University of Central Florida



CAP 4104 Human and Technology Interaction

Functional Requirements Document

Group <2>

Thomas Flannery

Brian Herrin

Jon Liauw

Brooke Norton

Jasmine Savwoir

Matt Taubler

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Description of Change | Author | Date |
|  |  |  |  |
|  |  |  |  |

CONTENTS

1. INTRODUCTION
   1. Purpose
   2. Scope
   3. Background
   4. References
   5. Assumptions and Constraints
   6. Document Overview
2. METHODOLOGY
3. FUNCTIONAL REQUIREMENTS
   1. Context
   2. User Requirements
   3. Data Flow Diagrams
   4. Logical Data Model/Data Dictionary
   5. Functional Requirements
4. OTHER REQUIREMENTS
   1. Interface Requirements
   2. Data Conversion Requirements
   3. Hardware/Software Requirements

1. **INTRODUCTION**

Welcome to the functional requirements made by Group 2 for the continued development of a videogame version of the classic game, Pictionary! The goal of the rest of the document will be to provide direction, with necessary specifics, to whichever team ends up having to develop Pictionary, so they will know exactly what Pictionary is and how they should continue this great game’s development. We at Group 2 are confident that these functional requirements can be used by any team made up of our fellow UCF classmates to guide the development process and lead to the success of this project!

* 1. **Purpose**

The application will be an emulation of the guessing word game Pictionary, where 4 or more players are required to draw, recognize images, and input guesses in under one minute. This is because the guess timer will be running at the same time as the draw timer. The objective of the game is to guess the correct word or phrase, then points are given based on how quickly a correct answer is provided. The users will pick the number of rounds before playing, and the team that has accrued the most points by the end is declared the winner.

The purpose of the FRD is to outline the functional specifications of “Pictionary”, input, and outputs, and descriptions of interactions between users and the application. In completing this document, consensus will be reached by the development team as to what is being designed and specific objectives and by the testing team as to what will be done on the unit and integration testing level. Finally, the stakeholders (in this case TAs/Professor) will receive clarity as to what exactly the product is. Specifically, this document will cover methodology and requirements, both functional and nonfunctional.

* 1. **Scope**

This document covers a general introduction (objective and purpose of software, outline), methodology, functional requirements (for the user, as well as models/diagrams), and other requirements (such as interface, hardware/software). Again, providing written descriptions and the creation of this document as a collective of the system objectives is integral for clarity in development, testing, and for the stakeholders.

* 1. **Background**

The team as a collective is involved in the creation of this document, with the purpose of achieving a consensus on the development objectives, testing plan, and overall design of the system to be produced. The responsibilities will vary from person to person but as a whole the team is responsible for the outline of objectives (this document), the implementation as a development team, and unit and integration testing.

* 1. **References**

Pictionary: How­to http://www.wikihow.com/Play­Pictionary

Example game: http://www.pinturillo2.com/

* 1. **Assumptions and Constraints** 
     1. **Assumptions**

People have the ability to get the system working on a functional interface, with drawings able to be understood by other players, even if they are drawn on a screen.

We are assuming the availability of at least a single computer, with the players playing the game as a standalone application that should not require the internet to run.

The market the application would be run in is assumed to be in the United States of America (USA) and policy decisions should be constrained by challenges of the USA legal system. The game should not be exactly like the version of Pictionary released by Mattel or other well known board game companies in order to dodge cease and desist orders that are commonplace in the USA.

* + 1. **Constraints**

The development of the game is limited based upon the due date to submit the project to Karin Whiting for the CAP 4104 class. The real business consequence that will happen should the due date be ignored is a lower grade for the project.

As a strategic decision, the functional requirements of this project need to consider the probable skills of any given development team that would be put in charge of implementing Pictionary from these functional requirements. CAP 4104 is available to Computer Science majors that have not passed the Foundation Exam. This prevents us from assuming that any given team in the class would be able to implement a version of Pictionary that involves very complex elements like internet access or running servers.

The implementing team can only be assumed to know Java and C, the two primary programming languages taught pre­Foundation Exam. Even from those two, only making Graphical User Interfaces in Java is taught sufficiently. Therefore, the functional requirements should be possible to develop in Java and with this level of Computer Science education in mind.

