

## **Z-Town Design Document**

Development Team:

Cameron Jett: Content Designer  
Ryan Southard: Systems Designer  
Drew Switzer: Content Designer  
Ben Trivett: World Designer

## **Table of Contents**

Introduction / Overview.....	3
Game State Diagram.....	4
Development Environment.....	5-6
Graphical User Interface (GUI).....	7-8
Terrain.....	
9	
Artificial Intelligence.....	10
Game Mechanics.....	11-14
Game Content.....	15
Audio.....	16
Input Control Devices.....	17
Timebox 1.....	18
Timebox 2.....	19
Timebox 3.....	20
Timebox 4.....	21
Timebox 5.....	22-23
Timebox 6.....	24

## **Introduction**

Z-Town is a 3rd Person Zombie shooter inspired from games such as Left 4 Dead and movies such as I Am Legend. It is the zombie apocalypse and you play the role of a survivor. The game setting takes place in an urban environment. There is a day and night state for the game. During the day, the zombie go into hiding to escape the light, giving the player an opportunity to explore the city for supplies to help them survive the zombie apocalypse. The player can find weapons and defense items to help them fight the zombies. At night time, hordes of zombies wander the city and assault the player's home base to try and feed on the player's brains. At night time the player has to fight off the zombie hordes until the break of day. Each day, the zombie hordes get stronger, until the player either gives up fighting or is overwhelmed.

## **Overview of Design Document**

This document will discuss all the design aspects of the Z-Town video game in detail. First we will go over the Game State Diagram and outline the flow of the game and how the user can move from one state to another. Next we will briefly discuss the development environment of the game. Then we will outline the design of the GUI of the game, including the start menu and player heads-up display (HUD). Next we will discuss how game assets, such as models, scripts, etc. will be created and collected. Then we will discuss some of the game assets in greater detail on how they are designed. First we will talk about the design of our terrain. Next we will discuss the design of our Artificial Intelligence elements of the game such as enemy and spawn behaviors. Then we will talk about the designed game mechanics for our game, which covers subjects such as zombie statistics, player statistics, weapon statistics, and next we will discuss the design of the various game content in our game such as the defense items, zombies, and player character. Then we will talk about the various audio elements of the game including music and sound effects. Then we will discuss the input device we are designing to be compatible with the game and the various controls. Finally we will talk about the work required for each iteration of our design process to produce a complete game.

