МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГУ»)

Факультет компьютерных наук

Кафедра информационных технологий управления

Создание игры «Arkanoid» на языке Python

Курсовая работа

09.03.02 Информационные системы и технологии

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Зав. кафедрой | \_\_\_\_\_\_\_\_\_\_\_ | Матвеев М. Г. | д.т.н., профессор | \_\_.\_\_.2023 |
| Обучающийся | \_\_\_\_\_\_\_\_\_\_\_ | Меркутова К.Д |  |  |
| Руководитель | \_\_\_\_\_\_\_\_\_\_\_ | Копытина Е.А. | к. т. н. старший преподаватель |  |

Воронеж 2023

Содержание

[Содержание 2](#_Toc153835635)

[Определения, обозначения и сокращения 3](#_Toc153835636)

[Введение 4](#_Toc153835637)

[1 Постановка задач 6](#_Toc153835638)

[1.1 Обзор аналогов 7](#_Toc153835639)

[1.1.1 Brick Breaker: 7](#_Toc153835640)

[1.1.2 Paddle 7](#_Toc153835641)

[1.1.3 Shatter: 7](#_Toc153835642)

[2 Анализ предметной области 8](#_Toc153835643)

[2.1 История создания игры 8](#_Toc153835644)

[2.2 Правила игры 8](#_Toc153835645)

[3 Реализация 9](#_Toc153835646)

[3.1 Средства реализации 9](#_Toc153835647)

[3.2 Логика приложения 10](#_Toc153835648)

[3.3 Реализация интерфейса 13](#_Toc153835649)

[Заключение 14](#_Toc153835650)

[Список использованных источников 15](#_Toc153835651)

Определения, обозначения и сокращения

В настоящем отчёте о курсовой работе применяют следующие термины с соответствующими определениями:

Рефакторинг - это процесс изменения кода программы с целью улучшения его структуры, читаемости, поддерживаемости и эффективности, без изменения её внешнего поведения. В основном, рефакторинг включает в себя переименование переменных и функций, разделение длинных функций на более маленькие, устранение дублирования кода, улучшение комментариев и документации [3].

Введение

Сейчас время компьютерных технологий и всего, что с этим связано - роботостроение, ракетостроение, создание игр, медицина и многое другое. Везде необходимо программирование. Люди, которые им владеют, называются программистами. Они создают игры, управляют роботами и космическими кораблями. Поэтому такие специалисты будут нужны всегда. В будущем свою жизнь я тоже хотел бы связать с программированием, поэтому уже второй год во время летних каникул обучался в школе программирования «Алгоритмика».

Арканоид - это классическая аркадная игра, которая была разработана компанией Taito в 1986 году. Она была создана как продолжение другой популярной игры от Taito под названием Breakout, выпущенной в 1976 году. Идея создания Arkanoid возникла у разработчиков, чтобы добавить новые элементы и улучшения к классическому геймплею Breakout. Создатели стремились сделать игру более привлекательной и разнообразной для игроков. Основная концепция Arkanoid заключалась в том, чтобы управлять платформой, отскакивающей мячик, и разрушать блоки, расположенные в верхней части экрана. Разработчики добавили различные типы блоков и специальные предметы, которые могли быть собраны игроком и давали дополнительные преимущества.

На сегодняшний день компьютерные игры занимают важное место в жизни человека. Они могут использоваться в качестве развлекательного времяпрепровождения или в обучающих целях, а профессия разработчика компьютерных игр является перспективной и хорошо оплачиваемой. Именно поэтому важно понять процесс алгоритмов и программировании.

Разработка игр в жанре Arkanoid до сих пор имеет актуальность и пользуется популярностью. Хотя оригинальная игра была выпущена более 30 лет назад, она оказала огромное влияние на индустрию видеоигр и стала классикой.

Сегодня множество разработчиков продолжают создавать новые версии и вариации игры Arkanoid, добавляя новые элементы геймплея, улучшенные графику и множество уровней. Арканоид вдохновляет творческие идеи разработчиков, и это позволяет игре оставаться актуальной и интересной для игроков всех возрастов.

Таким образом, разработка и модернизация игры Arkanoid продолжают быть актуальными и интересными для игроков и разработчиков. Эта классическая игра остается важной частью истории видеоигр и продолжает вдохновлять новые создания.