* 1. **Document Overview**

The document is broken down to 4 main parts:

(1) Introduction

(2) Methodology

(3) Functional Requirements

(4) Other Requirements

The introduction covers the scope and purpose of the FRD as well as background information, references, and assumptions and constraints about the development and testing process.

Methodology covers the overall approach of determining the FRD contents.

The Functional Requirements is broken down into multiple aspects, firstly a context diagram of the system to elucidate its functionality. User requirements provides details about varying permissions for different types of users, as well as high level functional requirements, and likely scenarios with regards to system interactions and user functions

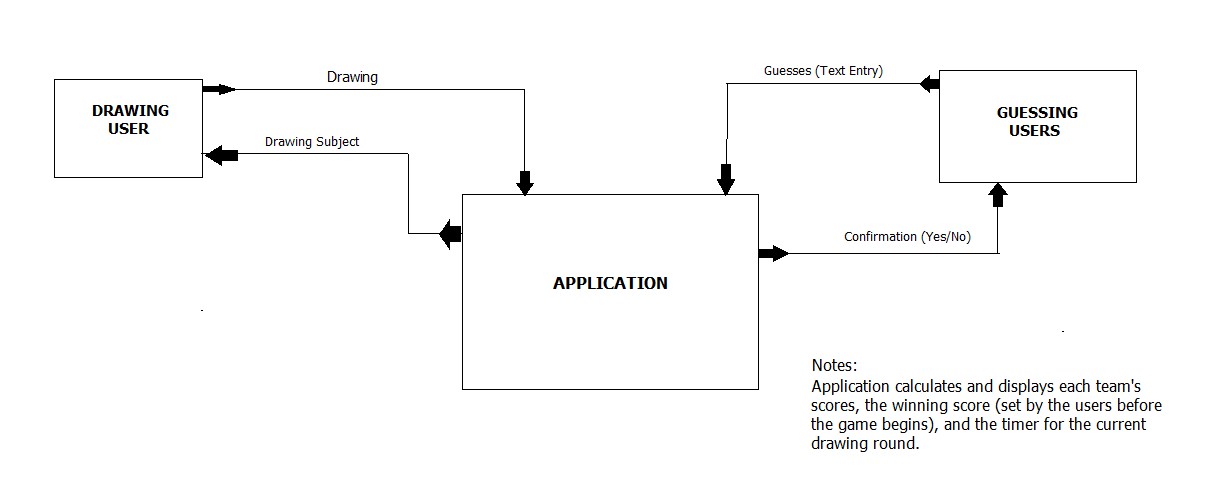
Finally, the Other Requirements section details interface, data conversion, and hardware and software requirements.

1. **METHODOLOGY**

In order to determine the FRD contents we have to fully understand both how the game works and how users will interact with it. We must first analyze the functions of the game, the interface it will function on, and system requirements for it to work. We have to determine the rules of the game, visual aesthetics, writing tools, timers, etc. Then we must analyze what type of audience will play the game, age range of users, difficulty of the game, and how to make it accessible over multiple devices. Rules have to be easy to understand and the game should have a simple design so anyone can play it. We must answer these questions:

* + 1. How will the game look?
    2. What are the rules of the game?
    3. Who will play this game?
    4. What devices will the game be accessible on?
    5. How will the interface work?

1. **FUNCTIONAL REQUIREMENTS** 
   1. **Context**



* 1. **User Requirements**

The users are split into teams, and take turns going through drawing/guessing rounds. Each drawing/guessing period for a team is a “turn.” A round is finished after each team has had their turn in that round. There are no administrator/owner user types needed for the game; however, at the beginning of a draw/guess turn, the active team will select a member to draw (the Sketching User), while the other team members act as Guessing Users. Once a Subject Card is randomly selected by the application, only the current Sketching User (and the application itself) knows what Subject Card was drawn.

* 1. **Logical Data Model/Data Dictionary**

No interfacing with databases will be necessary for this application. Data requirements are confined to the object­oriented classes needed to implement Pictionary as a GUI­driven application.

