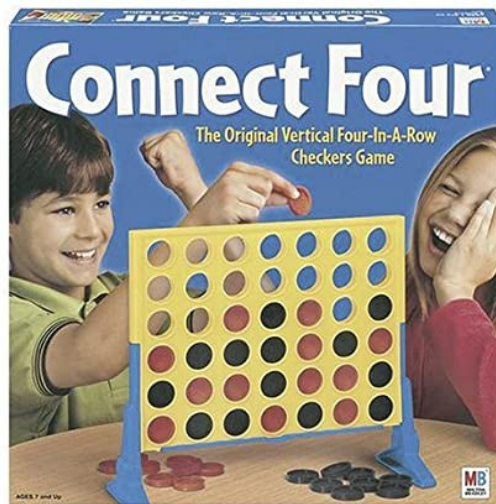


Final Project – Connect Four Game

GitHub repo: <https://github.com/ramirez-gabriela27/Connect-Four>

Gameplay Requirements:

1. Give a brief description of how your game is played and what (if any) game it is base off of.
Our game is based on Connect Four – grab a piece of your corresponding color (up to 3) and drop it down a row when it is your turn, the goal is to connect four of your pieces either vertically, horizontally, or diagonally, to win the game.
2. Your game must have some sort of “playing field” – a board, where cards are displayed, etc, showing the current state of the game. Briefly describe what yours look like.
Our board will consist of a “home screen” where the stats of the game will be displayed, and a 6x7 grid in which the players will place their pieces (as depicted below).



3. Your game must have a consistent theme. What will it be?
The theme will simply consist of the three piece colors (R,G,B) and a yellow board. It is a simple theme for a simple game.

4. You must have at least **3** different types of resource/game piece/card that are related to each other. What are they?

1. These should be represented by one struct/enum class/class that you will define

Each player will have 21 pieces at their disposal at the beginning of the game. The points they earn per rounds can be used as a resource to “buy” items.

5. Players must be able to acquire the items from #4. How?

The game pieces are distributed at the beginning of the game, consistently being 21 pieces per player. At the end of the round: If there are three players, the first to connect 4 would win 30pts and the remaining players will battle for 2nd and 3rd place (20pts and 10pts respectively) – if its a stale mate or ends in a draw, they get 15pts. From there, players can redeem points in the “home” for items in between matches.

6. You must have at least **2** different buildings/structures/upgrades/power-ups that are related to one another. What are they?

1. These should be represented either by one base class and one derived class or by different instances of the same class, as appropriate for your implementation.

Upgrades and Power-Ups will include: a space that will double your points at the end of the match if your piece is in it, the ability to remove any chip from the board, a space that allows you to take an extra turn if your chip lands in it.

7. Your game must be multiplayer. How many players will you allow?

Minimum of 2 players, maximum of 3.

8. Your game must have well-defined begin and end states. What are they?

Begin state: empty board

End state: players have connected four, or the board is filled up.

9. (Do you have any other features that you are hoping to incorporate?)

- Customized colors
- Leaderboard
- sounds/animations

Design Proposal

Classes needed:

- home screen
- game
 - pieces
 - enum class for colors
 - board
 - player
 - enum class for power-ups/upgrades/items

We will take advantage of the Prototype Design Pattern to go from match to match, keeping the pieces, players, and board consistent. We will also utilize the Flyweight Pattern to create and utilize the chips. As of right now, the way our game is set up, we don't need anything to be inherited – we could have the power-ups/upgrades/items inherit the piece class, but we are not fully set on that decision.

Checkpoint 1 – goals:

By April 23rd 2021, 11:59pm, we intend to have the GUI (with dialog boxes) and a general template for the classes done.

- Template of classes – with methods and member/fields declarations (h files). Essentially a skeleton of each class.
- Implemented dialog boxes for the menu – user will be able to see the Main Menu, select New Game, enter player info, see the board, and customize pieces.
- Implemented dialog boxes for the shop – user will be able to see the shop menu, item listings, player stats and inventory.