## **Game State Diagram**

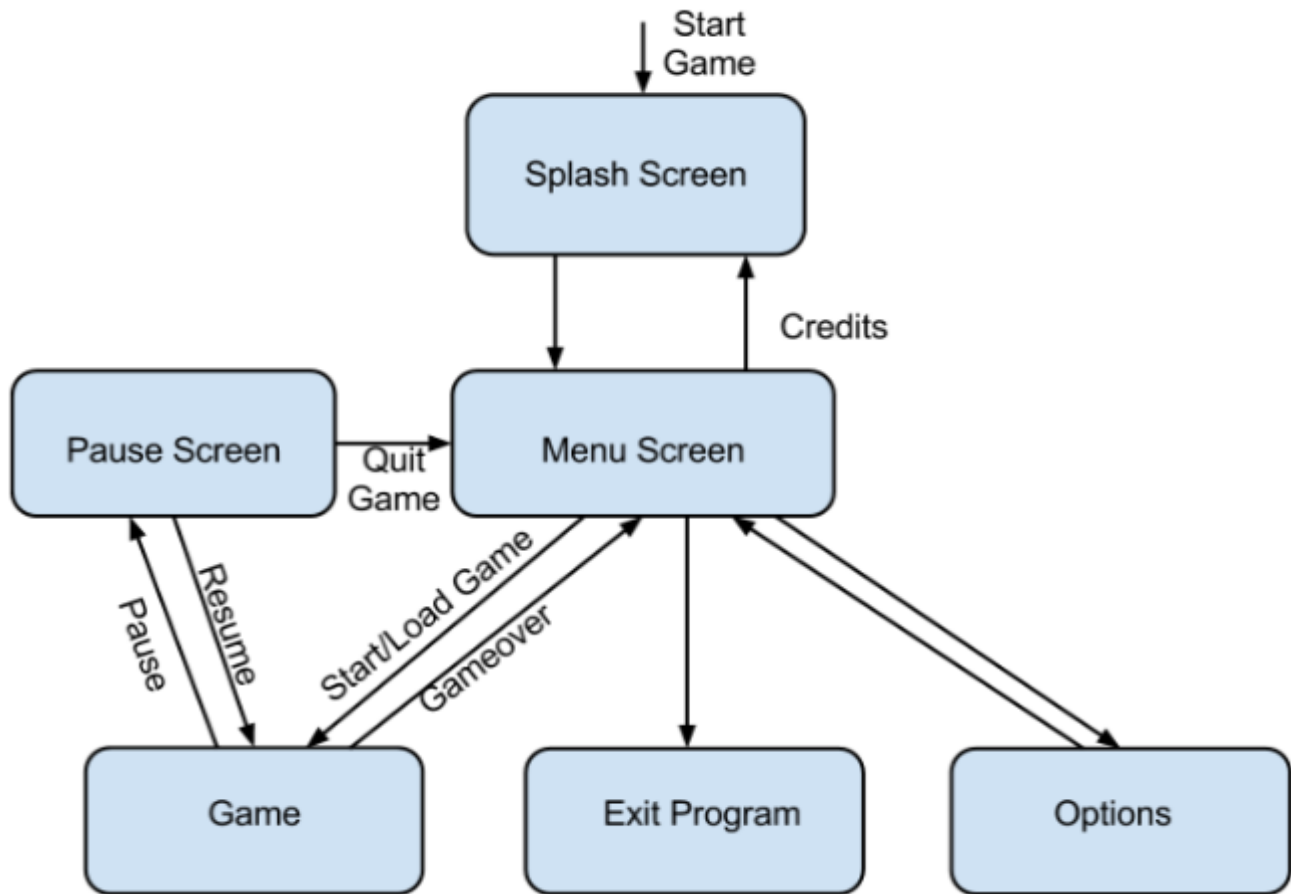


Figure 1.1 Game State Diagram

Figure 1.1 shows the Game State Diagram for our video game. This diagram demonstrates the flow of our game. The game is started through the Splash Screens or credit screens. This is just a few quick scenes displaying information about our design team. Then it moves into the Menu Screen. The Menu Screen is a scene with a GUI menu. From this Menu, we have several paths we can take. From the Menu we can go to the Options Screen, we can quit the game, or start or load a new game. The Options Screen is a GUI scene that allows the user to change some of the game mechanics. From the Options Screen the user can only return to the Menu Screen. The Exit Program state is not a scene in the game. It is merely the program shutting itself down and returning the user to their computer desktop. The Game state is where all the game play elements take place. From the Game state the user can pause the game and move to a Pause Screen state. In the Pause Screen state, the game is no longer progressing

and it remains paused until the user returns to the Game State by clicking a resume game button. The user also has the option to quit the game from the Pause Screen and return to the menu screen. Any game saving would also be done in the Pause Screen. The user may also reach the Menu Screen from the Game state if the player reaches a gameover, which in our game would be getting overwhelmed by the zombies.

## **Development Environment**

Z-Town was developed using the Unity game engine on Windows machines. Some of the computers used to develop the game were older computers with outdated graphics cards and processors. We designed the game so it would run smoothly on the older machine to help ensure that the game would run well on most computers. Scripting files were written in either C# or Javascript depending on the developer's preference, and we used the MonoDevelop editor provided by Unity to write the scripting files. The models we used for the game were either .fbx or .max file types because these are the only model types we found to be compatible with Unity. Textures for the game were created using Gimp 2.

## **Graphical User Interface (GUI)**

There were four different GUI scripts that had to be developed for the game. The four GUI's needed for the game are the menu screen, pause screen, options screen, and the heads-up display (HUD).

