CSCE 606 Software Engineering Spring 2020

Stress Learning Game Final Project Report

Team: Not Enough Pizza

1. Team Members and Roles
Hanyang Li (Product Owner)
Ping Lu (Scrum Master)
Xin Hu (Programmer)
Zhiyu Yan (Programmer)
Chaoyang Zhu (Game Testing)
Yipeng Lu (Game Design)
Mufeng Xie (Report Editor)

2. Links
GitHub repo:
https://github.com/pinglu0212/2020-Spring-CSCE-606
Full poster and demo link:
https://drive.google.com/file/d/1KRJPOsBVaCMYGldBg8Q88H-H7Z8xVkD2/view?usp=sharing
Poster link:
https://github.com/pinglu0212/2020-Spring-CSCE-606/blob/master/StressLearningGame_Poster.pdf

3. Summary
The stress learning game is designed to reinforce the learning of nervous system concepts for middle school students.

We constructed a file system for customers/instructors to customize the content of the game anyway they want. They can add or delete more questions for students in this file system by simply writing questions and hints in the configuration generation system user interface.

We improved the gaming experience of the Hangman word guessing game by increasing game play and visual effects for the Hangman user interface.

4. User Story

The customer wants an app that let the gamers/students guess words that are missing in a sentence related to the nervous system. The gamer is provided with an image and/or words as hints. We have completed all the user stories below systematically:

First, the customer wants an interactive and user-friendly Hangman gaming interface for students/gamers. There should be words and picture hints along with a sentence. The sentence has a few words missing and are to be guessed. Each gamer has 5 life points to lose in each round. If the gamer successfully click on the alphabets that are contained in the words before losing 5 life points, the page will direct the gamer to the next question. If the gamer click on an alphabet that's not in the words to be guessed, the gamer loses 1 life point. If the gamer can't guess the words correctly within 5 times, the game is over. The gamer can click on the "play again" button to have another round of game. We completed this step during iteration 0-3.

Secondly, the customer wants a configuration generation system that allows the instructors to easily add more questions to the system by directly filling out a form in the stress learning game file system. We completed and perfected this step during iteration 4 and 5.

Finally, the customer also wants us to deploy the app on the www.futuredogter website. We completed this step during iteration 5. Detailed implementations can be referred to in 9. Implementations of this report.

5. Overall Flow

The customer/instructor types in a question to the stress learning game configuration generation system, and the configuration generation system generates a JSON file which are stored in the configuration folder. The word guessing game randomly reads one JSON file from the configuration folder and display the corresponding question to the user/students. Both the Stress Learning Game File system and the Hangman Word Guessing Game consist of a .css file that help keep all the information in a properly displayed format, a JavaScript file that handles all kinds of events and a html file as the basis of the webpage. Details of the overall flow can be seen in Figure 1 below.

