Yang Student

Professor Kevin Dill

CS 4150/5150: Game Artificial Intelligence

March 5, 2021

Project 2: Collision Avoidance

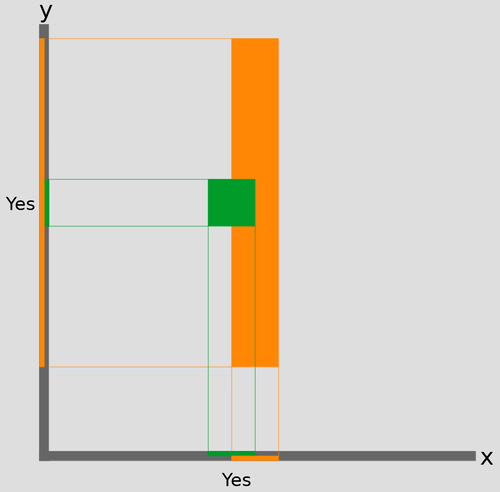
**What runs well**

I think I covered all the points in the homework requirements. I also added health bars to Entities for better visualization.

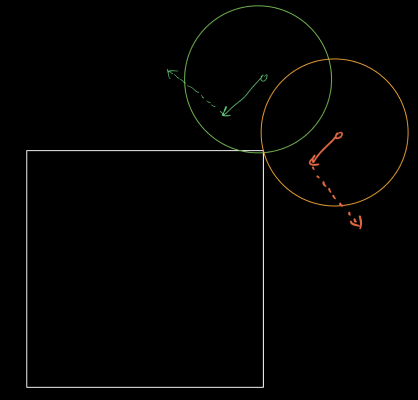
**Compile instruction**

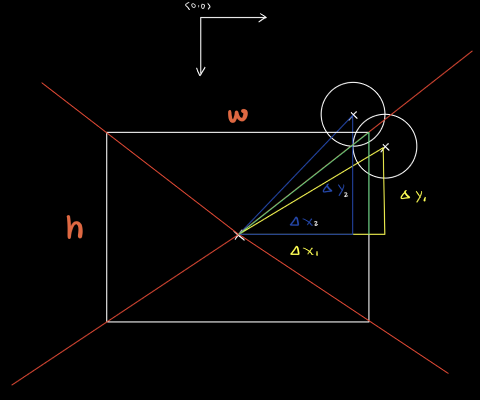
Compile in Visual Studio 2019 on Windows 10. The executable should be ./Debug/Game.exe.

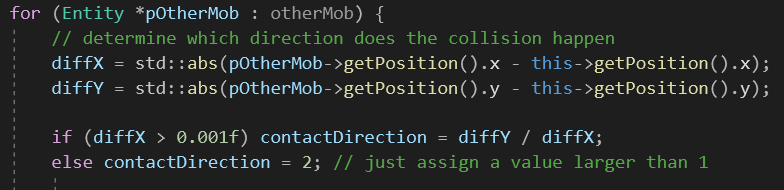
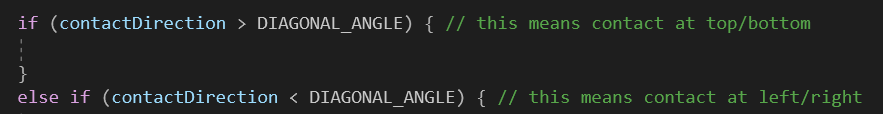
**Collision algorithm**

I used the Separating Axis Theorem (SAT) for box collision check. The theorem tells collision by confirming if the two boxes’ x and y ranges are overlapping with each other.



I also used an original idea (though I am certainly sure that I reinvented the wheel) for finding which side of the be-collided box does the incoming box hits. Here I will demonstrate it through a ball and paddle example.



Basically, the algorithm tells the collision direction (hitting top/bottom or left/right) by comparing two involving objects’ position difference ratio (delta y / delta x) and the be-collided object’s height / width ratio.

The DIAGONAL\_ANGLE is set to be 1, as all units in this project are squares.

**What could be improved**

1. Units sometimes wiggle very frequently, though shortly, when more than 3 same mobs collide with each other. Cases like this should be optimized to make the game runs more naturally.
2. The be-pushed units’ speeds are not always calculated according to their original speed. For example, when giants are moving around the buildings, I used a hard-coded speed, to avoid possible super slow motions. This should be improved for generalization purpose (adding more mobs won’t need more hard-coded bypass speed).
3. The graphics should be improved. Though it is not in the requirements, improvements like winning screen and basic UI are certainly doable and are great practices.