Emily Becher

CS161
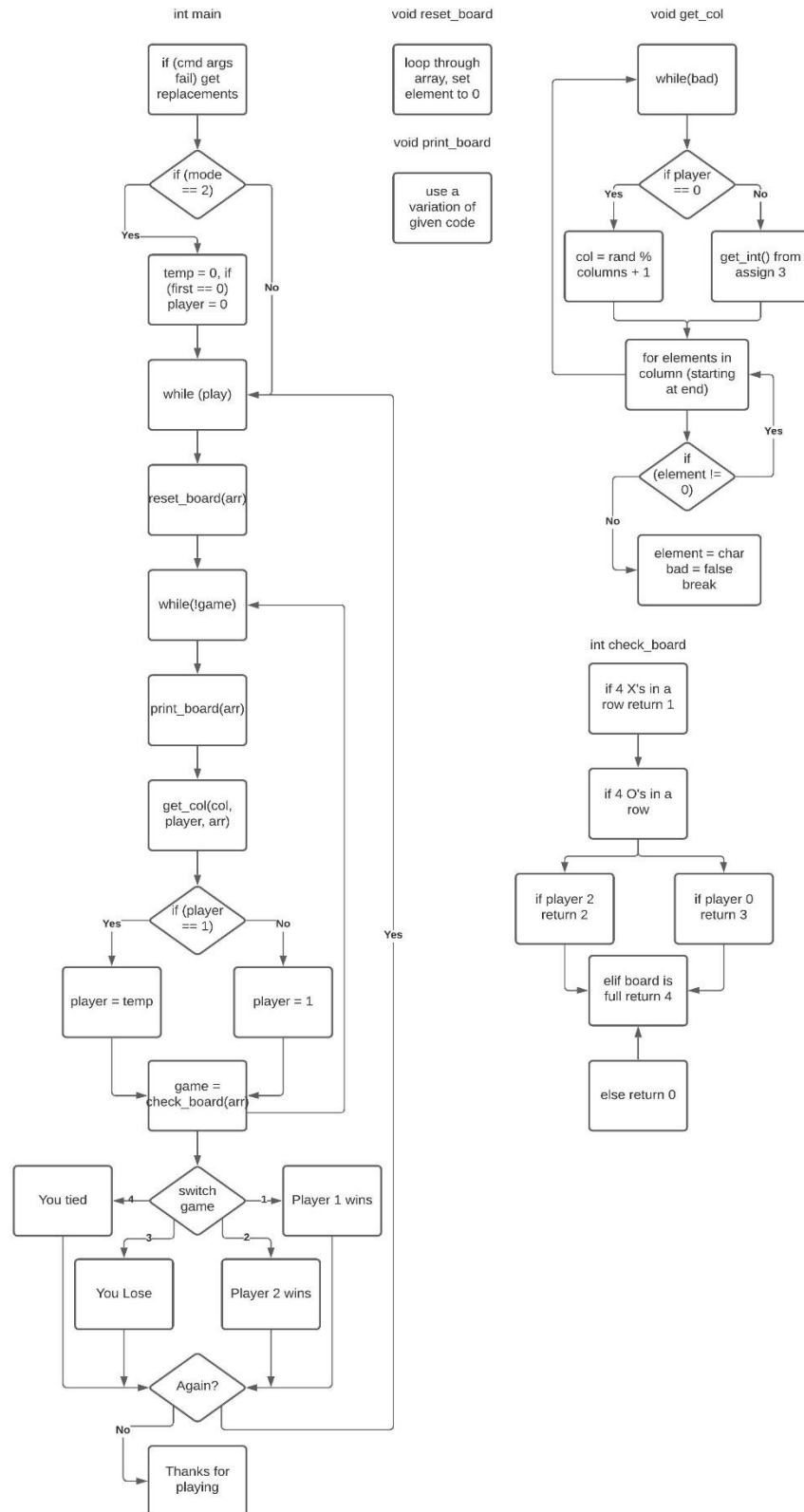
Connect Four Design Document

Problem Statement: Write a c++ program that plays a one or two person connect four game with the user. The subtasks for this program include:

- Get and check number of players (1-2) and board size from command line.
- Display board with numbered columns after each move.
- Prompt for inputs until user enters valid inputs.
- Create and adjust 2D dynamic array with user's moves.
- In one player mode have some algorithm for the computer's moves.
- Determine and output win, lose, or tie.