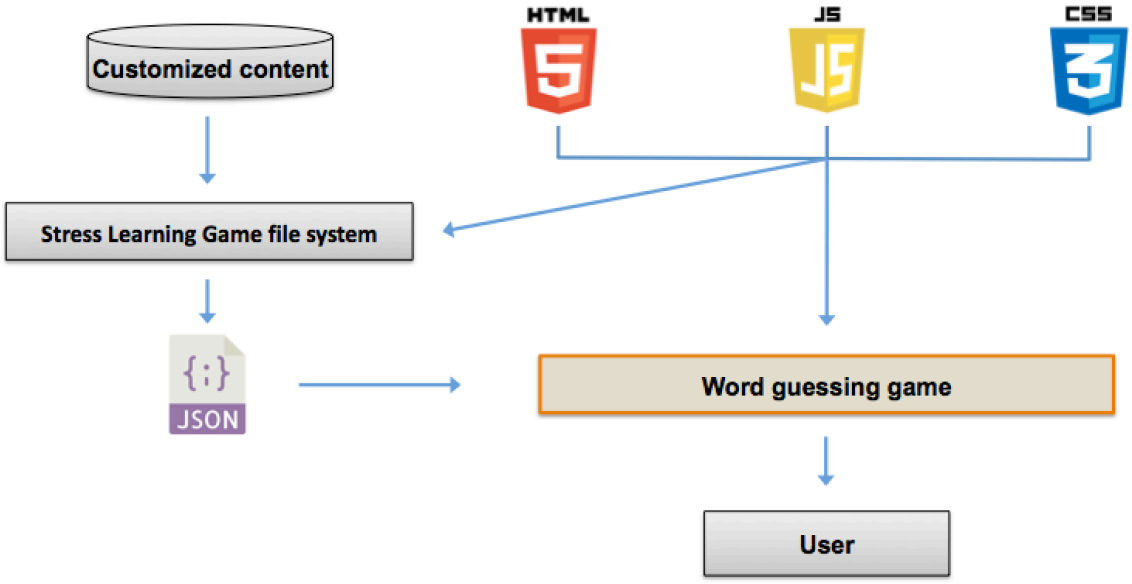

Figure 1. Overall Flow of the Game Design

6. User Interface of Hangman

The user interface of Hangman can be seen in the Figure 2 below as an example. The mechanics of the hangman word guessing game is:
Given a sentence with a few words missing, an image and a hint containing information about the missing words are shown, and the user/student has 5 chances to guess the missing words.

The rules for the game is explained as follows:
First, gamers need to click on the buttons containing alphabets that the missing word has.

Secondly, the number of lives decreases as the user/student makes wrong attempts. As the number of lives decreases, the figure becomes clearer and clearer, which makes guessing the words easier.

Thirdly, useful clues will show up when clicking on the missing words.

Finally, game is over when the number of lives decreases to 0. Click on "Play again" button to take another try. If the gamer clicked all the alphabets contained in the missing words before the gamer runs out of limited lives, the gamer wins the game and will be directed to the next game randomly chosen.

