YAHTZEE

Requirements Analysis Document

Team Name
Khang Phan
Duc Phan
David Recic
Eric Le

. Introduction

1.1. Purpose

The purpose of this project is to develop a computer game Yahtzee for Procrastination Pastimes (PP). This includes fulfilling the client's requirements and ensuring that the product delivered satisfies them.

1.2. Scope

The goal of the project is to create a version of dice game Yathzee running on computer. The game will follow the rule of the game mention in the references, and allow up to six players playing in turn in one computer.

1.3. Objectives and Success Criteria

The objectives and success criteria for the development of this game . The client has given several requirements that will need to be meet by by December 3, 2018. The project should incur no cost to both the client and the developers. The game should eventually be able to be developed into a web browser based application running on the CSE sever.

1.4. Definitions, Acronyms, and Abbreviations

1. Yahtzee: A dice game created by Milton Bradley.
2. CSE Server: The server that allows to test installation.
3. PP: Procrastination Pastimes
4. Categories: different combination of dices used to calculating scores.
5. Upper Section: A section displaying all six categories of sum dices from aces to sixes.
6. Lower Section: A section including poker-themed categories which specifies values of point.
7. Total score: total score of the upper and lower section.

1.5. References

1. Yahtzee rules: https://www.hasbro.com/common/instruct/Yahtzee.pdf

1.6. Overview

Building a computer game, Yahtzee, for PP. There will be several steps in developing this game. They are acquiring the necessary requirements, designing the system, building, debugging, deployment and maintenance. The game will be completed on December 2, 2018.

. Current System

2.1. Overview

1. This section provides the functional overview of the system.
2. This divides into two parts, functional requirements and non functional requirements.

2.2. Functional Requirements

1. Start a new game with one or more players (up to 6).
2. Each player at his / her turn:
   - ○ Throw / rethrow the dice up to 3 times.
   - ○ Choose which dice to be thrown, which on to be kept.
   - ○ Choose dice to unsave.
   - ○ Choose categories to score and end turn.
   - ○ Get to know his/her score, and scores of other players.
3. Surrender and withdrawn from the game.
4. Leaderboard to keep track of scores of each player.
5. The player is able to restart the game at any point.
6. Game music.
7. Functionality to change of the volume of the music.
8. Function to change language in setting.
9. One player ends his/her turn.
10. Add new feature to the game (the Tiny Straight score category).

2.3. Nonfunctional Requirements

2.3.1. Usability:

1. Interface is filled with at least five colors and background music.
2. Supporting multiple languages. E.g. Chinese, Vietnamese, Spanish, etc.
3. Scoresheet:
   - ○ Unscored and scored categories are distinguished.
   - ○ Upper section and lower section must be separate.
4. Players are distinguishable whether in their turns or not.

2.3.2. Reliability:

1. None.

2.3.3. Performance:

1. Response must be received within 1 second when user create an request / action.

2. All actions and manipulations in the game must be smooth without delaying.
3. The results on the leaderboard must be correct.

2.3.4. Supportability:

1. This game application is not dependent on any platform. It can be run on operating systems that support Unity Engine (Windows, OSX, Linux, Android, iOS,etc).

2.3.5. Implementation:

1. The project will be implemented in Unity.

2.3.6. Interface:

1. All graphic user interfaces will be created and designed using the Unity Engine.

2.3.7. Packaging:

1. None.

2.3.8. Legal:

1. Tools used for creating software must be under either open-source or educational license.
2. Images and music used are legally copied from source.

2.4. System Models

2.4.1. Scenarios

1. One player:
   - Sunday afternoon, after finishing the homework, Derrick wants to play some Yahtzee.
   - He open the app, choose one-player mode. He enters his name and starts the game.
   - In the first turn, he throws 5 dices and get 3 six, an one, and a two. Good start.
   - He keeps the six and throw 2 dices again. He gets a two and a three. Not happy, he tries the third time and has a five and a four.
   - Having no re-roll, he chooses “three of a kind” for 27 points.
   - Second turn, he has five ones at the first roll. He keeps the Yahtzee for 80 points. His total score is 107 now.
   - He gets very lucky at this game. After the last turn, he has total 243 points.
   - Satisfied, he closes the game.
2. Multiple players:
   - Three friends Bob, Phillip and Dave are so bored at the Friday party that they decide to go home and play Yahtzee.
   - They choose multiple-player mode and fill in their names in order to start the game.
   - The game starts. It’s Bob’s turn and he rolls 5 dices and get a two, a five, a three and 2 four. Phillip luckily gets 3 three, a one and a six. Dave gets 2 two, 2 five and a three.
   - Bob keeps 2 dices of four and rolls three others. He gets 1 six, 1 five and 1 three and score a Small Straight. Phillip keeps 3 dices and rolls 2

dices and he gets Yahtzee (5 three). Dave keeps 4 dices of 2 two and 2 five and roll one dice. He gets full-house (3 two and 2 five).

