

2. Разумников С.В. Разработка программного обеспечения для построения агрегированных рейтингов на основе метода порогового агрегирования / С.В. Разумников // Вестник ВГУ, серия: Системный анализ и информационные технологии. – 2021. – № 2. – С. 138–152.

3. Разумников С.В. Планирование развития облачной стратегии на основе применения многокритериальной оптимизации и метода STEM / С.В. Разумников // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2020. – Т. 23, № 1. – С. 53–61.

4. Разумников С.В. Модели, алгоритмы и программное обеспечение поддержки принятия стратегических решений к переходу на облачные технологии: монография / С.В. Разумников. – Томск : Изд-во Томского политехнического университета, 2020. – 176 с.

РАЗРАБОТКА ИГР НА PYTHON

И.Н. Газмарин², ученик 11 класса

Научный руководитель: Разумников С.В.^{а1}, к.т.н. доц.

¹Юргинский технологический институт (филиал)

Национального исследовательского Томского политехнического университета

652055, Кемеровская обл., г. Юрга, ул. Ленинградская, 26

²МБОУ «Гимназия города Юрги», 652057, Кемеровская обл., г. Юрга, ул. Московская, 48

E-mail: "razumnikov@tpu.ru

Аннотация: В статье рассказывается о возможностях создания игра на языке программирования Python. Приведен пример кода разработки игры игра "Прыгающий мячик" с использованием библиотеки Pygame.

Ключевые слова: игра, Python, разработка, библиотека Pygame.

Abstract: The article talks about the possibilities of creating a game in the Python programming language. An example of code for developing the game "Bouncing Ball" using the Pygame library is given.

Keywords: game, Python, development, Pygame library.

Введение

Разработка игр – это увлекательный и творческий процесс, который требует знаний в области программирования, дизайна и математики. Python – это язык программирования, который широко используется для разработки игр благодаря своей простоте и гибкости. В данной статье мы рассмотрим, как можно использовать Python для создания игр.

Основные понятия

Python – это язык программирования, который используется для создания различных приложений, включая игры.

Pygame – это библиотека для Python, которая предоставляет набор инструментов для создания игр.

Игровой цикл – это последовательность событий, которые происходят в игре.

Преимущества разработки игр на Python:

1. Простота использования Python является одним из самых простых языков программирования, что делает его идеальным для начинающих разработчиков игр.

2. Большое сообщество разработчиков. Python имеет большое сообщество разработчиков, которые готовы помочь и поделиться своим опытом.

3. Большое количество библиотек и фреймворков. Python имеет большое количество библиотек и фреймворков, которые могут упростить разработку игр.

Примеры использования Python для разработки игр:

1. Создание 2D-игр

2. Python и Pygame могут быть использованы для создания 2D-игр, таких как аркады, платформеры и головоломки.

3. Создание 3D-игр

4. Python может быть использован для создания 3D-игр с помощью библиотек, таких как Panda3D или PyOpenGL.

3. Создание многопользовательских игр

Python может быть использован для создания многопользовательских игр с помощью библиотек, таких как Twisted или Tornado.

Примером разработки простой игры на Python с использованием библиотеки Pygame может служить игра «Прыгающий мячик».

