MULTIPLAYER YAHTZEE

Blue Bulldogs

Anna, Tyler, Jonathan, and Hailey

PROJECT OVERVIEW

- Multiplayer Yahtzee

  - On every player's turn, they have the option to roll the dice three times

  - On each roll, possible scores are displayed in the current player's scorecard, the player can then choose which dice to keep and roll again

  - After rolling, the player must record a score on their scorecard, regardless if they have any points or not

  - The player can choose to count totals of matching dice, or score points for three of a kind, four of a kind, a full house, a small straight, a large straight, a yahtzee, or simply add total of all dice.

  - The player finishes the game when their entire scorecard is filled out.

- Limitations: No more than eight players
- Assumptions: 5 6-sided dice, 3 rolls per turn

FUNCTION REQUIREMENTS

- Let user pick number of players

- Iterate through players in the game

- Let players enter player names

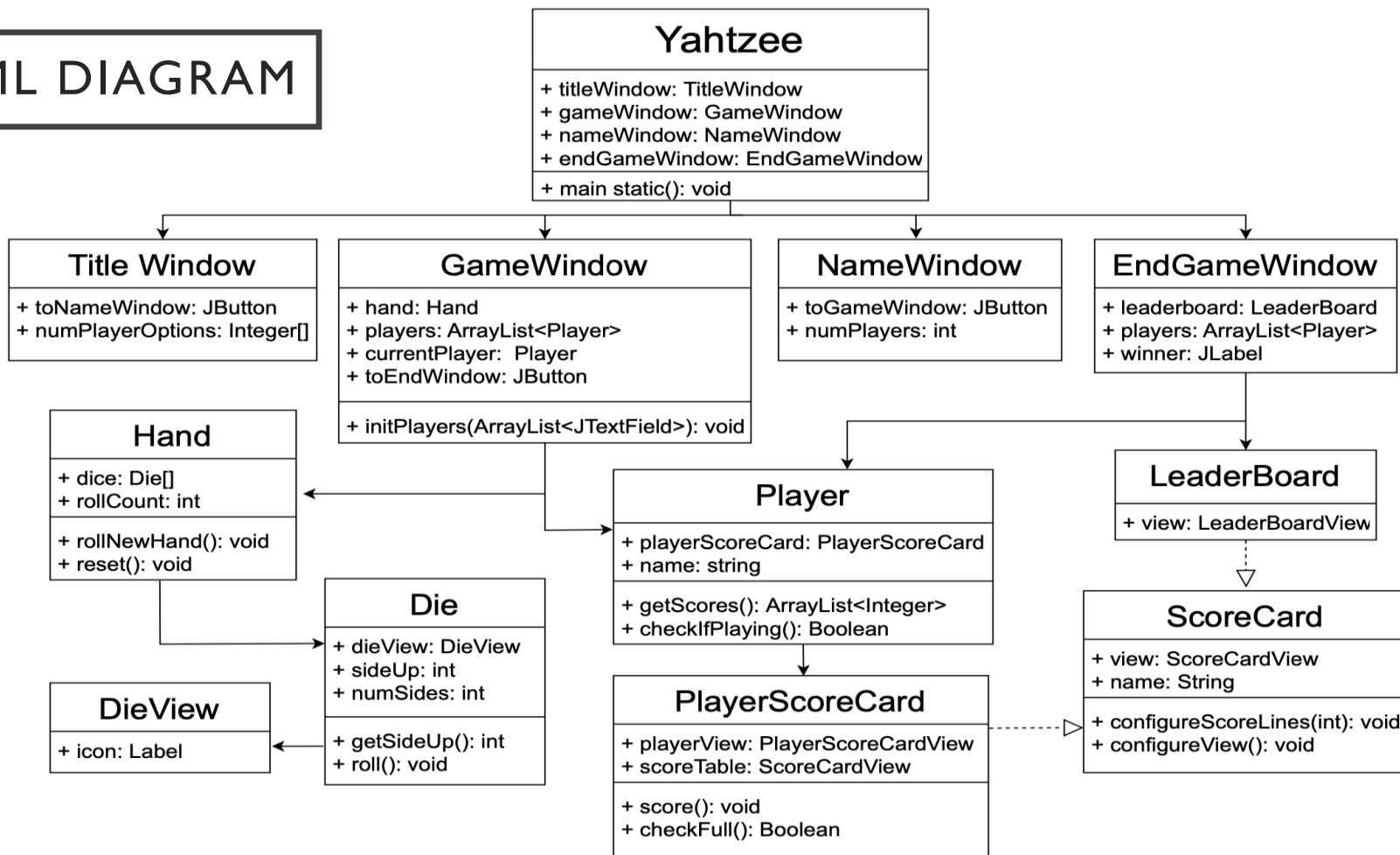- Let user navigate between windows

NON - FUNCTION REQUIREMENTS

- Create cohesive color scheme

- Make each player scorecard a different color

- Display winner in end screen

- Display what roll current player is on

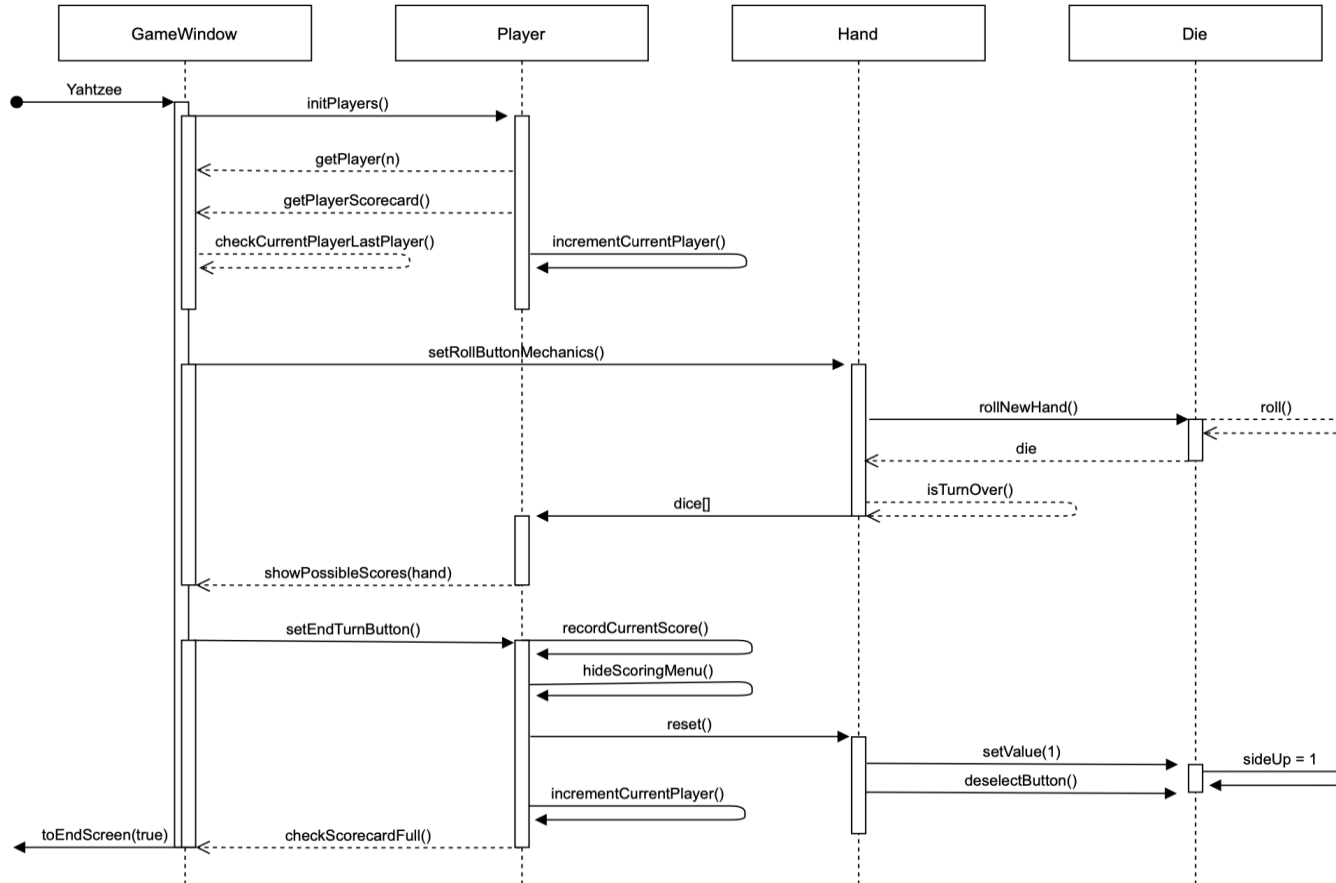- Display who's turn it is at top of screen

PROJECT SOLUTION APPROACH

- What are the major components in your solution design?

