

# Toasted Pygame Documentation

Saad Al-Mridha A013319129

Ramith Rajan A01316540

## File 1: game.py:

This file contains the main logic for the game. It uses the Pygame library to create a window, load images, and handle user input. The game has a player character that can move left and right using the arrow keys or the A and D keys on the keyboard. The player's goal is to avoid colliding with enemies that bounce around the screen. The player has three lives, and the game ends when the player loses all of their lives. The Game class is responsible for managing the game's state, including creating the player and enemies, handling collisions between objects, and updating the score and health. The game loop is also implemented in this class, which runs until the game is over.

## File 2: enemy.py:

This file defines the Enemy class, which represents an enemy object in the game. Each enemy has an image, position, and velocity, and can bounce off other enemies and the walls of the game window. The Enemy class contains methods for updating the enemy's position, handling collisions with other enemies, and drawing the enemy on the screen.

## File 3: player.py:

This file defines the Player class, which represents the player character in the game. The player can move left and right using the arrow keys or the A and D keys on the keyboard. The Player class contains methods for updating the player's position, handling collisions with other objects, and drawing the player on the screen. The player also has a health attribute that starts at 3 and decreases each time the player collides with an enemy. If the player's health reaches 0, the game is over.

## **SCREENS:**

### **File 4: menu.py:**

The menu.py file contains a MenuScreen class that represents the menu screen of a game. The class inherits from the pygame.Surface class, allowing it to be drawn onto the game window. In the init method, the screen attribute is set to the game window passed as an argument, and the next screen attribute is initialized to None. The menu method creates a pygame.Rect object to represent the play button, and enters a loop that handles events until the play button is clicked or the user closes the window. Within the loop, the screen is filled with a purple color, and the play button is drawn using a brown color for the button and white for the text. The image of Toastman is loaded and centered on the screen, and the text "Toasted" is also centered and rendered using a white color. Finally, pygame.display.flip() is called to update the game window.

### **File 5: game\_over.py:**

The game\_over.py contains a class GameOverScreen which inherits from pygame.Surface. The \_\_init\_\_ method initializes the class and stores the passed window and score variables. The game\_over method creates a game over screen where the final score is displayed along with a button to play again. The screen is continuously displayed until the player clicks on the play again button or quits the game. The pygame.event module is used to handle events like mouse clicks and the pygame.Rect class is used to define and handle the button. The screen is filled with the color red and the "Game Over" and score text is rendered onto the screen using different fonts. The play again button is also rendered onto the screen and it changes color when hovered over with the mouse. Finally, pygame.display.flip() is called to update the screen with all the changes made.