Asingh
Apr 6 · 4 min read · Listen

How to Create a Simple Puzzle Game with Python and Pygame

> *"This was created using ChatGPT."*

Puzzle games are a great way to exercise your brain and have fun at the same time. In this tutorial, we will show you how to create a simple puzzle game using Python and the Pygame library. By the end of this tutorial, you will have a working puzzle game that you can customize and build upon to create your own unique game.

Step 1: Install Pygame

Before we can get started with creating our puzzle game, we need to install the Pygame library. Pygame is a set of Python modules that enable you to create games and multimedia applications. To install Pygame, open a command prompt or terminal window and run the following command:

```
pip install pygame
```

This will download and install the Pygame library on your system.

Step 2: Set Up the Game Window

Now that we have Pygame installed, we can start setting up our game window. Open your favorite code editor and create a new Python file called `puzzle_game.py`. Then, import the Pygame library and set up the game window by adding the following code:

```
import pygame

# Set up the game window
pygame.init()
WIDTH, HEIGHT = 800, 600
screen = pygame.display.set_mode((WIDTH, HEIGHT))
pygame.display.set_caption("Puzzle Game")
```

In this code, we import the Pygame library and set up the game window with a width of 800 pixels and a height of 600 pixels. We also set the title of the game window to “Puzzle Game”.

Step 3: Set Up the Puzzle Board

Now that we have the game window set up, we can start creating the puzzle board. The puzzle board is a grid of squares that the player must fill in to complete the puzzle. In this tutorial, we will create a simple puzzle board with random squares filled in.

To create the puzzle board, add the following code after the game window setup:

```
import random

# Set up the puzzle board
board = [[random.randint(0, 1) for _ in range(6)] for _ in range(5)]
```

In this code, we import the `random` library and create a 2D list `board` that represents the puzzle board. Each element of the list is a 0 or a 1, representing an empty or filled

Now that we have the puzzle board set up, we can start drawing it on the game window. To draw the puzzle board, add the following code after the puzzle board setup:

```
# Draw the puzzle board
for row in range(5):
    for col in range(6):
        if board[row][col] == 1:
            pygame.draw.rect(screen, (0, 0, 0), (col * 100 + 50, row * 100 + 50, 5
        else:
            pygame.draw.rect(screen, (255, 255, 255), (col * 100 + 50, row * 100 +
```

In this code, we loop through the `board` list and draw rectangles on the screen using the `pygame.draw.rect` function. If an element of `board` is 1, we draw a black rectangle. Otherwise, we draw a white rectangle with a black border.

Step 5: Handle Events

Now that we have the puzzle board drawn on the game window, we need to handle events such as mouse clicks and key presses. In this tutorial, we will handle mouse clicks to fill in squares on the puzzle board.

To handle events, add the following code after the puzzle board drawing code:

```
            for col in range(6):
                if board[row][col] == 1:
                    pygame.draw.rect(screen, (0, 0, 0), (col * 100 + 50, row * 100 + 5
                else:
                    pygame.draw.rect(screen, (255, 255, 255), (col * 100 + 50, row * 1

        pygame.display.update()
```

In this code, we create a while loop that handles events until the user quits the game window. We check for two types of events: the `pygame.QUIT` event, which occurs when the user closes the game window, and the `pygame.MOUSEBUTTONDOWN` event, which occurs when the user clicks the mouse button.

If the user clicks the mouse button, we calculate which square on the puzzle board was clicked and toggle its value. Then, we redraw the puzzle board on the screen.

Step 6: Run the Game

Now that we have completed the puzzle game, we can run it by executing the `puzzle_game.py` file. Open a command prompt or terminal window and navigate to the directory where the file is located. Then, run the following command:

```
python puzzle_game.py
```

his will start the puzzle game window. Click on the squares to fill them in and complete the puzzle.

Conclusion

In this tutorial, we have shown you how to create a simple puzzle game using Python and the Pygame library. You can customize this game and add new features to make it your own. We hope that you have found this tutorial helpful and that it has inspired you to create your own games with Python and Pygame.