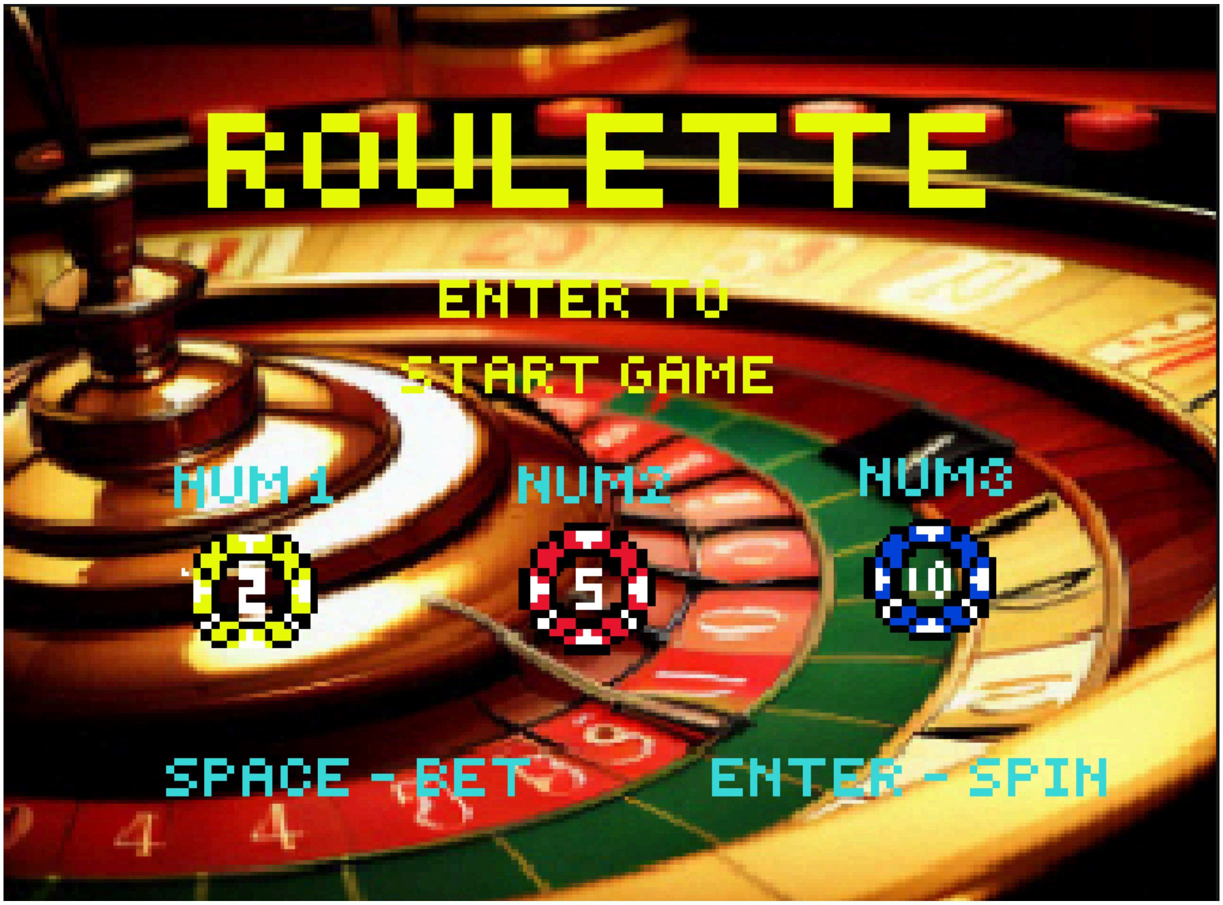

Eric Lou & David Huang

Project description:

Our project presents an interactive simulation of a traditional roulette game which incorporates the roulette betting systems. The game allows users to place bets on different outcomes and update game displays based on the result. This project provides an intuitive understanding of roulette game mechanics, payout ratios and betting strategies in a virtual environment.

Features:

Loading Screen:

Introduces user input keys to allow efficient user input and avoid confusion on game operations:

Press 1,2,3- choose the betting amount
Space- Betting
Enter - Start Roulette

Betting System: Users can place bets on a wide range of outcomes, including single number, colors(red/black), even or odd, and groups of numbers. The system allows for multiple bets per spin, which enables users to experiment with complex betting strategies.

Randomize Generate Outcome: The core of the game logic is randomized roulette number generation mechanism which ensures each spin is independent and unpredictable.

Payout Calculation: Depending on the result of each spin, the game will calculate winnings based on payout ratios corresponding to the amount and type of bets placed. These ratios reflect the odds and payouts of a traditional roulette, providing a real gaming experience.

Balance Management: A virtual balance system that tracks the user's bankroll which will update the balance based on the game outcome. This feature introduces the concept of risk management, which encourages players to bet more strategically.

User Interface: The game has a user-friendly interface that displays the betting table, balance and bet amount management. Result displays and balance updates also enhance user engagement and provide immediate feedback on results.

Result and Gameover Display: After each spin, the game presents a result page showing the winning number and color, won or lost and the total amount of credits that are won or lost. A gameover page appears when the user’s balance is below 0 and will restart the game.

Challenges and Solutions:

PS2 Mouse detection was part of the tasks that we wanted to incorporate in our game, however, after excess testing and debugging sessions on the PS2 mouse and interrupt, it failed to consistently provide user input upon command. Sometimes it will only provide location within a quarter of the screen. Our solution was to switch over to the PS2 keyboard, which is able to give consistent results in detecting user inputs.

Future Enhancements:

We would add roulette animation which spins instead of only the result display. We will add sound to the game to make the player have a better experience. We will add mouse clicks for navigating our table which makes our project more user friendly. We can also add history roulette results for users to make betting more strategic.

Conclusion:

Our project helps players experience the excitement of roulette with the convenience of virtual gameplay. This serves not only as an entertaining roulette game but also as an educational tool for understanding roulette mechanisms and betting strategies.

Attribution Table:

Student Names	Work Description
Eric	
Loading Screen	Designed loading screen graphics, loaded in when the program is started. Proceed to next stage when enter key is detected
Betting board & logic	Designed a betting board to be 320x240 pixels size, implemented logic in moving through different sizes of betting options to provide a smoother user experience. Detection of w,a,s,d keys are implemented for selecting different betting options. Provided visual indication on which betting block the user is on.
Bet placements	Detection on key presses 1,2,3 and spaces are implemented such that different chips are drawn onto the screen when the user wants to bet different amounts on each block. The chips colors are able to change dynamically when more chips are stacked upon one betting option.
Keyboard detection	Detection of all keys that are utilized for different purposes (load to play game and spinning) is implemented.
David	
Game Logic	After users place different amounts of bets on different outcomes, randomized Roulette Number Generation and Payout Ratio Calculation is calculated for each different type of bets.
Balance Amount	Hand draw text and digits to display user balance. User's Balance is tracked and updated throughout the game, including initial balance setup($100), winning and losing conditions(added all bet winning or losing amount) and when they are broken(gameover screen and go back to $100).
User Interface-Result Display and Gameover Page	Display on the outcomes of a roulette spin(e.g. The winning number, color) and the corresponding results for the user(win/lose status, amount won or lost). When the User's balance reaches zero, they will be told that they are broke and go back to $100.