CompTech 10
Term 3 Project:

A Tile Memory Game In Python

Evan Praël

May 2024

Topics:

- The idea for the project
- Game programming
- AI tools used
- Live demo of final game
- Behind the scenes

The Idea For The Project

In 2008, Lego Indiana Jones was released

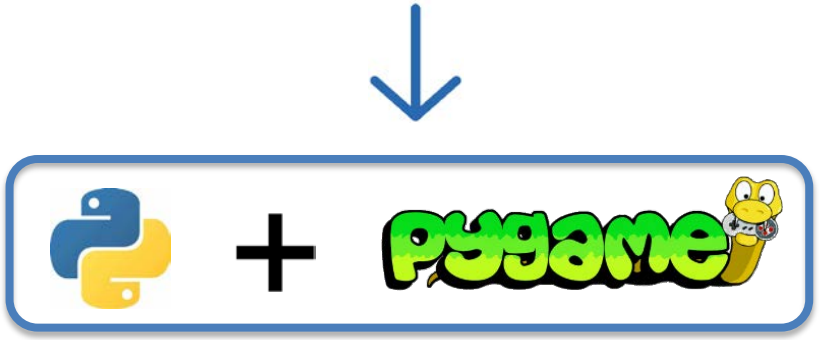
There is this tile puzzle that appears all over the game

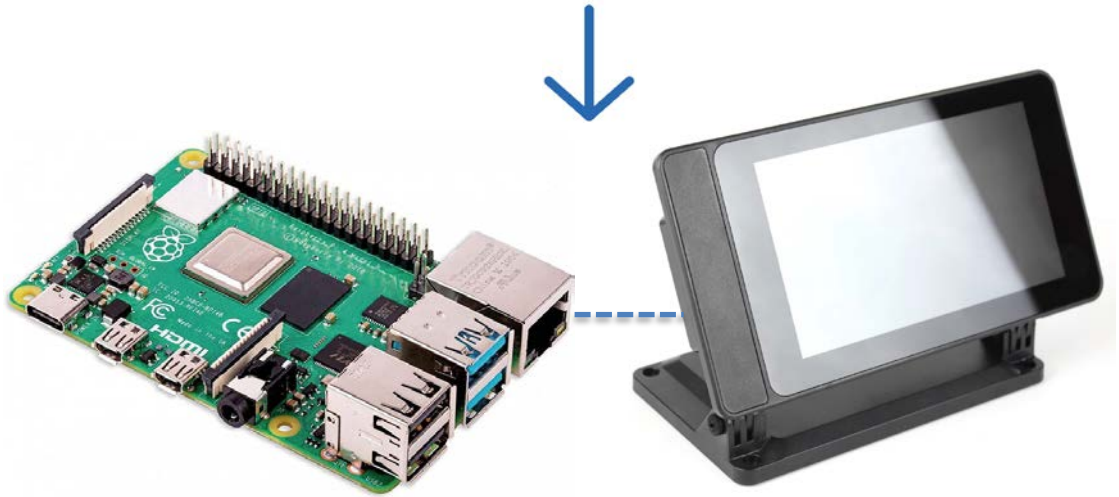
Sample Clip from Lego Indiana Jones

(view on Youtube)

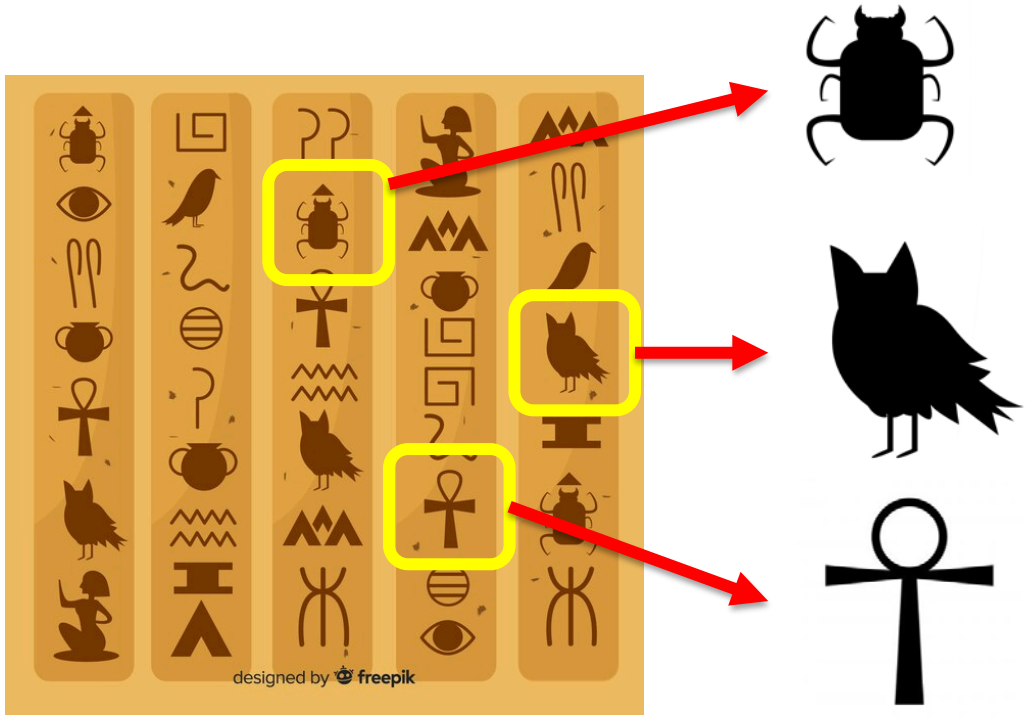
Project Tasks

1. Write the tile game in Python
2. Use the Pygame library for graphics
3. Run it on a Raspberry Pi
4. Connect a touch screen





Making The Gameboard



Online Clipart

- Hi-Res gameboard required
- Found Egyptian symbols online
- Created a new board

