My Code Projects

Project 1: *The Ranch*

The Ranch was the final project of semester 1 and of programming 1 where we were tasked with making any sort of game we could immagin. I build a collection style game where the player must round up as many cows as they can in 2 minutes. This game allowed me to explore the use of object oriented codding and making global character classes re-used throughout the code. The game features a start screen, a main game screen with timer and scoreboard and a game-over screen with the ability to replay or quit.

Technologies used

- Coded in Visual Studio 2022 *C#*
- Use of `Class` for Cows
- `DateTime` to keep track of game time and cow movement
- `ConsoleKey` to detect player input
- `String[]` allows many lines of text in one object
- `ForgroundColor` is used to color what is writen in console

Code Snippets

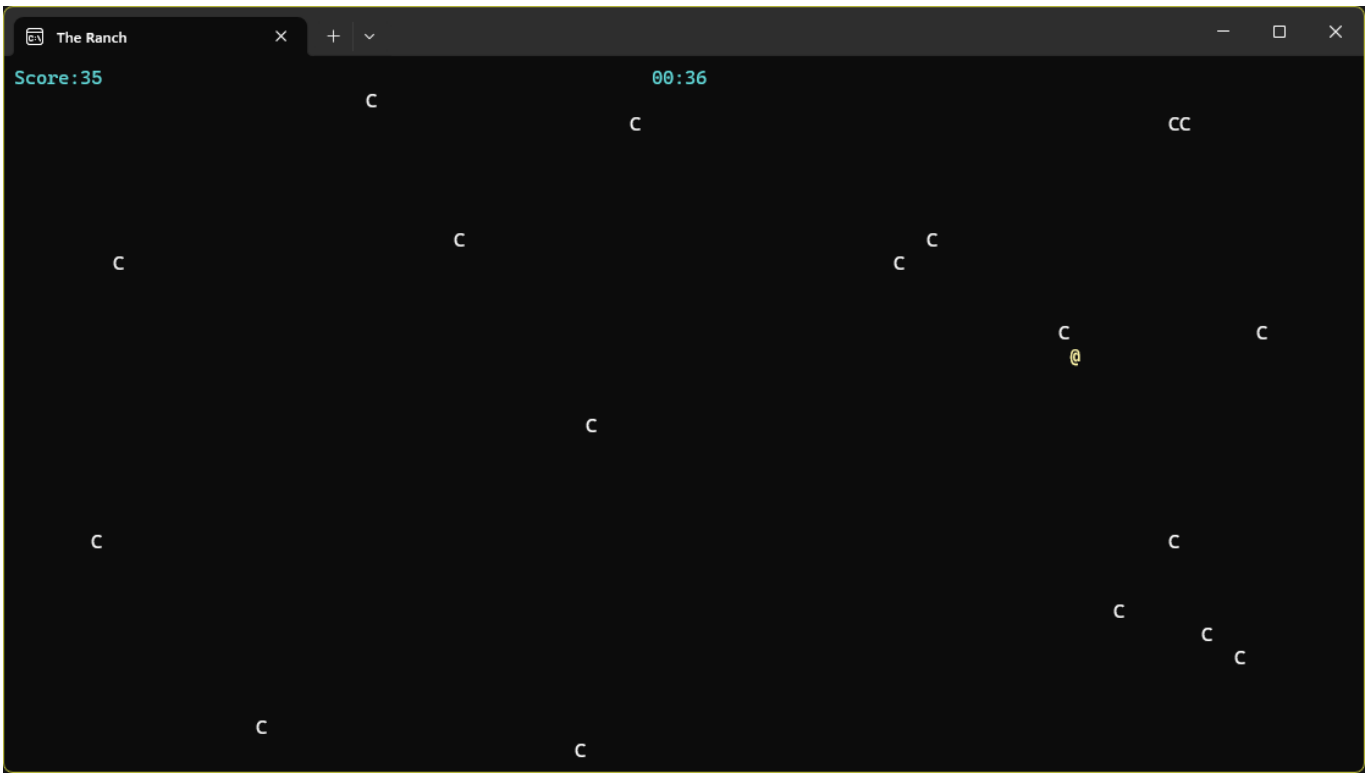
```
        class Cow
        {
            public Int32 x, y, oldx, oldy, offsetx, offsety, health;
            public char cowchar = 'C';


            public void MoveCow()
            {
                oldx = x; oldy = y;

                offsetx = Program.GlobalRandom.Next(-1, 2);
                offsety = Program.GlobalRandom.Next(-1, 2);


                x += offsetx;
                y += offsety;

                if (x < 0) { x = 0; }
                if (x >= Console.WindowWidth) { x = Console.WindowWidth - 1; }
                if (y < 1) { y = 1; }
                if (y >= Console.WindowHeight) { y = Console.WindowHeight - 1; }
            }

            public Cow()     // this is a constructor its supposed to have the
same name as the class
            {

                x = Program.GlobalRandom.Next(1, Console.WindowWidth);
```

