

Object-Oriented Project Part 2 - GROUP

Team: Nolawee Mengist

Shawn Polson

Theerarun Tubnonghee (Steve)

Jose Escobar

Title: Blackjack!

Project Summary: A replication of the Blackjack card game that a user will play against a computer dealer and (optionally) computer players. Our goal is to create a program that looks and feels like an exact digitization of the real game; our code will faithfully represent Blackjack and our computer dealers/players will play autonomously.

Project Requirements:

Business Requirements: *We have omitted the “Business Requirements” table here because we do not have any.*

User Requirements				
ID	Requirement	Topic Area	Actor	Priority
UR-001	As a player, I can view the rules of Blackjack.	(optional)	(optional)	(optional)
UR-002	As a player, I can pick how many computer players to play with.			
UR-003	As a player, I can be dealt cards			
UR-004	As a player, I can place bets using chips			
UR-005	As a player, I can know the total amount of “money” I own.			
UR-006	As a player, I play against a CPU dealer			
UR-007	As a player, I can “hit” during my turn.			
UR-008	As a player, I can “stand” during my turn.			

UR-009	As a player, I can “double” during my turn.			
UR-010	As a player, I can “split” during my turn if my first two cards are of the same denomination.			
UR-011	As a player, I can win the bet if I get “21”			
UR-012	As a player, I lose the bet if I “bust” over 21.			
UR-013	As a player, I can play through as many rounds as I want.			
UR-014	As a player, I can still lose to the dealer even if I get 21			
UR-015	As a player, I can drop out and end the game with my winnings any time			
UR-016	As a player, I can save the current state of the game and continue to play.			
UR-017	As a player, I can continue my saved state of the game			
UR-018	As a player, I can start a new game and play with some amount of chips.			

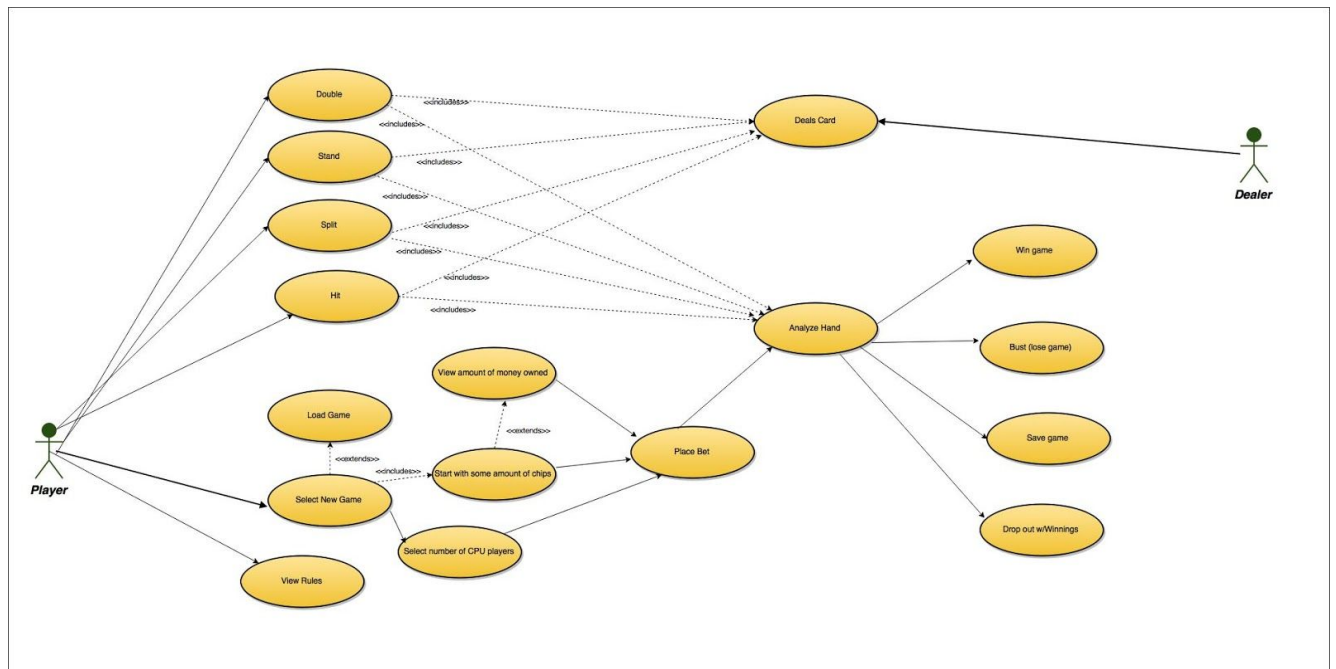
Functional Requirements				
ID	Requirement	Topic Area	Actor	Priority
FR-001	As a player, I lose if I run out of chips.	(optional)	(optional)	(optional)
FR-002	Card deck should be shuffled before play or when deck runs out			
FR-003	Dealer and computer players must			

