

IOS DEVELOPMENT ASSIGNMENT 2



Contents

I.	Introduction	2
II.	Rules	2
III.	Design.....	3
IV.	Media	7

I. Introduction

This game is a direct adaption of the dice game Yahtzee .The reason I choose a pre-existing game to adapt instead of coming of with a new idea on my own is due to me being unfamiliar with dice game and casino game in general so relying on game that has already been hashed out is a more effective choice. The reason for Yahtzee is that as single player dice game aim at children, it is very simple to do, and its nature meant that I could focus more on the game logic instead of trying to do complex animation.

The direct inspiration for the implementation is the specific iteration of Yahtzee found in Nintendo's Clubhouse Games: 51 Worldwide Classics which I had saw some of my favorite streamers played several times.

Overall, the app have almost fully adapted Yahtzee to a digital form with the only exception being the Joker rules which I forgot about during development.

II. Rules

Yahtzee involves a player rolling five dice and then comparing the results of their role to a score sheet with 13 categories each with its own scoring criteria.

The game consist of two main screens, the Dice roll screen and the Score sheet screen. In the dice roll screen, the player can choose to roll their dice. They can roll three times during which they choose to keep a dice by taping on it. Keeping a dice and pressing roll will lock that dice until the next turn. After all five dices are lock or the player use all three rolls, the player will have to fill a category in the score sheet to continue to the next turn. They will have to continue this until all categories all filled which correspond with 13 turns of play.

The score categories can be divided into two groups: Upper and Lower.

Upper groups consist of six simple scoring categories: Aces, Twos, Threes, Fours, Fives, Sixes. To fill these categories the Player only need to have at least one dice of the corresponding value i.e. [1] - > Ace, [2] -> Two and so on. The number of points earn in each category corresponds to the number of dice of those values. If the total score of the upper group is 63 the Player will earn a bonus +35 points.

The Lower groups consist of seven categories each with own rules:

- Three of a kind: If player's roll have three or more matching dices, they can fill this slot with the sum of all five dice.
- Four of a kind: If player's roll have four or more matching dices, they can fill this slot with the sum of all five dice.
- Full House: If player's roll have three matching dices of one value and two matching of another (for example three [2] and two [6]) they can fill this slot with a set score of 25
- Small Straight: If player roll have four dices with sequential value (1-2-3-4, 2-3-4-5 or 3-4-5-6) they can fill this slot with a set score of 30.

- Large Straight: If player roll have five dices with sequential value (1-2-3-4-5, or 2-3-4-5-6) they can fill this slot with a set score of 40.
- Yahtzee: If player's roll have five matching dices, they can fill this slot with a set score of 50.
- Chance: A free slot. Player can fill this slot with the sum of all five dices.

The goal of the player is to get the highest possible score that they can. The game is entirely dice base with no place for a difficulty selection due to being a single player game. There is no lose state for this game.

III. Design

a) General

The game rely on a central GameState object to keep all gameplay values and logic. All views draw their data from this object which is initialized with the diceRollView. The GameState is design in such a way that if all the data within it is saved, the game can just load the whole object and be able to continue the game where it left off. However, no functionality to save and load the GameState object is implemented.

GameState itself contain the various Boolean flags to determine which stage of gameplay it is in as well as the data of the dice and the 13 score groups. It also contain all the functions to determine if a set of roll can be used to fill any of the score group.

Similarly the high score list which is an array containing the MatchRecord object is pass to the HSVView on startup. However the functionality to save and load the array<MatchRecord> is not completed by the end of the project.

The game is design to be able to be with two people with all gameplay logic in place but since the view file for the 2Player match records is not complete, the ability to play that mode is disabled although it can be reenabled and played but not complete

b) DiceRollView

9:23

Current Turn: 1
Remaining Roll(s): 2



Roll Dice 🎲

Open Score Board 📝

Open Instruction 📖

This is the main game view of the app. The GameState is initialized alongside this view and is passed from here to all the child view. From this view, the player can open the instruction as a sheet that can be swipe down to close, the score sheet as a full screen popup that link to a Bool flag to open and close and when the isEndGame flag is trigger, it will call a version of the High Score View that will take in the data from GameState to append a new record to the high score list.

This view contain the visualization of the dice with functionality to lock the dice value by tapping directly on the icon which will change the image variables of the dice to a black version. Tapping the dice again without rolling will switch the image back to the white version. The function to switch dice image have its own sound effect.

The roll button that will call not just the dice functions but also all the score group checkers to ensure they received all the data before they render. The roll button will stop its roll function after running out of roll. The roll button have its own sound effect.






The view also contain its own music which will played until the player return to the main menu.

A hidden button to instantly end the game is available to be enabled in the code for testing purpose.

c) ScoreSheetView

9:25 📶 🔋

[Close Score Sheet](#)

Upper Point Group

Aces	1	Commit
Twos		Commit
Threes	3	Commit
Fours	8	Commit
Fives	10	Commit
Sixes	18	Commit
Upper Group Total Score:	36	

Lower Point Group

Three of a Kind	Commit
Four of a Kind	Commit
Full House	Commit
Small Straight	Commit
Large Straight	Commit

The most important view of the game. The view is actually a full screen pop up. Each Score group is actually a custom view that is link to a ScoreGroup Object within the GameState.

