**Object Pooling Implement:**

Object Pooling is implemented by taking a certain prefab and create multiple pools (I chose to do 10). After every 1 second the game runs it will activate a new duck from the scene and when the duck has been shot (mouse press) it will deactivate the duck from the scene and increase the score by 1. This benefits the game because it will need less information instead of having to keep spamming a new duck every 1 second it will just bring back the same 10 ducks being used which will need the game to require less memory usage. Once you have killed 10 ducks the game will be over and you win, if all 10 ducks are in the scene it will be game over and you lose.

Graphical user interface

Description automatically generated with medium confidence

There are some spikes when a new duck is active but besides that it will always stay smooth.

**Command Implement:**

Unfortunately, I was not able to invert the mouse position but instead I decided to invert the camera rotation to act as an invert mouse position. How I did this was I created a class that will take the basic functions (Execute, Redo, Undo) and overridden them to do what I want for this specific type of command. So, for this I made it so when the command is executed it will check if it wants to be inverted or not and if so, it will rotate the camera upside down. This command will be invoked when the player has missed the ducks 2 times in a row. In order for it to check it will see if you clicked on anything outside of the ducks and if so it will count it as a miss and when the miss counter reaches 2 it will invert the camera for 5 seconds.

**Game Management System:**

I believe to have been using multiple game management systems including input manager and score manager. The input manager will use the new input actions Unity system and will be used as a Singleton to know when it has been enabled or disabled. It will also use this to be able to take the controls and remember them so they can be used on other classes and functions. The score manager will keep track of the score of the player and when the player has hit 10 ducks it will tell the game that it is over, and the player has won. We use this as a singleton as well so that we can keep the score onto one class but change its values from other classes and conditions.