### **Menu Screen**

The menu screen is designed to use the GUI function that are built into Unity. Unity has built in GUI functions to make creating buttons, labels, themes, etc. easy. The Menu screen is a pretty simple GUI script with four buttons. The four buttons are the New Game, Options, Credits, and Quit Game buttons. Each button in the Menu screen use an OnClick listener so that it could perform the appropriate action when the button is clicked. When the Game, Options, and Credits buttons are clicked, the OnClick listener function will start a new scene by calling the Application.LoadLevel() function. This advances the user from the Menu Screen state to the appropriate next state (as seen in Figure 1.1) depending on which button the user clicks. When the user clicks on the Quit Game button, it simply calls Application.Quit() and ends the program.

### **Pause Screen**

The pause screen will be implemented by using a boolean value to determine whether or not the current state is the game state or the pause state. If it is the pause state, then it should set the time scale to zero so that the game is paused, and it should display some GUI buttons in a similar fashion to the menu screen so that the user can either resume the game or quit to the menu screen.

### **Options Screen**

The options screen will be designed in a similar fashion as the menu screen. We will use a script to display all the GUI buttons and their listeners. Then we will use another script to keep track of the game mechanics, such as the spawning rate, zombie health, player health, etc. Each of the options GUI buttons will allow the player to change the game mechanics so that when a new game starts, it uses the game mechanics the user has chosen in the options screen.

## **Heads-Up Display (HUD)**

The HUD is designed in a script using the OnGUI function. Using the built in GUI features in Unity, we will create a script that displays images and text to form or HUD. The images will be simple icons that describe the statistical information that the user sees displayed in the text. This will allow the user to quickly look at one of the text labels on the screen and decipher its meaning. A PlayerStatus.cs script is also needed to keep track of the player's statistics such as health, ammo, etc. so that the information can be properly displayed in the HUD. The HUD will be attached to the in-game camera object so the player always sees the HUD no matter what happens to them in the game state.



## Terrain

The majority of our games terrain will be an outdoor urban environment. The environment needs to be large and populated with buildings and streets. We decided that we would use a static environment, and luckily for our team, we were able to find a free model online that would suit our needs. The model we found online is very extensive and efficiently modeled, which does pose some problems.

One problem is that the city is almost too big. It would take the player a long time to explore the entire terrain. The game also has lower performance because the size of the city, so we will have to reduce the size of the city to the area that we want to include in our game.

Another problem is that some of the textures for the city have very low detail, and there are artifacts littered all over the city. We may be able to improve the textures of the city by including higher quality textures to the city terrain. We may also be able to remove some of the artifacts that we are seeing in the game.

One last problem is that the city is too clean for the zombie apocalypse. The city model looks very peaceful and serene, which is the complete opposite of what we want, so we will have to spend some time “destroying” the city to make it look like the zombie apocalypse. We don’t want to place random objects all over the terrain and make it look cluttered, but rather place meaningful obstacles that can make the gameplay more interesting and challenging for the player.

We also plan on having some indoor terrains. We want the player to be able to explore some of the buildings that they see on the outdoor urban terrain.

## **Artificial Intelligence**

Artificial intelligence is mostly used with the zombies in the game. The behavior and pathing of the zombies is controlled by the AI as well as how, when, and where the zombies are spawned. The behavior of each zombie is controlled by a pre-made script. These scripts have a predetermined movement speed for each zombie and well as a set of instructions as to how the zombie should move to the player (or do nothing). Most zombies will always try to go to the player. Spawning is done by placing zombies randomly between two radii from the player. This gives the illusion that the player is always being swarmed by zombies at every angle. During this placement, the AI also determines what type of zombies are placed (normal, fast, dumb, etc.). Currently, a set maximum number of zombies can be spawned at any time during the game. There are plans to make this number vary as the player progresses through the game, making it more difficult the longer the game goes on. There are also plans to include “boss” zombies every xth wave the player reaches. Since these zombies would be “smarter” and have unique behaviors, they would require a more detailed and laid out AI.