Understanding the Problem: The program must get inputs for the number of players, rows of the board, and columns of the board through command line arguments. If any of those inputs are invalid, then the program must get the inputs during runtime. In addition, the program must get inputs for who goes first, the players' moves, and if the user would like to play again. The program must output clear instructions and prompts for the user as well as output the board after each move and the result of the game. The program assumes that the user knows the rules and objective of the game. Constraints on the program include functions with no more than twenty lines of code.

Devising a Plan:

int main

```
if (cmd args
fail) get
replacements
        │
        ▼
   if (mode
    == 2)  ──── No ──┐
        │            │
       Yes           │
        ▼            │
temp = 0, if         │
(first == 0)         │
player = 0           │
        │            │
        ▼            │
   while (play) ◄────┤
        │            │
        ▼            │
  reset_board(arr)   │
        │            │
        ▼            │
   while(!game) ◄──┐ │
        │          │ │
        ▼          │ │
  print_board(arr) │ │
        │          │ │
        ▼          │ │
  get_col(col,     │ │
  player, arr)     │ │
        │          │ │
        ▼          │ │
   if (player      │ │
    == 1)          │ │
   Yes     No      │ │
    │       │      │ │
    ▼       ▼      │ │
player = temp  player = 1  │ │
    │       │      │ │
    └───┬───┘      │ │
        ▼          │ │
    game =         │ │
 check_board(arr) ─┘ │  Yes
        │            │
        ▼            │
   switch game       │
```

Yes: You tied ◄──4── switch game ──1──► Player 1 wins
     3 ──► You Lose   2 ──► Player 2 wins
     │   │
     You Lose ──► Again? ◄── Player 2 wins
                    │
                   No
                    ▼
              Thanks for
               playing

void reset_board

```
loop through
array, set
element to 0
```

void print_board

```
use a
variation of
given code
```

void get_col

```
    ┌──► while(bad)
    │        │
    │        ▼
    │    if player
    │     == 0
    │   Yes    No
    │    │      │
    │    ▼      ▼
    │ col = rand %   get_int() from
    │ columns + 1    assign 3
    │    │      │
    │    └──┬───┘
    │       ▼
    │  for elements in ◄──┐
    │  column (starting   │
    │  at end)            │
    │       │             │
    │       ▼             │
    │      if        Yes  │
    │   (element !=  ─────┘
    │      0)
    │       │
    │      No
    │       ▼
    │   element = char
    └── bad = false
        break
```

int check_board

```
if 4 X's in a
row return 1
      │
      ▼
if 4 O's in a
row
   │      │
   ▼      ▼
if player 2   if player 0
return 2      return 3
   │      │
   ▼      ▼
  elif board is
  full return 4
      │
      ▼
  else return 0
```

Testing:

Test Value (a test value that the user could input)	Expected Output (what I expect the program to output)	Match Expected (does plan match expected output)
./connect4 2 5 6	Program continues on to ask if the user wants to go first	
./connect4 3 21 5	Program prompts for number of players and rows	
Two – in row reprompt	Program reprompts since the input is invalid	
User completes a four in a row	Program outputs win message	
User enters column that is already full	Program reprompts and outputs that the column is full	
User enters 8 when there are only 6 columns	Program reprompts and outputs the number must be less than 7	
User moves to fill board without getting four in a row	Program outputs tie message	
1 – in play again prompt	Program resets the board and starts new game	
N – in play again prompt	Program outputs end message	