CMPT 276 – Introduction to Software Engineering

Zombie Dash Game

Phase One

by Group 6:

Jacky Li
Natalie Lo
Rebekah Wong
Huanyu (Eric) Zhou

Zombie Dash is a 2D arcade-style game arising in the setting of a daunting zombie apocalypse. In a world where an infectious outbreak has plagued all forms of life as you know it, you must stealthily weave your way out of the ruins of what you once called 'home' as one of the only survivors of the virulent eruption of zombies. Whether it be hastily gathering survival supplies or vigilantly avoiding the gaze of bloodthirsty zombies out for your flesh, Zombie Dash has it all. But wait—there's more! You've heard of a shelter housing survivors some ways away, and it seems like going there would be your best bet to stay alive. Collect all of the canned food along your trail and make it out to the exit to ensure your safety, and make sure that the traps scattered across the map—or even worse, the zombies themselves—don't get to you first.

The overall strategy for our implementation of Zombie Dash will follow an Agile software development lifecycle, as we are a smaller, tight-knit team with fewer developers and can thus be more flexible when discussing how to approach the requirements and adaptable with how we choose to implement our game. In addition, the Agile development process will also allow us to efficiently focus on individual activity segments, which is beneficial to our workflow since breaking down the project requirements and completing tasks one by one on a smaller scale would be more motivating than viewing the project as a larger, daunting task. We believe that the repeated tasks will ultimately help us identify any bugs earlier on by going through all of the stages for each individual activity and be able to fix them on the spot, rather than noticing them later once larger chunks of work have been finished.

The phases of our project will break down the game into smaller pieces, versions, or milestones where each will have a primary goal, and steps necessary to achieve completion of the overall project. We want to implement Alpha and Beta phases to ensure that the outcome is well-defined, with each phase entailing a reduced version of the game instead of an unfinished one. In development we expect to continue to monitor for any adjustments or refinements necessary and to improve or fix any software bugs. In our Alpha Phase and Phase One, we will primarily focus on requirement analysis where it begins with how the game software interacts with the primary actor through several use cases, what classes are required, and the attributes and methods inside those classes. Our system will have classes such as the game, environment, main character, enemy, and keyboard. The Environment class includes attributes of the game's maps such as the height and width of the game environment which will be represented by a 2D array, and methods necessary to set paths and game objects at specific locations on the grid. To implement the game map, each index of the array will be encoded with an ASCII value to indicate whether it is a player or enemy-walkable path, barrier, wall, punishment, or reward. The system also includes a Game class, which will be responsible for updating, rendering and maintaining all the game's objects in the system and reading responses from input devices. The Game class is required because since the game has multiple objects, it will need to loop through all the objects to update and render those objects to the screen. A method called render is responsible for computing an image from the contents of a scene. An example of a method is the keyPressed function that will take the player's input and apply it to the main character in the game by making the character move in the appropriate direction. The start, stop, and run methods inside the main class are also significant, because they will be in charge of how the game begins/renders and terminates. The team will then focus on the visual aspects of the game by designing its User Interface which will include textures for the character, enemy, backdrop, walls, rewards, punishments, and barriers, and add the finishing touches to create the apocalyptic world of Zombie Dash.