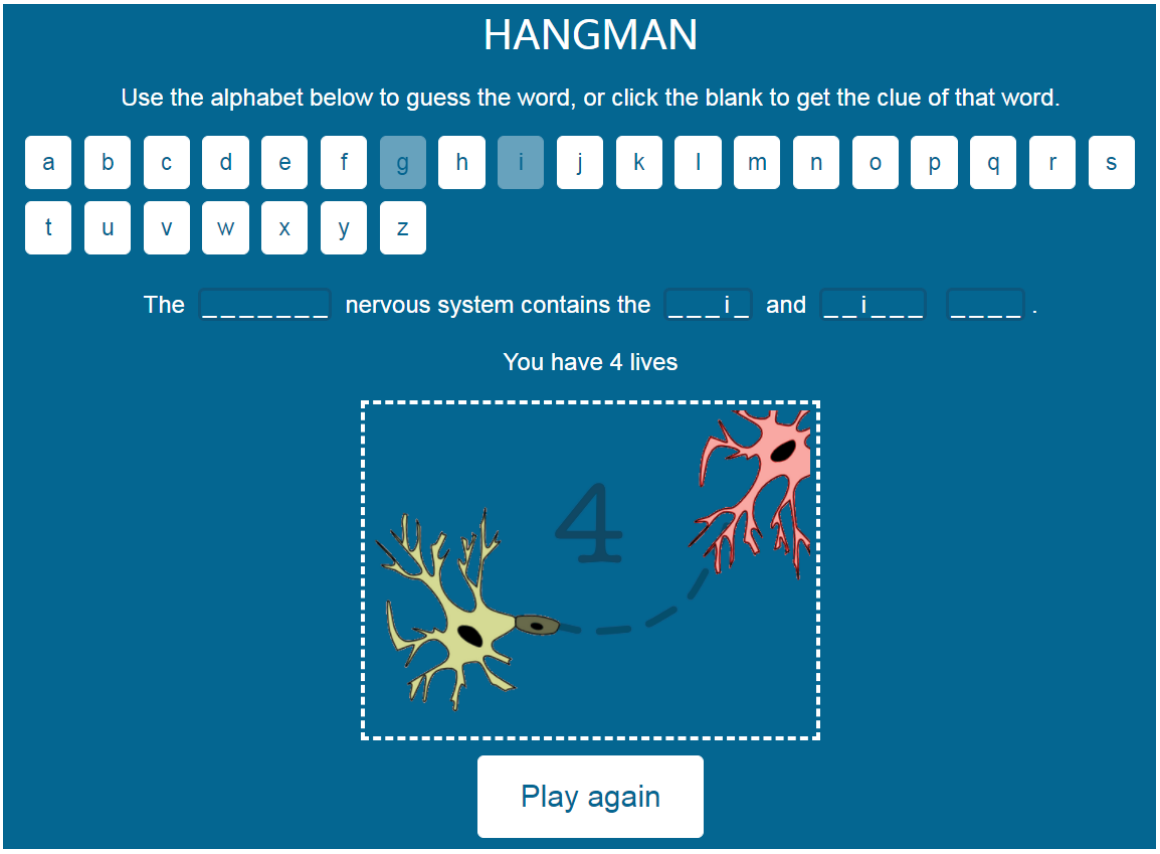

Figure 2. User Interface of Hangman Word Guessing Game

7. User Interface of Stress Learning Game Configuration Generation System

The Stress Learning Game Configuration Generation System (see Figure 3 below) is developed to allow the instructors to easily add more questions to the Hangman Game by directly filling out a form in this configuration generation system. The forms the instructors filled out in the configuration generation system will be treated as configurations for the Hangman Game interface. The configuration will be stored in a JSON file, which contains sentence, description, word, text hint, image URL.

To generate a configuration file, the instructor needs to fill out a whole sentence or a whole word in the “Sentence” box first, this sentence will be shown in the students’ screen except the missing words.

Secondly, the instructor needs to specify the “missing word” or the word to be guessed in the “Word” box.

Thirdly, the instructor can put a text hint, namely synonyms or descriptive words of the “missing word” in the “Text Hint” box.

Fourthly, the instructor can set up an image hint to be shown as an option. After typing in an image URL, a preview of the image will be shown in the "Image Show" box in order to help instructors to check whether they have chosen the correct image. The URL can be a website link of the image or it can be the full folder name/file name. If the instructors choose to use their own images as input, they can upload their own images to the folder /images.

Then the instructor can click on "add" or "delete" button to add more words to be guessed or remove some words.

Finally, the instructor can click on "SUBMIT" button to generate a JSON file.

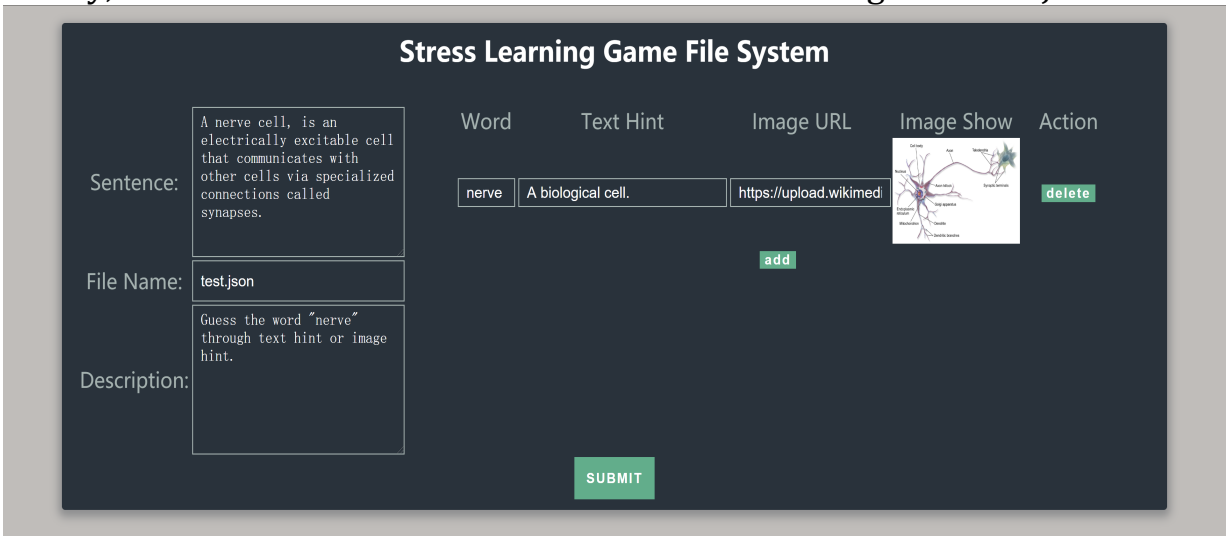

Figure 3. Stress Learning Game Configuration Generation System

8. Deployment on the Development Server

The server the client specified: futuredogter.com
The user: lawano1
The password: H-8Uqp5Z

