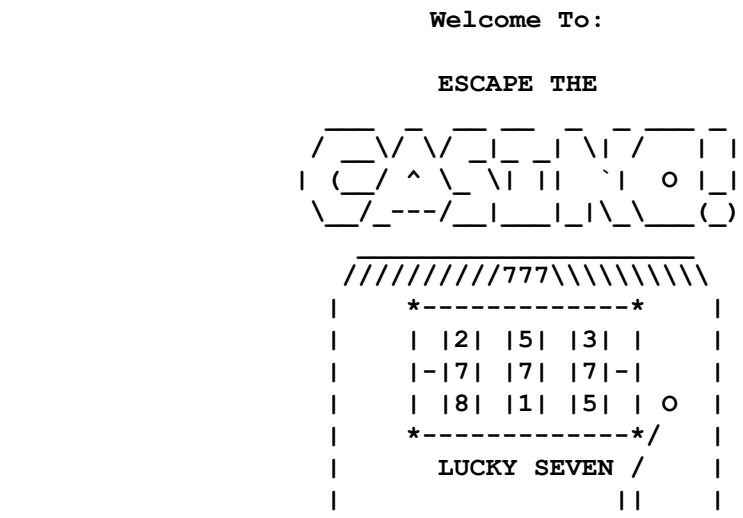
Team CarryOnWaywardCode APCS pd3
Alessandro Cartegni, Elvin Hwang, and Ricky Chen 1-06-17



Final Project Proposal

Casino-RPG

Overview

The project will play out as a simulation of a casino with some RPG elements (mostly for flair) that will focus on the player earning money to win. The game will incorporate 3 main games to earn the game's currency, those being Blackjack, Slots, and Roulette, which will implement interface Minigame. There will be a bad ending (the player runs out of currency) and a good ending (the player earns enough money to defeat the final boss), and a secret ending (HINT: It involves the food court). The games will mostly take advantage of Arrays and ArrayLists while the RPG elements will incorporate inheritance and interfaces.

BlackJack

One of the games you can play is Blackjack. A deck of 52 cards will be created at the start of every new visit and shuffled. The player will choose to either hit or stay, and once/if they stay the dealer will dealer cards until he wins or loses. The game is played with the end goal of beating the dealer not only at cards, but also in the RPG aspect with the game tokens won. Because it is the easiest to win at, this has a lower bet multiplier. This uses an array that is built and randomized for the deck, and different subclass methods for dealer and player, as the dealer goes last and always has to beat the players.

Slots

Another of the games is Slots. It will have a variable bet multiplier (depending on game state) and standard slot payouts. The Slot machine will be made using a 2D array with 3 arrays of 6 strings. Thus, the user must "pull the lever" to lock in a string as they cycle by for a total of three times. A power-up that the user may gain, Slow Time, will make the process easier to control. The bet multiplier will scale with money, essentially making the it harder to get a good return as your cash stack increases.

Roulette

For roulette, the player can choose any amount of numbers+colors to bet on, as long as they have the money to pay up. Game tokens are detracted whenever a number+color is bet on. After the bets are finalized by a terminal entry, the line "No more bets" will be made and there will be a loading time, indicating that the ball is being spun on the wheel. If the randomizer method gets a value equivalent to the bet number made, as well as the correct number, the player will be rewarded game tokens. If the player loses, then game tokens will not be rewarded and terminal will say "Tough luck, try again?" or something of that sort. This gamemode is the most challenging and therefore will have the highest multiplier. It uses Math.random() to generate the result, as well as direct user input for the number.

RPG

In the game, the 3 games will be guarded by 3 house members. To defeat them, the player will partake in a faux RPG battle where they must have the appropriate amount of coins to win, (Every battle will be an instance of RPG-Battle). There will also be a food court where various "cheats" will be made available. They will all be subclasses of superclass Burger and will be stored in private variables within the driver method with the minigames using a public mutator function. They include Future Sight (for BlackJack), Slow Time (for Slots), and Magneto's Hand (for roulette); a greater white probability. Also, the plot will be completely 100% serious. We swear.

Minimum Viable Product

If we were unable to achieve many of these ideas, then the least we can manage is to make a simple casino game. There will be a menu when running the game that will allow the user to choose between BlackJack, Slots, or Roulette. There will still be no money system, and no endings. Blackjack will instantiate an array representing only one suit of cards. Slots will use the more basic method of shuffling an array of strings. Roulette will still use Math.random(). The UI will be heavily undercut.