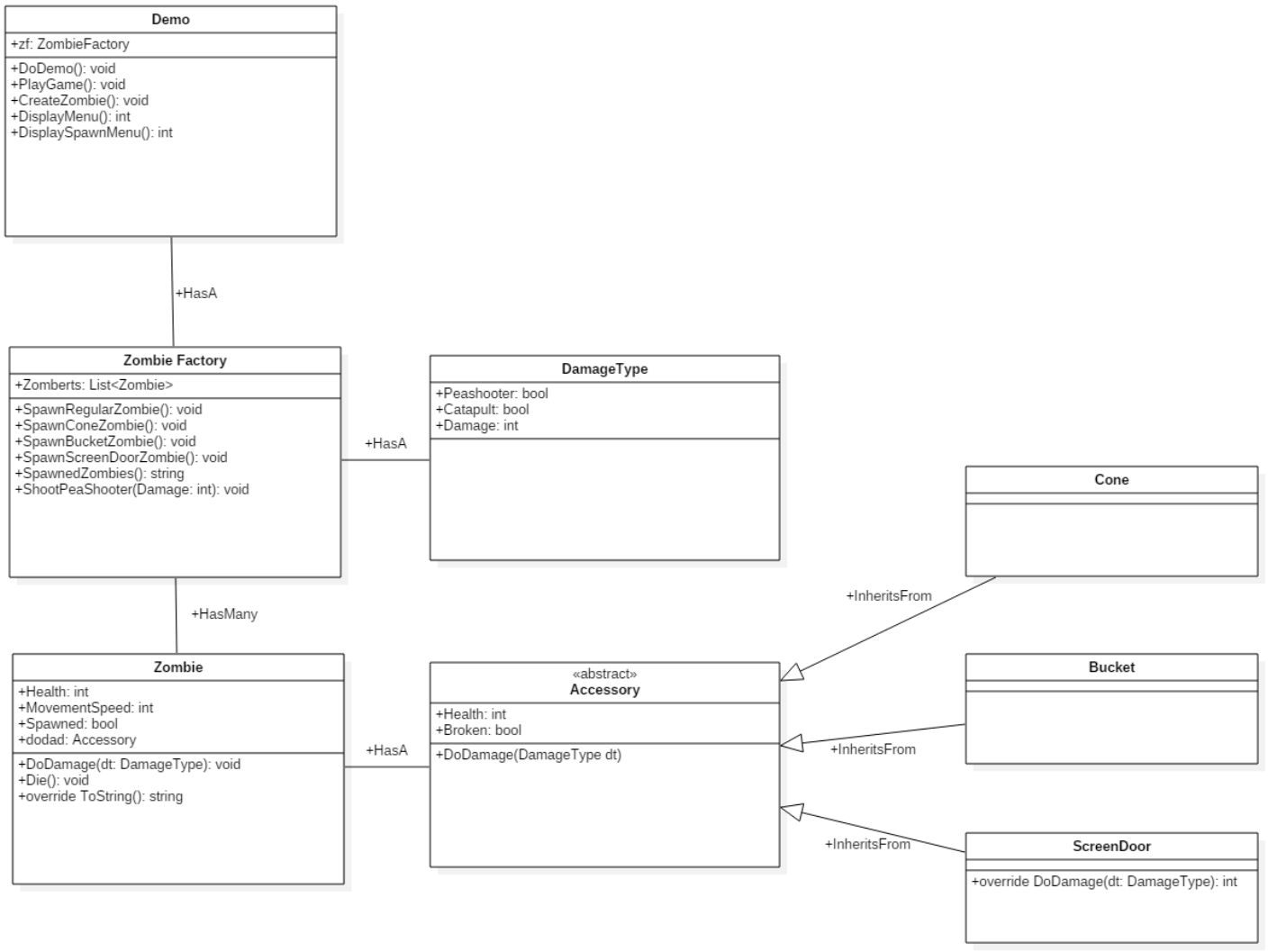
Justin Harper

WSUID 10696738

CS 487 Midterm

1. Create UML diagram for project



2. Write program to implement design for demo
    a. See Visual Studio Solution for my work

3. What would change if we reduce the damage of Peashooter to 10, instead of 25? Does this change impact your code and/or game logic at all? Write a short answer to this question, then accommodate this change into your program, allow the user to set a damage value after selecting "Demo game play". Also, specify any assumptions you made for this function.
    a. Currently I am just calling zf.ShootPeaShooter(25) on line 42 of my Demo.cs file. In order to change this value this hardcoded integer would need to be changed. I have commented out the old PlayGame() and implemented it with the changes. The new implementation begins on line 45 of my Demo.cs file.

4. Say we want to introduce a new plant: Watermelon. The way a Watermelon attacks is that it catapults a watermelon above and hit the Zombies from the top. Therefore, for Regular, Cone and Bucket Zombies, the Watermelon would function the same way as the Peashooter. However, for the Screen-Door Zombie, the Watermelon can skip the screen-door it's holding, and hit the Zombie directly.
   What does the new feature change about your program? Does the Composite pattern still work? DO NOT change your code for this question. However, write up a short explanation on how this new change would impact your design and implementation. What kind of modification would be necessary?
    a. I am using a DamageType object to determine if the damage is being caused by a peashooter. If the watermelon catapult attack is used, I would simply implement a ShootWatermelonCatapult function in my ZombieFactory.cs file. This function would be almost identical to the ShootPeaShooter function on line 58 of my ZombieFactory.cs file. I would simply call Zombert.DoDamage(new DamageType(false, true, Damage)); to do execute the catapult damage.

    b. The DamageType object can be extended to support many damage types which is why I chose to implement it. If we were to actually implement this game it would make determining the damage calculations simple.