Team Members:
Olivia Spartz
Reilly Ferraro

Introduction:
We created a game using Pygame. An apple moves across the screen and the user has to click it as many times as possible before time runs out.

Project Structure:
Main file titled “project”
- main.py

Subfolder titled “assets
- apple.png

Key Functionalities:
Game Initialization:
- pygame. init initializes pygame
- pygame.display.set_mode creates a game window
- pygame.display.set_caption sets the window title
- pygame.image.load(os.path.join(‘assets’, ‘apple.png’)) loads apple image
- Variables are sectioned off and defined

Game Loop:
- while run: ensures the game runs until the user quits
- clock.tick(FPS) controls game frame rate

Timer:
- A countdown timer is used with pygame.time.set_timer(pygame.USEREVENT, 1000)

Scoring:
- When the apple is clicked on, the counter increases by one and displays during the game as well as the game over screen

Apple movement:
- Apple moves and bounces off window edges by reversing its speed when it hits a boundary

Event Handling:
- Mouse interaction with play again and quit buttons with pygame.MOUSEBUTTONDOWN

Rectangle Collision Points:
- Using rect.collidepoint(mouse_pos) we created interactive quit and play again buttons

Game Over Screen:
- Game over screen appears when the timer runs out and displays the score as well as buttons for restarting or quitting

Rendering:
- The timer, apple, and score appear on the game window during play and is updated using pygame.display.flip()
- Displayed on game screen using window.blit
- The font is rendered using pygame.font.SysFont

Reset Function:

- If the user clicks the “Play Again” button at the end of the game the game variables reset and the game begins again

Exit/Quit Functions:

- Players can quit using the red circle at the top left of the window
- Players can quit using the escape key because of pygame.KEYDOWN and pygame.K_ESCAPE
- Players can quit at the end of the game using the red quit button

Known Problems:

- The way we calculated the space the apple took up assumed the apple was rectangular which is not true. A player could click slightly off the apple and still receive a point.

Potential future extensions:

- Possibly speeding up the apple movement as the score increases
- Allowing the user to select a difficulty level, that could affect the speed of the Apple and time
- Blinking apple image
- Game over if you fail to select the apple for a certain amount of time

Team Member Contributions:

Olivia Spartz:

- Known problems section of the report
- Potential future extensions section of the report
- Introduction section of the report
- Sources section of the report
- Did functions regarding the Apple portion of the code
- Voice over demo
- Added explanation next to code

Reilly Ferraro:

- Potential future extensions section of the report
- Key functionalities section of the report
- Sources section of the report
- The project structure section of the report
- Did initializing functions regarding the game window
- Set up variables regarding FPS, colors, sizes, font
- Did functions regarding game over screen
- Functions involving timer
- Added explanation next to code

References:

Countdown timer in Pygame - Stack Overflow - parts of the timer

https://www.youtube.com/watch?v=jO6qQDNa2UY - YouTube tutorial from modules - general usage

Pygame Front Page — pygame v2.6.0 documentation - general

pygame.time - also parts of timer

pygame module for loading and rendering fonts - fonts - specifically font.Sys.Font

PYTHON Trouble using escape key to exit - pygame - esc key to quit function

pygame.display - pygame.display.flip to update the window

The movement of the apple came from the ball game lesson in class

pygame.mouse — pygame v2.6.0 documentation - using the cursor to select buttons and apple

pygame.Rect — pygame v2.6.0 documentation - all of the rectangle collide-point functions (buttons)