1. Постановка задач

Игровое поле представляет собой прямоугольную область, разделенную на ячейки. В верхней части поля находится горизонтальная платформа, которую управляет игрок с помощью клавиатуры или мыши. В нижней части поля расположены несколько рядов кирпичей, которые должны быть разрушены путем отбивания мячика к платформе.

Задачи разработчика включают создание игрового поля, платформы, мячика и кирпичей, а также программировка логики взаимодействия между ними. Также может потребоваться создание уровней с разными сложностями и внедрение графического и звукового оформления игры.

**Основные задачи:**

* Ознакомиться с теорией;
* Создать игровое поле: разработка механизма, который будет отображать игровое поле и его состояние, а также определять какие блоки разрушены, а какие целы;
* Создание платформы-ракетки и шарика;
* Программирование движения платформы и шарика;
* Программирование окончания игры (выигрыш, проигрыш);
* Подведение итогов курсовой работы.

1. Обзор аналогов
2. Brick Breaker:

Это один из самых популярных аналогов Арканоида для мобильных устройств. Он предлагает похожий геймплей, где ты отбиваешь мячик, чтобы разбить кирпичики. Однако некоторые версии могут иметь ограниченное количество уровней или повторяющиеся уровни, что может привести к утомлению.

1. Paddle

Это тоже замечательный аналог Арканоида с уникальным дизайном и захватывающими уровнями. Однако некоторые игроки отмечают, что управление может быть несколько неудобным, что может повлиять на игровой процесс.

1. Shatter:

Эта игра представляет собой модернизированную версию Арканоида с улучшенной графикой и специальными эффектами. Она также включает в себя различные режимы игры. Одним из потенциальных минусов может быть то, что некоторые игроки считают ее слишком сложной, особенно на более высоких уровнях.

Каждый из этих аналогов имеет свои особенности и потенциальные минусы, но все они предлагают аналогичный геймплей Арканоида.

1. Анализ предметной области
2. История создания игры

«Арканоид» изначально разрабатывался для специализированных автоматов. Официальная дата выхода — 1986 год. Разработана японской игровой корпорации Taito, руководимой тогда советским гражданином Михаилом Коганом. Сегодня же классическая версия игры насчитывает более двухсот приложений. Но началось все значительно раньше.

В 1958 году физик Уильям Хигинботам решил доказать своим друзьям, что наука может быть веселой. Из различных радиодеталей создатель атомной бомбы собрал некое устройство и подключил к осциллографу.

На экране прибора появились ракетка и мячик. Создателю пришлось немедленно ограничивать доступ к своему детищу: появилось слишком много желающих протестировать новинку.

А в 1976 году Стиву Джобсу, будущему соучредителю Apple, поручили создать новую игру Breakout. Ее цель — бить мячиком по цветным блокам, пока не останется ни одного. Важно при этом, чтобы мяч не проходил мимо ракетки. Все эти игры можно считать прародителями будущего “Арканоида”.

1. Правила игры

Игрок контролирует небольшую платформу-ракетку, которую можно передвигать горизонтально от одной стенки к другой, подставляя ее под шарик, предотвращая его падение вниз. Удар шарика по кирпичу приводит к разрушению кирпича. После того как все кирпичи на данном уровне уничтожены, происходит переход на следующий уровень, с новым набором кирпичей. Есть и некоторое разнообразие: определённые кирпичи нужно ударять несколько раз, удар по некоторым кирпичам приводят к выпадению из них капсул-призов – приз активируется, если поймать такую капсулу ракеткой.

1. Реализация
2. Средства реализации

При создании игры используются такие библиотеки как:

* Tkinter, эта библиотека предустановлена на большинстве версиях Python и используется для создания самого оконного приложения;
* Time, в нашем случае будет использоваться, чтобы задать скорость мячу и платформе. Для установки зайдите в командную строку от имени администратора и напишите: pip install time проверьте что pip у вас установлен;
* Random, в нашем случае будет использоваться, что бы мяч отскакивал в разных направлениях. Эта библиотека так же предустановлена на всех версиях Python;
* Pygame, используется для создания графического интерфейса нашей игры. Эту библиотеку нужно скачать через командную строку, прописав: pip install pygame [1];

Среди преимуществ языка программирования Python можно выделить:

* Простота и читаемость кода;
* Быстрая разработка;
* Большая библиотека и экосистема;
* Кроссплатформенность.