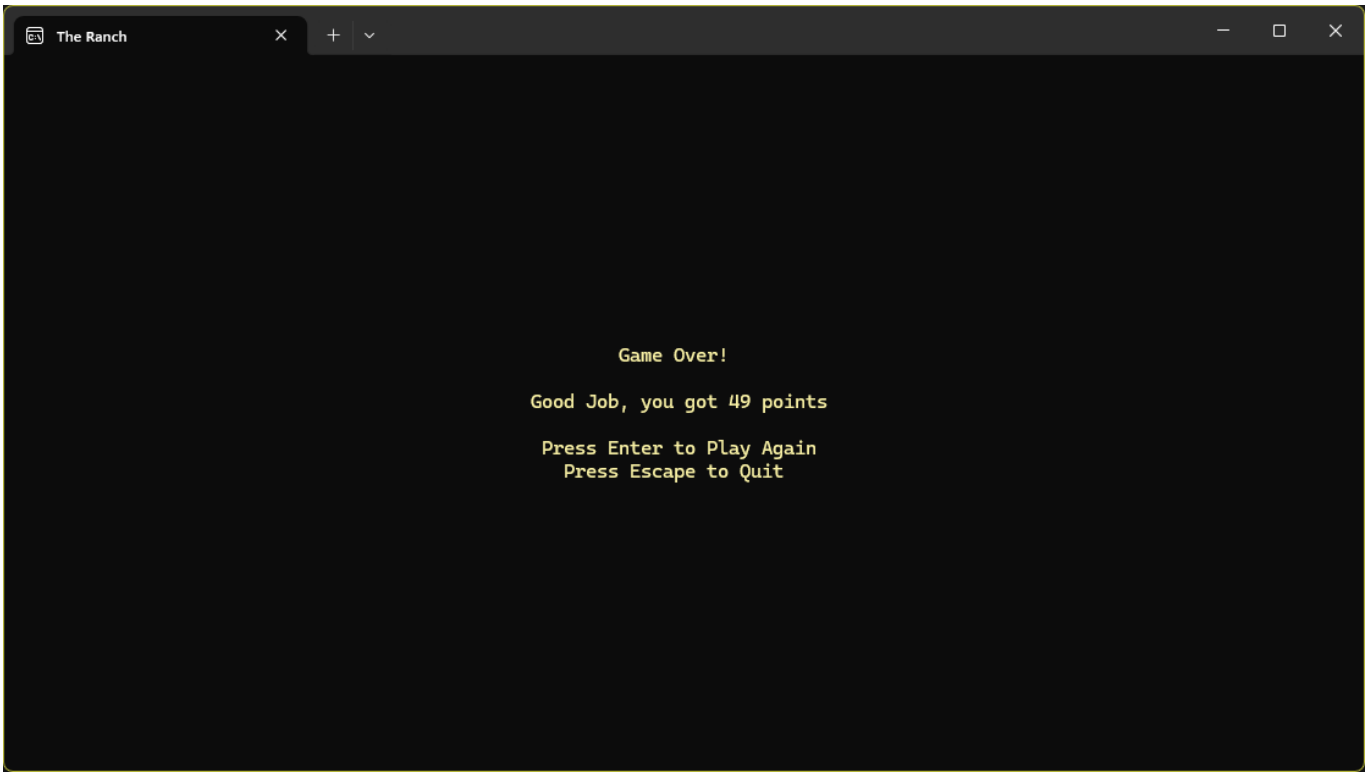
```
                y = Program.GlobalRandom.Next(1, Console.WindowHeight);
                health = Program.GlobalRandom.Next(1, 101);
            }

        }
```

```
static void DrawCenteredMenu(string[] lines)
{
    Console.Clear();

    Int32 screenWidth = Console.WindowWidth;
    Int32 screenHeight = Console.WindowHeight;

    Int32 menuHeight = lines.Length;
    Int32 topPadding = (screenHeight / 2) - (menuHeight / 2);

    Console.ForegroundColor = ConsoleColor.Yellow;


    for (int i = 0; i < topPadding; i++)
        Console.WriteLine();


    foreach (string line in lines)
    {
        Int32 leftPadding = (screenWidth / 2) - (line.Length / 2);

        if (leftPadding < 0) leftPadding = 0;

        Console.SetCursorPosition(leftPadding, Console.CursorTop);
        Console.WriteLine(line);
    }

    Console.ResetColor();
}
```

