

game.py

Scan details

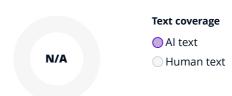
Scan time: Total Pages: Total Words: April 13th, 2024 at 9:18 UTC 6664 27

Plagiarism Detection



Types of plagiarism		Words
Identical	7.9%	528
Minor Changes	4.9%	328
Paraphrased	21.4%	1423
Omitted Words	0%	0

Al Content Detection



=Q Plagiarism Results: (66)

main.py

13.4%

https://raw.githubusercontent.com/kidscancode/pygame_tutorials/master/platform/working/main.py

KidsCanCode - Game Development with Pygame video series # Jumpy! (a platform game) # Art from Kenney.nl # Happy Tune by http://open...

【Pygame小游戏】你喜欢走迷宫?那错过这么好的就太可惜了~(AI版本) - 掘金

6%

https://juejin.cn/post/7065851266714304525

首页 首页 沸点 课程 直播 活动 ...

Pygame实战之迷宫游戏的实现-Pygame教程-码农之家

6%

https://www.xz577.com/j/284744.html

首页 计算机电子书 配套资源 课后答案 技术文档 软件教程 当前位置:主页 > python教程 > Pygame迷宫 Pygame实战 之迷宫游戏的实现 发布:2022-06-19 14...

3zqecxfpt - Python - OneCompiler

4.1%

https://onecompiler.com/python/3zqecxfpt

3zqecxfpt main.py 3zqecxfpt main.py import sys import pygame from scripts.utils import load_im...









Pygame Platformer 04 — Camera & Background | by Yen | Medium

3.8%

https://medium.com/@monlesyen/pygame-platformer-04-camera-background-893b61a37163

Yen

Open in appSign up Sign in Write Sign up Sign in Pygame Platformer 04 — Camera & Background Yen ·...

Pygame Platformer 07 — Particles. This is the note taken while I was... | ...

2.6%

https://medium.com/@monlesyen/pygame-platformer-07-particles-fed706d34c5d

Yen

Open in appSign up Sign in Write Sign up Sign in Pygame Platformer 07 — Particles Yen ·Follow 3 m...

Pygame Platformer 03 — Physics, Output data of rect, Jump | by Yen | M...

2.2%

https://medium.com/@monlesyen/pygame-platformer-03-physics-output-data-of-rect-jump-d75947dae66

Yen

Open in appSign up Sign in Write Sign up Sign in Pygame Platformer 03 — Physics, Output data of rect, Jump ...

3zyb72ffm - Python - OneCompiler

2.1%

https://onecompiler.com/python/3zyb72ffm

3zyb72ffm HelloWorld.py 3zyb72ffm HelloWorld.py import random class OnePieceGuessingGame: def...

python - Pygame font error - Stack Overflow

2.1%

https://stackoverflow.com/questions/28517979/pygame-font-error

Stack Overflow About ...

Pygame Platformer 02 — Collider, Utils, Entities, Tilemap & Render | by Y...

2%

https://medium.com/@monlesyen/pygame-platformer-02-collider-utils-entities-tilemap-1acc19db9323

Yen

Open in appSign up Sign in Write Sign up Sign in Pygame Platformer 02 — Collider, Utils, Entities, Tilemap &...

python - How to display text with font and color using pygame? - Stack O...

2%

https://stackoverflow.com/questions/10077644/how-to-display-text-with-font-and-color-using-pygame

Stack Overflow About ...











2%

https://raw.githubusercontent.com/kidscancode/pygame_tutorials/master/shmup/shmup-8.py

KidsCanCode - Game Development with Pygame video series # Shmup game - part 8 # Video link: https://www.youtube.com/watch?v=abm1VwF...

shmup-7.py

2%

https://raw.githubusercontent.com/kidscancode/pygame_tutorials/master/shmup/shmup-7.py

KidsCanCode - Game Development with Pygame video series # Shmup game - part 7 # Video link: https://www.youtube.com/watch?v=U8yyrpu...

Error in My Code - Python - Replit Ask

2%

https://ask.replit.com/t/error-in-my-code/2809

Loading ...

python - How do I display a score at the top of my screen in pygame? - St...

2%

https://stackoverflow.com/questions/59175392/how-do-i-display-a-score-at-the-top-of-my-screen-in-pygame

Stack Overflow About ...

3zxcqby7g - Python - OneCompiler

2%

https://onecompiler.com/python/3zxcqby7g

3zxcqby7g HelloWorld.py 3zxcqby7g HelloWorld.py import pygame import sys import random # Initi...

python - Pygame unable to open file - Stack Overflow

2%

https://stackoverflow.com/questions/45773229/pygame-unable-to-open-file

Stack Overflow About ...

Justin Pombrio

2%

https://justinpombrio.net/code/lights-out/

Justin Pombrio About Me brown-transcript.pdf Friends Pictures resume.pdf wpi-transcript.pdf Code EC2 Map PEG Con...







pygame学习(3) 画面和字幕 - 灰人 - 博客园

https://www.cnblogs.com/snailbuster/p/13121312.html

python - I'n creating my first game in pygame and i need help on showing...

2%

2%

https://stackoverflow.com/questions/70131211/in-creating-my-first-game-in-pygame-and-i-need-help-on-sho... Stack Overflow About ...