PyCharm - это одна из самых популярных сред разработки для языка программирования Python. Она предоставляет полноценную среду разработки с широким набором инструментов и функций, которые упрощают процесс написания, отладки и тестирования кода. В ней присутствует подсветка синтаксиса, авто дополнение, инструменты рефакторинга, возможность быстрого перехода к определению функций и классов. PyCharm также позволяет легко создавать и управлять проектами [2].

1. Логика приложения

Для реализации кода была импортирована библиотека pygame и использован стандартный шаблон, который создает нам пустое черное окно заданного разрешения.

Нужно создать основное поле игры (Листинг 1).

Листинг 1 – Создание поля

WIDTH, HEIGHT = 1200, 800

fps = 60

Далее необходимо создать шарик и платформу-ракетку (Листинг 2, листинг 3)

Листинг 2 – Создание шарика

ball\_radius = 20

ball\_speed = 6

ball\_rect = int(ball\_radius \* 2 \*\* 0.5)

ball = pygame.Rect(rnd(ball\_rect, WIDTH - ball\_rect), HEIGHT // 2, ball\_rect, ball\_rect)

dx, dy = 1, -1

Листинг 3 – Создание платформы-ракетки

paddle\_w = 330

paddle\_h = 35

paddle\_speed = 15

paddle = pygame.Rect(WIDTH // 2 - paddle\_w // 2, HEIGHT - paddle\_h - 10, paddle\_w, paddle\_h)

После нужно запрограммировать генерацию блоков, которые шарик будет разрушать (Листинг 4).

Листинг 4 – Программирование генерации блоков

block\_list = [pygame.Rect(10 + 120 \* i, 10 + 70 \* j, 100, 50)

for i in range(10)

for j in range(4)]

color\_list = [(rnd(30, 256),

rnd(30, 256),

rnd(30, 256))

for i in range(10)

for j in range(4)]

Также для того, чтобы платформа двигалась игра заработала, следует задать движение платформе по оси X, а движение шарика по оси X и Y (Листинг 5).

Листинг 5 – Задание движения

def detect\_collision(dx, dy, ball, rect):

if dx > 0:

delta\_x = ball.right - rect.left

else:

delta\_x = rect.right - ball.left

if dy > 0:

delta\_y = ball.bottom - rect.top

else:

delta\_y = rect.bottom - ball.top

if abs(delta\_x - delta\_y) < 10:

dx, dy = -dx, -dy

elif delta\_x > delta\_y:

dy = -dy

elif delta\_y > delta\_x:

dx = -dx

return dx, dy

Для того, чтобы игра закончилась следует написать код для завершения игры - выигрыш или проигрыш (Листинг 6).

Листинг 6 – Окончание игры)

if ball.bottom > HEIGHT:

print('GAME OVER!')

exit()

elif not len(block\_list):

print('WIN!!!')

exit()

1. Реализация интерфейса

Начав игру, пользователю будет представлено игровое окно, в котором присутствует само игровое поле, платформа ракетка, шарик, цветные фигуры.