	run through their turns when needed			
FR-004	Computer players must track their own total money and not bet past 0			
FR-005	Dealer and computer players must not hit when their currentTotal>=16			
FR-006	When player saves the game and exits, all totals and current cards in deck must be saved for reloading			
FR-007	User can navigate through the game.			

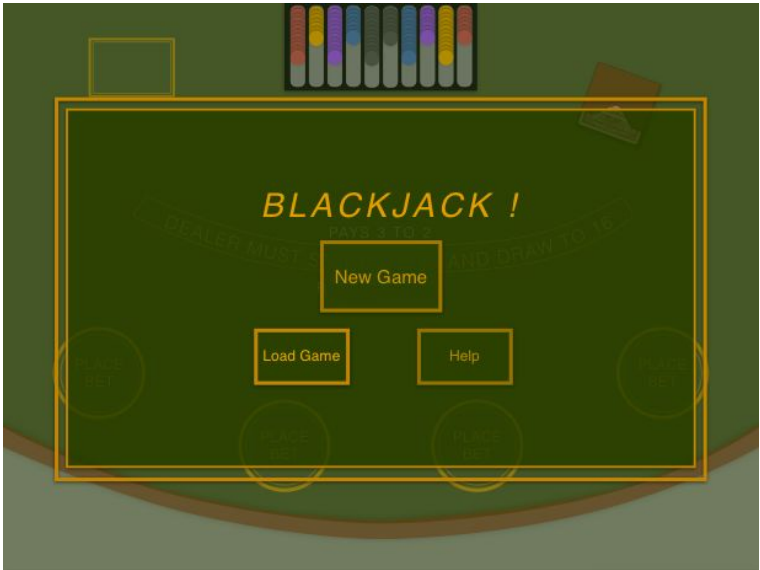
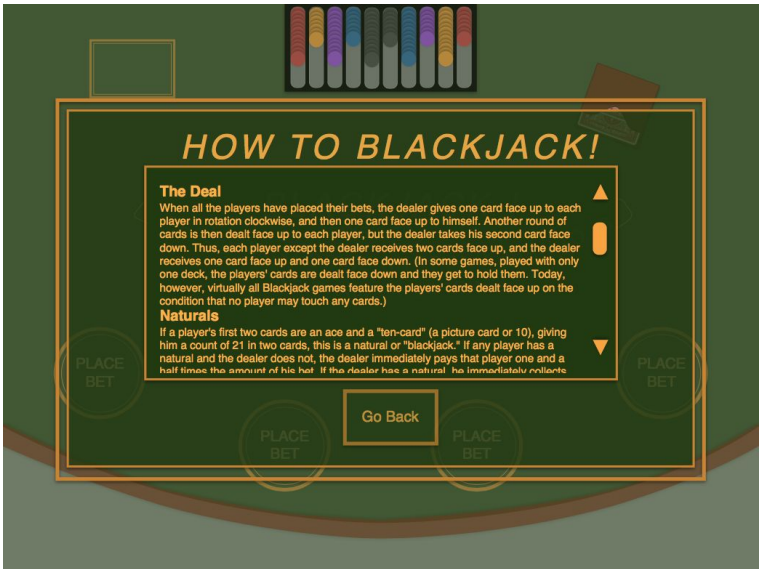
Non-Functional Requirements				
ID	Requirement	Topic Area	Actor	Priority
NFR-001	The game can be played on the command line before our GUI is implemented.	(optional)	(optional)	(optional)
NFR-002	The game should work properly each time it's opened			
NFR-003				