如何在游戏的屏幕顶部显示分数?-腾讯云开发者社区-腾讯云

2%

https://cloud.tencent.com/developer/ask/sof/108400094

腾讯云 开发者社区 文档建议反馈控制台 首页学习 活动 专区 工具 TVP最新优惠活动 文章/答案/技术大牛搜索 搜索关 闭 发布 登录/注册 首页学习 活动 专区 工具 TVP最新优惠活动 返回腾讯云官网 ...

Solved I need the solution in python. Please run the code in | Chegg.com

2%

https://www.chegg.com/homework-help/questions-and-answers/need-solution-python-please-run-code-thre...

Skip to main contentStudy TasksHomework help Understand a topic Writing & citations ToolsExpert Q&A Textbook Solutions Math So...

── pygame小游戏 视力诊断器——找到数字1 - 知乎

2%

https://zhuanlan.zhihu.com/p/146428068

切换模式 写文章 登录/注册 pygame小游戏 视力诊断器——找到数字1 Reader 这个人很神秘,什么也没写 设计理念(视 力诊断器——找到数字1) 随机生成40*40的数字7,其中包含一个数字1,找到它,证明自己的视力正常 特点: ...

game.py

2%

https://raw.githubusercontent.com/abdelatief/tictactoe-pygame/master/game.py

import pygame from sys import exit from logic import TicTacToe colors = { 'white': (255, 255, 255), 'blue': (0, 0, 200), ...

python - pygame font Attribute error - Stack Overflow

https://stackoverflow.com/questions/51585285/pygame-font-attribute-error

Stack Overflow About ...







2%

Python Pygame分数显示-腾讯云开发者社区-腾讯云

2%

https://cloud.tencent.com/developer/ask/sof/105663739

腾讯云 开发者社区 文档建议反馈控制台 首页学习 活动 专区 工具 TVP最新优惠活动 文章/答案/技术大牛搜索 搜索关 闭 发布 登录/注册 首页学习 活动 专区 工具 TVP最新优惠活动 返回腾讯云官网 ...

PrimarySettings.py

2%

https://raw.githubusercontent.com/faridhazratian/tank-trouble/main/primarysettings.py

from os import path import pygame vector = pygame.math.Vector2 # define colors based on (R, G, B) WHITE = (255, 255, 255) BLACK = ...

Reddit - Dive into anything

2%

https://www.reddit.com/r/learnpython/comments/sgbddp/urgent_help_with_python_sprite_movement/

Skip to main content ...

game.py

2%

https://raw.githubusercontent.com/mehedih/lafa/master/game.py

import pygame, random, pyrebase # import pygame and other modules from defaults import * # bring in the default params from sprites ...

Reddit - Dive into anything

1.9%

https://www.reddit.com/r/pygame/comments/15ldj02/typeerror_int_object_is_not_subscriptable/

Skip to main content ...

Light Gun Game Arcade: Space Protector

1.5%

https://courses.ece.cornell.edu/ece5990/ece5725_fall2018_projects/lab_yp373_sl2737/index.html

Toggle navigation ...

使用 Python 开始游戏开发 - HackerVirus - 博客园

1.5%

https://www.cnblogs.com/leo_wl/p/3248767.html









Week 4 - Section 2 - The Game

1.3%

https://elearn.ellak.gr/mod/book/tool/print/index.php?id=1910

Skip to main content Print book Week 4 - Section 2 - The Game ...

Pygame a primer - Learn how Pygame creates fully-featured g...

1.2%

https://sqlpad.io/tutorial/pygame-a-primer/

SQLPad Menu ...

avoid_v0.6.0.py

1%

https://raw.githubusercontent.com/gkalemis/avoid-rocks/master/avoid_v0.6.0.py

"ELLAK - Python Course Avoid the Rocks v0.6.0 "import pygame # Define some colors BLACK = (0, 0, 0) WHITE = (255, 255, 255)...

avoid_v0.7.0.py

1%

https://raw.githubusercontent.com/gkalemis/avoid-rocks/master/avoid_v0.7.0.py

"ELLAK - Python Course Avoid the Rocks v0.7.0 "import pygame # Define some colors BLACK = (0, 0, 0) WHITE = (255, 255, 255)...

game.py

0.9%

https://raw.githubusercontent.com/randhir408/sourceofcargame/main/game.py

""" @author: Randhir Kumar @email: randhirkumarsonukumar@gmail.com @date: 13-september-2020 """ import random from time import sleep ...

3yzqncfez - Python - OneCompiler

0.9%

https://onecompiler.com/python/3yzqncfez

3yzqncfez HelloWorld.py script1.py 3yzqncfez HelloWorld.py script1.py import HelloWorld.py HelloW...

python - Real time typing in PyGame - Stack Overflow

0.8%

https://stackoverflow.com/questions/22957726/real-time-typing-in-pygame

Stack Overflow About ...









Pooter.py

0.8%

https://raw.githubusercontent.com/exphat/binding-of-isaac/master/pooter.py

Pooter.py # Aaron Taylor # Moose Abumeeiz # # The pooter will follow the user similarly to a fly # from pygame import * from cons...

● [지금 무료] 파이썬 무료 강의 (활용편1) - 추억의 오락실 게임 만들기 (3시간) 강의 -...

0.7%

https://www.inflearn.com/course/%eb%82%98%eb%8f%84%ec%bd%94%eb%94%a9-%ed%8c%8c%ec%9d%b... 질문 게시판 고민...