## **Game Mechanics**

This section highlights the design plans for game mechanics of the Player, Zombie, Weapons, Spawner, Tower Defense, and the Day Night Cycle. This is very important to our game because if the game mechanics are not solid, the gameplay might be too difficult or too easy depending on how the Player, Zombie, Weapons, etc. are implemented. For example, if the Zombie damage is too high, the game might become too challenging, but if the damage is too low, the game might become less fun. Finding the right balance will most likely require a lot of tweaking as we progress.

### **Player**

The player will be able to control the main character via keyboard and mouse. With the mouse control your character will follow the mouse cursor and left clicking will fire bullets, right clicking will place tower defense items.

### **Zombie**

During the night phase of the game, zombies will spawn around the player and throughout the city. The zombies will chase down and try to kill the player. Some basic pathfinding is used to control the route that the zombies take to chase down the player. Using a simple random placement algorithm, the zombies spawn around the player. This placement currently needs to be worked with, to find the right distance from the player to spawn the zombies without them being too close or too far. Also, the zombies will be able to attack the player if they get within a close enough range. This will cause the player's health decrease each hit they receive from a zombie. There will also be some variation in the types and behaviors of the zombies. Some will move faster than the others while some may do nothing at all.

### **Weapons**

Guns use raycasting to simulate bullets fired, each gun has a certain fire speed, accuracy (bullet spread), and damage per bullet. Weapons include the pistol, AK47, shotgun, minigun, rocket launcher, and laser pistol.

## **Spawners**

Spawners are used to spawn the zombies and tower defense / pickup items throughout the city. The zombies are spawned around the player, randomly placed between two radii from the player. This will give the illusion of the player being swarmed by zombies from all angles. This same algorithm is used to spawn the zombies around the home base, that way the player is somewhat forced to stay near the home base and defend it. During this placement of the zombies, the type of zombies that are spawned are also determined (normal zombie vs. fast zombie). Spawners are also used to control the placement of Tower Defense and Pickup items. The timing of these spawners will have to be adjusted so that the player has to explore the city to find new items, but we don't want it to be impossible to find items. The placement of these items will also be important. We will most likely place item spawns in static locations so it is a challenge to find items.

## **Tower Defense/Pickup Items**

Tower defense items include placeable walls that will stop zombies until destroyed, barrels that explode when shot by the player, decoys that distract the zombies, turrets that automatically fire upon zombies within range, and a nuke that destroys all zombies on screen. The tower defense items will have to be modeled in such a way that they don't give the player too much of an advantage against the zombies, but they must also be models so they are not useless against the zombies. Finding the right balance will be important to making the Tower Defense items an important and fun part of the gameplay. Pickup items include all tower defense items, health increasing items, and each of the guns that adds ammo to that gun. The statistic of pickup items will have to be managed in such a way that the player is not rewarded too much or too little for finding a pickup item. We want the ammo that players find to provide enough ammo so that the player does not feel like they are constantly running out of ammo, but we don't

want to provide too much ammo for the weapons that are the most powerful against the zombies.

### **Day/Night Cycle**

The game is split into two main phases, day and night. Each phase will have a fixed duration and transition into the other when that time is up. Zombies will spawn during night cycles and actively pursue the player, during day cycles the zombies will not spawn and ignore the player or wander around the map. We want the day time to be long enough so that the player has time to explore the city without getting bored, and we want the nights to be long enough so that when the zombies attacks, it is challenging, but not so long that the game becomes impossible.

## **Game Content**

Our game is a mix between a standard top down arcade shooter (such games like Smash TV or Super Stardust), mixed with horror and tower defense elements.

The arcade side of the game is represented by the movement, gun firing style, and overall game feel. Also the fact that the game has no ending, however there is a zombie kill counter which acts as a high score.

The tower defense side of the game is represented by having a home base that you have to defend, and being able to put traps around the base to help you do so.

## **Audio**

The music will be low and in the background, a low mumbling that gets slightly louder as time goes by and possibly becomes filled with static during times when there are many enemies surrounding you. It should make the player feel tense in those situations.

