Project Backlog: Team 7

Will Enright, Zhaoji Jiang, Adam Petty, Dingfu Sun, Thomas Worns
January 29, 2016

PROBLEM STATEMENT

Everybody gets bored sometimes and a simple, easy, and fun way to solve that is by playing a game. We are planning on implementing a version of Pictionary. With the new recreation of this classic game, we hope to provide people with an entertaining way to connect with their friends and families.

BACKGROUND INFORMATION

People have searched for entertainment in various forms, from ancient games like chess to modern games like Cards Against Humanity. Both of those games requires setup and individual game pieces, but Pictionary does not. Pictionary is an excellent way to spend some time with friends and practice your art skills. Pictionary is an easy and fun game for all ages that is being brought back with this fresh implementation.

ENVIRONMENT

We plan on using electron to create our application. Electron is a framework for creating cross platform applications using web technologies. On top of that, we will use Angular for building using interfaces.

Backlog ID	Functional Requirements	Hours
1	As a user, I want a canvas to draw pictures on.	20
2	As a user, I want to be able to play with friends locally.	30
3	As a user, I want to be able to play with 3 or more friends online in a private lobby.	45
4	As a user, I want to be able to play with random players online in public lobbies.	8
5	As a user, I would like the ability to save drawings I liked.	10
6	As a user, I would like to have a slideshow featuring all the correct guesses after each round of the game.	14

7	As a user, I would like to have a scoreboard that keeps track of all the players' performance.	6
8	As a user, I would like to have sound effects.	12
9	As a user, I would like to have friends join my online game lobby.	6
10	As a user, I want word clues to be randomized during play.	5
11	As a user, I want to have a main menu.	10
12	As a user, I want to have the ability to send guesses.	6
13	As a user, I want to have game settings including: game timer, word categories, and difficulty levels.	20
14	As a user, I want to have a player name to be displayed.	6

NON-FUNCTIONAL REQUIREMENTS

- As a developer, I want the server to support more than 50 concurrent players.
- As a developer, I want the application to be cross platform.
- As a user, I want a visually appealing interface to promote intuitive gameplay.
- As a user, I want the game to run smoothly and be highly responsive.

USE CASES

Case: Playing (Artist Player)

Actor	**System**
	1. A randomized word clue is shown to the player, and the round timer begins.
2. Click and drag on the drawing canvas.	**3.** Line following cursor path appears.
4. Release mouse press on the drawing canvas.	**5**. Line following cursor stops appearing where mouse press ended.

Case: Playing (Guessing Player)

Actor	System
	1. Live drawing of artist player's word clue appears on the guessing player's canvas.
2. Types and sends a guess of what is being drawn by the artist player on the canvas.	**4**. Automatically matches the player's guess against the actual word clue.
	5. If the player's guess is correct, then a success sound effect is triggered, visual confirmation is displayed to all players, points are added to the artist and guessing player, all canvases are cleared, and a new, randomly different word clue is shown to the artist player. If the player's guess is incorrect, then an audiovisual failure effect is given to the guessing player, and the round of gameplay continues.

Case: Starting a local game

Actor	System
1. Opens application	**2.** Displays main menu with Options, New Local Game, and New Online Game commands.
3. Selects "New Local Game" command	**4.** Displays window with party size, difficulty, timer, and category options.
5. Selects confirm	**6.** Canvas and game interface are displayed, and the game begins.

Case: Joining a multiplayer game

Actor	**System**
1. Opens application	**2.** Displays main menu with Options, New Local Game, and New Online Game commands.
3. Selects “New Online Game” command	**4.** Displays Online Game window with menu options to host or join game.
5. Selects “Join Game”	**6.** Displays interface with options to either enter a private lobby passcode or randomly a public game lobby.

Case: Hosting a multiplayer game

Actor	**System**
1. Opens application	**2.** Displays main menu with Options, New Local Game, and New Online Game commands.
3. Selects “New Online Game” command	**4.** Displays Online Game window with menu options to host or join game
5. Selects “Host Game”	**6.** Displays window with settings for the game, including party size, word categories, game difficulty, timer configuration and a passcode for this private lobby.
7. Selects “Confirm”	**8.** Displays a lobby screen with all current players until designated number of players is reached. Once full, displays game interface and begins the game.

Case: Game ended

Actor	System
1. The final round of the game concludes.	**2.** Final scoreboard is displayed along with the winning player(s).
	3. Slideshow containing all correctly guess pictures is shown to all players.
4. User clicks on a drawing he would like to save.	**5.** “Save” and “Cancel” options are displayed to the user.
6. Clicks “Save”	**7.** The image file of the picture is saved to the user’s Downloads folder.

Case: Change game settings

Actor	System
1. While in a game lobby as host, click “Game Settings” button.	**2.** Game Settings menu opens with options for the game timer, word categories, and difficulty settings.
3. Select new desired game settings by clicking on appropriate buttons.	**4.** The newly-selected game setting(s) will now be highlighted.
5. Click “Save”	**6.** The newly-selected game setting(s) are applied and saved to the config file. The next game to initiate will reflect the new game settings.