Subject Card:

Subject Cards have a category and a list of four subjects. Categories include Person/Place/Animal, Object, Action, Difficult, and Free Choice. Excluding Free Choice cards, when a card is drawn, one of the five subjects on that card will be randomly generated for the current Drawing User to sketch on their team’s canvas. When a Free Choice card is drawn, the Sketching User may select any word or phrase to draw, they must input their word or phrase into the system.

Timer:

The timer counts down from 60 seconds once a Sketch/Guess Turn is initiated.

Canvas:

An editable, MSPaint­like area that the Sketching User uses during a Sketch/Guess/Turn. Only the active Sketching User may have edit access to their team’s canvas at a given time.

* 1. **Functional Requirements** 
     + Ability to draw an image
     + Set the total number of rounds for the game
     + Display how to play and ruleset
     + Use of timer for drawing/guessing turn
     + Ability for users to join a team.
     + Randomly select which team goes first
     + Alert team when timer has expired
     + Randomly select a category for each turn (Person/Place/Animal, Object, Action, Difficult, All Play, and Free Choice)
     + Generate words or phrases for teams to draw — plentiful and random enough to not draw the same word or phrase within a short time period or one game
     + Keep record of score for each team during game
     + Ability to simulate turns where a team is given a word or phrase and a timer begins for them to draw an image.
     + Ability to stop clock upon correct guess.
     + Determine if guess input is correct or not.
     + Tools for drawing images should be easy enough for anyone to use them

**3.4.1 Functional Requirements Group 1**

**Exhibit 1 ­ Ability to Draw an Image**

|  |  |
| --- | --- |
| Section/  Requirement ID | Requirement Definition |
| FR 1.0 | The system shall provide the ability to draw images on screen. |
| FR 1.1 | The system shall provide FR1.0 o n a canvas in­game (See 4.1 Interface Requirements ) |
| FR 1.1.1 | The system shall provide paint tools for the canvas mentioned in FR1.1 |
| FR 1.1.2 | The system shall provide mouse detection to use both FR1.1.1 and FR1.1. |
| FR 1.1.1.1 | The system shall provide the paint tool “Pencil” for FR1.1.1 |
| FR 1.1.1.2 | The system shall provide the paint tool “Eraser” for FR1.1.1 |
| FR 1.1.1.3 | The system shall provide the paint tool “Fill Bucket” for FR1.1.1 |
| FR 1.1.1.4 | The system shall provide the option to change paint color for FR1.1.1 |
| FR 1.1.1.4.1 | The system shall show what color is being used for painting for FR1.1.1.4 |
| FR 1.1.1.4.2 | The system shall show different color swatches used for painting for FR1.1.1.4 (See UI Plan) |

**3.4.2 Functional Requirements Group 2**

**Exhibit 2 ­ Calculate Score for Each Turn/Round and How Winning Round is Determined**

|  |  |
| --- | --- |
| Section/  Requirement ID | Requirement Definition |
| FR 2.0 | The system shall keep t rack of how many points each team has earned throughout the game. |
| FR 2.1 | The system shall determine how many points per correct guess based on how quickly they guessed correctly before the clock time ended. |
| FR 2.2 | The system shall create a final turn in the case of a tie, the fastest to correctly guess the image will be the winner. |

**3.4.3 Functional Requirements Group 3**

**Exhibit 3 ­ Display How to Play/Ruleset**

|  |  |
| --- | --- |
| Section/  Requirement ID | Requirement Definition |
| FR 3.0 | The rules shall be displayed at the beginning o f each game |
| FR 3.1 | There shall be an option to display the rules throughout the game |
| FR 3.1.1 | FR 3.1 shall be activated by pressing the “esc” key |

**3.4.4 Functional Requirements Group 4**

**Exhibit 4 ­ Input Guess**

|  |  |
| --- | --- |
| Section/  Requirement ID | Requirement Definition |
| FR 4.0 | There shall be a text box for team members to input guesses |
| FR 4.1 | The input will not accept numbers and non­letter characters |
| FR 4.2 | There shall be a button to submit a guess input |
| FR 4.3 | Pressing ‘enter’ within the text box also acts as a submission |
| FR 4.4 | Textbox should be cleared out after each submitted guess |
| FR 4.5 | Functionality to be able to match guess to current turn word or phrase |
| FR 4.6 | If submitted guess i s correct, timer ends and score is calculated based on time left to guess |