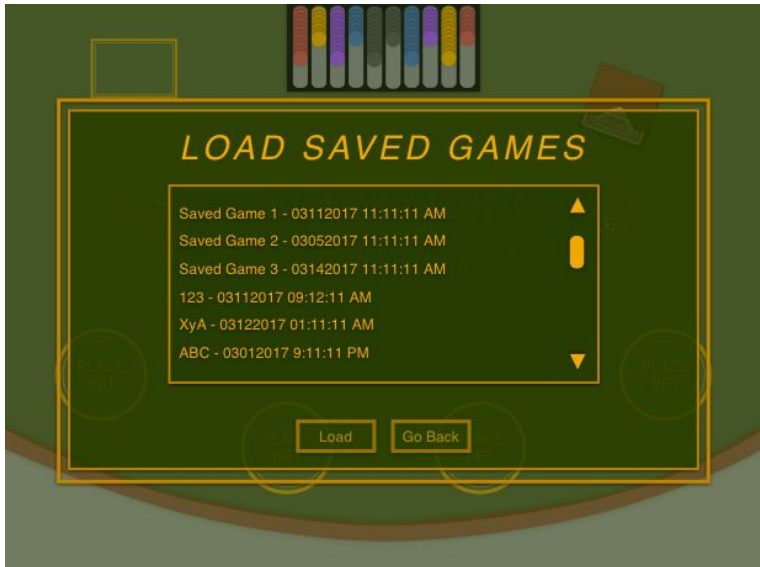

Use Cases:

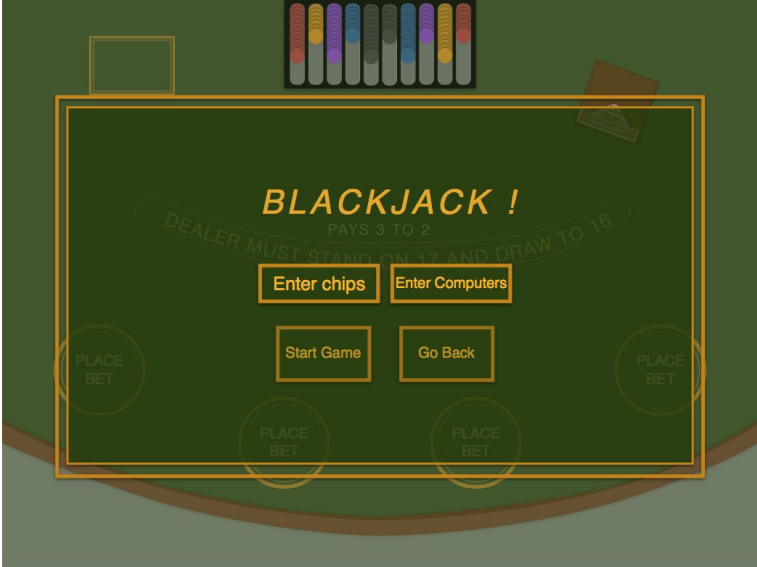
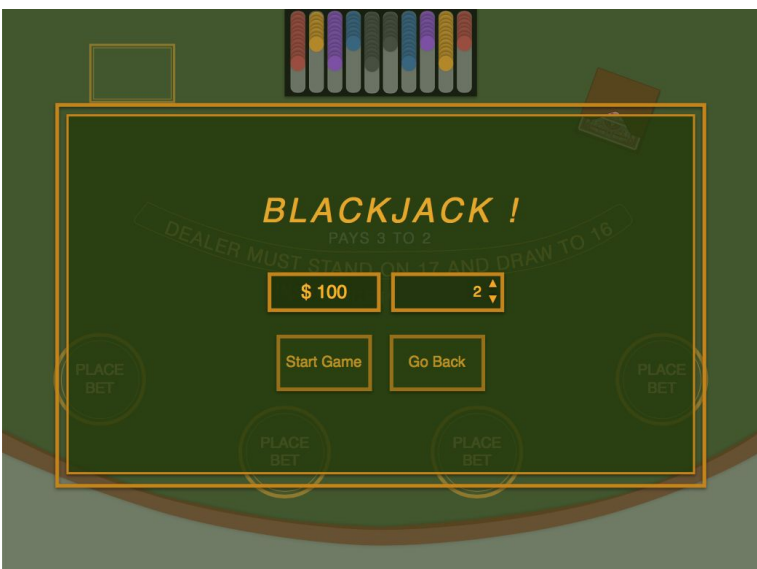
- Use Case Overview:







UI Mockups:



