Ryan Gavino, Dominic Orsi, Cameron Williamson, Connor Deide

Yahtzee Project Plan

Yahtzee Final Project Description

This final version of Yahtzee will simulate a full game of Yahtzee according to the original rule set provided for the first homework assignment. The program can handle performing turns with up to 4 players at once. The game will begin with the default ruleset where each player holds five six-sided dice in one hand and can roll them up to three times in a given round. At the start of each round, the first player will perform their full turn until after the player has chosen a row to fill with a score. Following this, the game will switch to the next player in order and continue to finish turns until every user's turn has completed. This will complete one round and begin another until the final player's scorecard has been completely filled. Once play has stopped, the program will declare the winner as the player with the greatest score among the participating users.

A full turn will begin with the current player being prompted to throw their first roll. Once the player hits the roll button, the dice will visibly randomize themselves to give the player their first given hand. They can choose to keep rolling for a better hand up to an additional 2 rolls after the first before they must move on to selecting a score. Also, before each additional roll the player can choose specific dice they wish to keep alongside any newly rerolled dice to increase their chances of getting a favorable hand. Once a hand is ready to be submitted, the user will see within the scorecard all of the possible scoring rows they are allowed to choose. The last thing for the current player to do is select a row to fill in and their turn will be over. After that player's turn is over, the current view of both the scorecard and the dice will be refreshed to show it is now the next player's turn, while also updating the total score display for the previous user. These turns will continue to iterate through to the last player's selection of their score, at which point that program will check if the game is over (i.e. all player's scorecards are filled up. If it isn't over, a new round begins starting with the first player and turns will continue as normal, filling in more and more rows in each other's scorecards. If the game is indeed over, the view will update and a game over message will be displayed to declare who won this session of Yahtzee with the highest total game score.

Functional Requirements

The app will allow the user to specify how many players will be participating, and allow them to enter names for each player.	
Priority	Medium
Purpose	Foundation for implementing multiplayer and user customization for Yahtzee
Inputs	Amount of players (1-4), names (Strings)
Operators	Name textboxes, Player
Outputs	Game will iterate through each player every round, scoreboard in GUI will display each player participating

For a player's turn to begin, the user will click the roll button to begin their turn.	
Priority	High
Purpose	The initial roll to officially begin their turn
Inputs	Clicking a button
Operators	Roll button, Hand, Die (5 of them)
Outputs	Every Die in the player's Hand will be rolled, updating both the model and the view

Before each additional roll, the player can select the dice that they wish to keep rather than reroll with the rest in their hand.	
Priority	High
Purpose	Integral feature for the dice-rolling part of Yahtzee
Inputs	Clicking (highlighting) dice, should only be able to select dice after initial roll
Operators	Hand, every Die
Outputs	Selecting dice to keep will visibly highlight the view of each die chosen. After selecting and then hitting the roll button, the keepers

	selected should not change.

The program must limit the player on how many rolls they can use before selecting a score	
Priority	Medium
Purpose	Prevents player from cherry picking dice for best possible score
Inputs	Counter for amount of rolls previously thrown
Operators	Roll button
Outputs	If the player has reached the maximum amount of rolls (default: 3 rolls), the roll button will be disabled and the player must select a score with their given hand

The scorecard should have selectable rows that allow the player to choose which possible score they want to fill in for their turn.	
Priority	High
Purpose	Represents the second half of a regular turn of Yahtzee
Inputs	Scorecard uses the current Hand to calculate possible scores. Each row that has not been previously filled will be selectable for the player to choose.
Operators	Hand, Scorecard, Scorecard rows
Outputs	After each roll, the Scorecard view should update with possible scores to choose. Once the player has clicked on a row, that row will be filled with the calculated score and update the total score.

