**Introduction:**

***What I want my final project to be like...***

* Player has properties like name, score, win (status whether they win the game or lose the game), and bet
* User gets to choose a game to play
* User will bet desired amount of point on the game
* User will lose their points if they lose the game, gains if they win the game depending on the rate of that game
* Each game will have a different interest rate
* Create an abstract class called MiniGames with a property rate
* Create classes for each game inheriting MiniGames class
* Create user interfaces for each game
* Create a list of players
* Arrange the order of the list so that it is in order of their final scores
* Displays the list of final scores
* User gets to keep playing, or create a new player, or exit the program

***Lists of things I want to include***

* Advanced classes
* Use of Object class
* LinkedList
* Static methods/properties
* Interface
* Generics
* Recursion
* Graphics (UI)

**Reflection:**

I achieved most of my plans in terms of the overall structure of this project: created Player class with properties that I intended to; made more than one game; introduced a betting system and ranking; made an abstract class 'MiniGames' (parent of classes for each mini game), and each class inherits this class's methods and properties as well as their unique ones. On the other hand, I couldn’t introduce recursion and interface to my project, although they were on my initial list to be included. This final project doesn't have a lot of visual components or animations, but I used ideas and contexts that we've learned in the 40s course (different types of classes, collections, dynamic structures, etc.) which helped me write code more efficiently.

***Things I learned / improved during the process of making this project***

* Commenting - I became more consistent with commenting my code compared to the beginning of the semester, and it helped me stay up to date with things that I'm working on.
* Organizing – separating into folders and classes helped me stay organized and it made it easier to detects/fix problems in the code
* Error checking - I came to think about edges more and pay attention to possible errors my code could have caused. I caught several defects in my code that I would have missed before I took this course.
* Efficiency - I think I was able to code more efficiently than before: I got better at the use of variables, methods and classes more appropriately which made it easier for me to edit things later or re-use them in different parts of the project later on.

***Things I could have done differently & Future improvements***

* Recursion - I wanted to use recursive methods in my final project but I couldn't.
* Time management – it took me longer than I thought to finalize my idea and start to actually shape its structure. I could have started writing my code earlier even if my idea didn't seem solid enough, which would have allowed me to have more time to improvise the project.
* User Interface - I wish I could make this project more appealing and look cool (animations, sound effects, etc.). It took me longer to construct the overall structure, and I didn’t have much time to elaborate how it appears.
* I stuck with simpler games like rock paper scissors and number guessing, but I would have challenged myself with more advanced ones. I first wanted to have a brick breaker, but I gave up considering the amount of time left after I finished making the main part of the project. Now I think I might have been able to do it using code repositories and it would have made this program more interesting.
* Errors – even though I tested my code after writing each of new codes, to see possibilities of errors, I believe there are still more errors that could potentially occur, and it requires even more testing so that it can ultimately deal with all edges.

As I compare myself to last year as a programmer, I recognized some improvement in my code. I can confidently say that I am a lot more comfortable with working on a project in general, and I built up my own habits as a programmer. This project has a numerous possibility to grow further and could be developed in different ways might have been more efficient, but overall, I am satisfied with what I have done to conclude this semester.