Photos of *The Ranch*


The Ranch
Welcome to The Ranch
Round up as many cows as you can in 2 minutes
Use the arrow keys to control your rancher
Round up Cows by moving over them
Press Enter to Start


The Ranch
Score:35
00:36



What I Learnt

> C# is a very versatile coding language that can be easily understood and mastered to use for complexe tasks. Modern video games have many layers of theses smaller, simpler lines of code running in tandem to create the more detailed game we play. By using object oriented code, the manipulation of these entities can become easier.

Project 2: *Cooler Math Games*

Cooler Math Games is a c# project that allowed me to understand the use of several key technologies in C#. This project explores 5 distinct games that utilize different opperators to accomplish a certain task. For example: random number generation for a guessing game, modulo to explore the collatz conjecture, switch cases to make a menu, and more.

Technologies used

- Coded in Visual Studio 2022 *C#*
- `Switch` used to monitor choices in menu system
- `Modulo %` used to check odd vs. even in collatz conjecture
- `while` loop used in coin game to loop through coin types
- Nested `if` loops used in oldest person game
- `Random rand` used to generate random number for guessing game

Code Snippets

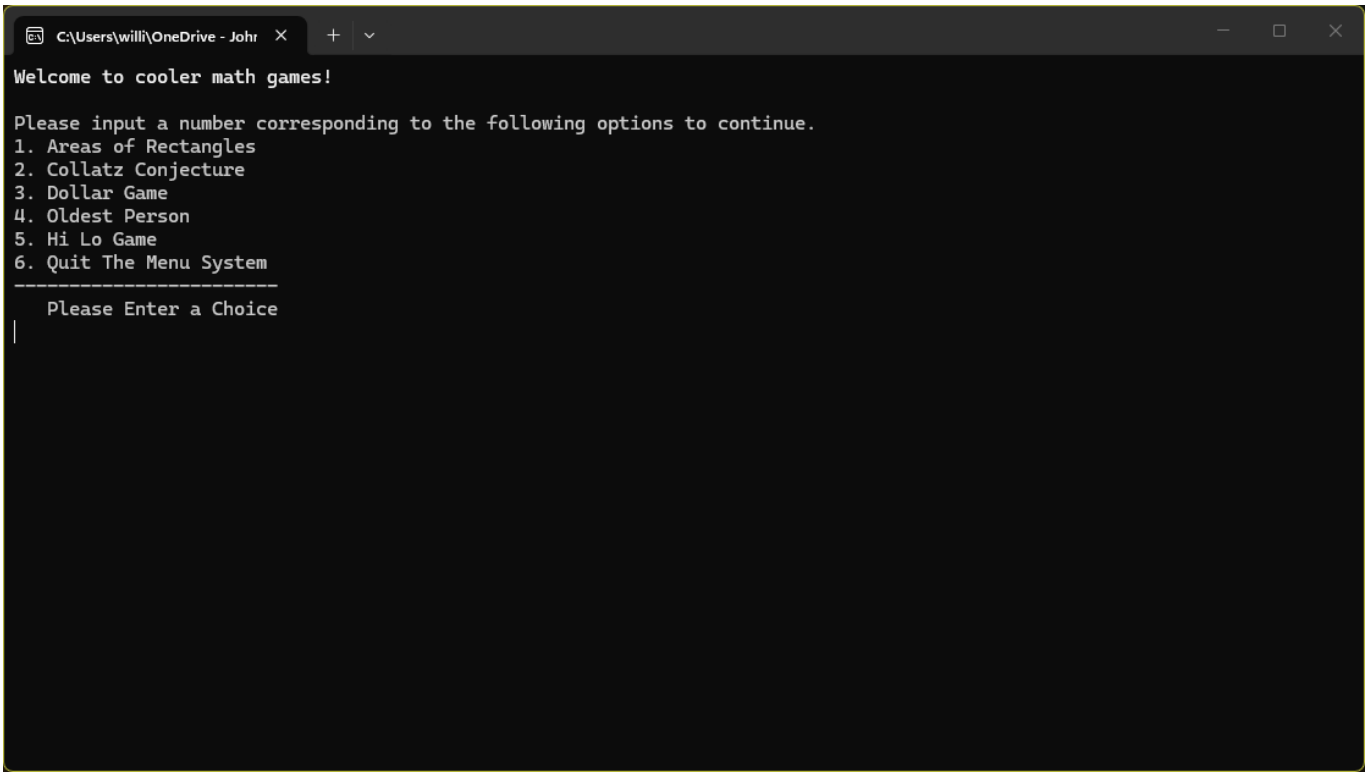
```
switch (choice)
{
    case 1:
```

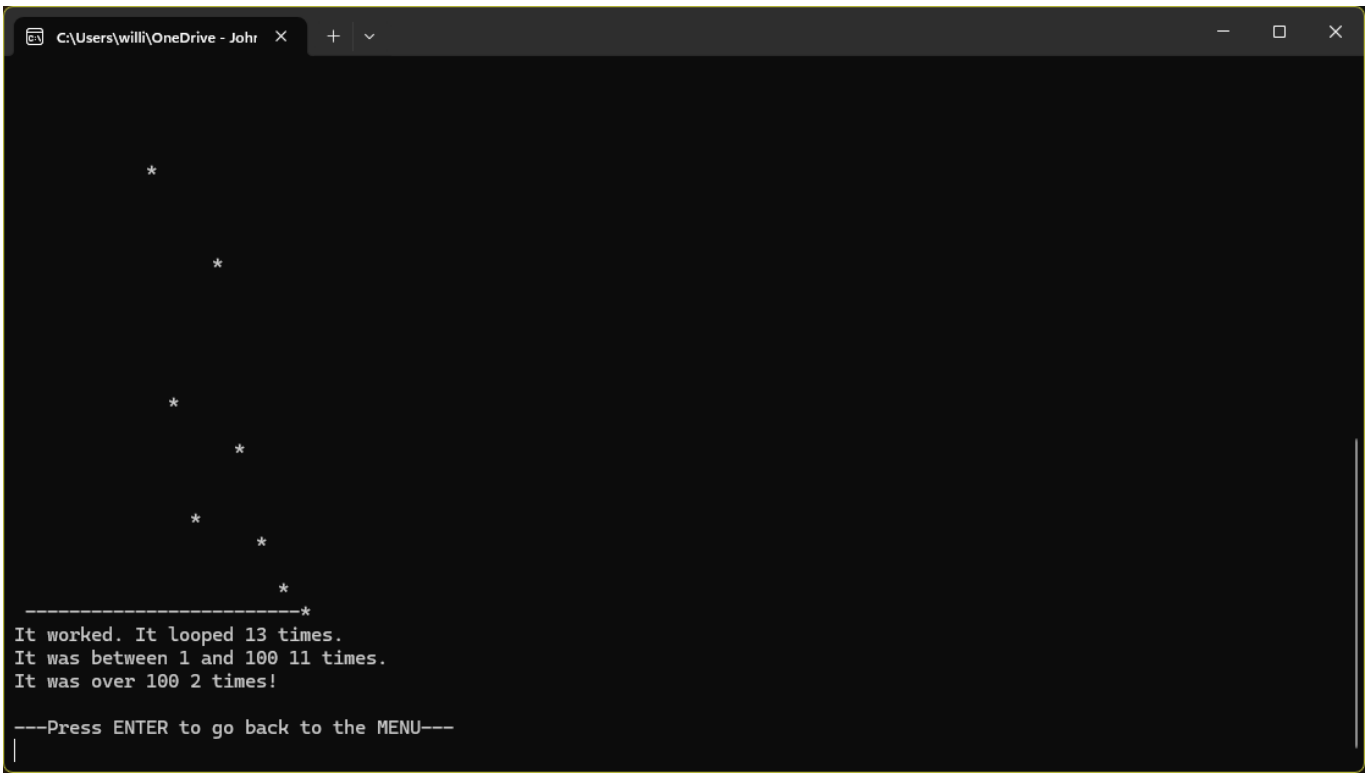
```
            Rectanglefull();

            break;
        case 2:

            Collatzfull();

            break;

        case 3:

            Dollargame();

            break;
        case 4:

            Oldestperson();

            break;
        case 5:

            Hilogame();

            break;
    }
```

