Professor Rawlins

**Wentworth Institute of Technology**

**Electrical Engineering and Technology**

**ELEC 3225-01 Applied Programming Concepts**

**<names> , Zach Ray**

**Casino Simulation**

**Synopsis [from project description on brightspace]:**

Simulate a casino environment from the standpoint of the casino management. Track money, games, users, etc. Will need many casino games (and multiple versions of the same games) to be run in a statistically accurate environment (blackjack, poker, roulette, craps, slots, etc.). Must have multiple users of different strategies and bankrolls. Track user gambling history (wins, losses, games). Record the amount lost/won individually and for the entire casino.

**Introduction**

Many components will be required to function together to simulate a casino environment. Just to name a few: we will need a database with multiple relations (games, users, money, etc.), classes to control user functionality as well as access to the database, and a clean and intuitive user interface. At the end of the semester, there should be multiple playable games – users should be able to sign in, view their winnings, play games, and bet in the casino. There should be a separate view for admins which allows one to see how the casino is performing. Admins should have a neat menu, be able to display info from the database, and update / remove entries in the database.

**Current Functionality (6/15/22)**

Functionality so far has been added in unison with the lecture topics. As of now, we have a database with two relations: player and slots. Each game will have their own relation – there is a base game model and table which is created when a game is added. There is an abstract base class – user. Player is inherited from user. Players currently store a unique id, user id, name, password, and winnings. Player also stores an integer to identify if they are banned or if they are an administrator. Players can be caught cheating and will be banned from playing games. Administrator functionality will give access to all of the back-end. It is easier to store an administrator, or few administrators, inside of the player relation – rather than create a whole new table just to store a few admins.

**Requirements analysis and definition**

**Major Components**

* Database with multiple tables
  + Player table (store user id, name, winnings, password, ban state, admin state)
  + Game table (store history of games, amount won, who won, when they won)
  + We should be able to track all player’s gambling history, gains/losses for each game, and be able to view different gains / losses over different time intervals.
* Multiple Interfaces – can be text or graphical
  + Player interface – access games, winnings
  + Game interfaces – each game will have their own unique interface
  + Admin interface – access and views on back-end
* Object Oriented Programming (OOP)
  + Base class User – derived classes for players and administrators
  + Games will be controlled through classes
* Playable games
  + Players should be able to sign into the casino and play games
  + Poker, roulette, blackjack, craps and slots
* Simulation
  + Should be able to simulate the casino as if 100’s of players were playing games and betting
  + Mimic real random games by the 100s
  + Integrate / change games and determine efficacy
  + Determine best number of each game to maximize profit

Multiple games are being created right now. We’re aiming to have five games – slots, poker, blackjack, craps, and roulette. Each game will have their own table. History of all the games will be stored here, and accessible from the admin. Each entry will store the player, how much they won, and a time stamp. Most games should have rules / variables which can be altered to test how profits change. The casino should eventually be able to be simulated with 100s of players, and we should be able to identify how changing rules impacts the casino. One view which the admin should have is gain / loss for any game over specific time intervals. Some games will involve betting which will either need to be an additional column in the games relation or a separate entity itself.

Multiple interfaces will need to be created. There should be a player interface – a way to play games, check account balance and make bets. Each game will need to have their own interface for playing the game. Some may be graphical. There should also be an admin interface. A way to add / remove players, track user gambling history, track dealers / players /games over some time period, a way to update rules and more.