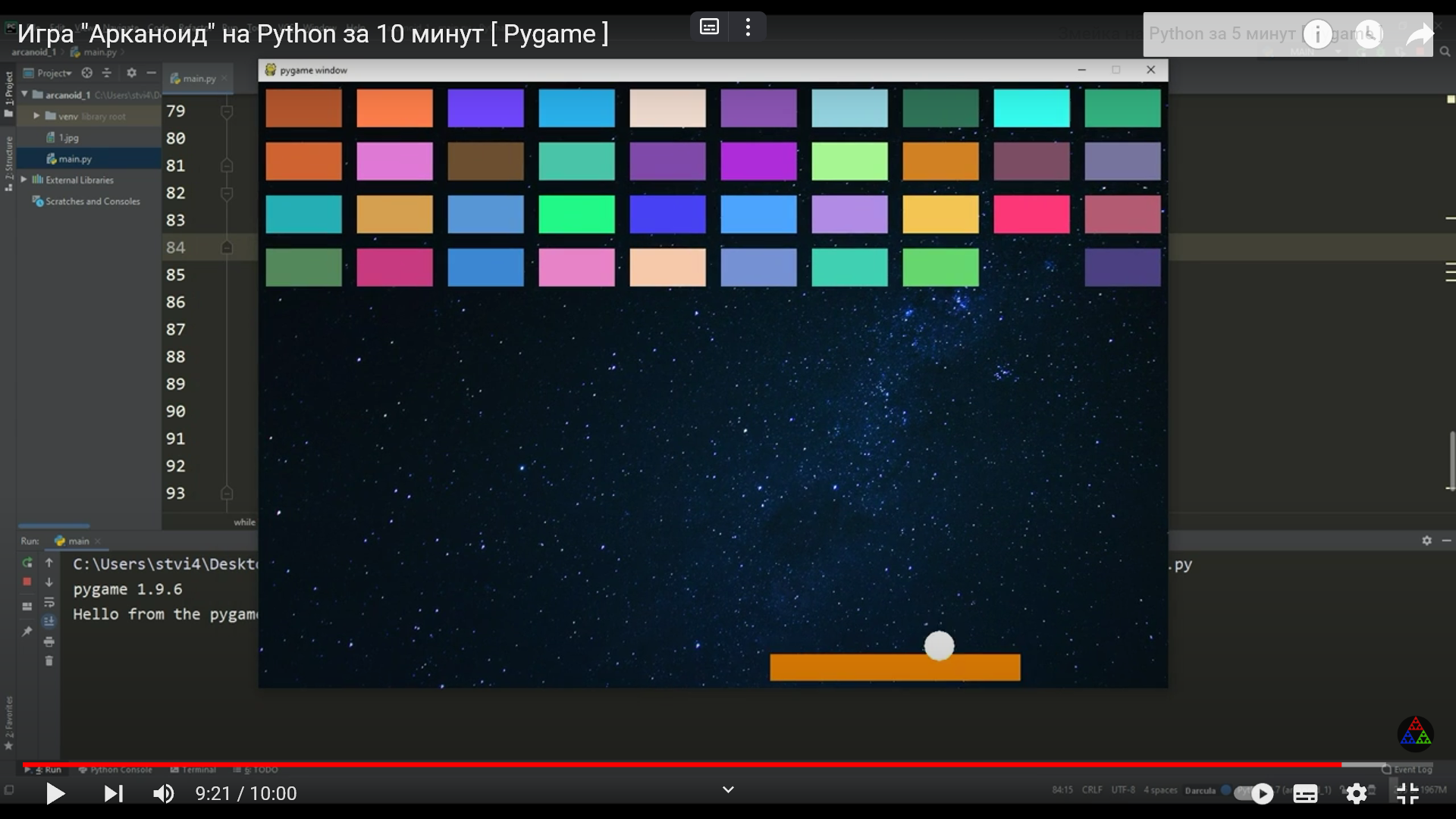
Интерфейс данной игры представлен на рисунке 1.

Рисунок 1 – интерфейс

Передвигая платформу-ракетку, игрок может отбивать шарик, который, сталкиваясь с блоками разрушает их. В случайном из блоков может выпасть какой-либо бонус. Его можно поймать с помощью платформы-ракетки. По окончании игры появится окно с проигрышем, если шарик не отобьется платформой-ракеткой, а с выигрышем, если все блоки будут успешно разрушены.

Заключение

В данном курсовом проекте была реализована игра «Арканоид» при помощи языка программирования Python. Пользовательский интерфейс был спроектирован с использованием библиотеки Pygame. В процессе выполнения курсовой работы были изучены принципы работы библиотеки Pygame, основные принципы игрового процесса классический игры «Арканоид», были разработаны игра и её пользовательский интерфейс. В перспективе можно добавить различные модификации игры, к примеру: новые блоки и их свойства, режимы игры, возможность самостоятельно выбирать уровень сложности.

Список использованных источников

1. Game Maker. – Текст: электронный // game-maker [сайт] - 2008-2020. – URL: https://game-maker.ru/ (дата обращения 25.09.23)
2. Возможности PyCharm. – Текст: электронный // JetBrains [сайт]. – URL: https://www.jetbrains.com/ru-ru/pycharm/features/ (дата обращения 18.09.2023)
3. Что такое рефакторинг кода и зачем он нужен. - Текст: электронный // Skillbox [сайт]. – URL: https://skillbox.ru/media/code/chto\_takoe\_refaktoring\_koda\_i\_zachem\_on\_nuzhen/ (дата обращения 18.09.2023)