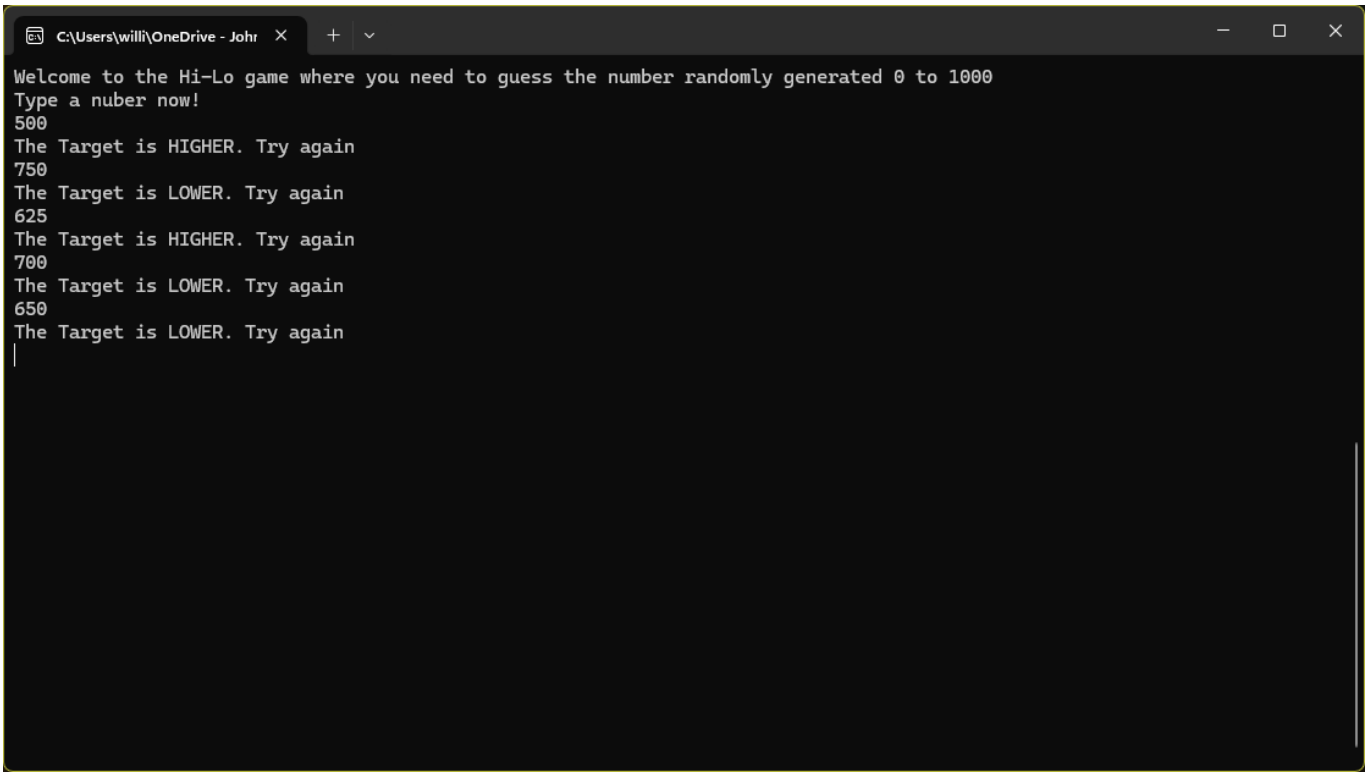
```
        static void Hilogame()
        {

            Console.Clear();
            Int32 target, guess, tries;
            tries = 0;

            Random rand = new();
            target = rand.Next(0, 1001);


            Console.WriteLine("Welcome to the Hi-Lo game where you need to guess
the number randomly generated 0 to 1000");
            Console.WriteLine("Type a nuber now!");
            do
            {
                tries++;
                guess = Realnumber();

                if (guess == target)
                {
                    Console.Clear();
                    Console.WriteLine("\n\n\nCongratulations, you got it!\nIt was
" + target + "!\nYou got it in " + tries + " Tries!");
                }
```

```
                else
                {
                    if (guess > target)
                    {
                        Console.WriteLine("The Target is LOWER. Try again");
                    }
                    else //this should mean it was less than target
                    {
                        Console.WriteLine("The Target is HIGHER. Try again");
                    }
                }



            } while (guess != target);


            Console.WriteLine("\n\n---Press ENTER to go back to the MENU---");
            Console.ReadKey();
        }
```

Photos of *Cooler Math Games*


C:\Users\willi\OneDrive - Johr
Welcome to cooler math games!
Please input a number corresponding to the following options to continue.
1. Areas of Rectangles
2. Collatz Conjecture
3. Dollar Game
4. Oldest Person
5. Hi Lo Game
6. Quit The Menu System
Please Enter a Choice


C:\Users\willi\OneDrive - Johr
It worked. It looped 13 times.
It was between 1 and 100 11 times.
It was over 100 2 times!
---Press ENTER to go back to the MENU---



What I Learnt

> This project was a perfect demonstration of the use case of these newly introduced techniques. The games where simple enought were we focused on the logic rather than the execution.