Computer sciense 5

**How I made RISK in C++**

horizontal line

# 

# Introduction

The question was why did I want to make RISK. It seemed like a daunting task, but I felt that it would be enjoyable to recreate one of my favorite boardgames from scratch! So I decided to create it in C++ as my project.

## How I came up with the logic.

This was one of the harder parts. As I went along I figured I’d add in what I needed as the project got more demanding. I used variables and arrays like plr1arm[] and plr3own[] to represent the size of the players territories and who owned what. I also made sure to keep all the arrays at a const int SIZE of 41, representing the 42 territories of the game. After that I went and added in an extra array later to account for the names of all the countries I would need to call in on the fly with a large array of strings called names[]. Then I set up the seed and the random number generator so that the AI would be capable of making their own choices. I also set up multiple functions in the header of the program, like nextto and control, mainly to make formatting easier instead of copypasting the headers for the countries everytime I prompted the user.

### Starting the game.

The hardest part was keeping track of all the countries. I did this by using each element of the array as a country, and filling in the data within the subscripts, such as the name of the country, who owns it, how many troops are in it, ect. Then I set up multiple for loops to keep the game going until the user either won or lost. After that I set up an if else statement at the start that takes user input and asks for a action, to either attack or pass turn. If the user chose to attack, a if else loop would occur where it uses the function enemyck() which takes the arrays of all players and compares the input made within it to see if the player landed on a neutral territory, a friendly territory, or an enemy territiroy. If they landed on a enemy territory, then they would have to fight said enemy with a dice roll, deiceded by the rand() function inbetween 1 and 6. If the player beat the enemy, he would take down some of his troops and, if the user beats out all the troops in a opponents territory, he claims that territory for himself. However, if the player loses the dice roll, the opponent cuts down some of the players own forces, and if the player loses all his total forces, he will lose the game. The bots follow roughly the same pattern among themselves with randomized inputs instead. Afterwards the program loops itself, rewards the players with additional troops, and the next turn starts again until one of the players wins!

Have fun!