Making The Gameboard

Lo-Res 😕

Hi-Res! 🙂

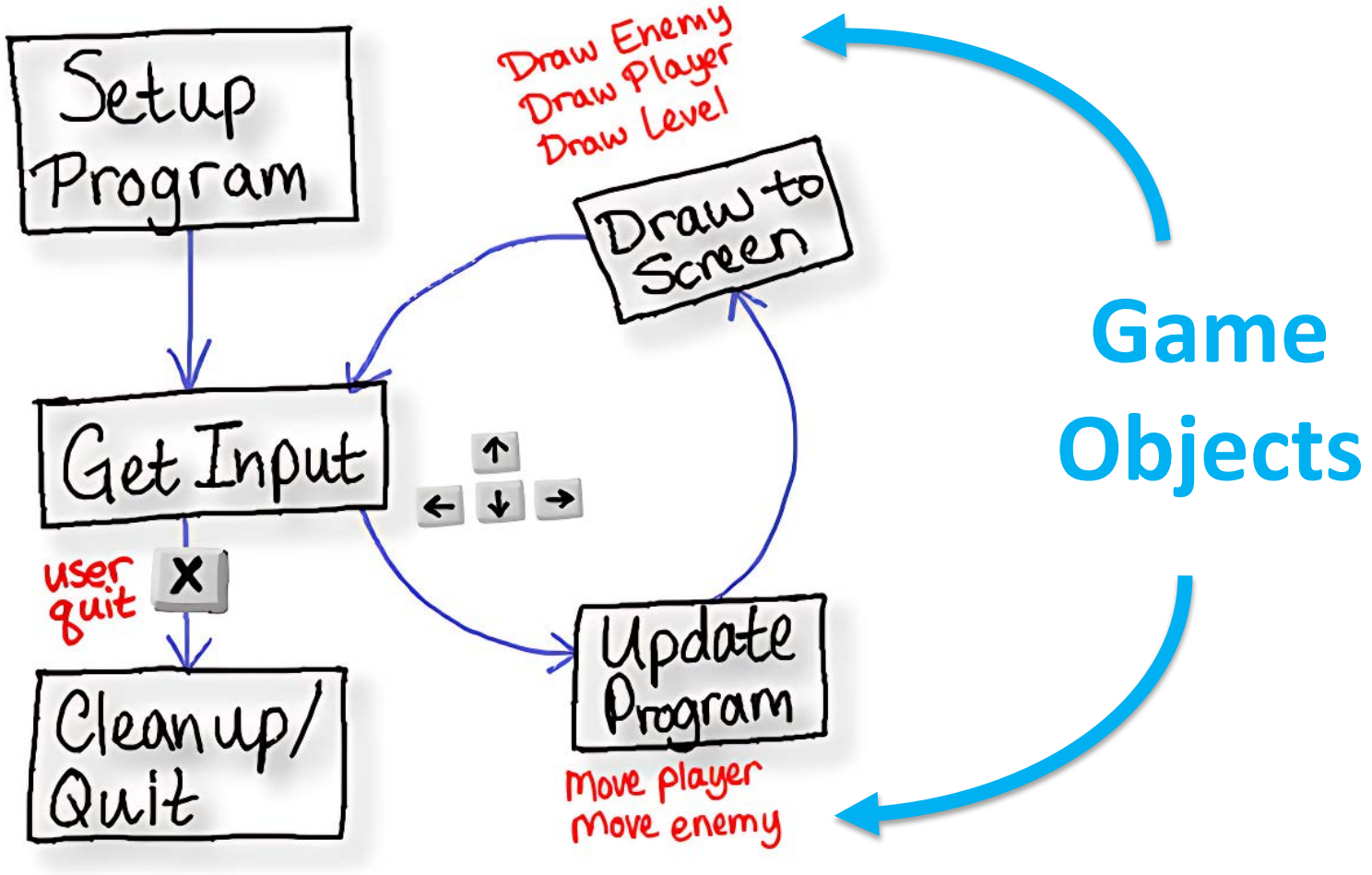
Making The Gameboard

Door Sign! 😊

Hi-Res! 🙂

Game Programming

The Game Loop:



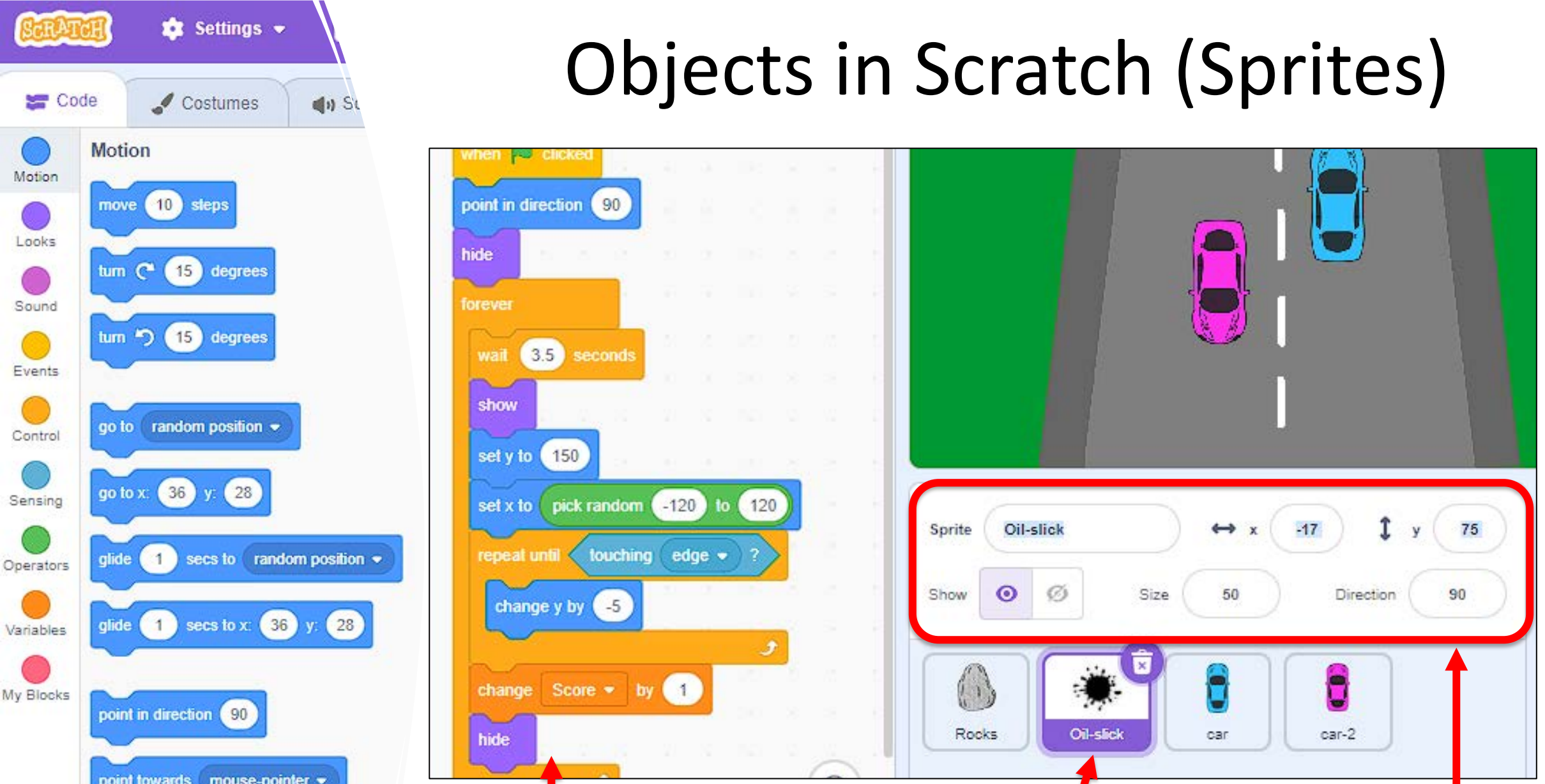
Game Objects

For Each Object On The Screen:

- Where is it? (x,y coordinates)
- Is it moving? In what direction?
- Is it fading, expanding, rotating?
- Etc, etc!!

Lots of stuff to keep track of!