There will also be zombie moaning in the background, as if there are many zombies around you or very far away.

Of course each gun will have its own bullet or rocket sound effect, as well as there being exploding sound effects for the exploding barrel and nuke. There will be a sound on the wall, turret, and decoy tower defense items when they are being attacked by zombies to indicate to the player that they are taking damage.

There will be a sound effect when either the player, a survivor, or a zombie takes damage, and a death sound effect for each.

## **Input Control Devices**

Keyboard and mouse will be used to control the player.

Keyboard + Mouse:

WASD - Movement

Left Click - shoot / cancel tower defense item placement

Right Click - preview / place tower defense item

M - view map

Esc - quit

Up/Down Arrows - change weapon

Left/Right Arrows - change tower defense item



## Timebox 1

For each timebox section we created a table to show each task we planned for the timeboxes. We also have estimated the effort required for each section by using a number between 1 and 5, with 5 requiring the most effort and 1 requiring the least amount of effort. A 5 will usually indicate tasks that will require the entire 2 week iteration to work of, and a 1 indicates a task that requires at most 3 days to complete. As you may notice, all the timeboxes do not include all of our designs for the game. There was not enough time in the semester to implement all of our designs, so all the timeboxes include the designs that we had the time to implement.

<b>Task</b>	<b>Details</b>	<b>Est. Effort Required</b>
Create Menu GUI	Implement a Menu GUI for the game. The Menu GUI should include: <ul style="list-style-type: none"><li>● Splash Screens for team credits</li><li>● Menu Screen with New Game, Options, Credits, and Quit buttons</li><li>● A Menu background</li><li>● Menu Sound effects for clicking and ambience</li></ul>	4
Basic Terrain	<ul style="list-style-type: none"><li>● Begin working on the basic layout of the terrain</li><li>● Find building and scenery models to use for the terrain</li></ul>	5
Basic Player Controls	<ul style="list-style-type: none"><li>● Implement a simple player control system. The player should be able to move using WASD</li><li>● Implement a camera script that follows the player around as they move</li></ul>	3
Basic Zombie AI	<ul style="list-style-type: none"><li>● Create a script for a basic zombie AI that chases the player</li><li>● Create a zombie prefab that should have some basic collision detection so it does not walk through terrain objects</li></ul>	3

## Timebox 2

Task	Details	Est. Effort Required
Basic Tower Defense/ Pickup Items	Implement basic tower defense and pickup items. The items should be able to: <ul style="list-style-type: none"><li>● Be displayed in the scene</li><li>● Be destroyed when the player collides with the item</li><li>● Be added to the player's inventory</li></ul>	5
Zombie AI	Implement different Zombie AI's. The implemented zombie behaviors should be: <ul style="list-style-type: none"><li>● Idle</li><li>● Wander</li><li>● Chase</li></ul>	4
Player Control and Weapons	<ul style="list-style-type: none"><li>● Add controls to allow the player to rotate by rotating the player so it always looks at the mouse cursor</li><li>● Find player and weapon models</li></ul>	4
Terrain	<ul style="list-style-type: none"><li>● Finalize the outdoor urban terrain structure<ul style="list-style-type: none"><li>○ Finalized outdoor terrain does not have to be polished and detailed with scenery</li><li>○ Building model positions need to be finalized</li></ul></li><li>● Search for scenery and building models that can be used for our terrain</li></ul>	5

### Timebox 3

Task	Details	Est. Effort Required
Tower Defense Items Placement Implementation	<ul style="list-style-type: none"><li>• The player should be able to place Tower Defense items that they have available in their inventory</li><li>• Tower Defense items places should be removed from the player's inventory</li><li>• Complete a spawner for tower defense and pickup items</li></ul>	5
Zombie AI	<ul style="list-style-type: none"><li>• Work on zombie physics, they should collide with walls and other obstacles</li><li>• Work on zombie pathfinding so they don't get stuck on obstacles</li><li>• Create a basic zombie spawner</li></ul>	4
Player Control	<ul style="list-style-type: none"><li>• Implement some XBox 360 control</li><li>• Add pistol and AK47 and ability to fire bullets</li></ul>	4
Terrain	<ul style="list-style-type: none"><li>• Start adding destruction details to the outdoor terrain</li><li>• Create one indoor terrain</li><li>• Work on improving any textures on terrain that cause artifacts or bad looking visuals</li></ul>	3
Player HUD	<ul style="list-style-type: none"><li>• Create a working HUD to keep track of items in the player's inventory</li><li>• Make sure icons in HUD are readable and easily understood</li></ul>	3