	Scenario	User Interface Response
1.1	Player opens Blackjack! Program. The UI displays Main Menu.	 <p>The main menu is displayed on a green background with a yellow border. At the top, there is a row of 13 colored vertical bars. Below them, the text "BLACKJACK!" is written in large, bold, yellow letters. Underneath, the text "PAYS 3 TO 2" is visible. In the center, there is a yellow box with the text "New Game". Below this, there are two yellow boxes: "Load Game" on the left and "Help" on the right. At the bottom, there are four yellow circles, each containing the text "PLACE BET".</p>
1.2	Player clicks Help button. The UI displays rules of Blackjack!	 <p>The "How to Blackjack!" screen is displayed on a green background with a yellow border. At the top, there is a row of 13 colored vertical bars. Below them, the text "HOW TO BLACKJACK!" is written in large, bold, yellow letters. In the center, there is a yellow box containing the following text:</p> <p>The Deal When all the players have placed their bets, the dealer gives one card face up to each player in rotation clockwise, and then one card face up to himself. Another round of cards is then dealt face up to each player, but the dealer takes his second card face down. Thus, each player except the dealer receives two cards face up, and the dealer receives one card face up and one card face down. (In some games, played with only one deck, the players' cards are dealt face down and they get to hold them. Today, however, virtually all Blackjack games feature the players' cards dealt face up on the condition that no player may touch any cards.)</p> <p>Naturals If a player's first two cards are an ace and a "ten-card" (a picture card or 10), giving him a count of 21 in two cards, this is a natural or "blackjack." If any player has a natural and the dealer does not, the dealer immediately pays that player one and a half times the amount of his bet. If the dealer has a natural, he immediately collects</p> <p>Below the text box, there is a yellow box with the text "Go Back". At the bottom, there are four yellow circles, each containing the text "PLACE BET".</p>



1.3	Player clicks Go Back to Main Menu. Then, Player decided to load saved games by clicking Load Game. The UI displays Load Saved Game Menu	
1.3.1	Player clicks on one saved game from the list, and clicks Load. The system loads the saved one and UI display the current state of saved game.	


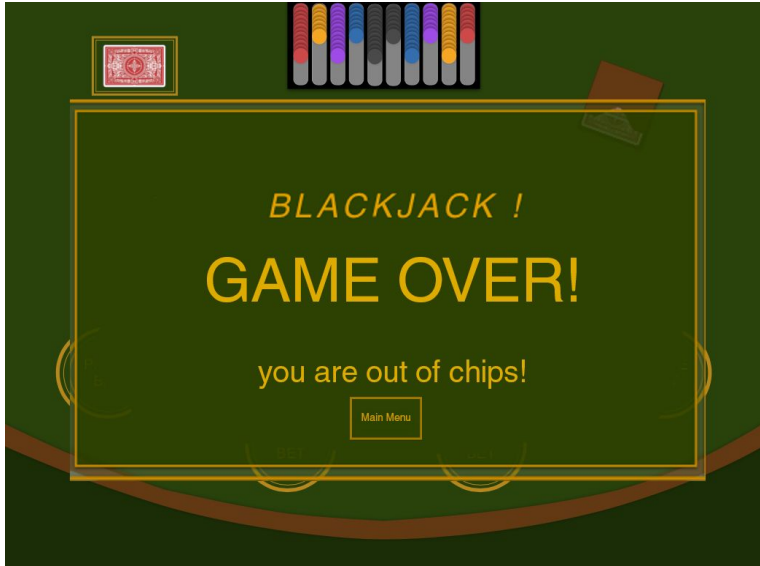
<p>2.1.1</p>	<p>In Scenario 1.1, Player clicks New Game to start a new game. The UI displays Options' requirement to start a game. Prompt Player to enter amount of starting chips and a number of computer players to play with.</p>	 <p>The screenshot shows a digital blackjack table interface. At the top center, there is a row of ten colored chips (red, orange, yellow, green, blue, purple, grey, black, white, and red). Below the chips, the text "BLACKJACK!" is displayed in large, bold, yellow letters. Underneath, it says "PAYS 3 TO 2" and "DEALER MUST STAND ON 17 AND DRAW TO 16". In the center of the table, there are two input fields: "Enter chips" and "Enter Computers". Below these, there are two buttons: "Start Game" and "Go Back". On the left and right sides of the table, there are circular areas labeled "PLACE BET".</p>
<p>2.1.2</p>	<p>Following, Player enters desired options. The UI real time responses back the entered options. If Player clicks Start Game, the UI will retrieve the information entered, and displays the empty table in 2.2.1. If Player clicks Go Back, the UI displays Main Menu.</p>	 <p>This screenshot shows the same digital blackjack table interface as the previous one. The input fields now contain the values "\$ 100" and "2". The "Start Game" and "Go Back" buttons are still present. The "Enter chips" and "Enter Computers" labels are no longer visible, indicating that the user has entered their values and the interface has updated.</p>