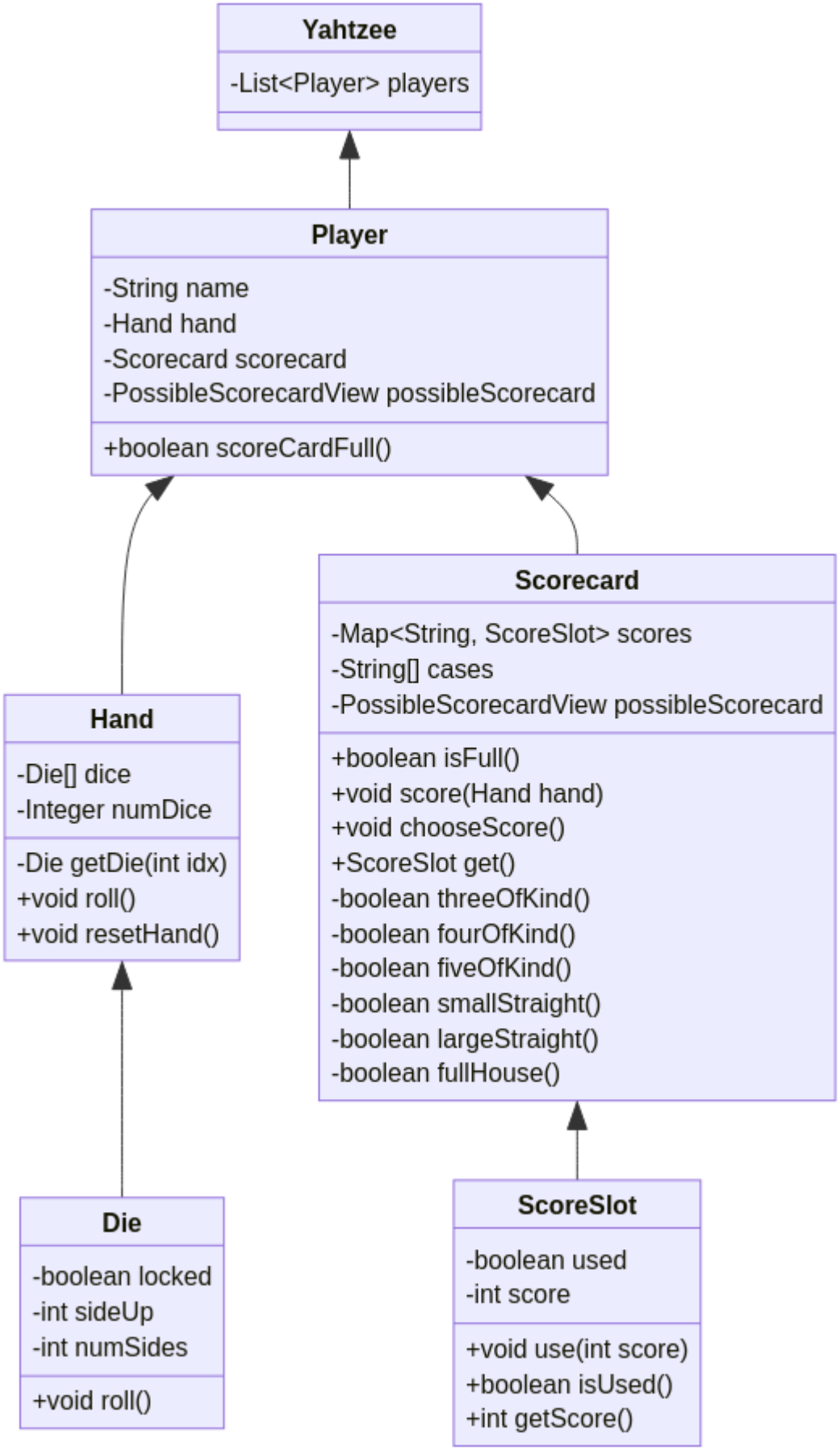
Once the current player has selected a scorecard row to fill, the program should swap to the next player in line and begin a new turn for them.	
Priority	High
Purpose	Iterates through the specified amount of

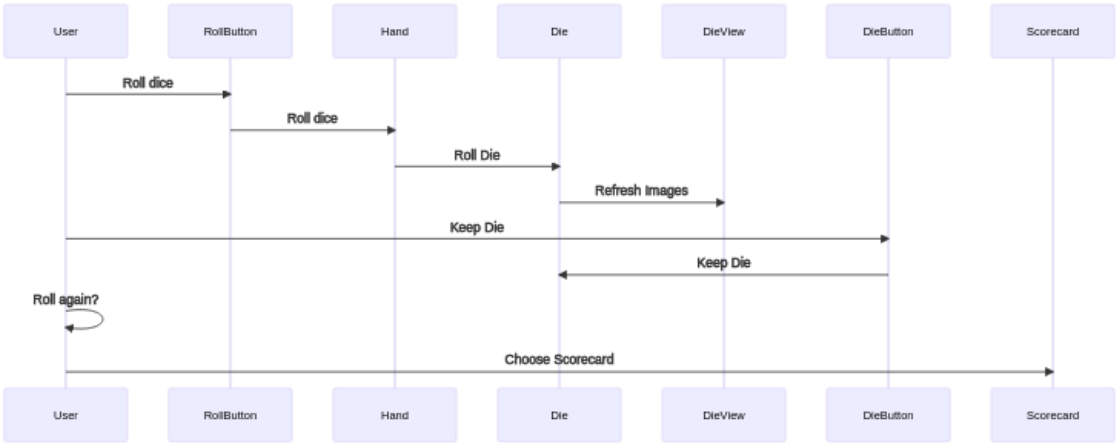
	players for multiplayer
Inputs	Scorecard, Player selecting a row to fill
Operators	Player, Scorecard, Hand, every view
Outputs	After the previous player has selected a score, the scorecard will swap to a new color alongside the new player's current scores. The Hand view will be refreshed indicating a new turn is ready to begin following the press of the roll button. Additionally, the total score display for the previous player is updated.