- ○ Next turn, Bob rolls 5 dices and get 2 two, a five, a six and a four. After he rolls, he click the scoreboard to see all scores. His score is 30, Dave's score is 25 and Phillip's score is 50.
- ○ Bob and Dave get bored by their bad luck and decide Phillip to be the winner.
- ○ They exit the game.

2.4.2. Use Case Model

1. Start game.
   - ○ Participants: group of players.
   - ○ Entry condition: Yahtzee is up and running.
   - ○ Flow of event:
     - ■ Choose vs "many players" mode.
     - ■ Input number of player
     - ■ Each player takes turn filling in their nickname and (optionally) changes avatar.
     - ■ Click start button.
   - ○ Exit condition:
     - ■ The game starts.
2. Roll dice and score.
   - ○ Participants: one or more players.
   - ○ Entry condition:
     - ■ Game has started.
     - ■ Only one player can throw the dice.
   - ○ Flow of event:
     - ■ First, player throw all 5 dices by clicking on the throw button on the screen.
     - ■ The screen responds by displaying the values and scores of the current dices in the corresponding unfilled categories.
     - ■ The interface will display choice of keeping dices that the player can make. Player may choose to keep from 0 to 5 dices, and reroll the rest.
     - ■ The screen displays the values of rerolled dices and then pops up a question asking if the player wants to reroll the dices.
     - ■ Player can choose yes or no and he or she may reroll at most twice.
     - ■ If yes, the screen displays again the values of rolled dices and then shows the categories that player can choose to score.
     - ■ Player click on the categories to confirm the score and the screen shows end turn.
   - ○ Exit condition:
     - ■ Scoresheet is updated.
     - ■ The game moves on next player's turn.

3. Exit game.
    - Participants: One player in the game
    - Entry condition:
        - Game is running.
        - It’s current player’s turn.
    - Flow of event:
        - Player click on resign button.
        - Game show a confirm message.
        - Player click yes to surrender and exit.
        - Player click no to cancel and keep playing.
    - Exit condition:
        - If there are multiple players in the game, the game move on with the next player’s turn.
        - If only one player left, he/she is the winner.