  - MVC pattern

  - Game Window Object Handles bulk of gameplay / component interaction

- What game/gui features did you really identify and work towards in your take on Yahtzee?

  - Multiple windows

  - Limit visibility

    - If the user isn't supposed to use a component yet, it isn't visible

UML DIAGRAM

Yahtzee

+ titleWindow: TitleWindow
+ gameWindow: GameWindow
+ nameWindow: NameWindow
+ endGameWindow: EndGameWindow

+ main static(): void

Title Window

+ toNameWindow: JButton
+ numPlayerOptions: Integer[]

GameWindow

+ hand: Hand
+ players: ArrayList<Player>
+ currentPlayer: Player
+ toEndWindow: JButton

+ initPlayers(ArrayList<JTextField>): void

NameWindow

+ toGameWindow: JButton
+ numPlayers: int

EndGameWindow

+ leaderboard: LeaderBoard
+ players: ArrayList<Player>
+ winner: JLabel

Hand

+ dice: Die[]
+ rollCount: int

+ rollNewHand(): void
+ reset(): void

Player

+ playerScoreCard: PlayerScoreCard
+ name: string

+ getScores(): ArrayList<Integer>
+ checkIfPlaying(): Boolean

LeaderBoard

+ view: LeaderBoardView

Die

+ dieView: DieView
+ sideUp: int
+ numSides: int

+ getSideUp(): int
+ roll(): void

DieView

+ icon: Label

PlayerScoreCard

+ playerView: PlayerScoreCardView
+ scoreTable: ScoreCardView

+ score(): void
+ checkFull(): Boolean

ScoreCard

+ view: ScoreCardView
+ name: String

+ configureScoreLines(int): void
+ configureView(): void

SEQUENCE DIAGRAM

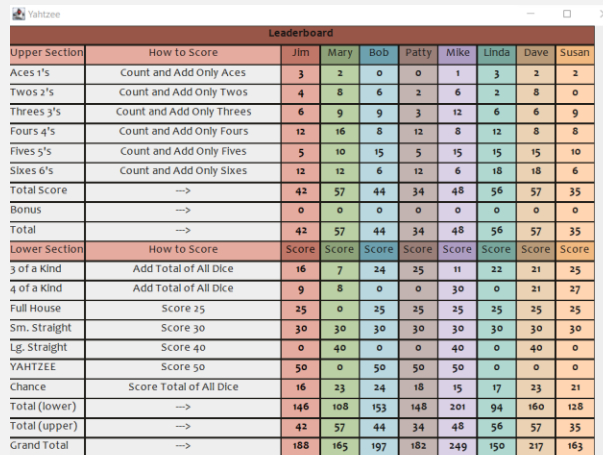# TEAM COLLABORATION APPROACHES

- Messaging tools we used: texting

- GitHub Issues:

  - Tried to make a milestone for every major project requirement and then broke it down in to smaller issues

- GitHub Branches

  - We tried to avoid any merge conflicts by assigning classes to work on

  - Generally careful about communicating any work done on other people's classes

- Did you code 100% separately, or did you do group hacking sessions?

  - Combination of coding separately and as a group

  - Fairly strong base code

TESTING, VALIDATION, AND ACCEPTANCE PLAN

- Testing Approaches:

  - Unit Tests

  - Integration Tests: MVC

  - Functional Tests: functional requirements were met

- What made a deliverable product:

  - Fulfilled all function requirements

  - Tests were built out for most areas of code

  - Satisfied with layout (fulfilled non-functional requirements)

LIVE DEMO TIME

- Major objects:

    - Player, Hand, Die, ScoreCard(s) (model)

    - Window(s) (view)

    - Buttons, Boxes, etc. (controller)

- Button presses do what??

SUMMARY

- Lessons Learned/Experiences:

- Git Issues are important for tracking progression

- Changing the main branch is not the ideal way to fix code

- Communication can make or break a project

- Improvements:

- Expand testing harness

- Add button on end screen to return to title screen

- Implement Lizard-Spock Yahtzee