**3.4.5 Functional Requirements Group 5**

**Exhibit 5 ­ Keeping Team Scores**

|  |  |
| --- | --- |
| Section/  Requirement ID | Requirement Definition |
| FR 5.0 | Users in teams are not strictly kept track of and teams go unnamed |
| FR 5.1 | A team is randomly selected to go first in the game. Notify the users which team is selected first. |
| FR 5.2 | Per each turn, a picture topic is randomly selected for the sketching team. |
| FR 5.3 | The system shall keep track and display the total score for each team |
| FR 5.4 | Team scores shall increment for each correct guess per turn |
| FR 5.5 | Score calculation: ceiling((60 ­ total seconds elapsed) / 2) |

**3.4.6 Functional Requirements Group 6**

**Exhibit 6 ­ The Timer Function and Alerting Team to Timer’s End**

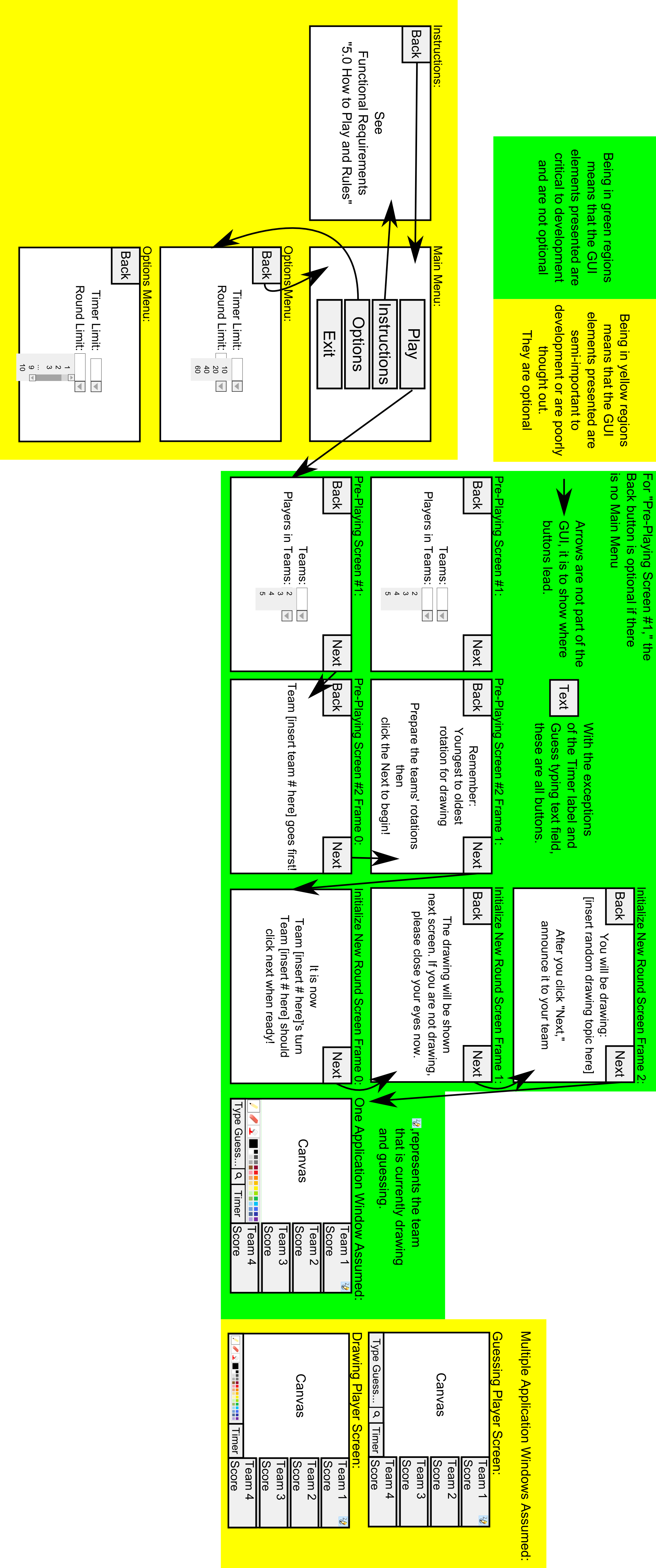
|  |  |
| --- | --- |
| Section/  Requirement ID | Requirement Definition |
| FR 6.0 | There should be a maximum amount of time for each turn of one minute |
| FR 6.1 | Timer should reset after each turn |
| FR 6.2 | There should be a buzzer sound that goes off when time runs out |
| FR 6.3 | Timer should stop if correct guess is provided |

