**Work in Progress Report #5**

**Major developments/breakthroughs(reference specific code please):**

This WiP I was able to get a lot cleaner movement in the velocity section of my program and I was also able to finish off the gravity component also. I used a timer for movement in the velocity section and I also used a timer in gravity to be able to make all of the objects drop.

**Major Challenges/setbacks( reference specific code please):**

The biggest setback I had was being able to use the same concept I used to make the sprite move and speed up in Velocity to also make it slow down after the key is released. I think I will be able to work on it though and have it incorporated in time for the final project.

**Any modifications to your specifications/release schedule:**

I’ll have to back up on my work in the functions side to finish off the slowing down aspect of the velocity section. I’m hoping to be able to have something going on in the functions side before the final presentation though.

**Description of your scratch/test program:**

**Describe the generic concept you needed to test out:**

TimerMovement: In this scratch I worked on using a timer to move a sprite. It would start a timer when you pressed a key and as you held it down it would move the sprite every time the timer increased a counter. Overall it made the movement of the sprite so much cleaner and less jerky.

ButtonTimerMovement: In this scratch I used a button to start a timer to move a sprite. When you clicked the button it would start a timer which would then make the sprite move every time it increased the counter and would only start when you clicked a button which would then stop the timer.

**Source any web site/book that helped you with that concept:**

N/A

**Describe the code and the lesson that you learned from it**:

Timer Movement: As you held down a key it would start a timer in panTimer which would increase a counter by 1 every time it ticked. In panDemo it would read in the counter from panTimer and would then increase its own timer by one. Every time the counter increased it would go to the setX or setY class in SprPerson and would increase or decrease it’s x or y.

Button Timer Movement: When you clicked a button it would start a timer which would increase the counter in panTimer. As long as the timer was going it would increase the counter which if it was greater than 0 (which is always) it would send to panDemo which in course would set the y co-ordinate in SprPerson to make it move. You could click the stop button which would then stop the timer and stop the sprite from moving.