PROJECT NAME: SNAKE GAME

By-Kunjesh and Shivesh

We have used some of the following modules, functions and widgetswhich are different from what we use regularly in python language.

The project is based on pygame module.

1 .init() - pygame.init() initialize all imported pygame modules and is a convenient way to get everything started.

2 .display.set_mode() - Setting the display mode in pygame creates a visible image surface on the monitor, or in simpler ways it is used to create the window which will open when you run the game

3 display.update() - updates the entire Surface, only if no argument is passed.

It is like after you write code or make any changes then you should use this module to govern those changes.

4 .time.Clock() - Creates a new Clock object that can be used to track an amount of time. The clock also provides several functions to help control a game's frame rate.(frame rate- it is the frequency at which consecutive images called frames appear on a display)

5 .blit():

The screen object represents your game screen. It is a thin wrapper around a **Pygame** surface that allows you to easily draw images to the screen ("**blit**" them). ... **blit**() accepts either a Surface or a string as its image parameter. If image is a str then the named image will be loaded from the images/ directory.

6 mixer.music.load():

This will **load** a **music** filename/file object and prepare it for playback. If a **music** stream is already playing it will be stopped. This does not start the **music** playing.

7 python.event.get():

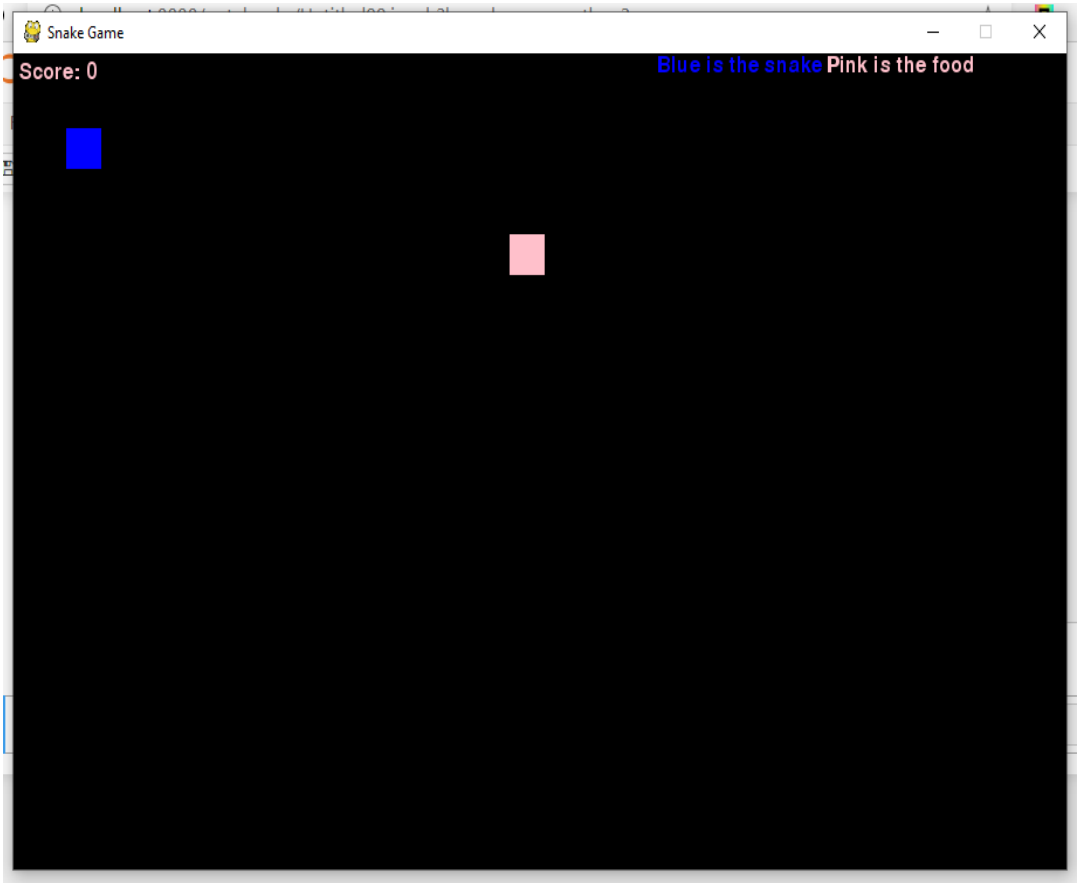
Since the game state is usually updated in response to events (such as mouse clicks or keyboard presses) or the passage of time, the game loop is constantly checking and re-checking many times a second for any new events that have happened. Inside the main loop is code that looks at which events have been created (with Pygame, this is done by calling the pygame.event.get() function). The main loop also has code that updates the game state based on which events have been created. This is usually called **event handling**.

8 abs():

The **abs()** function is used to return the **absolute** value of a number. ... The argument can be an integer, a floating point number or a complex number. If the argument is an integer or floating point number, **abs()** returns the **absolute** value in integer or float

Output Screenshot:

Initial Window



Output Window