
274                 rows = list()
275                 try:
276                     #see if any such emotions already exist. if so, reconstruct string, else
277                     simply add to stringand put back int
278                     Gcur.execute("""
279                         SELECT flag_types, human_corrected_emotion
280                         FROM eventdata
281                         WHERE sentence_hash = %s
282                         """, (senthash,))
283                     rows = Gcur.fetchall()
284                 except:
285                     print "failed just in getting pre-existing data from db for setting lags,
286                     in setFlagForEvent ->", stdsent
287                 for darow in rows: # should only be one but hey who knows
288                     #darow = Gcur.fetchone()
289                     #darow[0]
290                     flagString = ""
291                     newEmoCell = ""
292                     fspot = darow[0].find("type1")
293                     cspot = darow[1].find(data)
294                     if(fspot == -1):#this means it needs to have type1 set for the
295                     firsttime(since it doesnt already have 1), so ow flagString will get a value
296                     flagString = "[]type1[]"
297                     if(cspot == -1): #meaning there is no existence of a correction for this
298                     emotion yet, construct a new 1 with count 1 (emotion:1)
299                     if(len(darow[1])>0): #meaning already data there so append ";" plus
300                     1
301                     newEmoCell = darow[1] + ';' + data + ':1'
302                     else: #meaning nothing there so simple add emotion and 1
303                     newEmoCell = data + ':1'
304                     else:#meaning this has been corrected before, cuz the data(emotion) was
305                     found in this column. get the number for this and update it
306                     emotions = darow[1].split(';')
307                     newNum = None
308                     for lile in emotions: # find the number we want to inc, we knwo its
309                     in here
310                     nameNnum = lile.split(':')
311                     if(nameNnum[0] == data):
312                         newNum = int(nameNnum[1]) + 1
313                         break;
314                     if(newNum):#kinda dum check cuz always is here as doen by bigger IF-
315                     ELSE check above
316                     #we must separate the cell string into (begtilldata+';',
317                     updateddata,+';' end) if beg and end have dtat that is
318                     begExist = darow[1][:cspot]
319                     startOfEndExist = darow[1].find(';', cspot)
320                     if(startOfEndExist == -1):#meaning this is the last one or the
321                     first and only one (cuz no separtors after
322                     #endExist = None #so it doesnt need and ending to be added to
323                     it
324                     newEmoCell = begExist + data + ':' + str(newNum)

```

```

32         else:
33             endExist = darow[1][startOfEndExist:] #else it does need
34             closuer cuz somethign is after it
35                 newEmoCell = begExist + data + ':' + str(newNum) + endExist
36             try: #now try to put hte new EmoCell back into this (update?)
37                 Gcur.execute("""
38                 UPDATE eventdata
39                 SET human_corrected_emotion = %s, flag_types = CONCAT(flag_types, %s)
40                 WHERE sentence_hash = %s
41                 """, (newEmoCell,flagString,senthash))
42             except:
43                 print "db input erro atcoutning upnew emotions. surprise surprise"
44             #make sure changes get committed before exiting
45             try:
46                 Gameconn.commit()
47             except : # print error
48                 print "\nTriple Darn, error ocured when committing all new adds.
49
50 Details below:"
51             #print "\nActual Error--> "
52
53             # -if sentence thing, mark in db, count as well,nocount cuz after marked
54 one, it doesnt get reshown, so will always be 1
55             if(ftype == "type0"):
56                 #doesnt matter what is already there, cuz this can only get called once per
57 sentence, so just slap a "[]type0[]" in the update spot
58                 flagString = "[]type0[]"
59                 try: #now try to put hte new EmoCell back into this (update?)
60                     Gcur.execute("""
61                     UPDATE eventdata
62                     SET flag_types = CONCAT(flag_types, %s)
63                     WHERE sentence_hash = %s
64                     """, (flagString,senthash))
65                 except:
66                     print "db input erro at marking bad sentences"
67                 #make sure changes get committed before exiting
68                 try:
69                     Gameconn.commit()
70                 except : # print error
71                     print "\nQuadruple Darn, error ocured when committing a freakin FLAG.
72
73 Details below:"
74                 #print "\nActual Error--> "
75
76 #determine if its a valid sentence or not, spaces, subj and verb found (later and only if
77 reliable)
78 def sentOrNot(a):
79     a = a.strip()
80     if(a.find(' ') == -1):
81         return None
82     else:
83         return 1
84
85 #strip of end punctuation(no more), lowercase it (no more), strip ends, check if sent or
86 not(spaces, non-letter chars) and send back sent or null/None
87 def standardizeSent(a):
88     #if begin or end of sentence has punctuation take it off (non letter)
89     #find single quotes in sentence and replace with \ to escape it
90     #if any char in the sent is not a letter that is ok, prolly not a key word
91     #if ; char in the word, remove it (that is used in string going back to server (but
92 shouldnt matter cuz the first split should be on a diff char(like';;code;'))
93     ##     if(not a[-1].isalpha()):
94     ##         a = a[:-1]
95     ##     if(not a[0].isalpha()):
96     ##         a = a[1:]
97     #strip whitespace from both ends
98     a = a.strip()
99     a = a.replace("'", "\'")
100    #send whole sentence to lower case capitalization
101    #a = a.lower()
102    #check sent or not
103    son = sentOrNot(a)
104    if(son):
105        return a
106    else:
107        return None
108    #return this sentence or None based on if sentence or not

```

