Catherine Wallin
CSC-17A-49285
10/25/2020

Project 1:
The Game of Life

Introduction

I chose the board game, “The Game of Life” for my project.

It is a multiplayer game (2-4) players. Players progress through the board and encounter various life events such as marriage, careers, taxes, children, and more. Once the players retire, all of their earnings and debts are totalled and the player with the most money wins the game.

The game is a very popular family game to have fun.

Summary

Project Size: 363 Lines
Variables: 5

This project was very fun to start off with, but the further along I worked, the more challenging it became. I mostly utilized pointers and structures to store data and change data within the game.

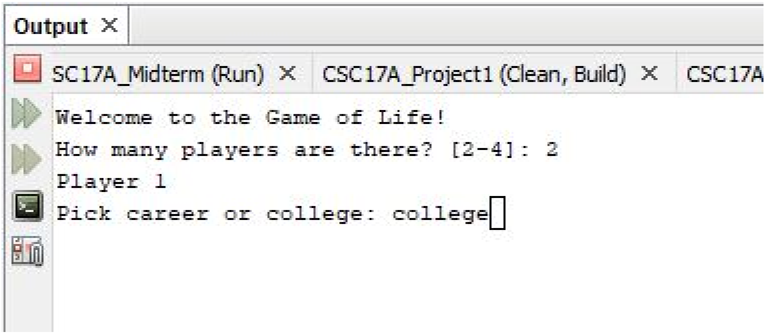
Description

I used structures to hold most of the game data, such as player data and the job data. The player “moves” through an array. Depending on the element in the array, they are guided to a function that goes through the desired set of instructions. At the end, the numbers are added up and the player with the most money wins.

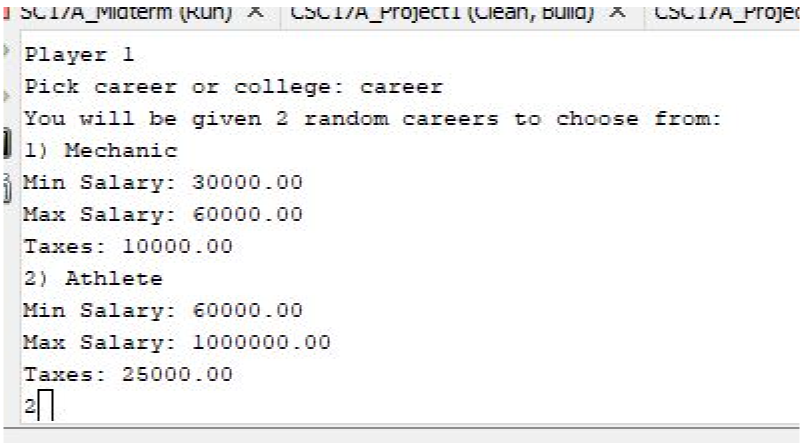
Sample Input/Output

Inputs:

1. The number of players [2-4]
2. “college” or “career”
3. 1 or 2 to select job
4. A filename to write the game data to, example: “gamedata.txt”

