**Change made from UML design:**

* Adding many constant which to the class which were not defined in the original UML. EG: ONESECONDS, ZEROPOINT4SECONDS in the World. I did not realize there were so many constant after actually implement the code
* Delete tempSprite[][]][ from Sprite Class, it turns out that using tempSprite[]][][ as buffer for temporary is meaning less, I could just do the sprite updating in the World. And it makes more sense, because in my design, every change in the sprite should notify world because they are all objects in the world.

**Any Difficultis you had during this Project:**

* Time limit. Though I drawed the UML first,

**One Key Piece of Knowledge you have Learned:**

* From thie project I have deeper understanding in polymorphism and inheritance. For instance, in the Project 1, I did not know whether child class method can change the instance variable which defined in the parent class. This caused me a lot of trouble in Project1, in which case I had to use a very weird logic to update my Player position, and tutor comment that “the logic in updating player is a bit weird”. However, this time, I learned that if a child class’s method want to change the variable that inherited from parent class, it should call the parent class’s setter to set it. In this way I could maintain the oop idea, while do the job.   
  Speak of Polymorphism, I was not sure when I have an parent class reference to a object, which actually was created as a child class object, how should I use child Class method. One way is to perform overriding in the child class, therefore when the object is called by the parent class reference, it will perform the lowest level of class’s method.

**Anything you would do differently if you did a similar Program:**

* Instead of Just drawing the UML, I should also consider the doing sudo code for some key methods. For example, the update method in the Program. The logic was quite confusing in my first attempt to do the project2B. I throwed away all OOP design and just tried to get everythin work. And after a few days, I could not recognize my code. Such rush to finish a function did not spped up my work, rather, I have to delete everything, and reconstruct my UML and Logic in update methods to make my work more clear, as well as setting up a clear logic level. Which, in the afterward work, I keep stick to the OOP idea as much as possible, and this saves me a lot of time because now I can treat many functions as a black box – all because I strictly consider the logic and what input, out should be. I wasted a lost of time to reconstruct my rush code, next time I should plan carefully first, then do the code. From this project, I have enhanced my understanding in the importance of design and OOP idea. Design will not waste your time, but provide clear logic and efficiency.