Nick Tansino

Prof. Brown

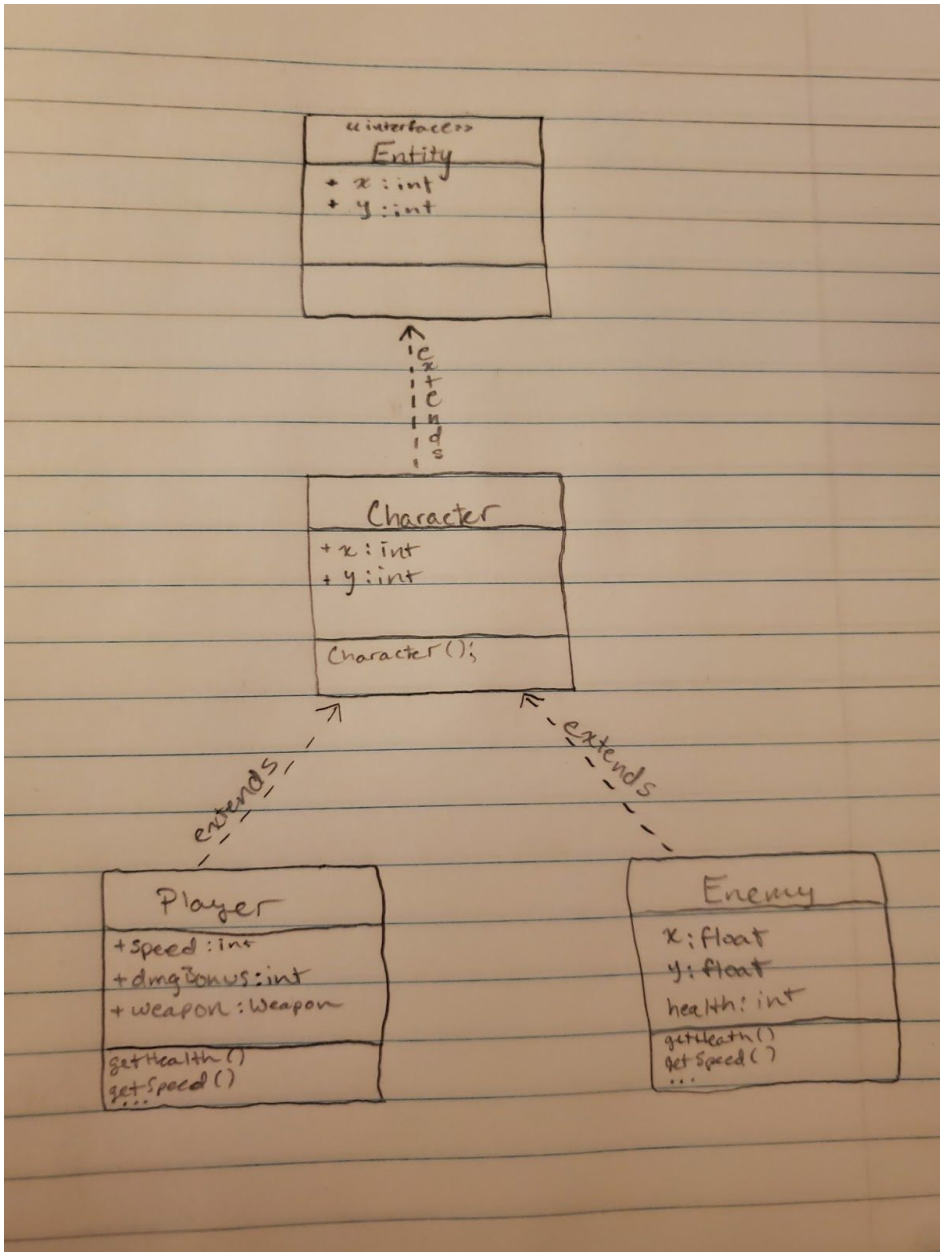
CSC 305

21 April 2020

Prototype Design Pattern:

The design pattern that I had picked for this assignment is a creational design pattern called the “Prototype” design pattern. This pattern is useful in many different situations, such as when the instantiating of certain objects in code can be too taxing for the program, which later leads to performance issues. This is precisely the reason that I chose to use this design pattern in my project. Pankaj of JournalDev states that the “..prototype design pattern is used when the Object creation is a costly affair and requires a lot of time and resources and you have a similar object already existing.” When there are multiple instances of objects in a game that could easily be extended off of one another, it makes a great deal of sense to make a “skeleton” object, that of which will help create objects of a similar type without overloading memory processing.

As previously stated, I found this to be a suitable candidate for my project due to the number of objects that we have in our game. Without creating some sort of placeholder object for a certain character, for instance (whether that character is a player or an enemy), there would need to be a lot more time to process each object individually, which is not beneficial. With this “skeleton-object”, things like processing time and memory management will run a lot more smoothly, given the implementation is correct. Below is where you will find a UML Diagram that clearly shows the implementation I plan to use.


«interface»
Entity
+ x : int
+ y : int
implements
Character
+ x : int
+ y : int
Character();
extends
extends
Player
+speed : int
+dmgBonus : int
+weapon : Weapon
getHealth()
getSpeed()
...
Enemy
x : float
y : float
health : int
getHealth()
getSpeed()
...

Work Cited:

M. O. Onarcan and Y. Fu, “A Case Study on Design Patterns and Software Defects in Open Source Software,” *Journal of Software Engineering and Applications*, vol. 11, no. 05, pp. 249–273, Nov. 2018.

“Prototype Design Pattern in Java,” *Prototype Design Pattern*. [Online]. Available: https://www.journaldev.com/1440/prototype-design-pattern-in-java. [Accessed: 23-Apr-2020].