RISK REMADE

Project Description

For this project, I will remake the game RISK, a strategy-based game centered around the goal of world domination. The game is conducted in a turn-based fashion and requires between 2-4 players. The project UI will overlay the Risk world map where players can click on certain countries in order to battle and quest for domination over other countries. A winner is decided once the world has been taken over by one player.

Competitive Analysis

Most projects that attempt to recreate a game this large use an abundance of classes in order to create continents, countries, players, troops, and many more objects within the game. The class structure of my project will be much less comprehensive as I believe only a couple of classes will be necessary. I believe only the continent and the player class will be necessary and all other objects such as countries and troops can be subcategorized attributes of the above classes. In addition, I believe it will be much simpler and I will run into less aliasing if I conduct turns and other moves using functions rather than class structures.

Other projects also put an intense emphasis on the AI that plays the game alongside a human player. This will be my goal after MVP and I will heavily use other coders' algorithms as inspiration for my own as they have used complex machine learning and neural networks to come up with the various variables that the AI considers and what weights to put to each variable.

Structural Plan

As mentioned above, my project will consist of 2 main classes - the continent class and the player class. These classes will hold the necessary information of countries, troops, colors, and possibly other additions later on. If necessary, I may use inheritance to create the country class as this would allow me to add additional functions necessary for battle. The battle mechanics will consist of multiple functions utilizing the global app variables to ensure values update throughout the game.

Algorithmic Plan

The trickiest part of my project will be implementing the battle feature in which multiple class objects will need to simultaneously be called in order to enact the battle. This battle feature will first use the random module in order to randomize a player's dice rolls based upon the number of troops in a given country. From there, the number of troops and the battle dynamics will be calculated and each country class object will be modified based on troop counts, the player in control, color, etc. This doesn't require many algorithms but does require calling multiple objects in succession which poses some challenges.

Timeline Plan

Nov 18	Nov 23	Dec 1
Have completed the setup of the game including creation of countries, distribution of countries, initial troop counts, UI for clicked on countries, etc.	Implement the battle function which allows for players to use the UI in order to interact with other countries. Be able to change each countries troop counts based on randomized dice rolls and switch possession from player to player.	Finalize the win condition of the game and make the UI more friendly. Display the battles in real time and make any mouse clicks more receptive.

Version Control Plan

To save my code, I simply save the python file everytime that I log off for the day and save that file into my icloud using the builtin software.

Module List - None

TP2 Design Updates - None

TP2 Design Updates - I decided not to pursue the AI option instead opting to make the user experience of the base game more enjoyable and less likely to result in crashes.