● [지금 무료] 파이썬 무료 강의 (활용편1) - 추억의 오락실 게임 만들기 (3시간) 강의 -...

0.7%

https://www.inflearn.com/course/%eb%82%98%eb%8f%84%ec%bd%94%eb%94%a9-%ed%8c%8c%ec%9d%b... 질문 게시판 고민...

圖 [지금 무료] 파이썬 무료 강의 (활용편1) - 추억의 오락실 게임 만들기 (3시간) 강의 -...

0.7%

https://www.inflearn.com/course/%eb%82%98%eb%8f%84%ec%bd%94%eb%94%a9-%ed%8c%8c%ec%9d%b... 질문 게시판 고민...

© Create Your Own Music Player in Pygame | by Mashiur Rahman | Mar, 20...

0.7%

https://medium.com/@01one/create-your-own-music-player-in-pygame-7d8d81d8580c

Mashiur Rahman

Open in appSign up Sign in Write Sign up Sign in Create Your Own Music Player in Pygame Mashiur Rahman ...

© Computer Science With Python and Pygame

0.6%

http://programarcadegames.com/python_examples/f.php?file=bitmapped_graphics.py

Toggle navigation ...

● 如何用Python创建自己的Dino Run?-睿象云平台

0.6%

https://m.aiops.com/news/post/11627.html

...









Computer Science With Python and Pygame

0.5%

http://programarcadegames.com/python_examples/f.php?file=move_keyboard.py

Toggle navigation ...

pygame - Python window opens and closes - Stack Overflow

0.5%

https://stackoverflow.com/questions/51639365/python-window-opens-and-closes

Stack Overflow About ...

My program is very laggy

0.5%

https://python-forum.io/thread-4391.html

View Active Threads V...

python - pygame - Unpress key on event.key - Stack Overflow

0.5%

https://stackoverflow.com/questions/51507113/pygame-unpress-key-on-event-key

Stack Overflow About ...

python - name 'event' is not defined not sure why im getting this - Stack ...

0.5%

https://stackoverflow.com/questions/67569585/name-event-is-not-defined-not-sure-why-im-getting-this and the state of the

Stack Overflow About ...

How to close audio file?

0.5%

https://groups.google.com/g/pygame-mirror-on-google-groups/c/xjsh9zs8j0u

Groups Conversations All groups and messages Send feedback to Google Help Training Si...

pygame.mixer.music — pygame v2.6.0 documentation

0.5%

https://www.pygame.org/docs/ref/music.html

pygame documentation Pygame Home | | Help Contents | | Reference Index ...









0.07

Adding Sound Effects And Music To Your Pygame Project - Code With C

0.5%

https://www.codewithc.com/adding-sound-effects-and-music-to-your-pygame-project/?amp=1

Code With C The Way to Programming ...

pygame.mixer.music — pygame v2.6.0 documentation

0.5%

https://www.pygame.org/docs/ref/music.html?highlight=pos

pygame documentation Pygame Home | | Help Contents | | Reference Index ...

Solved PYTHON SHAPE MOVING import pygame pygame.init() | Chegg.com

0.5%

https://www.chegg.com/homework-help/questions-and-answers/python-shape-moving-import-pygame-pyga...

Skip to main contentStudy TasksHomework help Understand a topic Writing & citations ToolsExpert Q&A Textbook Solutions Math So...

How to use the pygame.draw.rect function in pygame | Snyk

0.5%

https://snyk.io/advisor/python/pygame/functions/pygame.draw.rect

PyPI npm PyPI Go Docker Magnify icon All Packages JavaScript Python Go Code Examples JavaScript Python...

Pylance crashing for unknown reason · Issue #4712 · microsoft/pylance-r...

0.5%

https://github.com/microsoft/pylance-release/issues/4712

Skip to content ...

python - Pygame screen won't fill or draw text - Stack Overflow

0.4%

https://stackoverflow.com/questions/48064934/pygame-screen-wont-fill-or-draw-text

Stack Overflow About ...

● pygame 的声音播放 - 言若是诺 - 博客园

0.4%

https://www.cnblogs.com/liquancai/p/13236974.html

...









PYTHON游戏:冲刺弹球(pygame实现的游戏)_python游戏代码资源-CSDN文库

0.4%

https://download.csdn.net/download/gg 44886601/88722937

没有合适的资源?快使用搜索试试~我知道了~文库首页游戏开发其他PYTHON游戏:冲刺弹球(pygame实现的游戏) PYTHON 游戏:冲刺弹球(pygame实现的游戏) 共1...

如何立即产生我的迷宫,这样我就不用看它产生了?-腾讯云开发者社区-腾讯云

0.4%

https://cloud.tencent.com/developer/ask/sof/107678783

腾讯云 开发者社区 文档建议反馈控制台 首页学习 活动 专区 工具 TVP最新优惠活动 文章/答案/技术大牛搜索 搜索关 闭 发布 登录/注册 首页学习 活动 专区 工具 TVP最新优惠活动 返回腾讯云官网 ...

python - How to generate my maze instantly so I don't have to watch it G...

0.4%

https://stackoverflow.com/questions/68034054/how-to-generate-my-maze-instantly-so-i-dont-have-to-watch-i...

Stack Overflow About ...

fileserve.php?FID=35735

0.4%

https://www.ijraset.com/fileserve.php?fid=35735

9 VI https://doi.org/10.22214/ijraset.2021.35735 June 2021 International Journal for Research in Applied Science & Engineering Techno...

python - Keep Getting the error of "TypeError: 'bultin_function_or_metho...

0.4%

https://stackoverflow.com/questions/77214457/keep-getting-the-error-of-typeerror-bultin-function-or-metho...

Stack Overflow About ...

SatuMetsanHiiliTase.py

0.3%

https://heikki-valisuo.fi/xml/programming/simulaatioita/satumetsa/satumetsanhiilitase.py

coding: utf-8 "' Tämä on satumetsän peikkojen ohjelmoima simulaattori, jolla voi harjoitella peikkolaakson metsän hoitamista. " ...









```
import sys
import math
import random
import os
import pygame
import ison
from scripts.utils import load image, load pngs, Animation
from scripts.entities import PhysicsEntity, Player, Enemy
from scripts.tilemap import Tilemap
from scripts.clouds import Clouds
from scripts.particle import Particle
Main game module that integrates various components like menu, gameplay, and other game mechanics.
Utilizes pygame for rendering and managing game state, including entities, tilemaps, and interactions.
class Menu:
   Represents the main menu of the game, providing options like starting the game, viewing tutorials,
   adjusting settings, or exiting the game.
   def __init__(self):
        Initializes the game's main menu, setting up the pygame window, loading fonts, setting up button colors,
       and loading background images and sounds.
       pygame.init()
       pygame.display.set caption('JUMPY')
        self.screen = pygame.display.set_mode((1920, 1080))
        self.clock = pygame.time.Clock()
       self.font custom = pygame.font.Font(None, 70)
       self.white = (255, 255, 255)
       self.button color = (155, 155, 155)
       self.hover color = (130, 130, 130)
       self.quit_button color = (255, 0, 0)
       self.quit button hover color = (200, 0, 0)
       self.button_sound = pygame.mixer.Sound('data/sfx/click.wav')
        self.button sound.set volume(0.2)
       pygame.mixer.music.load('data/music.wav')
       self.bg image = pygame.image.load(os.path.join('data', 'pngs', 'bg.png')).convert alpha()
    def draw_text(self, text, font, color, x, y):
       Draws specified text on the screen at the given position.
        Parameters:
           text (str): The text to display.
            font (pygame.font.Font): The font used for the text.
           color (tuple): The color of the text.
            x (int): The x position of the text.
           y (int): The y position of the text.
        text surface = font.render(text, True, color)
        text rect = text surface.get rect()
        text_rect.center = (x, y)
       self.screen.blit(text surface, text rect)
    def draw button(self, text, rect, color):
       Draws a button with the specified text and color.
        Parameters:
           text (str): The text displayed on the button.
            rect (pygame.Rect): The rectangle defining the button's position and size.
            color (tuple): The color of the button.
        # Vykreslení zaobleného obdélníku tlačítka
       pygame.draw.rect(self.screen, color, rect, border_radius=20)
        # Vykreslení textu na tlačítko
       self.draw text(text, self.font custom, self.white, rect.centerx, rect.centery)
   def level selection(self):
       Displays the level selection menu, allowing the player to choose a game level.
       Returns:
```

```
int: The number of the selected level.
   self.screen.blit(self.bg image, (0, 0))
   button width = 300
   button height = 90
   level 1 button = pygame.Rect(1920 // 2 - button width // 2, 400, button width, button height)
   level_2_button = pygame.Rect(1920 // 2 - button_width // 2, 500, button_width, button_height)
   level 3 button = pygame.Rect(1920 // 2 - button width // 2, 600, button width, button height)
   running = True
   while running:
       mx, my = pygame.mouse.get pos()
       self.draw button("Level 1", level 1 button, self.hover color if level 1 button.collidepoint((mx, my)
        self.draw_button("Level 2", level_2_button, self.hover_color if level_2_button.collidepoint((mx, my)
        self.draw button("Level 3", level 3 button, self.hover color if level 3 button.collidepoint((mx, my)
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
               pygame.quit()
                sys.exit()
            if event.type == pygame.MOUSEBUTTONDOWN:
                if level 1 button.collidepoint((mx, my)):
                    self.button sound.play()
                    return 1
                elif level_2_button.collidepoint((mx, my)):
                    self.button sound.play()
                    return 2
                elif level 3 button.collidepoint((mx, my)):
                    self.button_sound.play()
                    return 3
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_ESCAPE:
                   menu.run()
       pygame.display.update()
def run(self):
   11 11 11
   The main loop for the game menu, handling button interactions and transitioning to other game states lik
    tutorial, settings, or starting the game.
   button width = 300
   button height = 90
   while True:
       self.screen.blit(self.bg image, (0, 0))
        self.draw text("Jumpy", self.font custom, self.white, 1920 // 2, 300)
       mx, my = pygame.mouse.get pos()
       play button = pygame.Rect(1920 // 2 - button_width // 2, 400, button_width, button_height)
       tutorial button = pygame.Rect(1920 // 2 - button width // 2, 500, button width, button height)
        settings button = pygame.Rect(1920 \ // \ 2 \ - button width // \ 2, 600, button width, button height)
       quit_button = pygame.Rect(1920 // 2 - button_width // 2, 700, button_width, button_height)
       play hovered = play button.collidepoint((mx, my))
       tutorial hovered = tutorial button.collidepoint((mx, my))
        settings hovered = settings button.collidepoint((mx, my))
       quit hovered = quit button.collidepoint((mx, my))
       self.draw_button("Play", play_button, self.hover_color if play_hovered else self.button_color)
        self.draw_button("Tutorial", tutorial_button, self.hover_color if tutorial_hovered else self.button_
        self.draw_button("Settings", settings_button, self.hover_color if settings_hovered else self.button_
        self.draw button("Quit Game", quit button, self.quit button hover color if quit hovered else self.qu
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
               pygame.quit()
                sys.exit()
            if event.type == pygame.MOUSEBUTTONDOWN:
                if play button.collidepoint((mx, my)):
                    self.button sound.play()
                    selected level = self.level selection()
                    if selected level is not None:
                       game = Game(start_level=selected_level - 1)
                        game.run()
                    self.state = 'level selection'
                elif quit button.collidepoint((mx, my)):
```

```
pygame.quit()
                                     svs.exit()
                              elif tutorial button.collidepoint((mx, my)):
                                     self.show_tutorial()
                              elif settings button.collidepoint((mx, my)):
                                     self.show settings()
              pygame.display.update()
def show tutorial(self):
       Displays the game's tutorial, explaining controls and gameplay mechanics to the player.
       The tutorial screen features a grey background with text instructions, enhanced by a brown border around
       self.screen.blit(self.bg image, (0, 0))
       self.button sound.play()
       button width = 300
       button_height = 90
       tutorial text = [
              "TUTORIAL:",
              "Arrow(UP)/Space - Jump",
              " Arrows(L,R)/(A,D) - Movement ",
               "X/(L-Shift) - Attack",
              "ESC - Exit",
       tutorial_font = pygame.font.Font(None, 40)
       tutorial height = len(tutorial text) * 40
       tutorial width = max([tutorial font.size(line)[0] for line in tutorial text])
       tutorial rect = pygame.Rect(1920 \ // \ 2 - tutorial \ width \ // \ 2 - 10, \ 1080 \ // \ 2 - tutorial \ height \ // \ 2 - 10, \ 1080 \ // \ 2 - tutorial \ height \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 10, \ 1080 \ // \ 2 - 
       background\_color = (128, 128, 128)
       border color = (139, 69, 19)
       back button = pygame.Rect(1920 // 2 - button width // 2, 800, button width, button height)
       running = True
       while running:
              for event in pygame.event.get():
                      if event.type == pygame.MOUSEBUTTONDOWN:
                             mx, my = pygame.mouse.get pos()
                              if back button.collidepoint((mx, my)):
                                     self.button sound.play()
                                    running = False
                      elif event.type == pygame.KEYDOWN:
                             if event.key == pygame.K_ESCAPE:
                                     self.button sound.play()
                                     running = False
              pygame.draw.rect(self.screen, border color, tutorial rect)
              inner_rect = tutorial_rect.inflate(-20, -20)
              pygame.draw.rect(self.screen, background color, inner rect)
              mx, my = pygame.mouse.get pos()
              back hovered = back button.collidepoint((mx, my))
              self.draw button("Back", back button, self.hover color if back hovered else self.button color)
              for i, line in enumerate(tutorial text):
                      self.draw\_text(line, tutorial\_font, self.white, 1920 // 2, 1080 // 2 - tutorial height // 2 + i
              pygame.display.update()
def adjust sfx volume(self, volume):
       Adjusts the volume of sound effects (SFX) in the game.
       Parameters:
              volume (float): The desired volume level for SFX.
       self.jump.set volume(volume)
       self.dash.set volume(volume)
       self.hit.set volume(volume)
       self.shoot.set volume(volume)
       self.ambience.set volume(volume)
def show settings(self):
       Displays the settings menu, allowing the player to adjust game settings such as sound volume.
```

```
self.button sound.play()
pygame.mixer.music.load('data/music.wav')
pygame.mixer.music.play(loops=-1)
self.jump = pygame.mixer.Sound('data/sfx/jump.wav')
self.dash = pygame.mixer.Sound('data/sfx/dash.wav')
self.hit = pygame.mixer.Sound('data/sfx/hit.wav')
self.shoot = pygame.mixer.Sound('data/sfx/shoot.wav')
self.ambience = pygame.mixer.Sound('data/sfx/ambience.wav')
button width, button height = 300, 90
volume slider width, volume slider height = 200, 20
sfx slider width, sfx slider height = 200, 20
volume_slider_x = 1920 // 2 - volume_slider_width // 2
sfx slider x = 1920 // 2 - sfx slider width // 2
volume slider y, sfx slider y = 400, 600
music volume, sfx volume = 0.5, 0.5
is_dragging_music, is_dragging_sfx = False, False
hover color = (130, 130, 130)
volume slider rect = pygame.Rect(volume slider x, volume slider y, volume slider width, volume slider he
sfx_slider_rect = pygame.Rect(sfx_slider_x, sfx_slider_y, sfx_slider_width, sfx_slider_height)
running = True
while running:
    self.screen.blit(self.bg image, (0, 0))
    pygame.draw.rect(self.screen, self.button_color, (volume_slider_x, volume_slider_y, volume slider wi
    pygame.draw.rect(self.screen, self.white, (volume slider x + music volume * volume slider width - 5,
    self.draw text("Music Volume", self.font custom, self.white, 1920 // 2, 300)
    pygame.draw.rect(self.screen, self.button color, (sfx slider x, sfx slider y, sfx slider width, sfx
    pygame.draw.rect(self.screen, self.white, (sfx slider x + sfx volume * sfx slider width - 5, sfx sli
    self.draw text("SFX Volume", self.font custom, self.white, 1920 // 2, 500)
    mx, my = pygame.mouse.get pos()
    back button rect = pygame.Rect(1920 // 2 - button width // 2, 800, button width, button height)
    is hovering back = back button rect.collidepoint((mx, my))
    back button color = hover color if is hovering back else self.button color
    self.draw button("Back", back button rect, back button color)
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
        elif event.type == pygame.MOUSEBUTTONDOWN:
            if back button rect.collidepoint(event.pos):
                self.button sound.play()
                pygame.mixer.music.stop()
                running = False
            elif volume slider rect.collidepoint(event.pos):
                is dragging music = True
            elif sfx_slider_rect.collidepoint(event.pos):
                is dragging sfx = True
        elif event.type == pygame.MOUSEBUTTONUP:
            is_dragging_music = False
            is_dragging_sfx = False
        elif event.type == pygame.MOUSEMOTION:
            if is dragging music:
                mouse_x, _ = event.pos
                music volume = max(0, min(1, (mouse x - volume slider x) / volume slider width))
                pygame.mixer.music.set volume(music volume)
            if is dragging sfx:
                mouse_x, _ = event.pos
                sfx_volume = max(0, min(1, (mouse_x - sfx_slider_x) / sfx_slider_width))
                self.adjust_sfx_volume(sfx_volume)
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K ESCAPE:
                self.button_sound.play()
                pygame.mixer.music.stop()
                running = False
    pygame.display.update()
```

....

Main game class, responsible for initializing the game environment, loading assets, and managing the game loop, including rendering and updating game states.

```
def _
     init (self, start level=0):
    Initializes the game, setting up the display, loading assets, and preparing the game environment.
    self.bg image = pygame.image.load(os.path.join('data', 'pngs', 'bg.png')).convert alpha()
    self.font custom = pygame.font.Font(None, 70)
    self.white = (255, 255, 255)
    self.button color = (155, 155, 155)
    self.hover color = (130, 130, 130)
    pvgame.init()
    pygame.display.set caption('JUMPY')
    self.screen = pygame.display.set_mode((1920, 1080))
    self.display = pygame.Surface((640, 360))
    self.clock = pygame.time.Clock()
    self.movement = [False, False]
    self.button_sound = pygame.mixer.Sound('data/sfx/click.wav')
    self.button_sound.set_volume(0.2)
    self.assets = {
        'decor': load pngs('tiles/decor'),
        'grass': load_pngs('tiles/grass'),
        'sky': load pngs('tiles/sky'),
        'large decor': load pngs('tiles/large decor'),
        'stone': load pngs('tiles/stone'),
        'player': load_image('entities/player.png'), 'background': load_image('background.png'),
        'background2': load image('background2.png'),
        'background3': load image('background3.png'),
        'clouds': load pngs('clouds'),
        'enemy/idle': Animation(load_pngs('entities/enemy/idle'), img_dur=6),
        'enemy/run': Animation(load pngs('entities/enemy/run'), img dur=4),
        'player/idle': Animation(load pngs('entities/player/idle'), img dur=6),
        'player/run': Animation(load pngs('entities/player/run'), img dur=4),
        'player/jump': Animation(load pngs('entities/player/jump')),
        'player/attack': Animation(load pngs('entities/player/attack'), img dur=1),
        'player/attack2': Animation(load pngs('entities/player/attack2'), img dur=1),
        'player/attack3': Animation(load_pngs('entities/player/attack3'), img_dur=1),
        'player/wall slide': Animation(load pngs('entities/player/wall slide')),
        'particle/leaf': Animation(load pngs('particles/leaf'), img dur=20, loop=False),
        'particle/particle': Animation(load pngs('particles/particle'), img dur=6, loop=False),
        'weapon1': load image('weapon1.png'),
        'fireball': load image('fireball.png'),
        'player/hit': Animation(load pngs('entities/player/hit'), img dur=5)
    self.sfx = {
        'jump': pygame.mixer.Sound('data/sfx/jump.wav'),
        'dash': pygame.mixer.Sound('data/sfx/dash.wav'),
        'hit': pygame.mixer.Sound('data/sfx/hit.wav'),
        'shoot': pygame.mixer.Sound('data/sfx/shoot.wav'),
        'ambience': pygame.mixer.Sound('data/sfx/ambience.wav'),
    }
    self.sfx['ambience'].set_volume(0.4)
    self.sfx['shoot'].set volume(0.4)
    self.sfx['hit'].set volume(0.15)
    self.sfx['dash'].set volume(0.3)
    self.sfx['jump'].set volume(0.4)
    pygame.mixer.music.load('data/music.wav')
    pygame.mixer.music.play(loops=-1)
    self.clouds = Clouds(self.assets['clouds'], count=16)
    self.player = Player(self, (50, 50), (8, 15))
    self.tilemap = Tilemap(self, tile_size=16)
    self.level = 0
    self.load level(self.level)
    self.fireball_hits = 0
    self.total enemies = 0
    self.enemies_killed = 0
    self.lifes = 2
    self.level = start level
    self.load level(self.level)
    self.button color = (155, 155, 155)
    self.exit button color = (255, 0, 0)
    self.exit_button_hover_color = (200, 0, 0)
    self.hover\_color = (130, 130, 130)
```

```
def draw text(self, text, font, color, x, y):
   Draws specified text on the screen at the given position. This method is needed for drawing button text.
    Parameters:
       text (str): The text to display.
       font (pygame.font.Font): The font used for the text.
       color (tuple): The color of the text.
       x (int): The x position of the text.
       y (int): The y position of the text.
    text surface = font.render(text, True, color)
    text_rect = text_surface.get_rect(center=(x, y))
    self.screen.blit(text surface, text rect)
def draw button(self, text, rect, color):
   Draws a button with the specified text and color. This method mimics the one from the Menu class.
   Parameters:
       text (str): The text displayed on the button.
       rect (pygame.Rect): The rectangle defining the button's position and size.
       color (tuple): The color of the button.
   pygame.draw.rect(self.screen, color, rect, border radius=20)
   self.draw text(text, self.font custom, self.white, rect.centerx, rect.centery)
def ingame menu(self):
    11 11 11
    Displays the in-game menu, allowing the player to resume or exit to the main menu.
    menu active = True
    while menu active:
        self.screen.blit(self.bg image, (0, 0))
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            elif event.type == pygame.MOUSEBUTTONDOWN:
                mx, my = pygame.mouse.get_pos()
                if resume_button.collidepoint((mx, my)):
                    self.button sound.play()
                    pygame.mixer.music.unpause()
                    return
                elif exit button.collidepoint((mx, my)):
                    self.button sound.play()
                    return 'exit'
        mx, my = pygame.mouse.get pos()
        button width, button height = 300, 90
        resume button = pygame.Rect(1920 // 2 - button width // 2, 400, button width, button height)
        exit_button = pygame.Rect(1920 // 2 - button_width // 2, 500, button_width, button_height)
        pygame.draw.rect(self.screen, self.button_color, resume_button, border_radius=20)
        pygame.draw.rect(self.screen, self.exit_button_color, exit_button, border_radius=20)
        if resume_button.collidepoint((mx, my)):
            pygame.draw.rect(self.screen, self.hover color, resume button, border radius=20)
        if exit button.collidepoint((mx, my)):
            pygame.draw.rect(self.screen, self.exit button hover color, exit button, border radius=20)
        self.draw text("Resume", self.font custom, self.white, resume button.centerx, resume button.centery)
        self.draw_text("Exit to Menu", self.font_custom, self.white, exit_button.centerx, exit_button.center
        pygame.display.flip()
        self.clock.tick(60)
def game over screen(self):
   Displays the game over screen and provides an option to return to the main menu.
   running = True
    while running:
        self.screen.blit(self.bg image, (0, 0))
        self.draw_text("GAME OVER", self.font_custom, self.white, 1920 // 2, 400)
        mx, my = pygame.mouse.get pos()
```

```
button width, button height = 300, 90
       back button = pygame.Rect(1920 // 2 - button width // 2, 500, button width, button height)
        if back button.collidepoint((mx, my)):
           pygame.draw.rect(self.screen, self.exit button hover color, back button, border radius=20)
           self.draw text("EXIT", self.font custom, self.white, back button.centerx, back button.centery)
       else:
           self.button sound.play()
           pygame.draw.rect(self.screen, self.exit button color, back button, border radius=20)
            self.draw text("EXIT", self.font custom, self.white, back button.centerx, back button.centery)
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                svs.exit()
           elif event.type == pygame.MOUSEBUTTONDOWN:
                if back button.collidepoint((mx, my)):
                    self.button_sound.play()
                    return
       pygame.display.flip()
       self.clock.tick(60)
def load level(self, map id):
   Loads a game level based on the specified map ID, setting up the environment, entities, and objects.
   Parameters:
      map_id (int): The identifier for the map to load.
   self.tilemap.load('data/maps/' + str(map_id) + '.json')
   self.enem = []
   self.leaf spawnerss = []
   for tree in self.tilemap.extract([('large_decor', 1), ('large_decor', 8), ('large_decor', 9), ('large_de
        self.leaf_spawnerss.append(pygame.Rect(1 + tree['pos'][0], 4 + tree['pos'][1], 30, 10))
   self.enem = []
   for spawners in self.tilemap.extract([('spawners', 0), ('spawners', 1)]):
        if spawners['variant'] == 0:
           self.player.pos = spawners['pos']
            self.player.air time = 0
       else:
           self.enem.append(Enemy(self, spawners['pos'], (8, 11)))
            self.total_enemies = len(self.enem)
            self.enemies killed = 0
   print(f"Total enemies loaded for level {map id}: {len(self.enem)}")
   self.fireball = []
   self.particles = []
   self.scroll = [0, 0]
   self.dead = 0
def run(self):
   Main game loop, handling events, updating game states, rendering the game world, and managing transition
   self.sfx['ambience'].play(-1)
   pygame.mixer.music.play(loops=-1)
   while True:
        if self.level == 0:
            self.display.blit(self.assets['background'], (0, 0))
        elif self.level == 1:
           self.display.blit(self.assets['background2'], (0, 0))
        elif self.level == 2:
            self.display.blit(self.assets['background3'], (0, 0))
       if not len(self.enem):
            if self.level < 2:</pre>
                self.level += 1
                self.total enemies = len(self.enem)
                self.load level(self.level)
            else:
               pygame.mixer.music.stop()
                self.game_over_screen()
                break
```

```
if self.dead:
    self.dead += 1
    if self.dead > 40:
        self.load level(self.level)
self.scrol1[0] += (self.player.rect().centerx - self.display.get width() / 2 - self.scrol1[0])
self.scroll[1] += (self.player.rect().centery - self.display.get_height() / 2 - self.scroll[1]) /
render_scroll = (int(self.scroll[0]), int(self.scroll[1]))
for rect in self.leaf spawnerss:
    if random.random() * 19999 < rect.width * rect.height:</pre>
        pos = (rect.x + random.random() * rect.width, rect.y + random.random() * rect.height)
        self.particles.append(Particle(self, 'leaf', pos, velocity=[-0.1, 0.2], frame=random.randint
self.clouds.update()
