Project Title:

# Casino Royale: A Digital Casino Simulation System

## Problem Statement

Many educational tools fail to combine learning and engagement. Most casino systems are profit-driven and involve real money, which can cause addiction. Learners who want to study casino logic, randomization, or probability lack safe, accessible tools for it.

The main problem with Casino systems is the cost and its instant gratification. The reward you could get from taking a big risk would cause a big rush of dopamine which is usually what keeps people coming back. They want to get more of that feeling but when you do something more often, the joy and pleasure you get from it is reduced or to a smaller degree. This leads you to do more for the same level of joy or pleasure and causes addiction.  This turns a fun luck game into an endless loop of losing.

Casino Royale fills this gap by offering a simulated, non-gambling version of popular casino games. It teaches how randomness, probability, and algorithms drive these systems. Removing actual money eliminates addictive behavior while encouraging logical and responsible learning. The project offers a safe, educational alternative that keeps the fun and challenge of real casino games.

## Project Objectives

Casino Royale aims to build an interactive casino simulation using programming techniques to model real casino behavior. The program will use modular design, random number generation, and conditional logic to simulate realistic outcomes. Users will see how computers process data and generate casino-like results.

Another goal is to apply probability concepts in realistic game simulations. Blackjack will use random number generation for dealing cards and calculating totals. The slot machine will use similar logic to display random outcomes.

The project will also promote responsible gaming awareness. It teaches how casino systems work from a technical and logical view, turning gambling mechanics into a practical learning experience.

This project will also teach financial responsibility. The money system is meant to teach players when they should stop spending, and to show how bad decisions can lead you down a horrible path. This will be measured using the file creation feature to make a counter on the computer of how many times the player walked away from the casino with cash.

## Planned Features

Casino Royale will feature two main games: blackjack and slot machines.

Blackjack: Simulates a game between the player and a digital dealer. Users will choose “hit” or “stand,” and the program will calculate totals and decide the winner. This part will demonstrate loops, conditionals, and randomization.

Slot Machine: Simulates a spin of reels with random symbols. It will calculate wins or losses based on probability. This section will show how randomization and pattern recognition work in programs.

The system will include a main menu for game selection, help instructions, and session resets. Text-based outputs will guide users through gameplay. The design will focus on clarity and logic, making it beginner-friendly.

We are planning on adding a money system to work with the games and to encourage financial responsibility. The money value will be decided by the player at the start. You gain and lose money by winning and losing bets.

## Planned Inputs and Outputs

The Casino Royale simulation depends on user interaction through straightforward inputs. Players begin by entering a username, setting a money value, and selecting a game from the main menu. In blackjack, players will input decisions such as whether to hit or stand. In the slot machine, players will choose when to spin the reels.

These inputs are processed by the program to determine each game’s outcome. The program will produce outputs that depend on user actions and randomly generated results. In blackjack, outputs include the display of cards, hand values, and a message indicating whether the player or dealer wins. In the slot machine, outputs display reel symbols and announce whether a winning combination was achieved. The program provides real-time feedback, allowing users to observe how inputs and algorithms produce dynamic outcomes.

## Logic plan

1. The menu will pop up and show you options. You will give an input to decide what game to play or if you read a guide.
2. The game checks the integer input and displays the chosen option

3. If the player chooses to play a game, the program asks for a bet input which will double and be added to your balance when you win or be lost upon losing. The bet cannot be higher than your current amount of cash. If the bet is worth 0, it will end the game to allow players to quit if they change their mind.
4. If the player chose Blackjack, The program will give cards from a deck list to the player and a dealer npc. The score of both players is calculated by summing up the value of their cards. The player can choose to roll another card to increase their score or to stop, finalize their score , and let the dealer roll. When the player stops, the dealer will keep rolling and stop if their score exceeds 17. The match will end when both stop rolling. The winner is determined by the one with more score and the loser is the one with the least score or had their score go over 21
5. For Slots, It rolls with a set of symbols and rolls 3 times. If the same symbol is rolled 3 times in a row, you win and lose if you don't roll the same symbol 3 times. The money system will be implemented into other games later.

## References

Code, B. (2024, June 4). *Let's code a beginners Python SLOT MACHINE 🎰*. Youtube. https://www.youtube.com/watch?v=f5J3YiZ3XX8

GeeksforGeeks. (2025, July 23). *Blackjack console game using Python*. https://www.geeksforgeeks.org/python/blackjack-console-game-using-python/

The Venetian Resort Las Vegas. (n.d.). *Craps Rules*. How to play craps. https://www.venetianlasvegas.com/resort/casino/table-games/craps-basic-rules.html