Once we are on the server, we would see a "stepstone" folder (might be inside the futuredogter.com parent directory, depending on your specific ftp client). We want to get to this directory: stepstone/workArea/NIH-SEPA-1/activityLib/sample1/.

Inside of this directory, there is an "apps" folder. This is where we place our mini-app stuff for testing and preview.

9. Implementation

Firstly, the user need to download all the files in our github website and unzip it.

Secondly, the user can deploy it by turning on the terminal and typing: python -m http.server 8888

Then, the user need to open the browser, and typing: http://127.0.0.1:8888/

Now you are ready to play the Hangman Stress Learning Game ran locally on your computer!

Alternatively, you can also add "Moesif Origin & Cors Changer" to your google chrome browser extensions (see Figure 4 below), turn it on, and open the link below to see the app in action on your browser: http://www.futuredogter.com/stepstone/playerShell.php?org=CET&sys=public.Latest&pool=TAMU-CET-1&resourceloc=www.futuredogter.com&resourceavatar=NIH-SEPA-1&resource=sample1&ppj=1_1_40

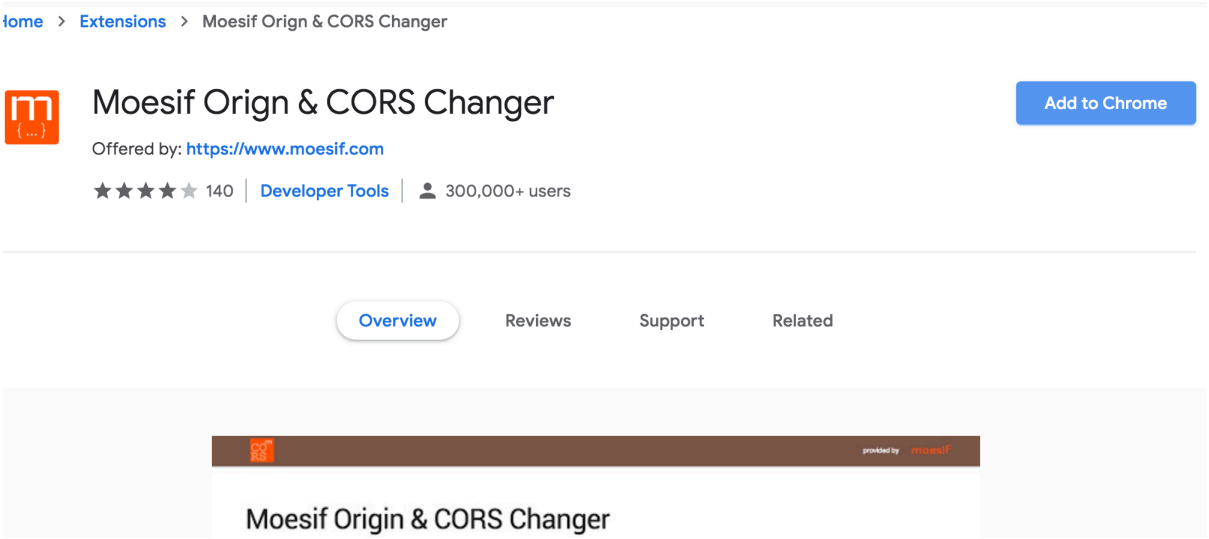

Figure 4. Download CORS on user's Chrome Browser

10. Meetings

Feb 24th at TAMU HRBB Building 5:30pm-6:30pm with the Customer
March 4th 4pm at TAMU HRBB Building 514F with TA
Every Friday 5pm on Zoom Internal Group Meeting
May 5th 1:30pm on Zoom with the Customer
May 5th 5pm on Zoom with TA