## Timebox 4

Task	Details	Est. Effort Required
Tower Defense Items Implementation	Implement the following placed tower defense items: <ul style="list-style-type: none"><li>● Wall</li><li>● Turret</li><li>● Gas Can</li><li>● Decoy</li><li>● Exploding Barrels</li><li>● Nuke</li></ul> Items need to have health stats, do damage to zombies, etc.	5
Zombie AI	Modify the AI so that the zombie spawners spawn zombies at a more reasonable rate and doesn't spawn zombies in unreachable locations.  Modify the individual zombie AI so that zombies attempt to attack the player if the player is within their reach.	4
Player Control	Add more weapons, add randomness to the bullet spread, and update graphics of bullets.	4
Terrain	Add another interior level Add ability to enter/leave buildings Place destruction models	3
Zombie Statistics Script	Create a script that keeps track of zombie statistics such as <ul style="list-style-type: none"><li>● health</li><li>● damage it deals</li></ul>	3

### Timebox 5

<b>Task</b>	<b>Details</b>	<b>Est. Effort Required</b>
Player Model and Animations	<ul style="list-style-type: none"><li>● the finalized player model needs to be completed</li><li>● the player animations must also be finalized</li><li>● some player animations include: running, walking, idle, struck, and killed animations</li></ul>	5
Terrain	<ul style="list-style-type: none"><li>● Design indoor level procedural generator to generate a large amount of indoor levels</li><li>● Add in the gas station</li></ul>	5
Zombie Mechanics	<ul style="list-style-type: none"><li>● Tweak the zombie mechanics so that health, attack, speed, etc are modeled in a way that the game is fun</li><li>● Tweak the zombie spawners in such a way that the game is fun</li><li>● Modify the zombie spawners that are outdoors so that they only spawn zombies when it is night time</li></ul>	5
Audio	<ul style="list-style-type: none"><li>● Find sound effects that we want to use for the game</li><li>● Add is some music to the game</li><li>● Begin working on sound effects for guns, zombies, explosions, etc.</li></ul>	3
Tower Defense/ Pickup Items Mechanic	<ul style="list-style-type: none"><li>● Tweak the tower defense mechanics so that tower defense items are modeled in a reasonable way that make the game fun</li><li>● Tweak the amount of items a player gets when they find pickup items in such a way that the game is fun</li></ul>	4

### Timebox 6

<b>Task</b>	<b>Details</b>	<b>Est. Effort Required</b>
-------------	----------------	-----------------------------

Terrain	<ul style="list-style-type: none"> <li>● Add in indoor levels using the indoor level procedural generator</li> <li>● Add static spawns to the indoor terrains</li> <li>● Add tower defense items and other pick ups to the indoor levels</li> </ul>	5
Audio Ambience	<ul style="list-style-type: none"> <li>● Create and Audio ambience that changes depending on situations the player is in</li> <li>● Make ambience audio change based on how the environment changes</li> </ul>	4
Audio Sound Effects	<ul style="list-style-type: none"> <li>● Add audio sound effects to all objects in the game</li> <li>● Add audio sound effects to player to make player feel human</li> </ul>	4
Zombie Models and Animation	<ul style="list-style-type: none"> <li>● the finalized zombie models needs to be completed</li> <li>● the zombie animations must also be finalized</li> <li>● some zombie animations include: running, walking, idle, attack, and killed animations</li> </ul>	5
Game Mechanics	<ul style="list-style-type: none"> <li>● Continue tweaking all game mechanics so the game is challenging and fun</li> </ul>	3

