Garrett McLaughlin

Professor Wood

Software Engineering

11/19/16

Zombie Simulator Project

My project uses the two given classes, ZombieSim and City, as well as three of my own which are Person, Human, and Zombie. Person serves as the super class to both Human and Zombie. It contains the move() method as well as the collisionDetection() method. The Human and Zombie classes in turn have their individual tracking/evade settings. My move method works by enabling a number 0-3. Zero corresponds to moving left, one to right, and so on. The directions is either made direction-- or direction++ in the case of hitting a wall or edge of the window. Walls are detected by using the walls[][] array as a parameter for the method and testing to see if wall[xPos][yPos] (where xPos/yPos is the position of the dot) is true.

My collision detection is fairly simple. I have two ArrayLists, one for humans and one for zombies. On top of this I have a temporary array[] of type Zombie. I use a nested loop to look through both ArrayLists and checks if (tempZombie.isCollidingwith(temp Human). This is a method in my Person class that simply checks if their xPos/xPos correspond to each other, as well as if they are adjacent, so one space away either left, right, up, or down. Instead of simply using zombieList.add(new Zombie) and humanList.remove(human) I need to use the temporary array to store the zombies that were collided with and then remove them after the nested loop. This is because an ArrayList cannot have elements added or removed while it is being looped

through. For whatever reason I chose to put this in the draw() method in City instead of update. In the future for organization sake it might have been better to play the method inside update().

For the updated features of clicking to add a zombie and hitting space my solutions are in the ZombieSim class. I implemented KeyListener and MouseListener so I was able to complete these. I use the method KeyPressed for checking spacebar, and MouseClicked to see if the mouse was clicked. If the mouse was clicked a new zombie is created inside the MouseClicked class and added to the zombieList ArrayList (which had to be made static). To reset the simulation first I check if spacebar was hit, then I activate the erase() method inside the City class. In this order the erase() method proceeds. First it clears each ArrayList, then clears the walls[][] array, then rebuilds a fresh array with randomBuildings(), then does drawWalls() to actually draw the walls, then re-populates the city with populate().

My special feature is “dance party zombies”. Pressing the ‘d’ character will at any time activate or deactivate this feature. I will let you see for yourself its effects but I use the Color.getHSBColor feature to create the effect.

Error Notes: At this time by tracking/runaway methods for zombies/humans is not working but I left versions of the method deactivated to give an idea of what I was going for. I have also found that if I click too fast to add a new zombie, or hit the spacebar too quickly it will cause an error.