Sprint 3 Release Document

Planned Requirements Delivered

Requirement 2.1

- Description: The GUI for Poker should show the public cards progressively dealt and the chips bet by each player.
- Story Points: 8
- Status: **COMPLETE** -- The GUI animates the dealing of public cards in a sequential manner, the chip counts for each player are tracked.

Requirement 2.2

- Description: User should be able to specify between 1 to 3 AI opponents.
- Story Points: 3
- Status: **COMPLETE –** The GUI contains a dial that allows a user to hide and display opponents, removing and adding them to the game.

Requirement 2.3

- Description: AI opponents must start with a random amount of chips.
- Story Points: 1
- Status: **COMPLETE --** AI opponents are assigned a random multiple of 50 for their chips upon creation.

Requirement 2.4

- Description: Each player must be initially dealt two cards at the beginning of a game.
- Story Points: 2
- Status: **COMPLETE –**The GUI and game logic support the creation and animation of dealt cards for each player.

Requirement 2.5

- Description: The game must track the number of chips each player holds.
- Story Points: 2
- Status: **COMPLETE –** Internal logic keeps track of each player's chips.

Requirement 2.6

- Description: The game flow must begin with three cards being dealt (flop), then one (turn), then one final card (river), with opportunities to bet after each phase.
- Story Points: 3

- Status: **COMPLETE –** Basic game flow is fully implemented, starting with the initial flop, followed by the turn, and finished by the river.

Requirement 2.7

- Description: AI opponents must have a strategy for deciding when to bet and how much to bet, considering their cards and the public cards dealt.
- Story Points: 13
- Status: **COMPLETE –** Opponents have a decision-making strategy that considers their best possible hand. It also incorporates randomness for more unpredictability.

Requirement 2.8

- Description: The user should be able to bet chips, either matching a current bet or raising the bet, during a round.
- Story Points: 3
- Status: **COMPLETE –** User's betting and raising functionality is fully implemented and accessed with a “bet” button.

Requirement 2.9

- Description: The user should be able to fold, forfeiting their hand and any bet chips during a game.
- Story Points: 3
- Status: **COMPLETE –** The GUI contains a fold button, which allows a user to fold from the current game. The game will still continue in the background, and the user will be given updates after each round.

Requirement 2.10

- Description: The user should be able to immediately bet all their chips during the game, going all in.
- Story Points: 3
- Status: **COMPLETE –** The GUI contains a flashing ALL IN button, which performs the desired functionality.

Requirement 2.11

- Description: The user should be able to check, doing nothing, during a round.
- Story Points: 3
- Status: **COMPLETE –** The GUI contains a check button, which allows the user to check.

Requirement 2.12

- Description: After final bets, all remaning (non-folded) hands are ranked based on standard hand precedence, the winner receiving all chips that were bet.
- Story Points: 5
- Status: **COMPLETE –** Winner determination and payout allocation are completely operational.

Requirement 2.13

- Description: AI opponents must be removed once they run out of chips.
- Story Points: 2
- Status: **COMPLETE –** AI opponent removal upon running out of chips is fully implemented.

Requirement 2.14

- Description: The user should be able to exit Poker to the main menu after folding or after a game concludes, leaving with their remaining chips.
- Story Points: 1
- Status: **COMPLETE –** The exit button permits the user to leave at almost any point. The user will have to forfeit any staked chips if they leave during a game, however.

Requirement 2.15

- Description: The user should be forcefully exited to the main menu upon losing all chips in Poker.
- Story Points: 1
- Status: **COMPLETE –** User removal is fully implemented upon losing all chips.

Requirement 2.16

- Description: Player wins and exits to the main menu if all AI opponents lose all of their chips.
- Story Points: 1
- Status: **COMPLETE –** The endgame scenario, where all opponents are defeated, is implemented, moving the player to the main menu, and allowing all opponents to reset.

Requirement 2.17

- Description: The rules must be accessible to the user through the interface.
- Story Points: 1
- Status: **COMPLETE –** The rules button is implemented and adequately explains the rules of Poker

Requirement 5.4

- Description: The card deck for Sabacc will consist of 62 cards, 30 positive and 30 negative, from values 1 to 10, with 3 of each corresponding to the suite, and two 0 cards.
- Story Points: 5
- Status: **COMPLETE –** The Sabacc Deck is fully implemented, with custom icons for each card to be used in the GUI.

Requirement 5.8

- Description: At the beginning of the game, all players should be dealt 2 cards, and another card should be dealt and displayed faceup representing the discard pile.
- Story Points: 5
- Status: **COMPLETE –** The full initialization sequence, including game logic and GUI animation steps, is implemented.

Requirement 5.9

- Description: A game of sabacc should proceed through three rounds, in which users are given the opportunity to draw, swap, stand, or junk.
- Story Points: 8
- Status: **COMPLETE –** The implementation moves through three rounds, after which point the game over sequence begins.

Requirement 5.11

- Description: When a player draws, a card should be dealt and added to their hand, and the player should be able to optionally add a card to the discard pile.
- Story Points: 5
- Status: **COMPLETE –** The full drawing with optional discard, complete with GUI animations, is implemented.

Requirement 5.12

- Description: When a player swaps, they may choose a card from their hand and swap it with the card at the top of the discard pile
- Story Points: 5
- Status: **COMPLETE –** The complete game logic and GUI animations for card swapping is fully implemented.

Requirement 5.13

- Description: When a player stands, they finish their turn without changing their hand.

- Story Points: 2
- Status: **COMPLETE –** The game logic for standing, as well as GUI processes, is fully implemented.

Requirement 5.14

- Description: When a player junks, all of their cards go to the discard pile and they are out of the game, forfeiting any chips bet.
- Story Points: 3
- Status: **COMPLETE –** GUI animations for junking is complete. After animations for giving up cards, the game concludes in the background, and the user is presented with the final results.

Requirement 5.17

- Description: Each player's chip balance should be tracked.
- Story Points: 1
- Status: **COMPLETE –** Chip tracking is fully implemented and integrated with the wider casino apparatus.

Requirement 5.18

- Description: The user should be able to bet chips, either matching a current bet or raising the bet, during a round.
- Story Points: 8
- Status: **COMPLETE --** Betting and raising is fully implemented, with buttons added to the GUI. The user bets after all AI make their bets, and the user is presented with a bet summary.

Requirement 5.19

- Description: After each round, there should be functionality for rolling 2 six sided dice, and the result should be displayed by the GUI.
- Story Points: 5
- Status: **COMPLETE –** Dice rolling is implemented. The dice are represented as two numbers on the screen. They are randomly generated at the end of each round.

Requirement 5.20

- Description: If the result of the 2 dice roll is doubles, players must discard all of their cards and redraw a new hand equal in card number size to their previous hand.
- Story Points: 5
- Status: **COMPLETE –** The “shifting” mechanism is implemented for all players. Full GUI animations and game logic actions are completed.

Requirement 5.21

- Description: At the end of the final round, the winner should be determined by adding the card values in a players hand together, and comparing them against each other.
- Story Points: 5
- Status: **COMPLETE –** Winner determination based on hand quality if fully implemented. In a complete tie between the user and an AI opponent, the user wins.

Requirement 5.22

- Description: The player closest to 0 wins. If two players are the same numeric distance away, the player with the closest positive value is the winner.
- Story Points: 2
- Status: **COMPLETE –** This component of winner determination is fully implemented, and factored into the winner determination process.

Requirement 5.23

- Description: If there is still a tie or more than one player gets 0, the hand with more cards wins.
- Story Points: 2
- Status: **COMPLETE –** This factor is also fully implemented in winner determination.

Requirement 5.24

- Description: The winning player should have the chips from the bets after each round added to their casino total.
- Story Points: 2
- Status: **COMPLETE –** Payout allocation is completely implemented.

Requirement 5.25

- Description: If the winning player wins on a "sabaac" which is when they have a final value of 0, the sabaac pot should be emptied and the chips held should be added to the player.
- Story Points: 3
- Status: **COMPLETE –** Checking and handling of the sabacc case is fully implemented, with the sabacc pot being added to the payout and reset to 0.

Requirement 1.2

- Description: Remove user from casino if user reaches 0 currency.
- Story Points: 1

- Status: **COMPLETE –** This feature is complete. When a user runs out of chips, they are forced to exit the casino program.

Additional Requirements Achieved

Requirement 5.1

- Description: The GUI for Sabacc will show the player's cards faceup, the cards of the other players facedown, the dice, and the number of chips contained in the Sabaac pot.
- Story Points: 8
- Status: **COMPLETE** – All specified elements in the GUI are completed implemented. One notable deviation is the publicity of cards that an opponent takes from the discard pile. These cards are seen by all, so they are not immediately covered after an opponent takes them, since all players “see” the opponent take from the discard pile.

Requirement 5.2

- Description: The GUI should display the discard pile, or an empty space if the discard pile is empty, and the chip values that have been bet for each game.
- Story Points: 8
- Status: **COMPLETE –** The discard pile, and bet values, including the game pot, sabacc pot, and the stakes for each player, are all shown in the GUI.

Requirement 5.3

- Description: The player should be able to choose between 1 and 3 AI opponents.
- Story Points: 5
- Status: **COMPLETE –** The same kind of dial implemented in Poker is implemented in Sabacc, allowing for addition and removal of players from the game. The initial requirement of up to 7 players was toned down to 3 due to GUI real estate constraints.

Requirement 5.5

- Description: The GUI should show the positive cards as green, the negative cards as red, 0 cards as any color, and the different suites as represented by a circle, triangle, or square.
- Story Points: 3
- Status: **COMPLETE –** Card graphic design is fully complete and matches specifications.

Requirement 5.6

- Description: To begin a game of sabaac, the player and every AI opponent must "buy in" and these chips will go to the sabaac pot.
- Story Points: 5

- Status: **COMPLETE –** Sabacc pot initial additions, as well as entry-fee deductions from active players, are fully implemented and reported in the GUI.

Requirement 5.7

- Description: The value in the sabaac pot should be randomized between accesses to the Sabacc mode and between games to simulate activity.
- Story Points: 2
- Status: **COMPLETE –** Sabacc pot randomization occurs every time the user leaves the Sabacc menu, simulating that it has been altered between accesses.

Requirement 5.10

- Description: In each round, the order of play should remain the same, and the user should play last.
- Story Points: 3
- Status: **COMPLETE –** The game order is fixed, starting with Chewbacca (left), Lando (center), Han (right), and ending with the user. The betting order is identical. If a player is defeated, they are excluded from the ordering.

Requirement 5.15

- Description: At the end of each round, the AI opponents should be able place a bet based on the quality of their hand, either matching a current bet or raising it.
- Story Points: 8
- Status: **COMPLETE –** AI betting decisions are implemented. AI have the capability to match or raise the current bet based on the value of their hand. If they can't match the bet, they simply bet what's left of their chip amount.

Requirement 5.16

- Description: AI opponents must initialize with a randomly generated chip balance.
- Story Points: 3
- Status: **COMPLETE –** The initial chip randomization for AI opponents is implemented. They are assigned a random multiple of 50 chips.

Requirement 5.26

- Description: After each game, the player should have the option to play again, or leave to return to the Casino.
- Story Points: 2
- Status: **COMPLETE –** Replayability and, naturally, exiting are fully implemented through the start and leave buttons on the GUI.

Requirement 5.27

- Description: Losing players forfeit all chips bet during the game. Once a player reaches a chip amount of zero, they are removed from the game(menu in case of the user)
- Story Points: 3
- Status: **COMPLETE –** Player removal in the instance of losing all chips is fully implemented. When an opponent is defeated, they are simply hidden from view and excluded from further games. When a user is defeated, they are forced to the main menu.

Requirement 5.28

- Description: The player is exited to the main menu with all won chips once all AI opponents are removed.
- Story Points: 2
- Status: **COMPLETE –** The total victory condition is fully implemented, and activates once all three opponents are defeated.

Requirement 5.29

- Description: The player should be met with genre-appropriate background music whilst playing sabacc.
- Story Points: 3
- Status: **COMPLETE –** Any time the Sabacc Screen comes into focus, one of four songs from the Star Wars franchise begins playing in the background. After a song ends, another (potentially the same) song is randomly chosen to begin playing. The music stops when the player returns to the main menu.

Story Points Promised: 122

Story Points Delivered: 177

Goals for Next Sprint

With this sprint, all initial requirements have been completed. As such, the final sprint will not have any strict, numerical requirements. **The primary goal** of the Final Sprint can be summarized as follows:

- Resolve any platform-volatile GUI complications.
    - Some GUI elements that appear acceptable on one platform (Mac, Windows, etc), appear differently on other platforms.
    - Examples: **main window chip amount, poker chip amounts**
- **Perform final verification and validation checks for all requirements.**

- **Complete the final project presentation.**