2.4.3. Object Model

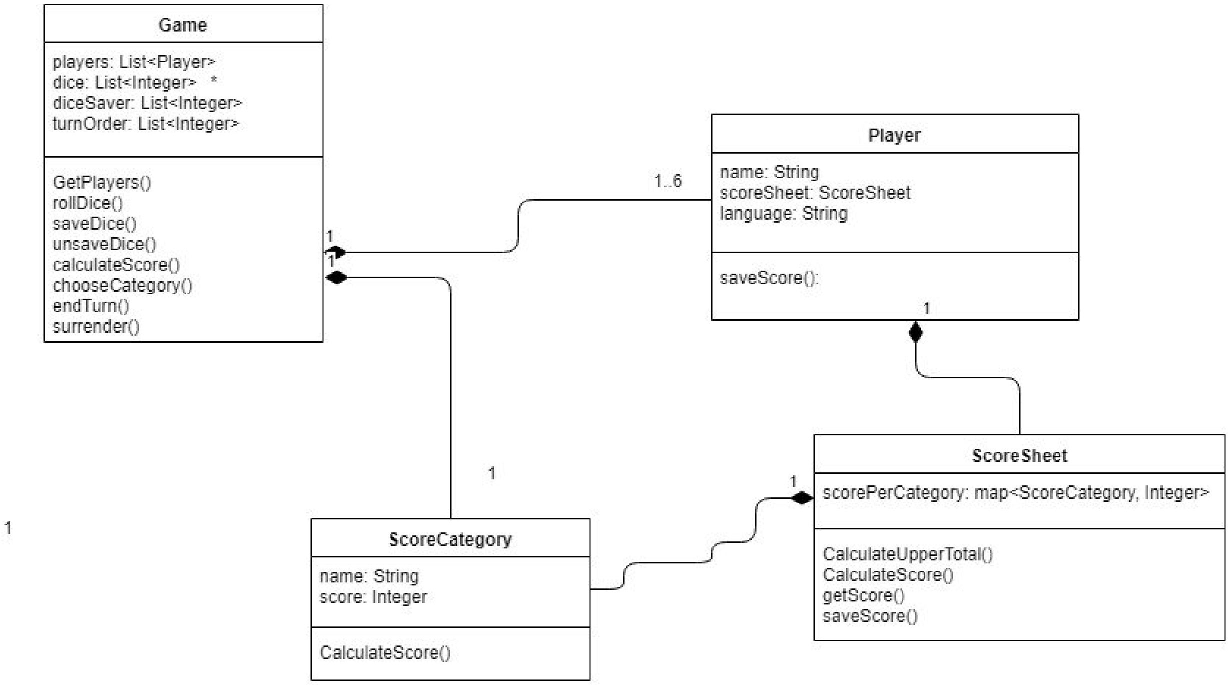

Figure 1: Object Model Diagram

2.4.4. Dynamic Model

1. Start game

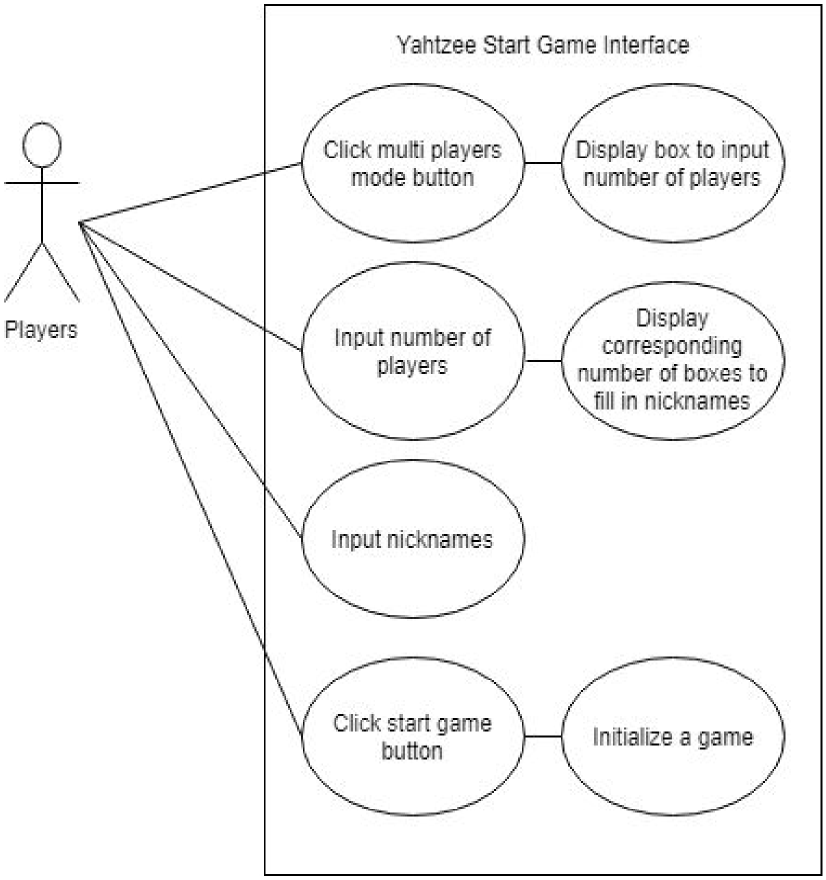
Yahtzee Start Game Interface
Players
Click multi players mode button
Display box to input number of players
Input number of players
Display corresponding number of boxes to fill in nicknames
Input nicknames
Click start game button
Initialize a game