self.clouds.render(self.display, offset=render scroll)
self.tilemap.render(self.display, offset=render_scroll)
for enemy in self.enem.copy():
    kill = enemy.update(self.tilemap, (0, 0))
    enemy.render(self.display, offset=render scroll)
    if kill:
        self.enem.remove(enemy)
        self.enemies killed += 1
if not self.dead:
    self.player.update(self.tilemap, (self.movement[1] - self.movement[0], 0))
    self.player.render(self.display, offset=render scroll)
for fireball in self.fireball.copy():
    fireball[0][0] += fireball[1]
    fireball[2] += 1
    img = self.assets['fireball']
    self.display.blit(img, (fireball[0][0] - img.get_width() / 2 - render_scroll[0], fireball[0][1]
    if self.tilemap.solid check(fireball[0]):
        self.fireball.remove(fireball)
    elif fireball[2] > 360:
        self.fireball.remove(fireball)
    if abs(self.player.dashing) < 50:</pre>
        if self.player.rect().collidepoint(fireball[0]):
            self.fireball.remove(fireball)
            self.fireball hits += 1
            self.lifes -= 1
            self.sfx['hit'].play()
            self.player.set action('hit')
            if self.fireball hits >= 2:
                self.dead = True
                self.lifes = 2
                self.fireball hits = 0
                for i in range(30):
                    angle = random.random() * 3.14 * 2
                    speed = random.random() * 5
                    self.particles.append(Particle(self, 'particle', self.player.rect().center, velc
for particle in self.particles.copy():
    kill = particle.update()
particle.render(self.display, offset=render_scroll)
    if particle.type == 'leaf':
        particle.pos[0] += math.sin(particle.animation.frame * 0.1) * 0.3
    if kill:
        self.particles.remove(particle)
self.screen.blit(pygame.transform.scale(self.display, self.screen.get_size()), (0, 0))
self.clock.tick(60)
for event in pygame.event.get():
    if event.type == pygame.QUIT:
       pygame.guit()
        sys.exit()
    if event.type == pygame.KEYDOWN:
        if event.key == pygame.K_ESCAPE:
            pygame.mixer.music.pause()
            action = self.ingame menu()
```

```
if action == 'exit':
                            return
                     f event.key == pygame.K LEFT or event.key == pygame.K a:
                        self.movement[0] = True
                       event.key == pygame.K RIGHT or event.key == pygame.K d:
                        self.movement[1] = True
                    if event.key == pygame.K UP or event.key == pygame.K SPACE:
                        if self.player.jump():
                            self.sfx['jump'].play()
                    if event.key == pygame.K_x or event.key == pygame.K_LSHIFT:
                        self.player.dash()
                if event.type == pygame.KEYUP:
                    if event.key == pygame.K_LEFT or event.key == pygame.K_a:
                        self.movement[0] = False
                    if event.key == pygame.K_RIGHT or event.key == pygame.K_d:
                        self.movement[1] = False
            font = pygame.font.Font(None, 72)
            enemy_info_text = f'Kills: {self.enemies_killed}/{self.total_enemies}'
            text_surface = font.render(enemy_info_text, True, (255, 255, 255))
            text rect = text surface.get rect(topright=(1880, 10))
            self.screen.blit(text_surface, text_rect)
            font = pygame.font.Font(None, 72)
            life info text = f'Lives: {self.lifes}/2'
            text surface = font.render(life info text, True, (255, 255, 255))
            text rect = text surface.get rect(topright=(1880, 80))
            self.screen.blit(text surface, text rect)
            font = pygame.font.Font(None, 72)
            life_info_text = f'LEVEL: {self.level + 1}'
            text surface = font.render(life info text, True, (255, 255, 255))
            text_rect = text_surface.get_rect(topright=(230, 10))
            self.screen.blit(text surface, text_rect)
            if self.dead == 1:
                self.lifes = 2
                self.fireball hits = 0
            if self.enemies killed + 1:
               pygame.display.update()
            else:
               pass
Entry point of the game. This script initializes the main menu and the game itself. The game loop continues to
execute, allowing the player to return to the main menu and restart the game as often as desired. The loop
terminates only when a specific 'action' (e.g., 'exit') is returned from the game, signaling the script to end.
The loop follows these steps:
1. Initializes and displays the game's main menu.
2. Waits for the user to navigate the menu and start the game.
3. Initializes the game and starts the game loop.
4. The game loop runs until an 'exit' action is returned.
5. If 'exit' is not returned, the loop starts over from step 1, allowing the player to play again or quit.
   __name__ == "__main__":
while True:
       menu = Menu()
       menu.run()
       game = Game()
       action = game.run()
       if action != 'exit':
```

if __name_

break