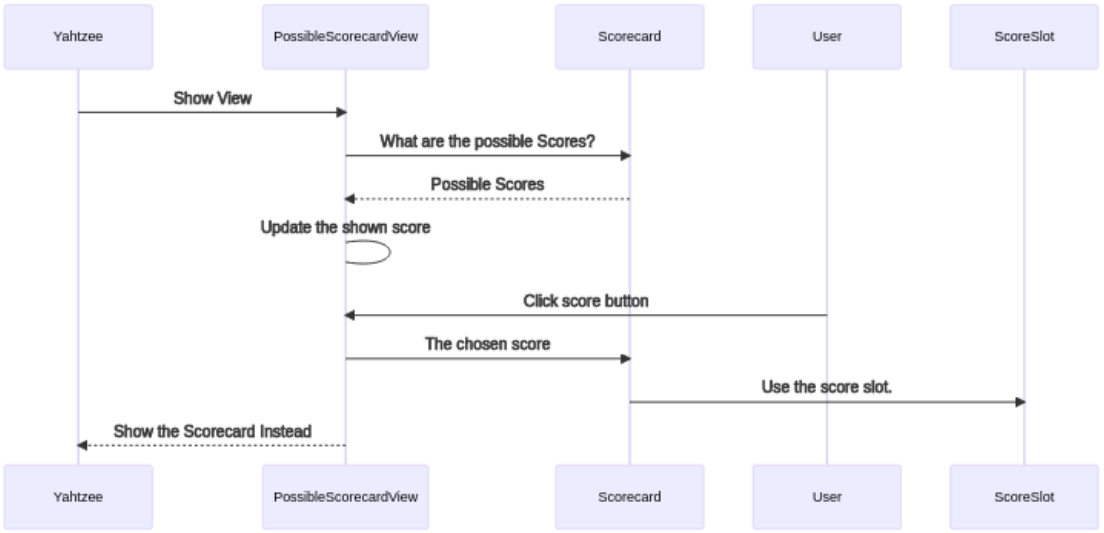
After the last player in order has ended their turn, the program will check if the game is over.	
Priority	High
Purpose	Method of exiting the game loop
Inputs	Checks how many rows are filled in the last player's scorecard
Operators	Scorecard, Player
Outputs	If the game is not over, the game will start a new round by going back to the first player and iterating through the player list once again. If the game is over, all buttons will be disabled and a game over message will be displayed.

Once the game is over, the game will declare the winner as a game over message.	
Priority	Low
Purpose	Somebody has to win right?
Inputs	Total score from each player's scorecard
Operators	Scorecard, Players
Outputs	A message will pop up saying, "[insert name] is the winner with a score of [score]!"