Objects in Scratch (Sprites)

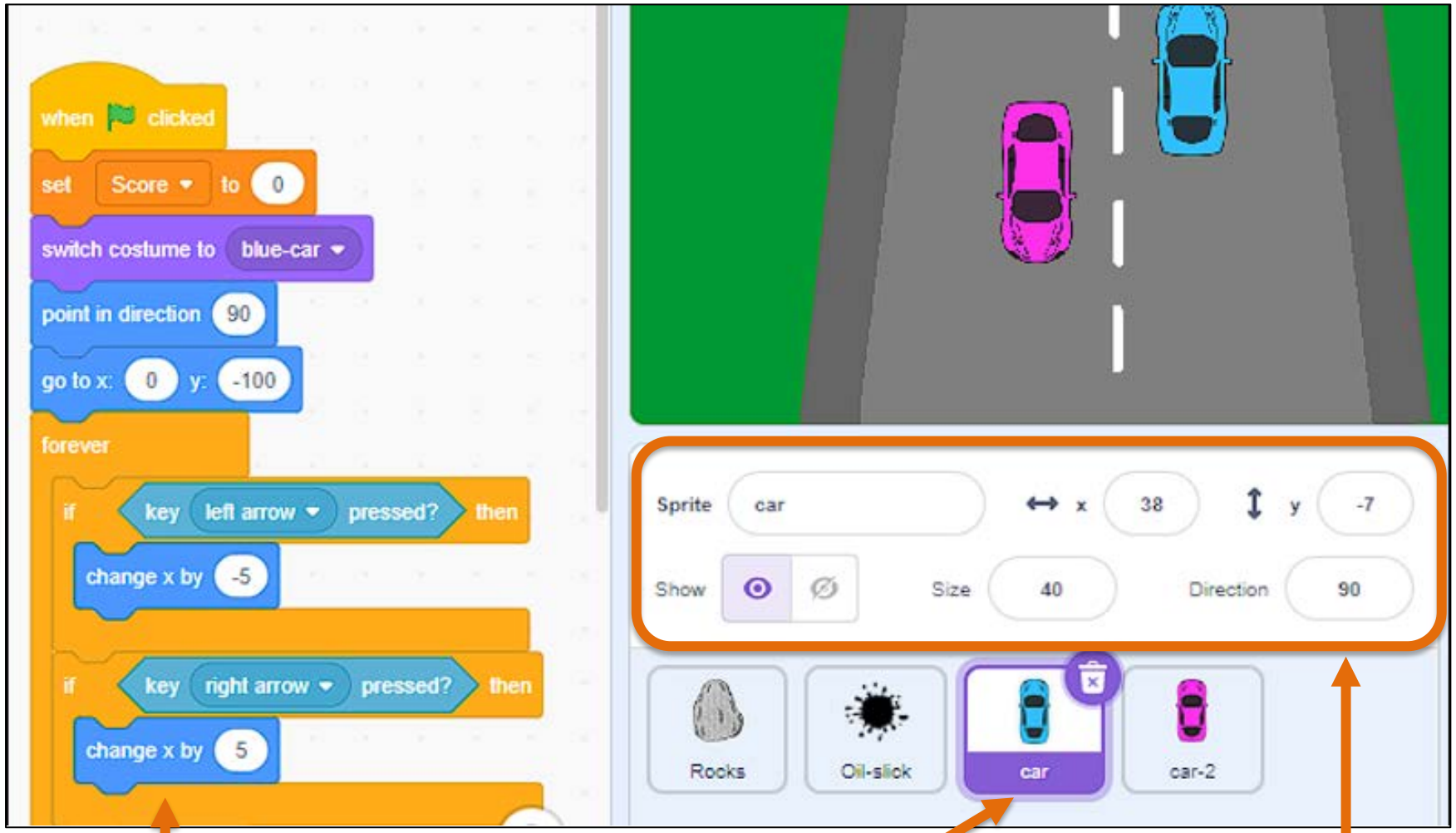


Oil Actions

Oil Object

Oil Properties

Car Object - Scratch



Car Actions

Car Object

Car Properties

Car Object - Python

```
class Car:
```

Class Definition

Car Object

```
    speed = 0
    started = False
```

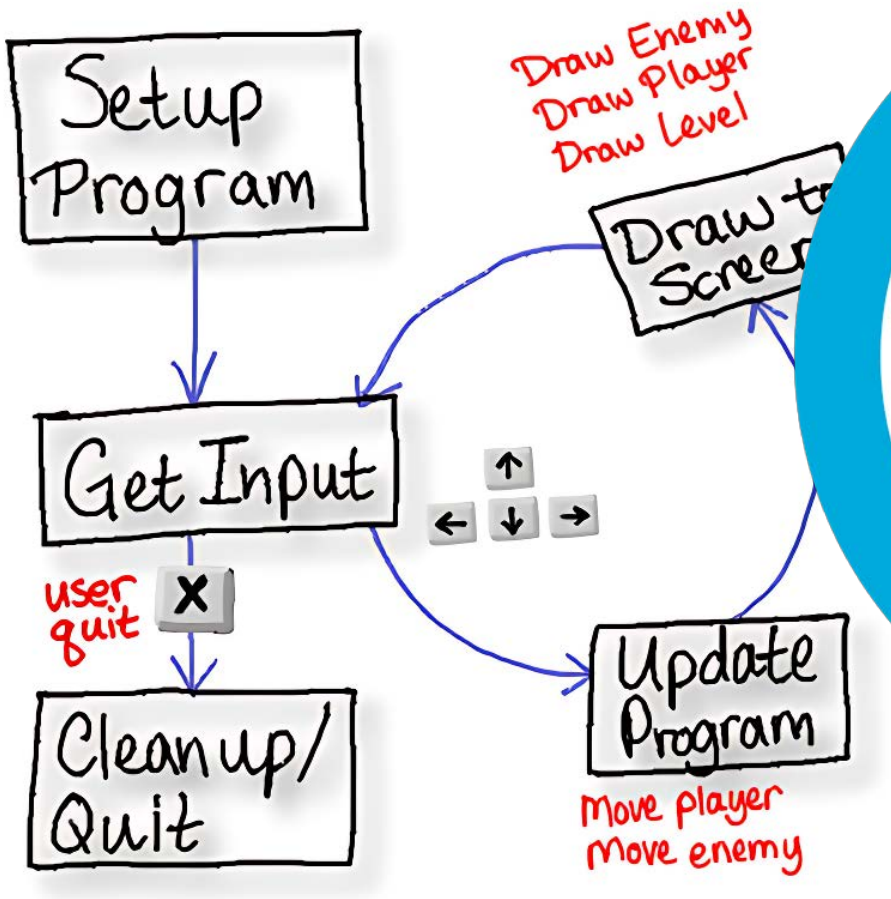
Car Properties

```
    def start(self):
        self.started = True
        print("Car started, let's ride!")

    def increase_speed(self, delta):
        self.speed = self.speed + delta
        print('Vroooooom!')

    def stop(self):
        self.speed = 0
        print('Halting')
```

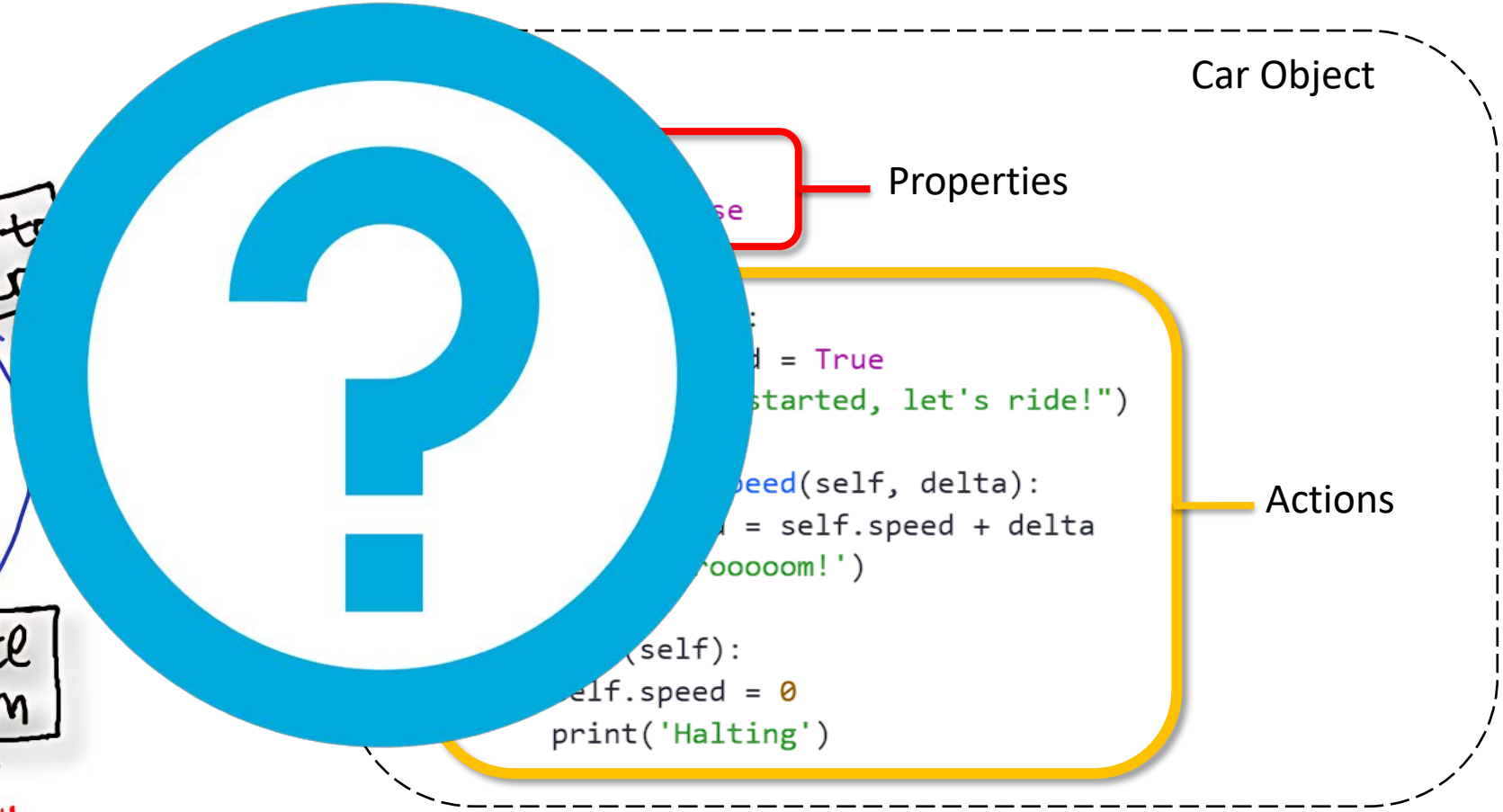
Car Actions

So How Do I Combine:

Game Loop



Objects



First Stop:

ChatGPT

Prompt:

Write a tile memory game in python using the tiles from this 3x3 game board (image uploaded). The game starts with the board displayed in the center of the screen. The computer then picks 4 random tiles and highlights each one on the board in sequence. The highlight happens in the form of a quick pulse. The user then clicks on the same tiles in the same order, each click highlighting the tile in the same manner as the computer did. If the user gets the order wrong or picks a wrong tile, the computer will say so and the game starts over. If the user gets it right, the game board is replaced with a random image of an Egyptian artifact. This image comes from a file stored in a sub directory of the game folder and any image in this folder can be displayed. The game then starts over. The program should be written using a standard game loop. Represent game objects like the board and the tiles with classes. Have the classes use update and draw methods to handle animations. Use a font that works on Windows and Raspberry Pi.

ChatGPT

Response:

To create this tile memory game in Python, we'll use the Pygame library, which is perfect for handling game loops, animations, and user input. Here's a step-by-step guide to implementing the game as described:

1. Initialize Pygame and set up the game window.
2. Create classes for the game objects: the board, the tiles, and the game manager.
3. Handle the game logic, including the sequence highlighting, user input, and result checking.
4. Add the functionality to display a random image of an Egyptian artifact upon successful completion.
5. Implement a main game loop to keep the game running and updating.

Below is the complete code to create this game:

160 lines of code!
With objects & a game loop!

```
import pygame
import random
import os
from pathlib import Path

# Initialize Pygame
pygame.init()

# Constants
SCREEN_WIDTH, SCREEN_HEIG
BOARD_SIZE = 3
TILE_SIZE = 150
MARGIN = 10
HIGHLIGHT_COLOR = (255, 2
BACKGROUND_COLOR = (255,
TILE_IMAGES = ["/mnt/data

# Load the Egyptian artif
ARTIFACT_FOLDER = "artifa
ARTIFACT_IMAGES = [os.pat
```

```
# Classes
class Tile:
    def __init__(self,
        self.rect = py
        self.image = p
        self.highlight

    def draw(self, scr
        if self.highli
            pygame.dra
        screen.blit(py
self.rect.topleft)

class Board:
    def __init__(self)
        self.tiles = [
        for i in range
            for j in r
```

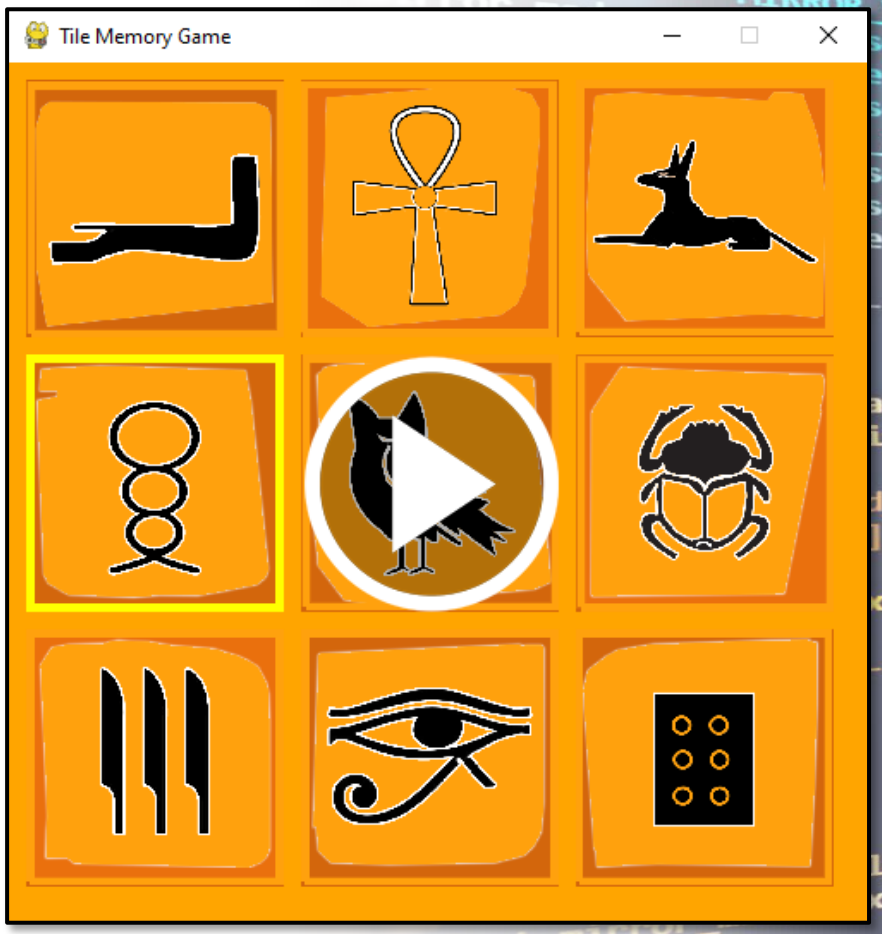
```
class GameManager:
    def __init__(self):
        self.board = Board()
        self.sequence = []
        self.user_sequence = []
        self.show_sequence = True
        self.sequence_index = 0

    def start_new_game(self):
        self.sequence = random.sample(self.board.t
        self.user_sequence = []
        self.show_sequence = True
        self.sequence_index = 0

    def update(self):
```

ChatGPT

A working starter app!

Some fixes required:

- Tiles weren't highlighting at first
- The artifact image needed resizing
- The board didn't reset between games
- Demo of ChatGPT Code



(view on Youtube)

Not the whole project!
But a great starter!

Full ChatGPT session

Things I Added

- A startup page
- A menu screen
- Difficulty levels (easy, medium, hard)
- Sound fx from the original game
- Temple music
- Pyramid chamber backgrounds
- Animations

And here's the final game...

The Final Game

On Windows

(view on Youtube)

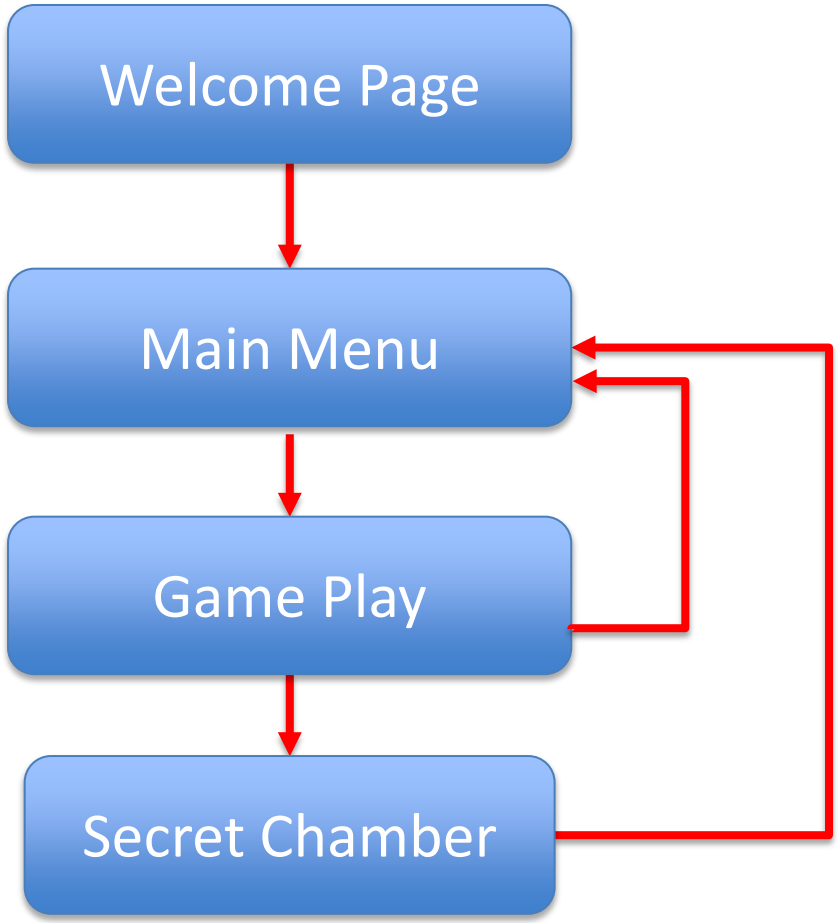
Demo now

On Raspberry Pi + Touch

(view on Youtub

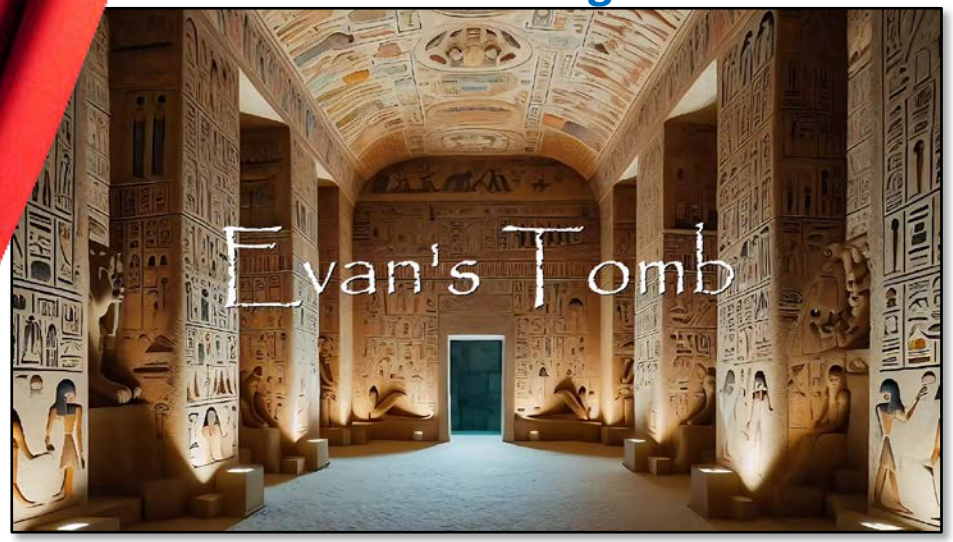
Try after

Behind The Scenes - Game Screens



Behind The Scenes - Game Screens

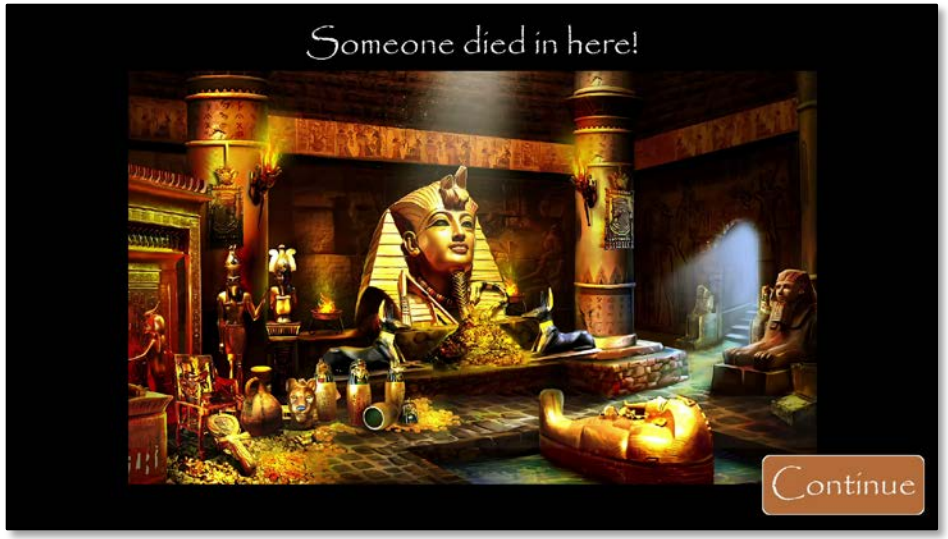
Welcome Page



Main Menu

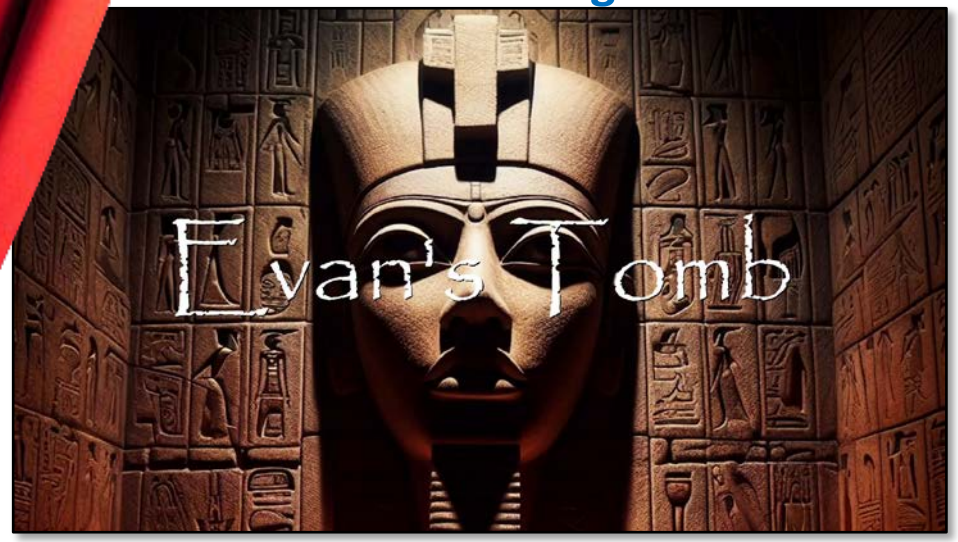
Game Play

Secret Chamber

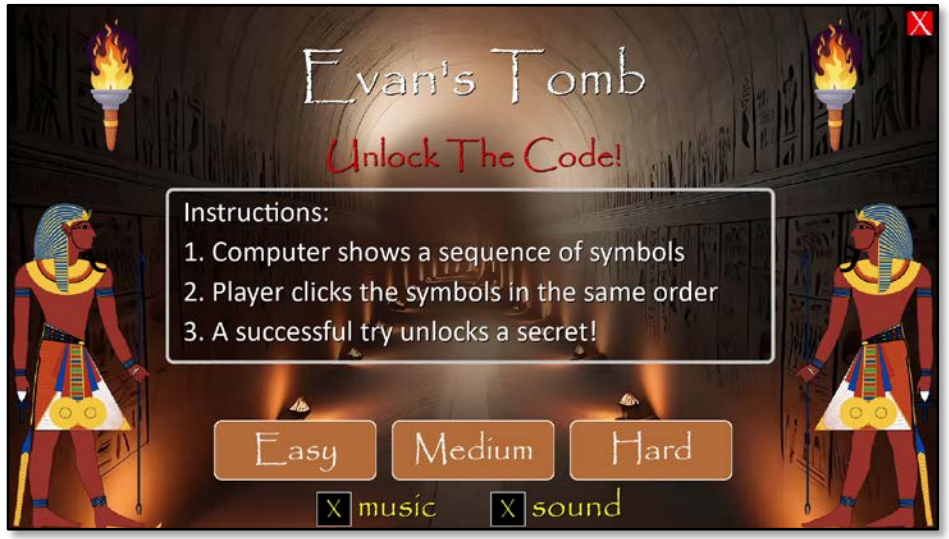


Behind The Scenes - Game Screens

Welcome Page

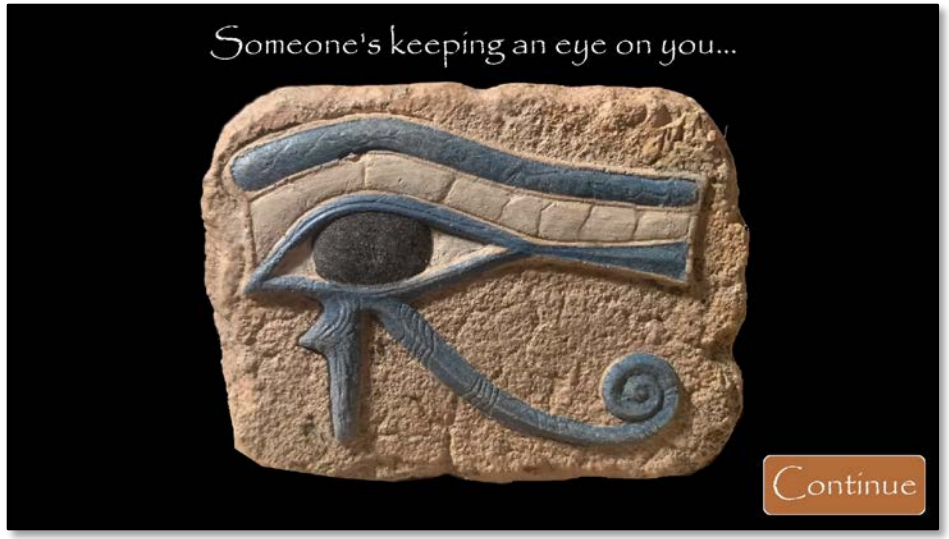


Main Menu



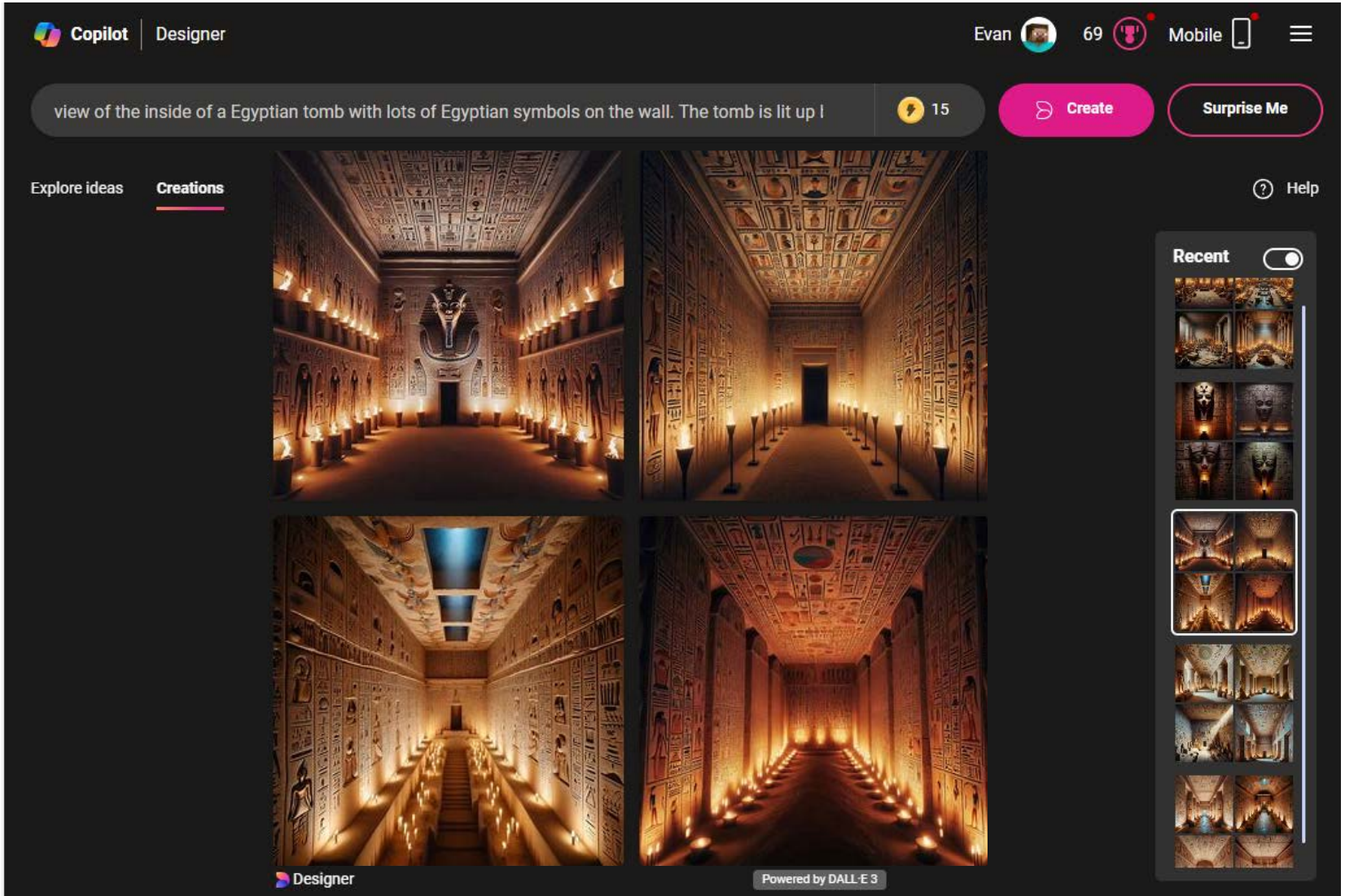
Game Play

Secret Chamber

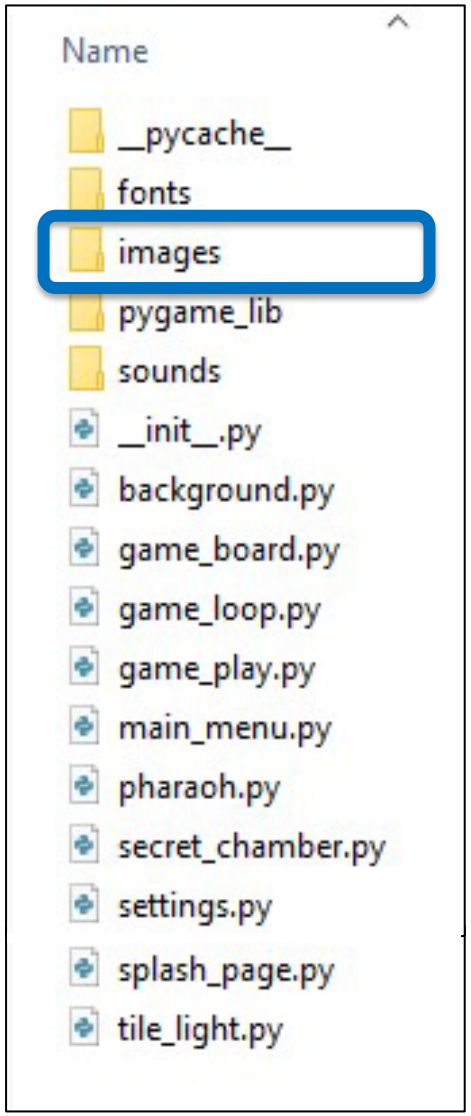


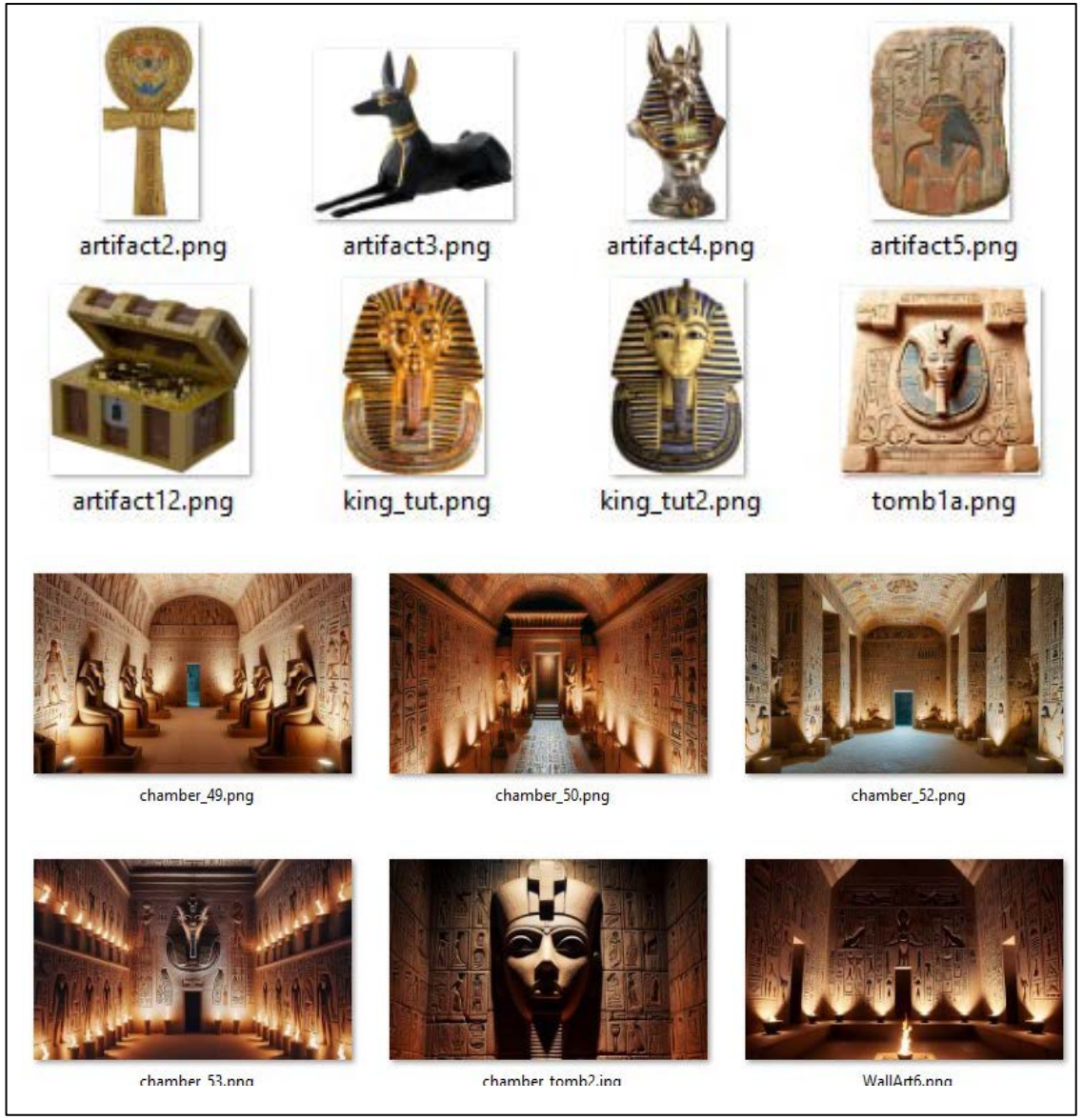
Behind The Scenes – More AI Tools

Making backgrounds with Bing Image Creator

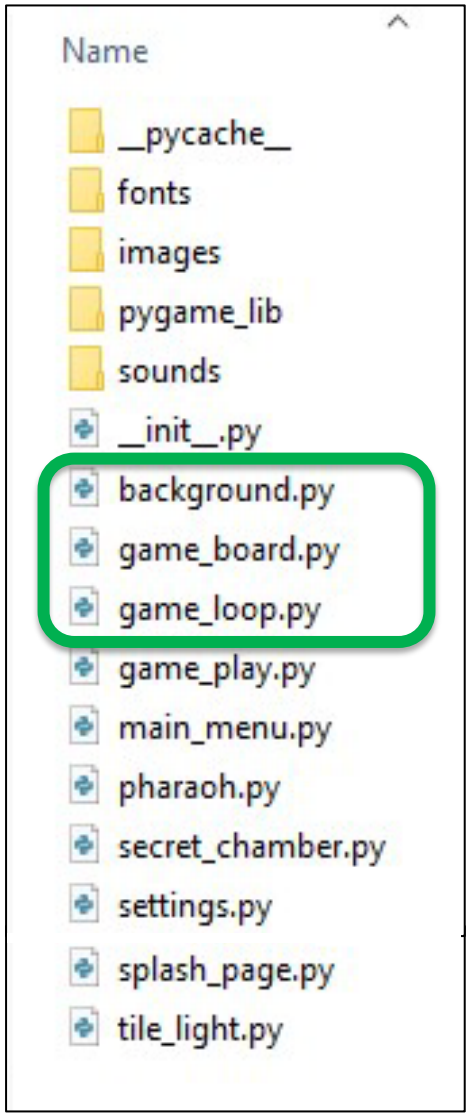


Behind The Scenes – Game Files





Behind The Scenes – Python Code

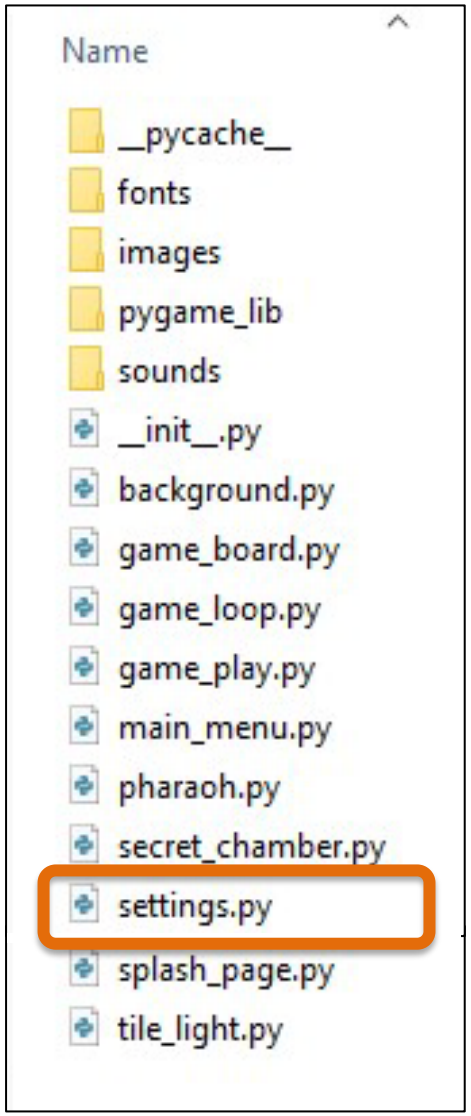



```python
class Background:
    def update(self, events):
        for event in events:
            if event.type == pygame.MOUSEBUTTONDOWN:
                if self.exit_button.is_clicked(event.pos):
                    self.game_exit = True
```

```python
class GameBoard:
    def setup_game_board(self, zoom_level=100):

        self.game_board = self.game_board_full.copy()

        if zoom_level != 100:
            scale_x = self.game_board.get_width() * zoom_level // 100
            scale_y = self.game_board.get_height() * zoom_level // 100
            self.game_board = pygame.transform.scale(self.game_board,
```

```python
class GameLoop:
    def load_background_music(self):
        if self.settings.background_music_sound != "" and \
           os.path.exists(self.settings.get_soundPath(self.settings
            pygame.mixer.music.load(self.settings.get_soundPath(sel
            pygame.mixer.music.set_volume(self.settings.initial_mus
            self.settings.background_music_available = True
        else:
            self.settings.background_music_available = False
```

Behind The Scenes – Game Settings




```python
class Settings:
    game_title = "Evan's Tomb"

    # fonts
    #----------------------------------------
    default_font = 'papyrus'
    instructions_font = 'calibri'

    # animations
    #----------------------------------------
    pharao_speed = 1                    # 0=still, 5=fast
    gameboard_zoomout_duration = 5      # 1=fast, 10=slow
    tile_pulse_duration_per_level = [0.4,0.3,0.2]           # use 0.2-0.

    # sounds
    #----------------------------------------
    sound_fx_on = True
    background_music_on = True
    initial_music_volume = 0.2
    background_music_sound = "background - temple of light.ogg"
    tile_pulse_sounds = ["tile_pulse_1.ogg", "tile_pulse_2.ogg"]
    menu_hover_sound = "menu_hover.ogg"
    tile_selector_sound = "tile_selector.ogg"
    rumbling_sound = "rumble3.ogg"
    temple_chant = "deep_voice_chant2.ogg"
    secret_chamber_sound = "deep_voice_chant2.ogg"
```

Acknowledgments

Lots of help from lots of places!

AI Tools:

- ChatGPT for creating the first cut of the game and answering many coding questions
- Github Copilot for helping to write even more code and adding comments too
- Powerpoint Designer for helping to make nice looking slides
- Bing Image Creator for creating images of Egyptian tombs & chambers

Game Audio:

- Soundtrack: Temple of Light (Youtube)
- Temple Chant: Tibetan Monks (Youtube)
- Sound FX: Original Lego Indiana Jones game

Gameboard Remake:

- Egyptian Symbols from Freepik

Coding:

- Many online tutorials on pygame & objects
- If it wasn't for the AI tools and my dad's help, I would not have gotten this done. 🙏

Acknowledgments

Online References

Tutorials

- Official Pygame Documentation
- Making Games With Pygame
- Official Pygame Tutorials
- Pygame - Game Programming Primer
- Objects And Classes

Pygame Audio:

- Playing multiple audio tracks
- Playing an audio file
- Using pygame mixer

Sample Code & Games

- Free Python Games
- Top Games Made with Pygame
- Pygame Examples
- Pygame Built-in Examples
- 2020 Pygame Projects

Egyptian Symbols & Images

- Egyptian Symbols @ Freepik
- Egyptian Artifacts On Bing
- Bing Image Creator

Full Project On GitHub

https://github.com/eprael/EvansTomb

Thanks for Watching!