Danny Nsouli

Software Engineering Zombie Simulator Design Document

Human Class

My Human class begins with a constructor that holds a human's standard attributes, which are its x position y position and direction, represented by a char variable. Next is the move method, which will move the human in the direction it is facing with a 10% chance of changing direction. I selected a random number 'r' between 0 and 9 using the helper class. I then use an if statement that moves the human once space in the direction it is facing if the number 'r' is larger than 0. If the number 'r' is 0 then another random number will be picked as 'd' and that will determine which direction it will randomly turn to with multiple if else statements underneath depending on what 'd' is randomly chosen as. The next method is checkSurroundings, which if a human sees a zombie 10 spaces or less away (not including the space adjacent to it) from the direction it is facing, it will move 2 spaces away in the opposite direction. What I have done is made a collection of similar if statements to consider all of the different cases for the zombie and human locations. In the first if statement, I check the direction of the human and then see if the 10 spaces in that direction (not including the space adjacent to it) has a zombie by finding the difference between all the zombies in the list, which I am cycling through for this method. Also when checking for zombies in the y direction, I had to check if zombie and human had the same x coordinates because otherwise all zombies above or below the human x coordinate would be targeted (and vice versa for checking for zombies in the x directions). Then if the condition was satisfied, I would change the human's direction to the opposite direction, and then check whether the space it would run away too would be occupied by a wall or exceeds the limits of the city grid space. If the conditions are satisfied the human will run away to that location.

Zombie Class

The Zombie Class is a subclass of the human class because it shares the same attributes and overall functions. A zombie will have an x and y coordinate on the grid and a direction. Its move method is the same as the human move method, however, the value for choosing the random number of 'r' is changed from 10 to 5 because zombies will move in the direction they are facing but with a 20% chance of turning/changing direction rather than a 10% chance like the humans. The checkSurroundings method has been changed from the human version. It works in the same way with very similar if statements, however, once a zombie sees a human within 10 spaces of it, it will not change direction but rather just simply move towards the human. The infect method will check if a human is anywhere around each zombie (meaning adjacent/one space difference in location). If this condition is satisfied then the human will be removed from the humans ArrayList and its coordinates and direction variables will be saved as temporary variables and placed into a new zombie, which will be added into the zombie ArrayList.

City Class

Firstly added to this class was the populate method, which took in an integer number of People variable as a parameter. For each person, the method would go through a for loop and calculate a random x coordinate, y coordinate, and direction (by randomly picking between 4 integers and choosing a character for the direction based off that and if statements). Then a while loop will check if there are walls in that position and if there are then it will pick another random set of x and y coordinates until that human can be placed in a proper location. After this is done, the loop ends and the human will be added to the ArrayList of humans. The populateZombie method works in the same way, however, there is no initial for loop since we only are trying to start with one zombie at a random location. There was a pre made method for adding random buildings into the program. There is also the draw method that will call individual character draw methods to visually represent the simulator. For the drawHumans method, I went through the ArrayList of humans and set the pen color to white, then drew a dot in the that color at each of their x and y location, taken from each index of the list. The same was done for zombies, however, zombies are colored red. DrawWalls was also a pre made method. The update method goes through each ArrayList for the characters in the simulator and allows them to continuously act. For zombies, it checks their surroundings, moves, then infects if humans are in the square adjacent to them. For humans, it only updates their checkSurroundings method and their move method. The resetSim method will reset the simulator by regenerating humans and buildings and adding 1 zombie. It firstly clears the zombie and human ArrayLists and the walls. Then it repopulates the zombies and humans with their populate methods and generates random buildings with the randomBuildings method.

ZombieSim Class

For the GUI requirements I made this class implement MouseListener, KeyListener, and ActionListener to manipulate their methods. In the keyTyped method, I made an if statement that would check if the keyCode of the key input from the keyboard was 0 meaning to check if the user clicked the space bar. This would then reset the similar by calling the resetSim method. MouseClicked will add a zombie to an empty location with the click of the mouse on the screen. Firstly, it checks if there are no walls in its away of where you are trying to place the zombie and then if there aren't it will create a new zombie with those coordinates and add on a random direction for it. The coordinates of the mouse click must be divided by the value DOT_SIZE because we need the added zombie to be of the same dot size as everything else for proportions sake (1 dot unit according to the other characters in the simulator).

Added Functionality: Hunter Class

Within the zombie apocalypse, few brave souls are ready to take back what is still the world of humanity or at least what's left of it. They break out of their group huddle and attack the dead! I created another subclass of human called Hunter. A Hunter will move the same way as a human. However, its checkSurroundings method is different. If it sees a zombie in the direction it is facing at least 10 spaces away from it, it will move in that direction towards the zombie.

However, like zombie, instead of having an infect method, it has a killZombie method. If a zombie is adjacent to the hunter, that zombie is removed from the zombie ArrayList. Zombies cannot infect the hunters because they are too skilled and are able to take the zombies down first. Everything else is the standard when putting hunters into the City, they have their own populate method, which will spawn 4 hunters at the start of the simulator. They have their own draw method as well, which marks them as green dots on the screen.