

# CPS 108 VOGGA Individual Design

Name: Donghe Zhao

Date: April 6<sup>th</sup>

Genre: Fighting Game

Teammate: Hareesh, Helena, Chen, Peggy, Wendy, Hui

UTA: Nathan, Tanner

GitLink: <https://github.com/hareeshganesan/Vooga/tree/aibranch/src/PhysicsEngine>

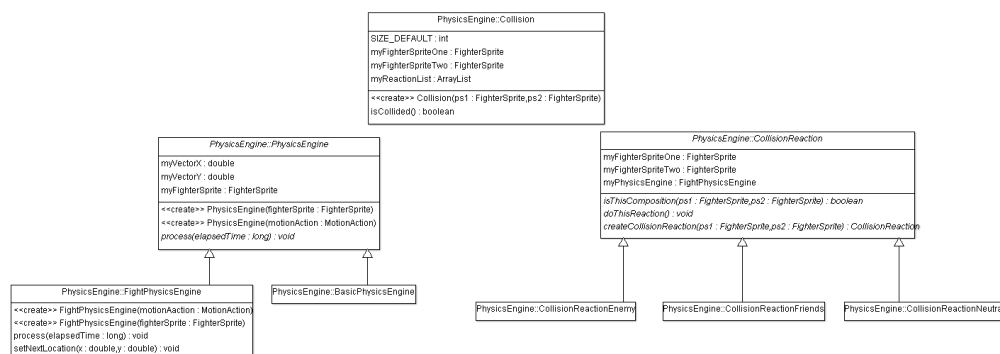
## Improved:

At first I wrote a lot of codes on the movement, but after discussion, we decide to do all the movements outside of physics engine. So physics engine just calculates the new location, and set it. This means almost all my work was meaningless. However, that's fine. Now the calculation is not a very hard part, I can pay more attention to the collision.

Now the responsibility of physics engine is very clear: get the MotionAction, which also means I can get the specific fighter, its direction and speed; check whether there is a collision; if a collision happens, give feedback to physics engine to do the reaction; else we just check whether the new position is inside of the bound; if it is out of bound, decrease its length of movement to make the sprite just move to the bound. So this basic process is very clear now.

For the collision part, I used Factory Method Pattern to create a kind of CollisionReaction, then let its subclasses to decide which specific reaction it needs to be created. After creating the specific collision reaction, I gave some basic feedback movement based on these two sprites. For example, when the two fighters collide, both of them will stop.

This is the UML of the codes which I have written.

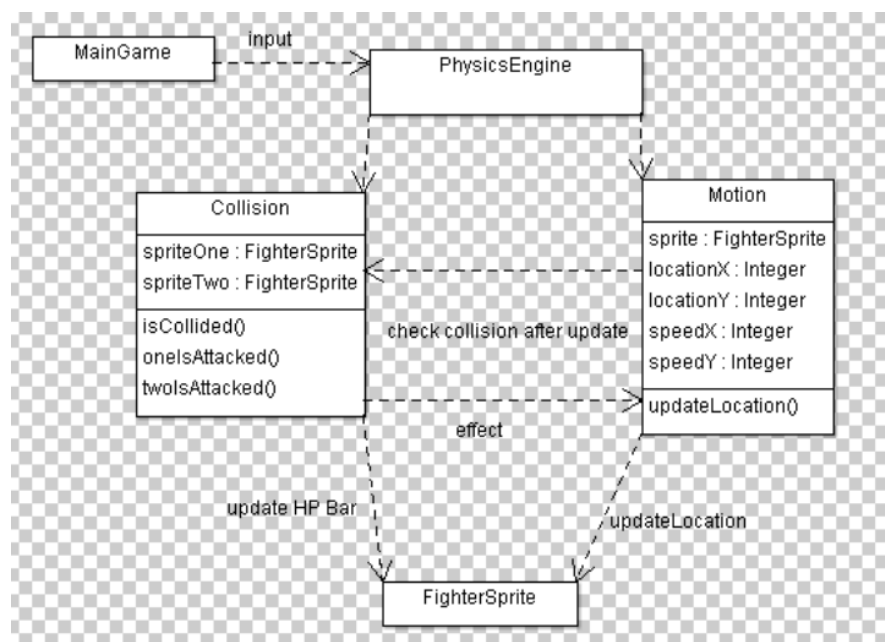


Therefore, the basic skeleton of Physics is much clearer than before. Although my part did not work very fancy, I did a lot before we got our decision. In the next week, I would like to implement more and add some more features

### Proposal:

In this project, I am responsible for “Physics Engine” part which includes “Motion” and “Collision”. The main function of “Motion” is to calculate and update the location of every FighterSprite. While the main function of “Collision” is to check whether there is collision between two groups of FighterSprites, and identify which one is attacked. After that, there is to reflect to the attacked FighterSprite.

Now, let’s talk this process in detail. We get “input” from the MainGame, the “input” includes the action from players, such as what key the players press now, and the system timer that helps physics engine to check and update every specified time. When Physics Engine gets this input, it will calculate the new position, if the new position is out of the background, we should force it to background. In other case, the new position is just the former position plus the speed times direction times eclipsedtime. After we calculate the new location, we check whether there is any collision between two groups of FighterSprite in our screen with their new location. If there is no collision, we just need to wait for next input from MainGame. While, if there is some collision, we should figure out which side is attacked, and how much damage it should get. And then send its feedback to the FighterSprite. The FighterSprite will update its HP bar, location and so on. Here is the UML of this part.



However, there are really some tough problems that need me to consider. For example, as UTA Nathan said “what if the fighting game is in the water how to calculate location”. It is a very good question, which reminds me that I really need to consider many special cases. I think basically I plan to use Strategy method to solve

this problem in the high level, that means I think of a lot of solutions for each specific case. This is really a very complicated parts. I cannot come up all the cases now, so I have to cover some cases now and consider others in the next couple of weeks. I believe I can do it well. Thank you!