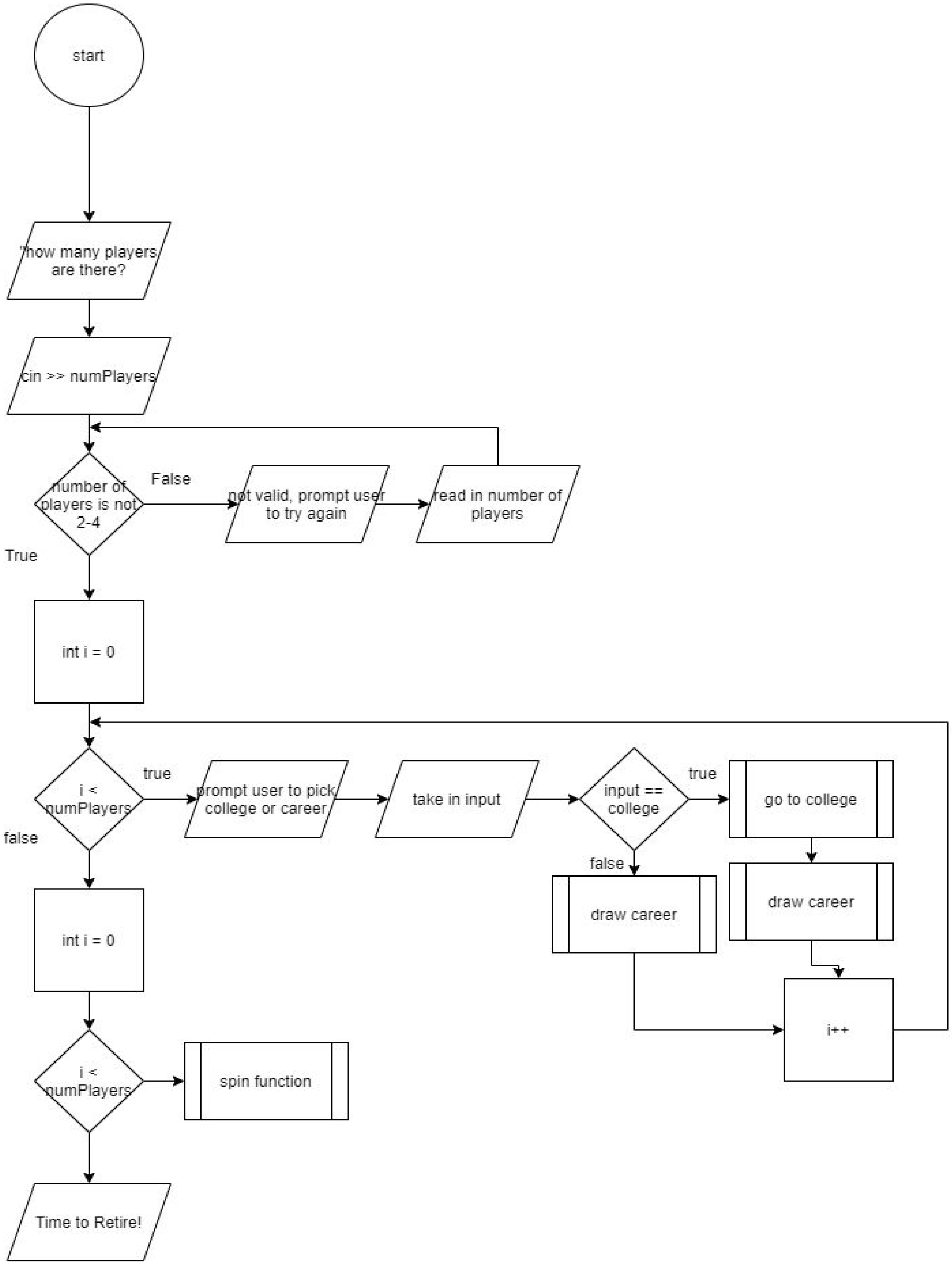


```
Welcome to the Game of Life!
How many players are there? [2-4]: 2
Player 1
Pick career or college: college
```



```
Player 1
Pick career or college: career
You will be given 2 random careers to choose from:
1) Mechanic
Min Salary: 30000.00
Max Salary: 60000.00
Taxes: 10000.00
2) Athlete
Min Salary: 60000.00
Max Salary: 1000000.00
Taxes: 25000.00
2
```

Flowcharts


start
"how many players are there?
cin >> numPlayers
number of players is not 2-4
False
not valid, prompt user to try again
read in number of players
True
int i = 0
i < numPlayers
true
prompt user to pick college or career
take in input
input == college
true
go to college
draw career
false
draw career
i++
false
int i = 0
i < numPlayers
spin function
Time to Retire!

