

Автономная некоммерческая общеобразовательная организация «Физтех-лицей» имени П.Л. Капицы (АНОО «Физтех-лицей» им. П.Л. Капицы)

**XX научно-практическая конференция
«Старт в инновации»**

ИГРА «ТЕТРИС»

Выполнили: Соловьева Виолетта,
Семенова Юлия

Руководитель: Зыонг К.К.

Московская область, г. Долгопрудный
2020 г.

Оглавление

Вступление.....	3
Цель и задачи проекта.....	5
Теория.....	6
Основная часть.....	8
Выводы.....	10
Использованная литература.....	11

Вступление

Игра Тетрис имеет долгую историю: впервые она была написана в 1984 году советским программистом Алексеем Пажитовым. В то время Пажитов разрабатывал технологии искусственного интеллекта и распознавания речи, и в своей работе использовал головоломки, в частности пентамино — кирпичики разных форм, из которых складывались фигурки — одна из самых популярных головоломок того времени. Он пытался автоматизировать укладку этих фигурок: в этих опытах и появилась идея Тетриса — чтобы фигурки сами падали, а заполненные ряды исчезали.



Кирпичики Пентамино

Игра сразу же приобрела популярность и распространилась сначала по стране, а вскоре и по всему миру. Версии Тетриса создавались в Британии и США, улучшались графика и звук игры. Также Тетрис имел огромную популярность в Японии, где в то время компания Nintendo работала над созданием портативной консоли GameBoy, для которой решили разработать свою версию Тетриса.



GameBoy с тетрисом

В последующие годы Тетрис был реализован для многих устройств: от калькуляторов до карманных ПК.

Интересно, что для IBM PC, первого персонального компьютера, игра была переписана на языке Turbo Pascal простым 16-летним школьником Вадимом Герасимовым. Вдохновившись его примером, мы решили тоже попробовать свои силы и написать собственную версию игры Тетрис, используя более современный язык программирования — Python 3, и библиотеку PyGame.

Несмотря на то, что игра появилась уже довольно давно, она не теряет своей актуальности по сей день: уже в 2014 году Хэнк Роджерс, глава The Tetris Company, утверждал, что количество платных загрузок игры для мобильных телефонов превысило 425 миллионов.

Тетрис не перестает развиваться: в 2019 году Nintendo выпустили Tetris 99 – многопользовательскую версию Тетриса. Сам Алексей Пажитов уже 10 лет работает над созданием многопользовательской версии игры, но до сих пор она не была выпущена.

Цель и задачи проекта

Главной целью нашего проекта является создание собственной версии Тетриса, с авторским оформлением и музыкальным сопровождением.

Для достижения этой цели мы поставили перед собой следующие задачи:

- изучить необходимую теорию, а именно:

библиотеку PyGame и основные алгоритмы для работы с ней;

объектно-ориентированное программирование в Python.

- Придумать алгоритм для написания программы, которая будет выполнять всю необходимые функции, а именно:

при открытии программы будет появляться меню, на котором будут расположены кнопки «Играть» и «Выход»; переход между ними будет осуществляться при помощи стрелок клавиатуры;

После нажатия кнопки «Играть» начинается стандартная игра Тетрис;

Управление в игре производится стрелками. Во время игры пользователь может видеть свой рейтинг, а также лучший рейтинг;

В случае проигрыша при нажатии любой кнопки клавиатуры пользователь вернется в главное меню.

- Реализовать алгоритм на языке программирования Python 3;
- Оформить интерфейс игры, добавить музыкальное сопровождение;
- Протестировать программу, убедиться в ее работоспособности и соответствии ожидаемому результату.
- Разработку проекта будем вести на github.com, параллельно осваивая работу в нём (как обновлять изменения кода, написание кода несколькими разработчиками, различные ветвления кода, слияние ветвей и т.д.).

Теоретическая часть

- **Функции** — именованные блокам кода, предназначенные для решения одной конкретной задачи. Чтобы выполнить задачу, определенную в виде функции, вы указываете имя функции, отвечающей за эту задачу. Если задача должна многократно выполняться в программе, вам не придется заново вводить весь необходимый код; просто вызовите функцию, предназначенную для решения задачи, и этот вызов приказывает Python выполнить код, содержащийся внутри функции. [4]
- **Объектно-ориентированное программирование** по праву считается одной из самых эффективных методологий создания программных продуктов. В объектно-ориентированном программировании вы пишете классы, описывающие реально существующие предметы и ситуации, а затем создаете объекты на основе этих описаний. При написании класса определяется общее поведение для целой категории объектов. [4]
- **Функция, являющаяся частью класса, называется методом.** Все, что вы узнали ранее о функциях, также относится и к методам; единственное практическое различие — способ вызова методов. Метод `__init__()` — специальный метод, который автоматически выполняется при создании каждого нового экземпляра на базе класса. Имя метода начинается и заканчивается двумя символами подчеркивания; эта схема предотвращает конфликты имен стандартных методов Python и методов ваших классов. [4]