Figure 2: Use Case 1 Diagram

2. Roll dice and score

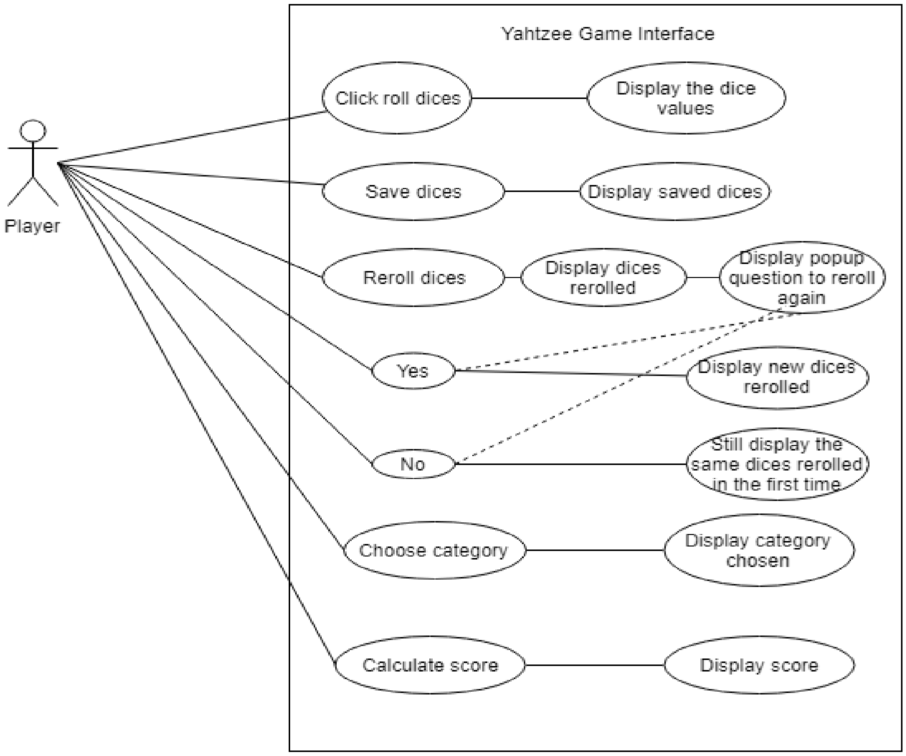

Figure 3: Use case 2 Diagram

3. Exit game

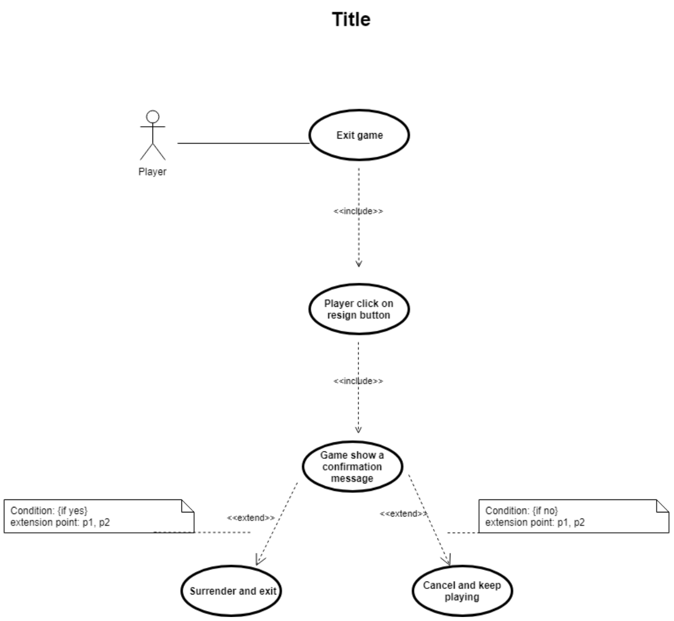

Figure 4: Use case 3 Diagram

2.4.5. User Interface: Navigational Paths and Screen Mockups

1. TBD.

. Proposed System

3.1. Overview

1. This section provides the functional overview of the system.
2. This divides into two parts, functional requirements and non functional requirements.

3.2. Functional Requirements

1. Start a new game with one or more players (up to 6).
2. Each player at his / her turn:
   - Throw / rethrow the dice up to 3 times.
   - Choose which dice to be thrown, which on to be kept.
   - Choose dice to unsave.
   - Choose categories to score and end turn.
   - Get to know his/her score, and scores of other players.
3. Surrender and withdrawn from the game.
4. Leaderboard to keep track of scores of each player.
5. The player is able to restart the game at any point.
6. Game music.
7. Functionality to change of the volume of the music.
8. Function to change language in setting.

9. One player ends his/her turn.

. Glossary

1. TODO: