Project 3 Requirements

Foundations

Requirement 1.1

The program simulates a user's experience with a casino. Naturally, the user must start out with some arbitrary amount of chips (1000), which they can take with them throughout the four games. The wins and losses of a user must be carried into further games. In other words, winning and losing chips alters a global chip value that is used by all games.

Requirement 1.2

This is a low-priority requirement that focuses more on the program as a whole, not any specific game. Once a player has been kicked to the main menu upon losing a game, the program should also forcefully exit when the user loses all chips. There should be a message displayed to the user that explains the situation (no chips, get out), and the program should give the user enough time to read the form before exiting (5-7 seconds most likely).

Requirement 1.3

The main menu of the digital casino. This will be the opening screen for players on start up, and should display the title, the games, their chip amount, and a button to close the program. Clicking on any of the buttons with game titles should change the screen to that respective game screen.

Requirement 1.4

A traditional 52 card playing deck with four suits (hearts, diamonds, spades, and clubs) and 13 ranks (Ace, 2, 3-10, Jack, Queen, King) must be implemented. There must be functionality to draw a random card from the deck, removing it as well. This object should be able to be used for BlackJack, Poker, and possibly Sabacc.

Requirement 1.5

It must be possible to “shuffle” the deck. When the deck is shuffled, all previously removed cards should be restored. Essentially, shuffling must reinstantiate the card deck to promote replay ability.

Poker

Requirement 2.1

The GUI must contain the basic essentials of the Poker game. It must display the public cards as they are dealt. They can either be hidden until they are dealt(unveiled) or added to the GUI in an animated manner. Similarly, there must be some way to display the current number of chips bet by each player, either as an aggregate total(the pot that the players stand to win) or as individual figures. Similarly, the chips bet can be represented either in a numerical form or by the presence of chip icons.

Requirement 2.2

AI opponents must be present in the Poker game. They can optionally be represented with avatars if possible. More importantly, the user should be able to specify the initial number of AI opponents in the game. Once specified, only the proper amount of AI opponents should be displayed (no grayed out unused opponents). This initial number should not be able to be changed once the user first starts a game of poker. For instance, after one game of Poker, the user should not be able to remove any other opponents. However, if the user chooses to leave the Poker mode entirely, the AI opponents should be completely reset, allowing the user to initially remove opponents once again.

Requirement 2.3

AI opponents should be initialized with a randomized balance of chips (between 1000-10000 chips). This balance should be reset every time the player reenters Poker mode. Effectively, each new entry into the Poker mode spawns new AI opponents with a new chip balance.

Requirement 2.4

Each player, both human and AI, should receive two cards upon the start of a game. The cards for each AI opponent should be displayed with the card face down, preventing the player from seeing the cards. However, the player should easily be able to see their own cards (display the proper card face in the GUI). The cards should arrive to each player in an animated fashion, originating from a location on the GUI representing the deck of cards.

Requirement 2.5

The game must track not only the chip balance of the human player, but also the chip balance of all the AI players throughout the Poker games.

Requirement 2.6

After the initial card dealing, three public cards must be made immediately visible in the center of the GUI. This is the flop phase. After the flop, all players, human and AI, must be prompted for game action. After all player make an action, another public card must be added to the center of the screen, known as the turn. All players must again be prompted for action. Finally, a fifth card should be added to the center (the river phase), prompting the players for action. These public card additions should be done in an animated manner, with the cards originating from the card deck location on the GUI.

Requirement 2.7

AI opponents must have some discernible strategy when determining whether to bet, check, or fold. This decision should consider the best possible poker hand using their private cards and the public cards displayed. If their best possible hand is within the middle third (~4-6) of the hand precedence hierarchy, they should check, betting no chips. If their best possible hand is in the top third, they should match the current bet plus some random percentage of their current chip balance. If their best hand is in the lower third of the precedence, they should fold if they are in the river phase. Otherwise, they should just check.

Requirement 2.8

After a phase, the user must have the ability to bet chips, with some limits. They must bet a minimum of the maximum bet placed by another player during this phase. If they have insufficient chips, they should not be able to bet. They should be free to bet over the current maximum, raising the bet. If so, any opponents going after the user would also be bound by this new maximum. This action should update a GUI element displaying the current bet maximum if applicable and the element displaying the current pot for the game.

Requirement 2.9

Alternatively, the user must also be able to fold during a phase. This effectively forfeits a game, losing all chips they currently bet. For time-saving, this action will effectively end the game. The turns and bets must continue in the background so that the proper AI opponent gets their updated chips. However, the user should not be forced to sit through this process, instead receiving some message conveying the winner and their new chip amount.

Requirement 2.10

During betting, the user should be given a special “all-in” option, which will immediately put all of their chips into the game’s pot. They will not be permitted to bet after this, nor will any opponent during the phase unless they can match or raise the bet.

Requirement 2.11

Conversely, the user should be permitted to check, doing nothing, after a phase. This will simply initiate the next phase of the game, unless an AI opponent goes after the player.

Requirement 2.12

After the final round of actions following the river phase, all hands should be made visible in the GUI. The best hands of each player should be ranked, with the highest-ranking player receiving all chips in the pot. The rankings should use the precedence described in the following website https://www.pokernews.com/poker-hands.htm. The chip rewarding could be implemented in an animated manner or through a message displayed to the user.

Requirement 2.13

After an AI opponent folds or loses a game of Poker, their chip balance should be checked. If it is zero, they should be “removed” from the table, making their avatar invisible. A message should be displayed to the user after such an event, congratulating them for defeating an opponent.

Requirement 2.14

The user should be able to freely exit to the main menu upon folding or finishing a game of Poker, leaving with their proper amount of chips. If they chose to leave during the game without folding, they must forfeit any chips bet during the game.

Requirement 2.15

Like the AI opponents, the user must be removed to the main menu after folding or losing results in a chip balance of zero. A message should be displayed to the user explaining this before forcing them to the menu.

Requirement 2.16

If all AI opponents are removed from the GUI upon losing all their chips, the player should be rewarded with a message congratulating them for winning. Then, they should be forced back to the main menu, so that new AI opponents can be regenerated.

Requirement 2.17

There must be a button in the GUI that displays a message explaining the basic rules, game flow and moves, and the hand precedence of Poker.

BlackJack

Requirement 3.1

During Blackjack, the player must be able to allocate an amount of chips as their bet before a given round of Blackjack. This means allocating until they've put in a max bet or until they've started the game with the appropriate Start button. This must be implemented cautiously so that when the player hits 0 chips after putting in a max bet, they are not booted from the game for having no chips to bet.

Requirement 3.2

There must be a button in the GUI to begin a game of Blackjack. When pressed, the player and dealer must be issued two cards randomly from a common deck. These dealing should be animated as well. This promotes fairness, and allows the game to proceed.

Requirement 3.3

The GUI must display every card dealt to the player in its revealed state, so that the player can make sound decisions. The first of the dealer's cards should be hidden, or face down, until the player stands.

Requirement 3.4

During Blackjack, after a game has started and the player or dealer have been dealt their hands, the hit button must become available to press. When the hit button is pressed, the player must be given a new card to their hand and then have it animated onto the screen for the player to see. Then, logic must be checked to determine whether the player has "busted", scoring over a 21. If a hit results in a bust, the other player wins.

Requirement 3.5

During Blackjack, assuming the player has started the game not lost by achieving a score of 21, the stand button must be available to press. When the stand button is pressed, no new cards will be issued to the player.

Requirement 3.6

The score used to determine the winner of Blackjack must be dependent on the hand's cards. Each number card's value is equivalent to their number. Face cards (jacks, queens, and kings) all have a rank of ten. The ace can have a rank of one or eleven. When calculating the score of a hand, aces must be assigned in a manner that gets the score closest to 21 without going over.

Requirement 3.7

When calculating the sum of a player or dealer's hand, should the sum exceed the value of 21, even when considering alternate values of all present aces, that player or dealer instantly loses the game.

Requirement 3.8

After the initial dealing, the game flow must begin by allowing the player to hit. Hitting will only become unavailable once the player busts or stands. When the player stands, the dealer must then reveal their hidden card and MUST hit until their score is greater than a soft 17 (two-card hand that has an ace is considered a "soft hand"). Even if the dealer's hand is still less than the player's (ex. Dealer has 18 and can still hit, but player has 19. The dealer should just hit anyway to have the best chance of winning) the dealer must stand regardless. This is to model rules present in certain real-world casinos. The game flow must proceed exactly in this order, with no way to circumvent it.

Requirement 3.9

Assuming neither the dealer nor player have achieved a “bust”, the final winner of blackjack is determined by evaluating the score of each hand, per requirement 3.6. If the player’s score exceeds the dealer’s, or the dealer busted previously, the player is the winner.

Requirement 3.10

In the event where the player triumphs over the dealer and wins with their hand, they receive double their initial bet.. For example, if the player has 50 chips and bets 25, they go up to 75 (their post bet balance would be 25, and their winnings would be 50). In other words, the player wins double their chip bet amount.

Requirement 3.11

In the events where the dealer triumphs over the player, or the player busts (exceeds 21) while opting to hit, the game displays a loss message, and the Blackjack game resets for a new round. No chips are removed from the player’s chip balance, but no chips are added as well. This means when a player starts with 50, bets 25, and loses, they remain at 25 chips total remaining.

Requirement 3.12

In the Blackjack GUI, there must at all times be a button to leave the game and return to the main menu screen. This should also reset any properties of Blackjack to their default or starting configuration, allowing a new game to be played in return. It should also be noted that when leaving in the middle of a bet, the player will not regain the chips they bet with, and they will instantly be counted as forfeit.

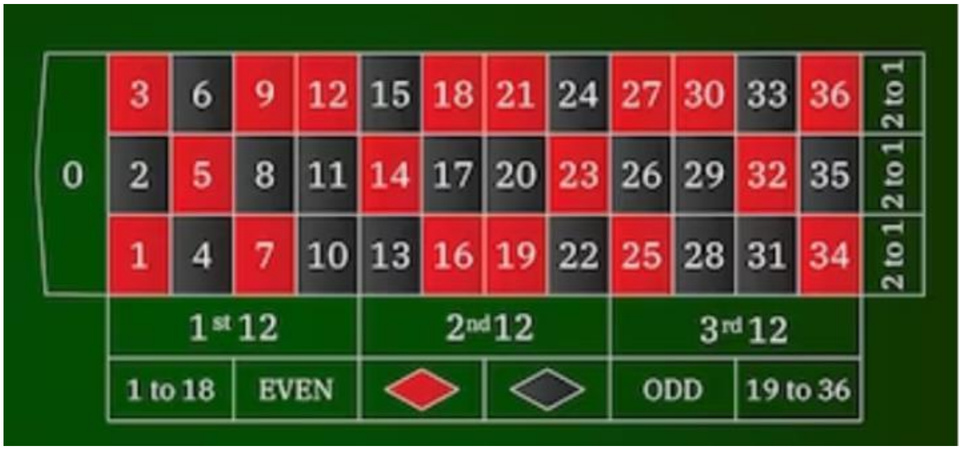
Requirement 3.13

In the Blackjack GUI, there must at all times be a button to display the basic rules of the game to new players or players in need of a reminder. This should include game objectives, card values, player actions, dealer logic, and win against dealer requirements. These rules will appear as a separate window that the player can simply click out of by selecting “Ok.”

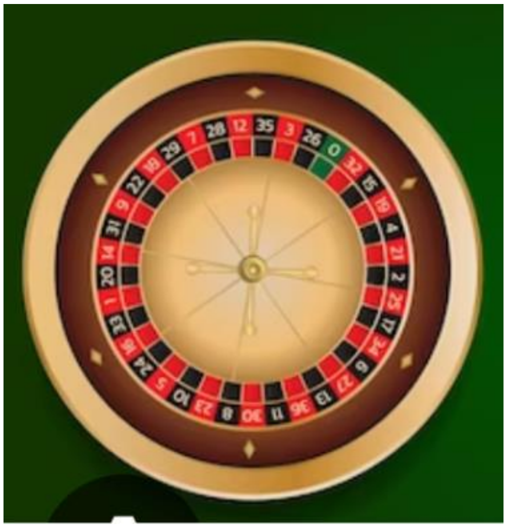
Roulette

Requirement 4.1

The GUI for roulette must display, alongside control buttons like “start” and “leave”, two main items. First, it must present a wheel graphic that is capable of spinning. The wheel must have numbered slots from 0 to 36(see below), with numbers being of the colors shown below. Second, it must present a European styled roulette board, with all the elements shown in the graphic below.



0	3	6	9	12	15	18	21	24	27	30	33	36	2 to 1
	2	5	8	11	14	17	20	23	26	29	32	35	2 to 1
	1	4	7	10	13	16	19	22	25	28	31	34	2 to 1
	1st 12				2nd 12				3rd 12				
	1 to 18		EVEN		♦		♦		ODD		19 to 36		



Requirement 4.2

A user playing roulette must only be allowed to bet from the supply of chips they take with them into the game. Of course, any chips being bet should not be immediately considered “lost”. They should only be considered lost upon losing the gamble or upon forfeit.

Requirement 4.3

A user should be able to place chips, either through dragging or by selection, directly onto a number space on the board. This placing does not inherently need to be an animation, but the bet must be recorded for later review. A bet on a single number should be mapped to a payout thirty-five times the initial bet.

Requirement 4.4

A user should also be able to place chips on the border between two number slots. This specific betting place should be the midpoint of the adjacent border. This signifies a bet that the wheel will land on either of these numbers. This bet should map to a payout seventeen times the initial bet.

Requirement 4.5

A user should be able to place chips at the “far” edge of a three numbered row(furthest from the 12 boxes). This is a bet that the wheel will land on one of the numbers in the row. This bet must map to a payout eleven times the initial bet. The midpoint of the edge should be used as a betting point to prevent overlap.

Requirement 4.6

A user should be able to place their amount on the intersection point of four numbered spaces, preferably in a space that avoids overlap with other betting spaces. This signifies a wager that the wheel will land on one of these four numbers. This bet maps to a payout of eleven times the initial bet.

Requirement 4.7

Due to the existence of the zero space at the left end of the board, it should be possible to place chips on the intersection point between either the 0, 1, and 2 spaces; or the 0, 2, and 3 spaces, denoting a bet on the wheel landing on one of the three numbers. This bet should map to an eleven times payout.

Requirement 4.8

At the far edge of the board, the same area described in requirement 4.5, placing a chip on two adjacent spaces should be considered as a bet on both rows(wheel lands on one of the six numbers in the rows). This maps to a five times payout.

Requirement 4.9

A user should be able to place chips on the red or black box, signifying a bet that the wheel’s number will be black or red. This bet must map to a one times payout(if the user bets one chip, they will receive a chip on success—they have not been deducted a chip as of yet).

Requirement 4.10

A user should also be able to place chips in the odd or even boxes, signifying a bet that the resulting number will be odd or even respectively. This also results in a one times payout.

Requirement 4.11

The player should be able to place chips in the “1-to-18"--signifying a bet that the resulting number is between 1 and 18—or the “19-to-36"--a bet that the number is between 19 and 36—boxes. This bet must map to a two times payout.

Requirement 4.12

The player should be able to place chips in the 1st, 2nd, or 3rd 12 boxes, signifying a bet that the resulting number falls between the first, second, or third twelve number intervals. This bet maps to a two times payout.

Requirement 4.13

The player should also be able to place chips in the boxes to the right in the provided image. These spaces bet on the corresponding column, and map to a two times payout.

Requirement 4.14

After a user is content with their bets, they should be able to press a button to spin the wheel. The GUI must depict the wheel actually spinning and landing on a number slot. This resulting number must be randomly determined for maximum fairness.

Requirement 4.15

There is a special case when the wheel lands on 0. In this instance, all bets placed in the even box must be considered to be lost.

Requirement 4.16

Once the wheel has finished and a resulting number has been chosen, this number must be used to validate or invalidate all bets placed by the user. Any bets that include this number must be rewarded according to the payouts specified above.

Requirement 4.17

Any bets that do not include the wheel-selected number must be deducted from the user’s chip balance. A user should only be subject to forceful exiting to the main menu after all awarding and deductions have occurred.

Requirement 4.18

The user must be free to return to the main menu at any time in the process. Doing so must reset the roulette state. However, any chips placed on the board will be considered forfeit and deducted from the user’s balance if they are leave before or while the wheel spins.

Requirement 4.19

The user must have available to them a button that will allow them to view a brief introduction to roulette, its rules, and payouts upon request.

Requirement 4.20

The user would benefit from some sound effects to support a more immersive experience. When the roulette wheel is spun, a “spinning” sound effect should play for the duration of the spin.

Requirement 4.21

The user should be able to press a button to receive a summary of all the bets they currently placed. This will help the user remember which bets they have placed, so that they can understand the payout they stand to win.

Sabacc

Requirement 5.1

When playing Sabacc, a user must be able to view their cards as face up. The cards initially dealt to the opponent should be covered, but any cards the opponents take from the discard pile, being publicly visible, need not be covered. The player should be able to see the result of the end-of-round dice rolls at all times. They also must be able to see the chip amount within the sabacc pot.

Requirement 5.2

In addition to those elements, the GUI must display all cards in the discard pile as visible, being a pile, the cards are allowed to be stacked on top of each other, but the top card must always remain visible. The player must also be able to see the game pot amount, as well as the chips and stakes of all players, human and AI.

Requirement 5.3

Upon entering the Sabacc screen, the player must be allowed to specify the number of opponents they will be initally playing against for the duration of their stay in the Sabacc mode. This number must range between 1 and 3. Once the first game of Sabacc has begun, this option must be removed, and will remain so until the user leaves. After all, a player has no power to force an opponent to leave during or after a game. After the user leaves, the AI opponents may be fully reset.

Requirement 5.4

The card deck for Sabacc differ significantly from the cards used in Poker and Blackjack. A Sabacc card deck must contain sixty-two cards. Sixty of those cards are divided into circle, square, and triangle suits. Each suit should have cards numbered from –10 to –1 and from 1 to 10. Negative cards should be red in color, and positive cards should be green in color. There should also be two 0 cards, which can be any color.

Requirement 5.5

The graphical design of card icons must be consistent with the specifications dictated in Requirement 5.4

Requirement 5.6

At the beginning of a game of Sabacc, all players must deposit 50 chips into the “sabacc pot”. This value should not be considered part of their stake, as this amount can only be won back in the scenario described in requirement 5.25.

Requirement 5.7

Every time the user accesses the sabacc mode, the sabacc pot should be reinitialized with a new amount, which should be a multiple of 50. This is to simulate games that may be going on while the user is away, as someone might have won the sabacc pot. This adds immersion to the game.

Requirement 5.8

At the beginning of a Sabacc game, each player should receive two cards(GUI not important in this sprint). There must also be a public card dealt to represent a discard pile.

Requirement 5.9

After an initial deal, a sabacc game must be dealt in three rounds. In each round, each player must be prompted for a game action (draw, swap, stand, or junk).

Requirement 5.10

The playing, and betting, orders must remain constant throughout the games. First, the leftmost opponent should play. Then, the opponent in the center, the opponent on the right, and then the player. This fixed ordering gives immersion for the player. Of course, any opponents that have been defeated can be excluded from the ordering.

Requirement 5.11

A player choosing to draw a card must have a randomly selected card added to their hand. After receiving a card, the player must be given the option to release a card, placing it at the top of the discard pile.

Requirement 5.12

Alternatively, a player should be able to exchange a card in their hand with a card at the top of the discard pile (not focused on GUI management yet).

Requirement 5.13

A player should also be able to chose not to do anything during their turn, choosing to stand instead.

Requirement 5.14

A player should also be able to junk. In this instance, each card in their hand is added to the discard pile, and the player is considered to be out of the game, forfeiting any possbility to win chips.

Requirement 5.15

After plays have concluded, each active opponent must decide whether to match the current bet or raise it. If they don’t have enough chips to even match, they should simply bet whatever remaining chips they have. Otherwise, they should be more likely to raise the bet based on how close their hand’s value is to 0. If they decide to raise, the amount they raise by should also be based on their hand's quality. However, there should still be a random element to the betting, to make the user's experience more realistic and interesting.

Requirement 5.16

When an opponent is initialized, they should be given a randomly-generated amount of chips. These chips should be a multiple of 50. Randomizing the starting chip balance gives a more realistic feel to the game, as it is very rare for all players to start with the same amount of chips.

Requirement 5.17

The game must track the chip balance not only of the user but any AI players as well.

Requirement 5.18

Similar to poker, a user should also be given the opportunity to bet some of their chips during their turn. They must at least match the current maximum bet during the round, but they have the option to raise it as high as they want.

Requirement 5.19

After all turns in a round of Sabacc, two six-sided dice should be rolled (don't worry about the GUI). In the final sprint(or this sprint in time), the GUI must display the results of the dice rolls.

Requirement 5.20

In the event that the double-dice roll results in the same two numbers, all players must discard their hands into the discard pile. Afterward, they must redraw a hand equal in length to their previous hand.

Requirement 5.21

After all rounds are complete, the total value of each player's hand, determined using the number on the card, is used as the deciding metric for victory or defeat.

Requirement 5.22

The player with a total closest to zero wins the game. In the event that two players have the same absolute value from zero (such as 1 vs –1), the player with the positive sum wins the game.

Requirement 5.23

In the event that multiple players have a sum of 0, the player with the most number of cards in their hand is the winner.

Requirement 5.24

Naturally, the player who wins the Sabacc game must receive all chips bet during the Sabacc game. These chips must be added to their total (the casino total in the user's case).

Requirement 5.25

A "sabacc" is defined as a score of zero. In a later sprint, there will be a Sabacc pot, which will collect the entry fee required to enter a game of Sabacc. While the full implementation of the sabacc pot is not covered by this requirement, a player who wins on a "sabacc" should also be rewarded the entire contents of the sabacc pot, adding it to their chip balance.

Requirement 5.26

Naturally, the user should always have the ability to exit to the main menu at the end of a game. They must also be allowed to leave during a game, but they forfeit any chips they bet during the game. This adds further realism to the game, since you can't just pull out without consequence.

Requirement 5.27

When a player, human or otherwise, loses the game, they lose any of the chips they staked towards the game, as well as they’re entry fee. Once a player loses all their chips, they should be removed from the game. Opponents can merely be removed from sight and from subsequent games. The player, upon losing all chips, must be forcefully exited to the main menu. This promotes realism, as you can’t play any more without chips.

Requirement 5.28

In the event that all three opponents have run out of chips and have won, the player has won at the “table”. They should be congratulated and exited to the main menu, allowing the opponents to return with more chips.

Requirement 5.29

For better immersion, the user should be treated to at least one musical track from the Star Wars franchise. This music should only play when the user is within the sabacc mode, nowhere else.