This custom view display the scoregroup name a dynamic rendering of the points. If the group is not selectable then no number is display. If the group is selectable, a gray number linked to the group currentValue variable is display. If the group is filled, a black number is display. In the example in the left, since the dice roll contain a [3] dice, the score group for Threes has a gray 3. If the player click on the commit button, it will assign the currentValue to the finalValue of the active player and change the flag linked to the diceRollView which will close the Score Sheet. The commit button will only work if the group isSelectable Boolean is true.

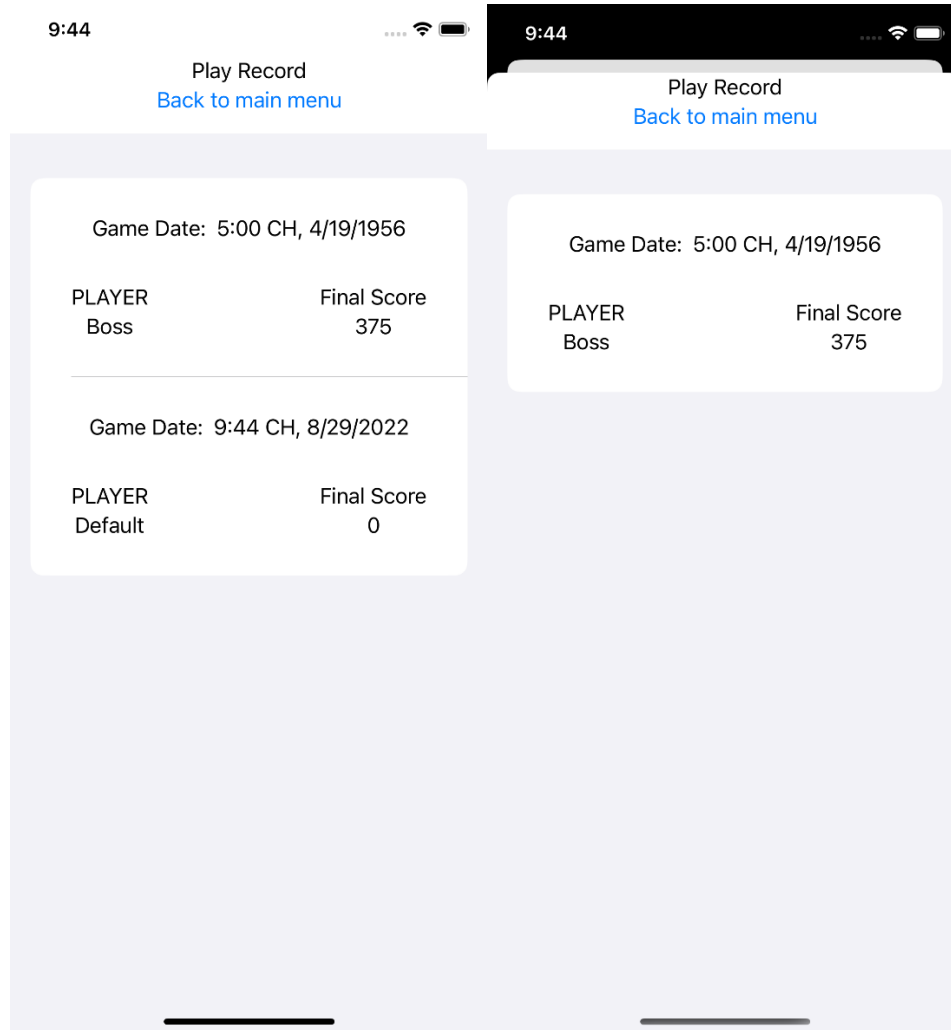
The Open Score Board button and Close Score Sheet button have a shared sound effect.

The commit button when click in its selectable state have its own sound effect.

Due to explicit use of black colored text, the view is not readable in dark mode.

A hidden button to debug the diceView ability to automatically end by adding 5 turns is available to be enabled in the code for testing purpose.

d) HighScoreView



The data use to populate the HSView is stored in the MainMenuView and either passed directly into HSView if its open at the main menu or is passed through a binding in the diceRollView which then passed it to the HSView. In the main menu, HSView is a called as a sheet and can be close by swiping down. The HSView that appeared after the game end can only be closed by clicking the “Back to main menu” button.

e) Unimplemented/Unfinished views

A view for storing and displaying 2 Player match is not fully implemented along with its corresponding subview.

f) Known Bugs.

IV. Media

See Demo.mp4 for full representation of a full game

9:42



[SINGLE PLAYER](#)

TWO PLAYER - TBA

[HIGHSCORE](#)

MATCH HISTORY - TBA

[How To Play](#)

Game by Do Ha Minh Long - S3634734