2.2.1	Player is in the first game. The Table will be empty. The UI displays current amount of chips/money Player owned.	 <p>The image shows a digital blackjack game table. At the top center is a chip rack with 10 chips in various colors (red, yellow, blue, green, purple, grey). Below it, the text "BLACKJACK !" is displayed in a large, stylized font, with "PAYS 3 TO 2" underneath. A curved banner below that reads "DEALER MUST STAND ON 17 AND DRAW TO 16". In the center of the table, there are four circular buttons, each labeled "PLACE BET". To the right of these buttons, the text "Current: \$ 100" is displayed. The table has a green felt background with a brown border at the bottom.</p>
2.2.2	After Player clicks PLACE BET, the UI prompts Player to enter the bet using keyboard. The UI displays the amount of bet, and prompts Player to start a round by clicking DEAL.	 <p>The image shows the same digital blackjack game table as in the previous state. The chip rack and the top text are identical. However, the "PLACE BET" buttons are now arranged differently, with one button in the center displaying "\$5". To the right of this button, a small "DEAL" button is visible. In the bottom right corner, a "SAVE GAME" button has appeared. The "Current: \$ 100" text remains. The table has a green felt background with a brown border at the bottom.</p>

<p>2.3</p>	<p>Promptly, the dealer distributed two cards to each player according to the rules. The UI will displays the Table with cards in hands of players but not displays dealer's hand if the dealer is not blackjack!</p>	 <p>The image shows a digital representation of a blackjack game table. At the top center, a banner reads "BLACKJACK !" in large yellow letters, with "PAYS 3 TO 2" and "DEALER MUST STAND ON 17 AND DRAW TO 16" in smaller text below it. Above the banner is a rack of 10 colored chips (red, orange, yellow, green, blue, purple, grey, black, white, red). To the left of the banner is a red card box. To the right is a brown card box. Four players are seated around the table, each with a hand of two cards and a bet amount in a yellow circle: Player 1 (left) has Ace of Spades and 2 of Hearts, bet \$5; Player 2 has Ace of Spades and 2 of Hearts, bet \$20; Player 3 has Ace of Spades and 2 of Hearts, bet \$20, with "DOUBLE" and "STAND" buttons; Player 4 (right) has Ace of Spades and 2 of Hearts, bet \$5, with a "DEAL" button. The dealer's hand is not visible, only a red card box is shown.</p>
<p>2.3.1</p>	<p>If the dealer has blackjack!, the game is end. The UI displays dealer's hand. Then the dealer claims and pays players. Promptly, the next round start (bring back 2.2.2).</p>	 <p>The image shows the same digital representation of a blackjack game table as the previous one, but with the dealer's hand revealed. The dealer's hand consists of the Ace of Spades and the King of Hearts, totaling 21. The banner "BLACKJACK !" is still present, along with the rules text. The players' hands and bets remain the same: Player 1 (\$5), Player 2 (\$20), Player 3 (\$20 with "DOUBLE" and "STAND" buttons), and Player 4 (\$5 with "DEAL" button). The rack of chips and card boxes are also visible.</p>

<p>2.3.2</p>	<p>If the dealer does not have blackjack!, each player can deal more or stand or double. The UI displays options of the Player.</p>	
<p>2.3.3</p>	<p>If there is an empty seat, HumanPlayer can split the card if the dominating cards are the same number. The UI displays options of the Player.</p>	

<p>2.3.4</p>	<p>After splitted, Player can stand/deal, and is need to confirm options on both hands by clicking on stands. The UI displays options of the Player. Then the dealer reveals his hand and claim/pay players. Promptly, the next round start (bring back 2.2.2).</p>	
<p>2.4.1</p>	<p>During each placing bet state (2.2.2), HumanPlayer will be allowed to save the game. The UI displays SAVE GAME button to be clicked as an option.</p>	

2.4.2	<p>Promptly, after Player clicked SAVE GAME (in 2.4.1), the UI display Saving Game menu. Player can enter the desired title of saving game, and click Save to complete the process. Player can click Go Back, Player will be brought back to the current game (in 2.4.1). Caution: The Table will not save the game unless Player click Save.</p>	
2.5	<p>If unfortunately you run out of chips, game is over! The UI responses GAME OVER! Player can click Main Menu.</p>	

Data Storage: *We do not have plans for using a database. We plan on implementing “save game” by saving/loading our model (table) object; to have that be the object that persists).*

Class Diagram:

