****

Programmer’s Manual

Hannah Roach

Dill Junye

Elijah Roberson  
CSC 478: Software Engineering Capstone  
11/1/2018

**CAPSTONE PROJECT**

**TABLE OF CONTENTS**

Page #

[1.0 Programmer’s Manual 3](#_Toc532139725)

[2.1 Requirements Documentation 3](#_Toc532139726)

[2.2 Design Documentation 9](#_Toc532139727)

[2.3 Source Code 15](#_Toc532139728)

[2.4 Test Plan and Test Results 20](#_Toc532139729)

[2.5 List of Known Bugs and Issues 22](#_Toc532139730)

# Programmer’s Manual

## 2.1 Requirements Documentation

1. **Introduction**
   1. **Scope of the Product**

This program should work with any platform, Windows, Unix, Macintosh, etc. It will function as a standalone application and will require no interaction among other software. Python will be used for this project, there will be 1 user player, no AI or computer player and the software will have a GUI.

* 1. **Definitions, acronyms, and abbreviations**

**Bullets:** A projectile for firing from a rifle, revolver, or other small firearm, typically made of metal, cylindrical and pointed, and sometimes containing an explosive. Bullet will only kill one zombie.

**Invasion:** An unwelcome intrusion into another's domain.

**Level:** An increment in difficulty when all of the zombies have been killed.

**Lives:** The number of times the shooter is allowed to die without the game ending.

**Mega Bullet:** bullet that will kill all zombies in its path

**Player:** A person taking part in a sport or game.

**Shooter:** The person shooting the bullets; i.e., the player.

**Zombie:** A person or reanimated corpse that has been turned into a creature capable of movement but not of rational thought, which feeds on human flesh.

* 1. **References**

pngimg.com. PNG images: Zombies. Retrieved from <http://pngimg.com/imgs/fantasy/zombie/>

python. Downloaded from <https://www.python.org/>

pygame, pygame documentation. <https://www.pygame.org/docs/ref/draw.html>

bullets.py Computer Science with Python and Pygame. <http://programarcadegames.com/python_examples/f.php?file=bullets.py>

opensource.com. Resize image using Python. <https://opensource.com/life/15/2/resize-images-python>

Matthes, E. (2016). Python Crash Course. San Francisco: no starch press.

The Invent with Python Blog. <https://inventwithpython.com/blog/2012/08/09/source-code-makeover-square-shooter-part-1/>

Beginning Game Programming for Teens with Python. <https://www.raywenderlich.com/2795-beginning-game-programming-for-teens-with-python>

1. **General Description**
   1. **Production perspective**

This software is being built for our semester group project. A very large portion of our grade will be determined by this project that we will work on as part of a team. Our team’s goal is to build a high-quality software application that has been thoroughly tested and documented. The goal is to end up with a product that is of high quality.

* 1. **Product function**

This software is a zombie-shooter game. There is 1 player who acts as the shooter and the objective of the shooter is to eliminate the zombies by shooting them. The shooter dies if a zombie touches him, or if the zombie reaches the bottom edge of the screen. The game is over after the shooter dies four times. When the shooter kills all of the zombies, the shooter moves on to the next level. There are 15 levels in the game. If the shooter wins each level, (s)he wins the whole game. The zombies get smaller with each new level and it is harder to kill all of the zombies. With each level, the shooter is given unlimited bullets, but only a set number of powerful bullets.

* 1. **User characteristics**

People who enjoy playing video games.

* 1. **General constraints**

This game cannot run over a network

The features do not run as smoothly on Windows computer.

The game can only run on a computer with an operating system.

We were not able to successfully create an executable file, so the end user will need to have python and pygame installed on their computer.

The user will need to know how to use a command prompt or terminal to run the game.

* 1. **Assumptions and dependencies**

The end user would need python install to run this, or any IDE that supports python.

The end user will need to know how to install pygame

The end user will need to know how to use a command prompt or terminal

1. **Specific Requirements**

* The game will consist of a shooter, bullets and zombies

|  |  |
| --- | --- |
| **Shooter** | |
| **Requirement ID** | **Description** |
| 6.0.0 | There should exist a shooter. |
| 6.0.3, 6.0.4, 2.0.1, 2.0.2, 2.0.5 | The shooter should be able to move vertically and horizontally. |
| 6.0.1 | The shooter will begin each game in the bottom center of the screen. |
| 6.0.5 | The default facing position for the shooter is facing up at 90 degrees. |
| 6.0.4, 2.0.5 | The shooter should be able to rotate 180 degrees and 45 degrees to the left and right of the 90 degrees default position |
| 5.0.0. | The shooter should have four total lives. |
| 2.1.4, 2.1.6, 2.1.7 | The shooter dies when a zombie hits the bottom of the screen. |
| 2.1.4, 2.1.7 | The shooter dies when a zombie collides with the shooter. |
| 2.0.3, 2.0.4, 2.0.2, 2.0.1 | The shooter should be able to shoot unlimited bullets. |
| 2.0.3 | The shooter should be able to shoot a fixed number of mega bullets |

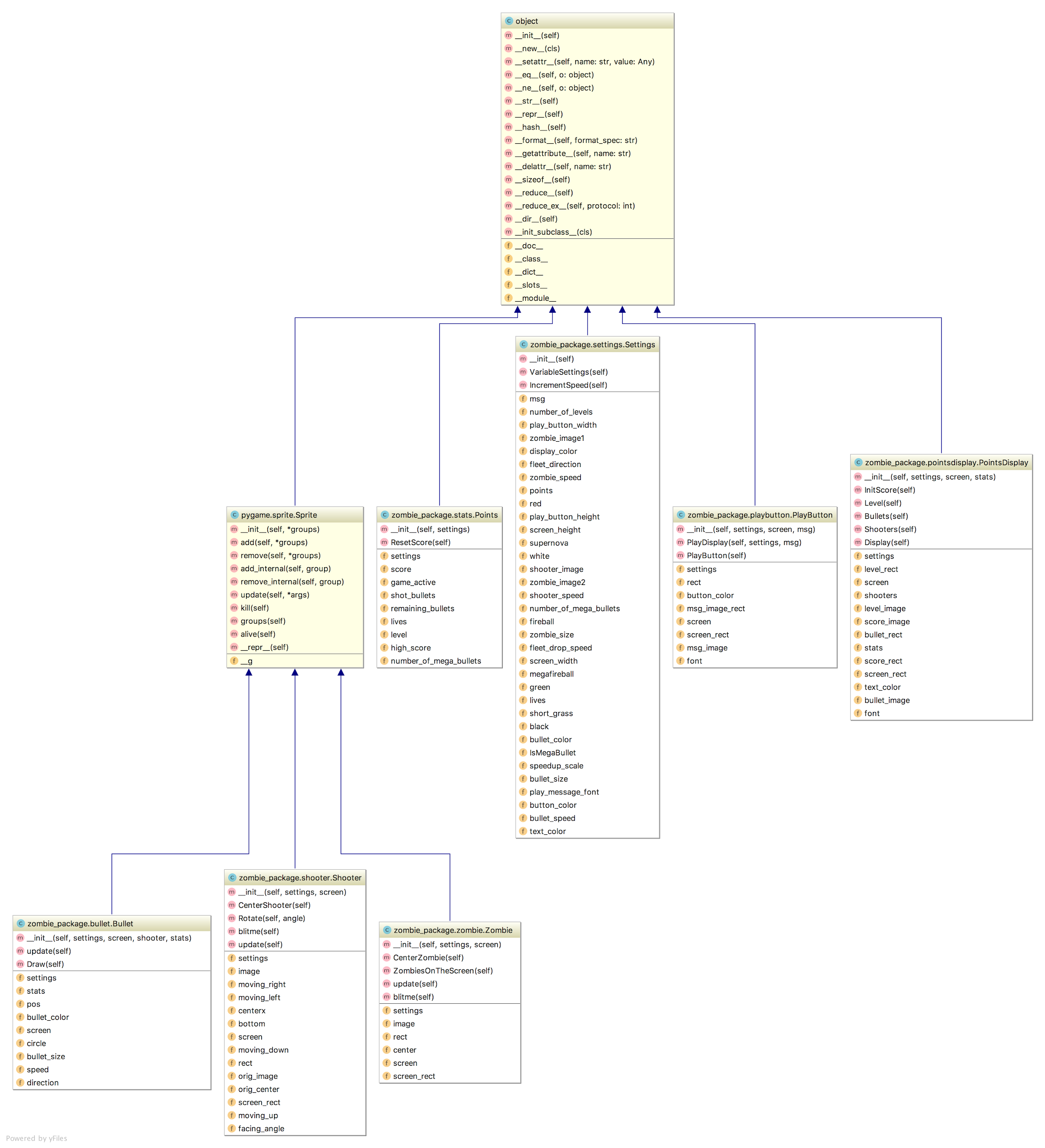
|  |  |
| --- | --- |
| **Bullets** | |
| **Requirement ID** | **Description** |
| 1.1.0 | There should exist bullets |
| 1.0.1, 1.0.2, 2.1.0 | The bullets should be able to move in any direction the shooter aims |
| 2.0.3 | There should be an unlimited number of regular bullets |
| 5.0.1 | There should be a limited number of mega bullets 5.0.1 |
| 7.0.1, 7.0.0 | There should be a count of mega bullets and how many the shooter has used displayed on the screen |
| 2.0.3 | If the shooter runs out of mega bullets, the shooter does not get new mega bullets until the next level, even if the shooter dies |
| 2.1.1, 2.1.2 | The shooter can earn additional mega bullets as (s)he progresses through the levels of the game |

|  |  |
| --- | --- |
| **Zombies** | |
| **Requirement ID** | **Description** |
| 8.0.0 | There should exist zombies |
| 8.0.3 | The zombies should move down from top to bottom of the screen |
| 8.0.3 | The zombies should move in a random zigzagging pattern like zombies |
| 2.1.7 | There should be a collection of zombies that get incrementally smaller with each level |
| 2.1.7 | With each level, there should be incrementally more zombies to kill |
| 2.1.0 | The zombie dies when it is hit with a bullet |
| 2.1.6 | The zombies should remain on and not wander off of the screen |
| 2.1.5, 8.0.2 | If a zombie hits the side of the screen, it will change directions |
| 2.0.9 | When a zombie is killed, the score is incremented by one |
| 8.0.1 | With each level, the zombies will be centered toward the top |

|  |  |
| --- | --- |
| **Functionality and Display** | |
| **Requirement ID** | **Description** |
| 3.0.0, 3.0.1 | The game with have a “Play” button |
| 3.0.2, 2.1.9, 9.0.0 | The user should press the play button to start the game |
| 2.1.8 | When the game is over, the game should be reset |
| 5.0.2 | There should be 15 levels of the game |
| 2.1.8, 2.1.3 | When the shooter wins level 15, the game is over and the shooter wins |
| 4.0.0, 4.0.5, 7.0.1 | The levels and points should be displayed on the screen |
| 2.0.0, 2.0.9, 4.0.1, 4.0.5, 7.0.1 | The score will be displayed and will increment by one with each zombie hit |
| 2.0.0, 2.0.9, 4.0.2, 4.0.5, 7.0.1 | The level will be displayed on the screen and will increment by one with each new level attained |
| 2.0.0, 2.0.9, 4.0.3, 4.0.5, 7.0.1 | The number of Mega Bullets will be displayed on the screen and will decrement by one with each Mega Bullet fired |
| 2.0.0, 2.0.9, 4.0.4, 4.0.5, 7.0.1 | The number of lives remaining will be displayed as shooter images on the screen and will decrement by one with each life lost |

## 2.2 Design Documentation

1. **Architecture diagrams**



1. **Pseudocode**

|  |  |
| --- | --- |
| **Shooter** | |
| **Requirement ID** | **Pseudocode** |
| **6.0.4** | **The shooter should be able to move vertically and horizontally.** |
| if (key == right arrow)  move the shooter right  if (key == left arrow)  move shooter left  if (key == up arrow)  move shooter up  if (key == down arrow)  move shooter down | |
| **6.0.1** | **The shooter will begin each game in the bottom center of the screen.** |
| set (center of shooter image) == center of screen  set (bottom of shooter image) == bottom of screen | |
| **6.0.5** | **The default facing position for the shooter is facing up at 90 degrees.** |
| set default angle = 0.0 | |
| **6.0.4, 2.0.5** | **The shooter should be able to rotate 180 degrees and 45 degrees to the left and right of the 90 degrees default position** |
| if key is down  if (key is x)  rotate shooter 45  if (key is c)  rotate shooter 315  if (key is z)  rotate shooter 90  if (key is v)  rotate shooter 270  if key is up  if (key was c)  rotate 315  if (key was z)  rotate 90  if (key was v)  rotate 270  if (key was x) rotate 45 | |
| **5.0.0.** | **The shooter should have four total lives.** |
| settings shooter lives = 3 | |
| **2.1.4, 2.1.6, 2.1.7** | **The shooter dies when a zombie hits the bottom of the screen.** |
| if (bottom of zombie image > bottom of screen)  remove zombie from screen | |
| **2.1.4, 2.1.7** | **The shooter dies when a zombie collides with the shooter.** |
| if (collides with zombie)  decrement number of lives | |