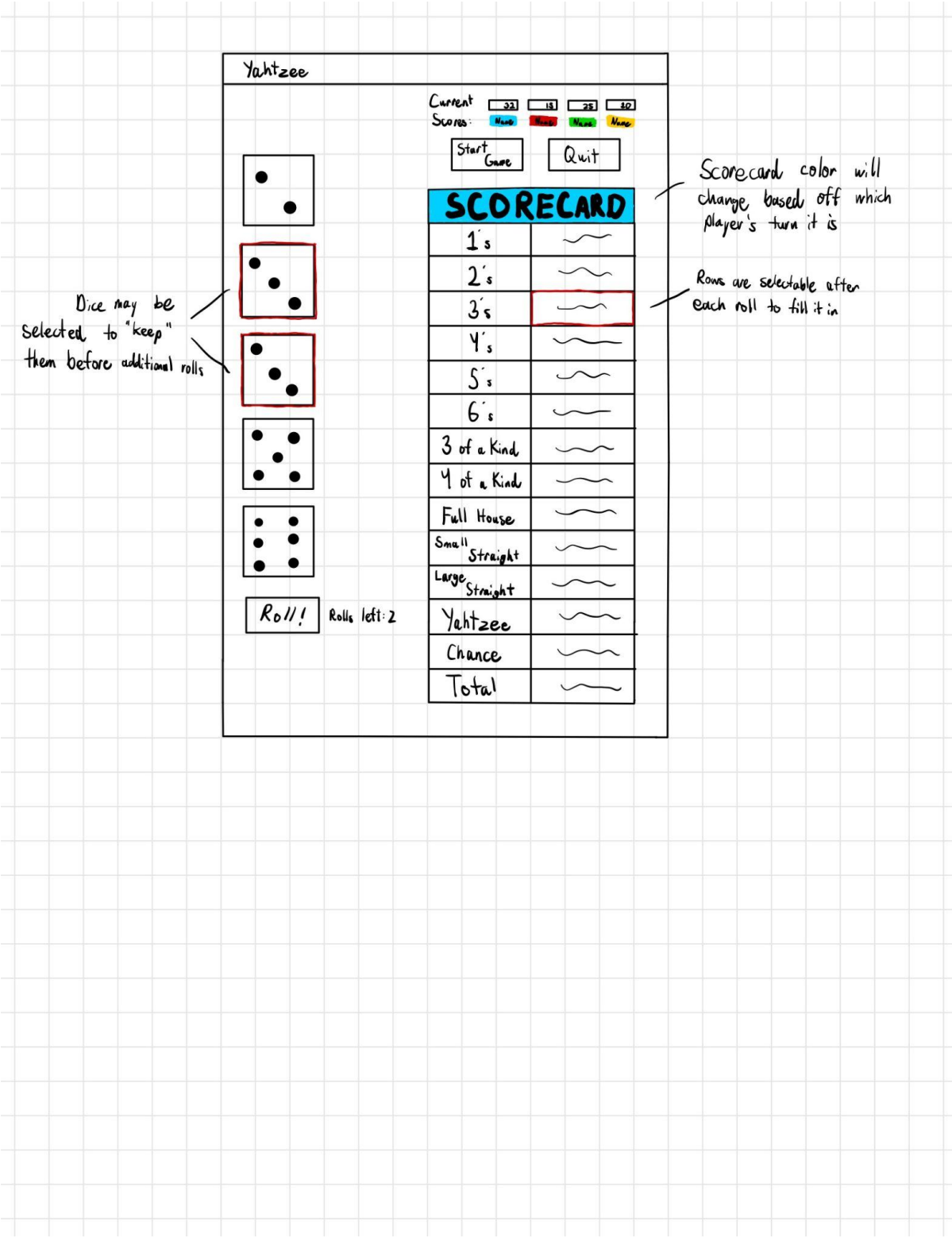
At any point in the game, a player can stop the program with a Quit button.	
Priority	Low
Purpose	Offers option of ending play early
Inputs	Clicking Quit button and not just the X button in the top right of the window
Operators	Quit button
Outputs	The game will terminate.


Yahtzee
-List<Player> players
Player
-String name
-Hand hand
-Scorecard scorecard
-PossibleScorecardView possibleScorecard
+boolean scoreCardFull()
Scorecard
-Map<String, ScoreSlot> scores
-String[] cases
-PossibleScorecardView possibleScorecard
+boolean isFull()
+void score(Hand hand)
+void chooseScore()
+ScoreSlot get()
-boolean threeOfKind()
-boolean fourOfKind()
-boolean fiveOfKind()
-boolean smallStraight()
-boolean largeStraight()
-boolean fullHouse()
Hand
-Die[] dice
-Integer numDice
-Die getDie(int idx)
+void roll()
+void resetHand()
Die
-boolean locked
-int sideUp
-int numSides
+void roll()
ScoreSlot
-boolean used
-int score
+void use(int score)
+boolean isUsed()
+int getScore()


User
RollButton
Hand
Die
DieView
DieButton
Scorecard
Roll dice
Roll dice
Roll Die
Refresh Images
Keep Die
Keep Die
Roll again?
Choose Scorecard
User
RollButton
Hand
Die
DieView
DieButton
Scorecard


Yahtzee
PossibleScorecardView
Scorecard
User
ScoreSlot
Show View
What are the possible Scores?
Possible Scores
Update the shown score
Click score button
The chosen score
Use the score slot.
Show the Scorecard Instead
Yahtzee
PossibleScorecardView
Scorecard
User
ScoreSlot

Yahtzee Mock GUI Design



Future Project Plan (Milestones/Issues)

- Create dev branch separate from main
- Stub out planned UML classes + functions
    - Look over UML diagrams one more time
- Implement unit testing methods
- Compare base code from previous Yahtzee's
    - Implement the Hand and Scorecard classes
- Implement Player class
- Begin working on GUI objects (Buttons and views)
- Playtest the program (try to break our own game)
- Prepare for final presentation