**Project 1**

**Connect 4 Game**

**CIS-17c**

**Alec Shahverdian**

**Due Date: 5/01/21**

**Introduction**

* I am coding a Connect Four game in C++. It is a game concept I am very familiar with, and so I felt it would be appropriate to attempt it in this project. I spent about two and a half weeks in between work and all my other classes working on this project. It contains 650 lines of code. The github location is <https://github.com/AlecShah/Cis17c.git> .

**Approach to Development**

* For developing the game, I wanted to mainly focus on developing a good gameplay cycle where to players can play without interference one after another until the game determines a victor, than the game can instantly be replayed. That was where I started.
* I attempted to use as many concepts as I could while sticking strictly to a format that would output a well-functioning connect four game, which led to some issues when trying to meet all the requirements in the final functioning program.
* Functioning bot was planned but was unfortunately unable to be correctly implemented in time.
* Unfortunately, the game concept could not realistically be fleshed out into a project with 750 lines of code within the given time, and I could not realistically innovate or restart, so I had to concede with the project finished where it was at just short of 750.

**Game Rules**

* The game is played with 2 players
* Players will take turns placing “tokens” in a 6 x 7 grid
* Tokens drop to the bottom of available spaces in that column
* Player wins if they connect 4 tiles either vertically, horizontally, or diagonally
* If all spaces are filled with no victor, it’s a draw

**Description of Prominent Code**

* Organization for code was planned and executed as follows…
* Struct for player info
* Main where rules and instructions are output and main gameplay loop is executed
* Restart function
* Drop token function
* Instantly display rules function
* Display and fill board functions
* Checking for four tokens connected and if tokens are below placed token
* Win condition function

**Sample Output**

(Output Depicts output during gameplay)

Player One please enter your name: Alec

Player Two please enter your name: Carl

|.......|

|.......|

|.......|

|.......|

|.......|

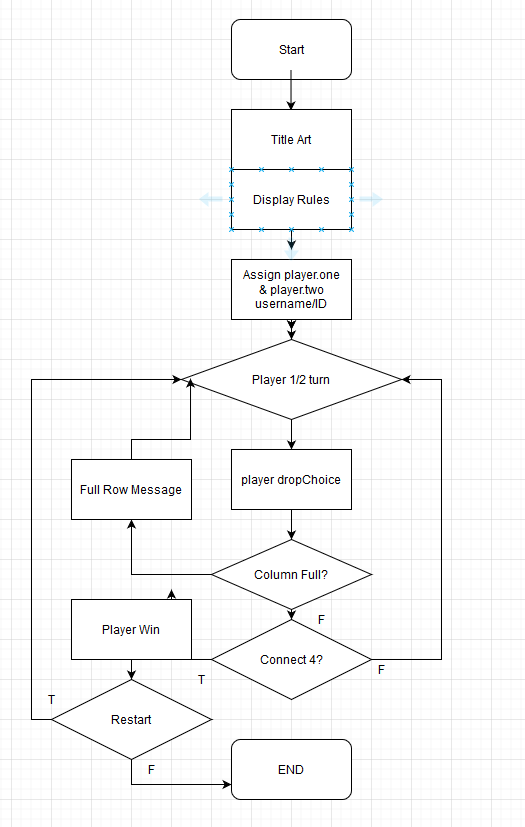
|.......|

Alec's Turn Please enter a number between 1 and 7:

**Pseudo Code for Main Gameplay Loop**

* Run
* Output Title art and Rules
* Output asking for user 1 and 2 names
* Input user 1 and 2 names
* While Player 1 and 2 turn:
* Output asking for player drop choice
* Player inputs drop choice
* If column is full:
* Output full row warning message
* Repeat turn
* If connect 4
* Output win message (end turn)
* Output asking for restart
* If restart, then start 1st turn again
* Else, end program

**Flow Chart for Main Gameplay Loop**



**Container Classes**

* List
* Set
* Queue and stacks planned to be used for tracking moves made during the game

**Algorithms:**

**Non-mutilating**

* Wanted to use count for when the player wanted to see how many moves were made during the match.

**Mutilating**

* Wanted to use remove in part with a function for redoing a move that would get rid of the move that was tracked in through a queue

**Organization**

* Used a struct that comprised of players’ info which consisted of their chosen username and the representation for their chips (X and O)