|  |  |
| --- | --- |
| **Bullets** | |
| **Requirement ID** | **Pseudocode** |
| **1.0.1, 1.0.2, 2.1.0** | **The bullets should move in any direction the shooter aims** |
| Set direction equal to the direction of the shooter  As shooter direction updates, update direction of shooter | |
| **5.0.1** | **There should be a limited number of mega bullets 5.0.1** |
| if (mega bullet is fired)  decrement mega bullet by one | |
| **7.0.1, 7.0.0** | **There should be a count of mega bullets and how many the shooter has used displayed on the screen** |
| if (mega bullet is fired)  decrement mega bullet image by one | |
| **2.1.1, 2.1.2** | **The shooter can earn additional mega bullets as (s)he progresses through the levels of the game** |
| If level increase by x amount  Increment bullet by one | |

|  |  |
| --- | --- |
| **Zombies** | |
| **Requirement ID** | **Pseudocode** |
| **8.0.3** | **The zombies should move in a random zigzagging pattern like zombies** |
| if (random number is less than 70 and zombie is on the screen)  move the zombie down the screen by random number/y  if (random number is less than 50 and zombie is on the screen)  move zombie to the right by random number /x  if (random number is greater than 50 and zombie is on the screen)  move zombie to the left by random number/x  where x and y are numbers | |
| **2.1.7** | **There should be a collection of zombies that get incrementally smaller with each level** |
| If level increments  Decrease zombie size | |
| **2.1.7** | **With each level, there should be incrementally more zombies** |
| Find the new size of zombie  Divide by the space available find number of rows possible  Populate the space with zombies | |
| **2.1.0** | **The zombie dies when it is hit with a bullet** |
| if (bullet collides with zombie)  remove bullet from screen  remove zombie from screen  increment score | |
| **2.1.5, 8.0.2** | **If a zombie hits the side of the screen, it will change directions** |
| Zombie hits side of screen  Zombie changes direction | |
| **2.0.9** | **When a zombie is killed, the score is incremented by one** |
| If zombie hit side of screen  Turn zombie around | |

|  |  |
| --- | --- |
| **Functionality and Display** | |
| **Requirement ID** | **Pseudocode** |
| **3.0.2, 2.1.9, 9.0.0** | **The user should press the play button to start the game** |
| Push play  Game starts | |
| **2.1.8** | **When the game is over, the game should be reset** |
| Game ends  Push play | |
| **5.0.2** | **There should be 15 levels of the game** |
| Settings levels = 15 | |
| **2.1.8, 2.1.3** | **When the shooter wins level 15, the game ends and the shooter wins** |
| Shooter wins level 15  Push play | |

## 2.3 Source Code

**Start Game**

**def** StartGame():  
 *# Requirement ID: 9.0.0  
 """This method initializes the game settings. It also starts and ends the game."""*

**Bullets**

**class** Bullet(pygame.sprite.Sprite):  
 *"""This is the bullet class."""* **def** \_\_init\_\_(self, settings, shooter):  
 *# Requirement ID: 1.0.0*

**def** update(self):  
 *# Requirement ID: 1.0.1  
 """This method updates the bullets location on the screen."""*

**def** Draw(self):  
 *# Requirement ID: 1.0.2  
 """This method draws the bullet on the screen"""*

**Shooter**

**class** Shooter(Sprite):  
  
 **def** \_\_init\_\_(self, settings):  
 *"""This class controls the shooter"""  
 # Requirement ID: 6.0.0* super(Shooter, self).\_\_init\_\_()

**def** CenterShooter(self):  
 *# Requirement ID: 6.0.1  
 """This method recenters the shooter. With each new level, the shooter is recentered. """*

**def** Rotate(self, angle):  
 *# Requirement ID: 6.0.2*

**def** blitme(self):  
 *# Requirement ID: 6.0.3  
 """This method refreshes the image """*

**def** update(self):  
 *# Requirement ID: 6.0.4  
 """Update the shooter location. """*

**Zombies**

**class** Zombie(Sprite):  
 *"""This class creates a template for the zombies"""* **def** \_\_init\_\_(self, settings):  
 *# Requirement ID: 8.0.0  
 """This method initializes the settings for the zombies. """* super(Zombie, self).\_\_init\_\_()

