**Final Product Definition and Scrum Team Plan**

**Submitted by**

Team West

**Submitted on**

May 26, 2020

**Team Members**

Jeremy Gallagher

Hailey Gibson

Jon-Erik Prichard

**1. Final Product Name and Description**

Product Name: WordWizard

Word Wizard is a single player anagram puzzle game designed to operate on devices utilizing the Android mobile operating system. Anagram puzzles are displayed in grid a format with each grid displaying 16 random letters. Players select letters on the grid that form words based on data from a chosen English dictionary API.

Players score points based on the per letter score and the character length of the word. Letters are scored by difficulty of use where vowels are scored low while rare-use letters such as Z and Q are scored high. Puzzle grids are timed at 60 seconds per grid which generates several scoring categories such as Total Number of Words, Average Score Per Word, Total Grid Score, Highest Word Score and Longest Word.

Although players compete individually, scores are registered for each category displaying game winners and high achievers. Scores are registered as scores for the Day, Month and All-Time.

**2. Final Functionality Overview**

WordWizard is a touch-screen game and operates by user interaction touching the screens of Android devices with a touch-enabled screen. The product does not function via keyboard, mouse or voice input. Word Wizard functions include the following:

* Touch activated buttons
* Letter randomizer algorithm
* Puzzle grid timer
* English word database
* Player register
* Player score register
* Scoring system
* Puzzle grid display
* Grid/game timer
* Player achievement badges
* Pause game, with reset or quit
* Reset the letters at the cost of points or time
* Score multiplier letters
* Danger letters or traps that cause the other letters to flash or disappear for a set amount of time.

**3. Final Overview of Architecture**

Word Wizard is designed for use in Android OS devices. The primary program language for Android APPs is Java but is more efficient using the Java derivative, Kotlin. The product will utilize a full Android Studio and Android Developer Tools tech stack including SQLite for the database.

4. **Final Scrum Team Members**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Role | Hours/Week | Responsibility | Skillsets |
| Jeremy Gallagher | Scrum Developer | 40 | Product development, Function development and integration, database integration, API integration | Java programming, development IDE management, function integration |
| Hailey Gibson | Product Owner | 40 | Project management as Product Owner, Algorithm integration | Project Management, Java programming, Algorithms |
| Jon-Erik Prichard | Master | 40 | Project management as Scrum Master, UI Design and programming. Secondary database programming | Graphic Design, Front-end (UI) programming, database design and programming |

**5. Final Virtual Daily Scrum Meetings**

* Location: Discord
* Daily, 5:00 PM PST
* Duration: 15 minutes or less

**6. Product Features to be Delivered**

1. As a user/player, I want puzzles to be generated at random so that each puzzle is a new experience.
2. As a user/player, I want functions to be responsive and perform as intended so that game play is possible.

**7. Backlog for Each Sprint**

[There will be three 2-week sprints and one 1-week spring. The features or user stories picked during a sprint can change but work on this must commence this week. Provide adequate detail in your descriptions. Scrum teams are to be agile and this initial plan could change as you go through the product development cycle - but it is good to give this some thought now.]

**Sprint 1** 5/27/2020 – 6/9/2020

1. Puzzle Grid Function: Programing to load random letters into puzzle grid
2. Puzzle Grid UI: Graphic design of puzzle grid, letter display and alerts
3. English Word Database: Acquisition and initial integration of English word database API used to affirm and register valid words as selected
4. Letter Randomizer Algorithm: Algorithm to randomly select 16 letters from the English alphabet for display in puzzle grid
5. Touch Activated Buttons Function: Change-state functionality for touch-activated buttons (systemwide)
6. Touch Activated Buttons UI: Graphic design for touch-activated buttons (systemwide)
7. User Interface Workflow Chart: User story detailing screen flow and game element transitions
8. User Interface Wireframe(s): Basic wireframe graphic design of WordWizard game interface detailing game element placement

**Sprint 2** 6/10/2020 – 6/23/2020

1. Quit Game Function: Method to stop game, no score registered
2. Reset Puzzle Grid Function: Method to reset letters displayed on grid
3. Pause Game Function: Method to pause game, including by device interruption
4. Game Scoring System Function: Method to register game scores
5. Game Scoring System UI: Graphic design and animation of scoring system
6. Puzzle Grid Counter Function: Method to number randomly assembled grids
7. Puzzle Grid Timer Function: Method to time game play
8. Puzzle Grid Timer UI: Graphic design of game timer

**Sprint 3** 6/24/2020 – 7/7/2020

1. Player Register Function: Method to register player information and identity
2. Player Register UI: Graphic design for player register
3. Player Score Register Function: Method to register player game scores
4. Player Score Register UI: Graphic design for player score register
5. Player Achievement Badges Function: Method to recognize player achievement
6. Player Achievement Badges UI: Graphic design for player achievement system

**Sprint 4** 7/8/2020 – 7/14/2020

1. Reward/Danger/Trap Letter Function: Method to generate letters displayed on puzzle grids that present dangers, traps and rewards
2. Reward/Danger/Trap Letter UI: Graphical design and animations for reward/danger/trap letter system
3. APP Deployment: Process to deploy WordWizard for live access

**8. Screenshots of Visual Studio with Agile Tool (or Suitable Substitute)**

[Use this section to provide the requisite screenshots of the platform, with explanation of each image.]