**3.4.7 Functional Requirements Group 7**

**Exhibit 7 ­ Picture Topic Generation**

|  |  |
| --- | --- |
| Section/  Requirement ID | Requirement Definition |
| FR 7.0 | Upon beginning of turn, user must generate a random category to play from. These include Person/Place/Animal, Object, Action, All Play, Free Choice. Upon selection of a category, system should generate a word or phrase that matches the category type. Free Choice allows user to pick the word or phrase which they must input into the system upon selection. |
| FR 7.1 | Only the sketching user may see the word or phrase to draw as this the essential component of the game. |
| FR 7.2 | The system shall generate words or phrases for each team to draw each turn of play. |
| FR 7.3 | Words or phrases will be uncomplicated to accommodate all ages. |
| FR 7.4 | Should generate words or phrases plentiful and random enough to not draw the same word or phrase within a short time period or one game |
| FR 7.5 | Should be able to draw an image hinting or depicting the word or phrase |

1. **OTHER REQUIREMENTS** 
   * Reliable
   * Effective and efficient to use
   * Safe to use
   * Has good utility
   * Easy to learn
   * Easy to remember how to use
   * Fun to use



**4.1 Interface Requirements**

The User Interface picture provided above should adequately show the overall flow of the program’s transitions to different aspects of the game. The development team is allowed to have creative input regarding the color scheme of the User Interface, the way any given button looks and behaves, and any music/sounds that occur in the game. Regarding color scheme of the User Interface, the picture provided is intentionally made to be a simple black and white color scheme. This is to make understanding the elements within very clear. Each frame of the project may have its own color scheme or the whole project can have one color scheme.

Regarding the way any given button looks and behaves, an example could be when a button is clicked, the development team may decide if clicking the button causes a sound to occur or how the button’s image changes when clicked and held down (there should be some form of response!). The image used for any given button (provided it does not obscure the labeling of the button) is open for the development team to decide. The development team may decide the text style for labels and buttons (again provided that it does not obscure the message of the label or button).

The program should be in English. Support for other languages will be out of the scope of these functional requirements.

The program is expected to take up the whole screen. Support for other sizes will be out of the scope of these functional requirements.