**def** CenterZombie(self):  
 *# Requirement ID: 8.0.1  
 """Center the zombies. With each new level the zombies are recentered. """*

**def** ZombiesOnTheScreen(self):  
 *# Requirement ID: 8.0.2  
 """Which direction the zombies are moving. If it hits the end of the screen, change directions"""*

**def** update(self):  
 *# Requirement ID: 8.0.3  
 """Update the zombies on the screen. Create a random walk."""*

**def** blitme(self):  
 *# Requirement ID: 8.0.4  
 """Refresh the screen."""*

**Stats**

**def** \_\_init\_\_(self, settings):  
 *"""This method initializes the score. """  
 # Requirement ID: 7.0.0*

**def** ResetScore(self):  
 *# Requirement ID: 7.0.1  
 """This method resets the score. """*

**Settings**

*# Requirement ID: 5.0.0  
# The shooter should have four total lives*self.lives = 3

*# Requirement ID: 5.0.2  
#number of levels in the game*self.number\_of\_levels = 16

**def** SetSpeed(self):  
 *# Requirement ID: 5.0.3  
 """Sets the initial speed settings. """*

**def** ResizeZombies(self):  
 *# Requirement ID: 5.0.4  
 """This method resizes the zombies and is called with each new level. """*

**Points Display**

**class** PointsDisplay():  
 *"""Displays points on the screen"""  
 # Requirement ID: 4.0.0*

**def** InitScore(self):  
 *# Requirement ID: 4.0.1  
 """Displays the current score"""*

**def** Level(self):  
 *# Requirement ID: 4.0.2  
 """Displays the current level"""*

**def** Bullets(self):  
 *# Requirement ID: 4.0.3  
 """Displays the number of bullets remaining """*

**def** Shooters(self):  
 *# Requirement ID: 4.0.4  
 """Displays the shooter lives. Display one image per life remaining."""*

**def** Display(self):  
 *# Requirement ID: 4.0.5  
 """Blits image"""*

**Play Button**

**class** PlayButton():  
 *"""Class for the play button"""* **def** \_\_init\_\_(self, settings):  
 *# Requirement ID: 3.0.0  
 """This method initializes the play settings. """*

**def** PlayDisplay(self, settings, msg):  
 *# Requirement ID: 3.0.1  
 """Defines the font, color and box for the play message"""*

**def** PlayButton(self):  
 *# Requirement ID: 3.0.2  
 """Play button"""*

**Methods**

**def** UpdateScreen(background, bullets, play, display, settings, stats, shooter, zombies):  
 *# Requirement ID: 2.0.0  
 """Refreshes items on the screen."""*

**def** EventCheck(bullets, play, display, settings, stats, shooter, zombies):  
 *# Requirement ID: 2.0.1  
 """Check for Key events. This method checks if a key was pressed on the keyboard  
 of if a mouse button was clicked."""*

**def** KeyDownCheck(bullets, event, settings, shooter, stats):  
 *# Requirement ID: 2.0.2  
 """Check for Key Down Event. This method checks if a key was pressed on the keyboard and if so, moves the  
 shooter, rotates the shooter, or shoots a bullet."""*

**def** ShootMegaBullets(stats, settings, shooter, bullets):  
 *# Requirement ID: 2.0.3  
 """Shoot the mega bullets."""*

**def** ShootBullet(settings, shooter, bullets):  
 *# Requirement ID: 2.0.4  
 """Shoot ordinary bullets."""*

**def** KeyUpCheck(event, shooter):  
 *# Requirement ID: 2.0.5  
 """Check key up events. This method checks if a key was lifted. """*

**def** UpdateObjects(bullets, display, settings, stats, shooter, zombies):  
 *# Requirement ID: 2.0.6  
 "This method initializes the bullet settings, calls the UpdateBullets and UpdateZombies methods."*

**def** UpdateZombies(bullets, display, settings, stats, shooter, zombies):  
 *# Requirement ID: 2.0.7  
 """This method updates the zombies on the screen, checks if a zombie has collided with a shooter, and  
 ensures that the zombies do not go off the screen.  
 """*

**def** RemoveBullets(bullets):  
 *# Requirement ID: 2.0.8  
 "This method removes bullets that have gone off the screen."*