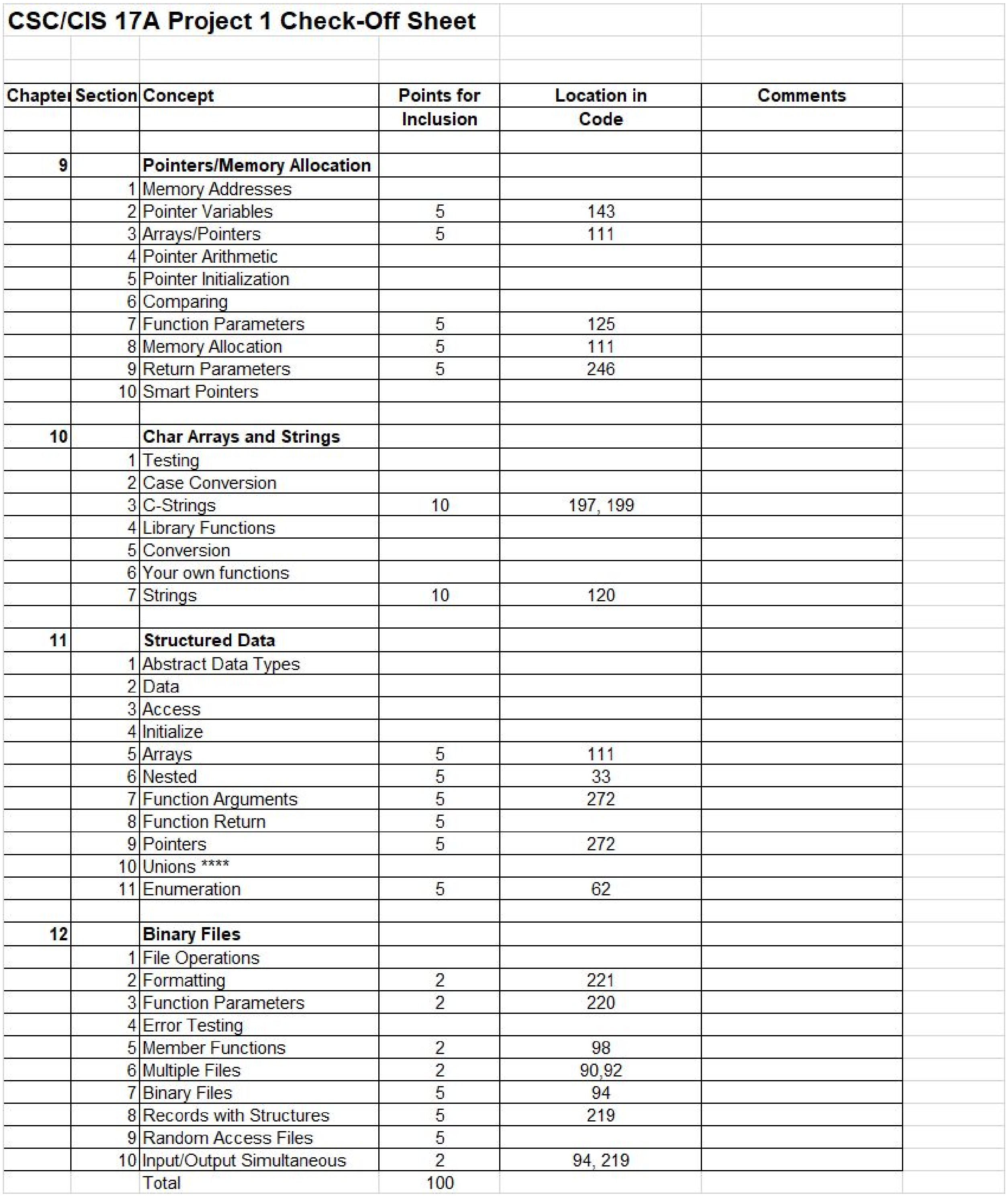
Pseudocode

```
Ask user for number of players
Read from the keyboard the number of players (to use to loop)

Loop through players
    Pick college or career
    Generate two jobs based on career path
    Read from the keyboard the user selection
    Store job and its members inside the player structure

Loop through turns until a player has reached the end of the board
    Loop through players
        Random number generator
        Add random number generator to the player’s current place
        Use the added numbers to “move” to that element of the array
        Execute the function that coordinates to that element

Add together the total money and debts of each player
Write the game stats to a file
```

Variables

Int maxBoardPos ~ to determine the players’ positions on the board, indicating when to stop the game
Int numPlayers ~ used to loop through the different players when taking turns in the game

Checklist (file in project folder)



CSC/CIS 17A Project 1 Check-Off Sheet

Chapte	Section	Concept	Points for	Location in	Comments
			Inclusion	Code	
9		**Pointers/Memory Allocation**			
	1	Memory Addresses			
	2	Pointer Variables	5	143	
	3	Arrays/Pointers	5	111	
	4	Pointer Arithmetic			
	5	Pointer Initialization			
	6	Comparing			
	7	Function Parameters	5	125	
	8	Memory Allocation	5	111	
	9	Return Parameters	5	246	
	10	Smart Pointers			
10		**Char Arrays and Strings**			
	1	Testing			
	2	Case Conversion			
	3	C-Strings	10	197, 199	
	4	Library Functions			
	5	Conversion			
	6	Your own functions			
	7	Strings	10	120	
11		**Structured Data**			
	1	Abstract Data Types			
	2	Data			
	3	Access			
	4	Initialize			
	5	Arrays	5	111	
	6	Nested	5	33	
	7	Function Arguments	5	272	
	8	Function Return	5		
	9	Pointers	5	272	
	10	Unions ****			
	11	Enumeration	5	62	
12		**Binary Files**			
	1	File Operations			
	2	Formatting	2	221	
	3	Function Parameters	2	220	
	4	Error Testing			
	5	Member Functions	2	98	
	6	Multiple Files	2	90,92	
	7	Binary Files	5	94	
	8	Records with Structures	5	219	
	9	Random Access Files	5		
	10	Input/Output Simultaneous	2	94, 219	
		Total	100		