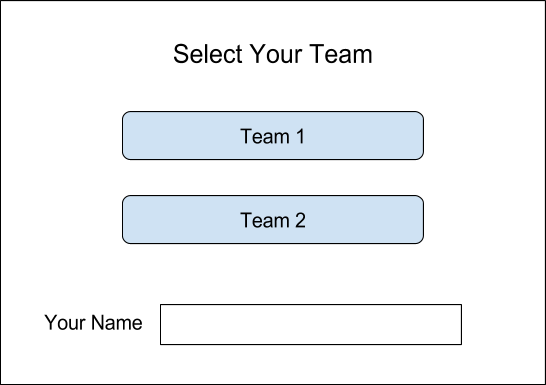
**4.1.1 How to Play and Rules**

1. Players must select the total number of rounds they would like the game to continue for. A single player may make this selection by selecting 1 to 10 rounds. The default behavior is 10 rounds.
2. Players are to be divided into teams by selecting from the possible teams. The first sketching artist is the youngest player on the team and the order of determining the sketching artist goes next youngest to oldest from there. The sketching artist is the person who attempts to illustrate the word using the interface to draw it. Everyone else on the team will try to guess the word that the sketching artist draws.
3. A team will be randomly selected to go first.
4. At the beginning of a team’s turn to sketch, a category card will be randomly generated for the team. The category card will determine which word or phrase the sketching artist has to draw. The different categories are Person/Place/Animal, Object, Action, Difficult words, and Free Choice. Only the sketching artist may see the word or phrase to be drawn. Upon Free Choice, the sketching artist may choose any word or phrase they would like to draw. The sketching artist must input their selected word or phrase into the provided field.
5. Once the sketching artist is given their word or phrase, the timer will start as soon as the next button is pressed and the canvas is shown. The sketching artist must begin to draw their image using the canvas while their team members try to guess the word or phrase they are trying to draw. Team members must guess through the guess input field and submitting their answer by clicking the provided button or hitting the enter key.
6. Points will be given to the team depending on how quickly they guess the word or phrase correctly. Upon a correct guess, the timer will stop, the score will be added to the team’s total score, and the team’s turn will be complete. It will now be another team’s turn until all have had their turn in order to complete a round. If a team does not succeed in guessing the word or phrase, 0 points will be given to the team.
7. Game play will continue in such way until the total number of rounds have been reached. The team with the highest score at the end of the rounds will be declared the winner.
8. Should a tie occur, then the game ends in a tie.

**4.1.2 Another Possible UI Plan**

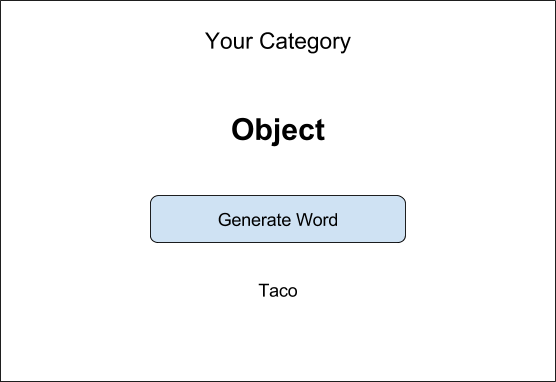
A different UI plan from the previous, the development team may choose to follow elements from either of these plans should they contradict.

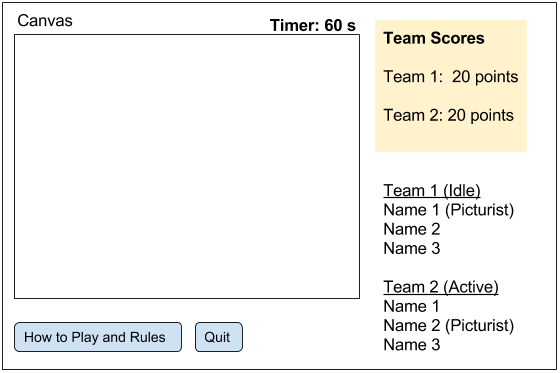




Sketching Artist Screen

Note: the word appears after generate button is clicked





**4.2 Hardware Interfaces**

This interface should be supported by any interface with a touch screen, or screen and drawing tool (such as a mouse) as Pictionary requires very little other than that. The user would also need some way to input a guess, such as a keyboard. The elements of the interface should be able to be accessed with either a click or a touch/tap.

**4.3 Software Interfaces**

The only applications which the subject application may need to interface with are web browsers, or to be really specific, operating systems.

**4.4 Communications Interfaces**

Should the development team wish to include local networks, the communications interfaces need to include a way to communicate over local networks if the game is played with other players that are not in the same room as the initiating player.

The communications interfaces should also be able to communicate across devices so players can play either online or on mobile devices.

For mobile connectivity, an external API to establish P2P connectivity is likely required (Ex: Multipeer Connectivity for iOS).

**4.5 Data Conversion Requirements**

There will be no legacy data conversions because the game data will only be stored during a single game session. Storing high score data for a party game like Pictionary is virtually pointless, because it can certainly be cheated.

Teams and scores will only be stored during the current game session.

Legacy data will include all the words or phrases associated with each category. The categories include Person/Place/Animal, Object, Action, Difficult words, and Free Choice. There must be at least 100 stored words or phrases under each category except for Free Choice. Upon each turn, when a category is generated, a word or phrase must be generated from the associated category. It is up to the development team to determine what these words or phrases will be for each category.

**4.6 Hardware/Software Requirements**

If the development team decides to include online support for the application, the system should be able to function across various electronic devices. It should be accessible online, and on iOS and Microsoft devices. It should work with mobile internet or data connection.

Otherwise, the system should be on a computer with a keyboard and mouse.