**def** IncrementScore(collisions, stats, display=**None**):  
 *# Requirement ID: 2.0.9  
 """If a zombie is killed, increment the score by 1."""*

**def** UpdateBullets(bullets, display, settings, stats, shooter, zombies):  
 *# Requirement ID: 2.1.0  
 """This method updates bullets on the screen and checks if they have hit a zombie. If the bullets have  
 hit a zombie, increment the score. If all of the zombies have been killed, reset the board and increment the  
 level. """*

**def** InitializeBulletSettings(stats, settings):  
 *# Requirement ID: 2.1.1  
 """Initialize bullet settings. Each level will give the shooter more mega bullets.  
 be given a special mega bullet that will be a random size larger than the first mega bullet."""*

**def** MegaBulletSettings(settings, level, color):  
 *# Requirement ID: 2.1.2  
 """This method sets the size and color of the mega bullet"""*

**def** CheckForWin(stats, settings, button, zombies, bullets, shooter):  
 *# Requirement ID: 2.1.3  
 """This method checks if the level is a winning level and if so prints a win message."""*

**def** ShooterDown(bullets, display, settings, stats, shooter, zombies):  
 *# Requirement ID: 2.1.4  
 """Checks if a shooter was hit by a zombie and if so decrements the lives."""*

**def** ZombiesOnScreen(settings, zombies):  
 *# Requirement ID: 2.1.5  
 """Zombies change direction when hit the edge of the screen"""*

**def** ZombiesOffTheScreen(bullets, display, settings, stats, shooter, zombies):  
 *# Requirement ID: 2.1.6  
 """This method checks if the zombie hits the bottom of the screen. If the zombie hits the bottom of the  
 screen the shooter dies. """*

**def** ZombieCollection(settings, shooter, zombies):  
 *# Requirement ID: 2.1.7  
 """Zombie Collection. This method determines how many zombies to put on the screen.  
 With each level the zombies get smaller and more zombies fit on the screen. """*

**def** ResetGame(zombies, bullets, settings, shooter, stats):  
 *# Requirement ID: 2.1.8  
 """Reset items on the board. This method is called for each new level. """*

**def** Play(bullets, display, mouse\_x, mouse\_y, play, settings, stats, shooter, zombies):  
 *# Requirement ID: 2.1.9  
 """What to do when play button is pressed."""*

## 2.4 Test Plan and Test Results

1. **White Box Testing**

To run the white box tests, please do the following:

1. Open a terminal or command prompt window
2. Make sure that Python is properly installed:

**python –version**

1. If Python is installed, run:

**pip install nose**

1. cd into the directory of ZombieInvasion/tests and run:

**nosetests**

\*For reference using “nosetests”, please see <https://nose.readthedocs.io/en/latest/>

1. **Black Box Testing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case # | Requirement Tested | Rationale | Input(s) | Expected Output | Passed? |
| 1 | 3.0.0, 3.0.1 | Check if the game starts/start button works | User click the play button when prompted to start the game | Game starts when play button is clicked | Yes |
| 2 | 4.0.0 | Check if the scoreboard display properly | Scoreboard is displayed at the start of the game | The scoreboard increments or function when the user gets a kill | Yes |
| 3 | 2.1.4 | Test if the game ends when the user runs out of life lives | Life points are set to three 4 at the start of the game | If the user life points falls to/below zero, game ends | Yes |
| 4 | 2.1.0, 2.0.8 | Check if the bullets disappears of the screen | User shots are not staying on the screen when he shoots | Bullet disappears when it hits a zombie or miss the target | Yes |
| 5 | 2.1.6 | Check if the zombies disappears/dies when they are shot | Zombies are not alive when a bullet touches them | Zombies dies when the user bullets hits them | Yes |

## 2.5 List of Known Bugs and Issues

* If the user presses any of the keys that rotate the shooter, before the user presses the play button, the shooter will get stuck at top left corner of the screen
* If the shooter loses the game and then clicks “Play” again, the zombies do not reset to their original size, but will continue where they left off at the smaller size.
* If the shooter runs out of mega bullets and dies, the mega bullets are not refilled for the next life. This makes winning a level at the higher levels difficult if the shooter has lost a life.
* We were not able to get an executable file to work properly. This means that the user will have to have python and pygame installed on their computer. The user will then need to run the game through the command line or through and IDE.