Вот пример кода на языке Python, который реализует такую игру:

```
import pygame
import random
# Инициализация Pygame
pygame.init()
# Создание экрана
screen = pygame.display.set_mode((640, 480))
pygame.display.set_caption('Прыгающий мячик')
# Создание спрайта мячика
ball_image = pygame.image.load('ball.png')
ball = pygame.transform.scale(ball_image, (32, 32))
ball_rect = ball.get_rect()
# Создание спрайта платформы
platform_image = pygame.image.load('platform.png')
platform = pygame.transform.scale(platform_image, (64, 32))
platform_rect = platform.get_rect(center=(320, 240))
# Создание спрайта препятствия
obstacle_image = pygame.image.load('obstacle.png')
obstacle = pygame.transform.scale(obstacle_image, (32, 32))
obstacle_rect = obstacle.get_rect(center=(320, 360))
# Создание спрайта кнопки "Play"
play_image = pygame.image.load('play.png')
play = pygame.transform.scale(play_image, (64, 32))
play_rect = play.get_rect(center=(320, 448))
# Функция для движения мячика
def move_ball(ball_rect, platform_rect):
    ball_rect.y += 2
    if ball_rect.colliderect(platform_rect):
        ball_rect.y = platform_rect.top - 1
    if ball_rect.top > 480:
        return False
    return True
# Функция для движения препятствия
def move_obstacle(obstacle_rect):
    if obstacle_rect.left < -32:
        obstacle_rect.left = 640
    obstacle_rect.left -= 2
# Функция для движения платформы
def move_platform(platform_rect):
    if platform_rect.left < -64:
        platform_rect.left = 640
    platform_rect.left -= 1
# Основной цикл игры
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        if event.type == pygame.MOUSEBUTTONDOWN:
```

```
if play_rect.collidepoint(event.pos):
    running = True
    play_rect.center = (320, 448)
else:
    running = False
screen.fill((0, 0, 0))
screen.blit(platform, platform_rect)
screen.blit(obstacle, obstacle_rect)
if running:
    screen.blit(ball, ball_rect)
    if move_ball(ball_rect, platform_rect):
        ball_rect.y += 2
    move_obstacle(obstacle_rect)
    move_platform(platform_rect)
pygame.display.flip()
pygame.quit()
```

В этом примере мы создаем игру, в которой мячик прыгает по платформе и избегает препятствия. Код использует библиотеку Pygame для создания экрана, загрузки изображений и обработки событий мыши. Функции `move_ball`, `move_obstacle` и `move_platform` отвечают за движение объектов на экране. Основной цикл игры проверяет события мыши и обновляет экран.

Заключение

Разработка игр на Python – это простой и эффективный способ создания игр. Python имеет большое сообщество разработчиков, большое количество библиотек и фреймворков, что делает его идеальным для разработки игр. Примеры использования Python для разработки игр включают создание 2D-игр, 3D-игр и многопользовательских игр.

Список использованных источников:

1. Разумников С.В. Алгоритм и программное обеспечение для построения агрегированных рейтингов / С.В. Разумников // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2021. – Т. 24, № 2. – С. 39–44.
2. Разумников С.В. Разработка программного обеспечения для построения агрегированных рейтингов на основе метода порогового агрегирования / С.В. Разумников // Вестник ВГУ, серия: Системный анализ и информационные технологии. – 2021. – № 2. – С. 138–152.
3. Разумников С.В. Планирование развития облачной стратегии на основе применения многокритериальной оптимизации и метода STEM / С.В. Разумников // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2020. – Т. 23, № 1. – С. 53–61.
4. Разумников С.В. Модели, алгоритмы и программное обеспечение поддержки принятия стратегических решений к переходу на облачные технологии : монография / С.В. Разумников. – Томск : Изд-во Томского политехнического университета, 2020. – 176 с.

ПРОГРАММИРОВАНИЕ МАТРИЦЫ ПОПАРНЫХ СРАВНЕНИЙ НА ЯЗЫКЕ C#

К.К. Первалов², ученик 11 класса

Научный руководитель: Разумников С.В.^{а1}, к.т.н. доц.

¹Юргинский технологический институт (филиал)

Национального исследовательского Томского политехнического университета

652055, Кемеровская обл., г. Юрга, ул. Ленинградская, 26

²МБОУ «Гимназия города Юрги», 652057, Кемеровская обл., г. Юрга, ул. Московская, 48

E-mail: ^arazumnikov@tpu.ru

Аннотация: В статье рассматривается инструмент в области принятия решений – матрица попарных сравнений. Предложен пример программной реализации на языке C#. Представлено диалоговое окно программы для проведения расчетов.