PyGame

- В языке Python присутствуют различные модули, такие как `random`, `math` или `time`, которые обеспечивают дополнительные функции вашим программам. Библиотека `Pygame` включает в себя набор модулей для создания графики, проигрывания звука, обработки ввода компьютерной мыши и другие вещи. [3]
- В сердце каждой игры лежит цикл, который принято называть «игровым циклом». Он запускается снова и снова, делая все, чтобы работала игра. Каждый цикл в игре называется *кадром*. В каждом кадре происходит масса вещей, но их можно разбить на три категории:
 1. **Обработка ввода (события).** Речь идет обо всем, что происходит вне игры — тех событиях, на которые она должна реагировать. Это могут быть нажатия клавиш на клавиатуре, клики мышью и так далее.
 2. **Обновление игры.** Изменение всего, что должно измениться в течение одного кадра. Если персонаж в воздухе, гравитация должна потянуть его вниз. Если два объекта встречаются на большой скорости, они должны взорваться.
 3. **Рендеринг (прорисовка).** В этом шаге все выводится на экран: фоны, персонажи, меню. Все, что игрок должен видеть, появляется на экране в нужном месте. [6]
- Почти каждая программа, созданная с использованием библиотеки `PyGame`, должна начинаться строкой
`import pygame, sys`
Это утверждение импортирует модули `pygame` и `sys`. Все необходимые функции для графики, звука и других свойств находятся в модуле `pygame`. Прежде, чем вызывать другие функции библиотеки, необходимо вызвать функцию `pygame.init()`
Далее необходимо создать окно игровой программы (объект `pygame.SURFACE`), и указать его размеры:
`pygame.display.set_mode((WIDTH, HEIGHT))`
После остается вызвать функцию:

```
pygame.display.set_caption("My Game")
```

(название программы передается как аргумент);

А далее начинается бесконечный игровой цикл *While True*, внутри которого и находится вся игра. [3]

Основная часть

Для запуска кода нам потребуются 12 файлов с различными классами (AnimatedSprite, AssetManager, Board, Config, Sprites, shapes и так далее), весь интерфейс (лист со спрайтами, требующимися для анимации в основной части игры) лежит в папке data. Также необходимы файлы с музыкальным сопровождением (для той или иной ситуации в игре) и тестовый файл с текущим рейтингом (который будет обновляться по мере достижения игроком текущего максимального рейтинга и будет оставаться неизменным при перезапуске игры). Кратко опишем принцип работы каждого из классов:

- Класс AnimatedSprite отвечает за нарезку листа с готовыми спрайтами (рисунками человечка в различных состояниях) на отдельные кадры для его дальнейшей анимации.
- Класс AssetManager отвечает за работу с файлами, которые требуются для интерфейса игры. В данном классе происходит проверка на то, существует ли файл, который мы собираемся достать из папки data, а также каждый из файлов проходит процедуру удаления фона (через встроенный в ругаме параметр colorkey), чтобы при отображении на экране фон какого-либо объекта не накладывался на заливку экрана.
- Класс Board – один из ключевых классов, отвечает за отрисовку доски, фигур на ней, обработку событий (при любом событии клавиатуры доска обновляется и перерисовывается заново). Также этот класс импортируется при проверке на то, возможен ли ход (возможно ли сдвинуть фигуру так, как этого хочет пользователь).
- Класс Config – “подручный” класс, нужен для более удобного доступа к некоторым текущим переменным и параметрам во время работы программы, состоит из нескольких функций, каждая из которых возвращает какие-либо глобальные значения.
- Класс game_config отнаследован от класса Config, передаёт в него такие глобальные значения, как ширина и высота окна, размер основного шрифта во время работы программы.
- Файл main.py - именно этот файл отвечает за запуск всей программы, именно от него идут все сообщения во все классы, запускаются те или иные функции. Состоит из двух частей – первая запускает начало программы, открывает и отрисовывает главное меню, обрабатывает события с клавиатуры. Вторая часть ссылается на класс Tetris и также обрабатывает события клавиатуры, проверяет параметр running, булево значение которого означает, нажал ли пользователь на кнопку выхода из программы и нужно ли завершить работу программы.
- Класс main_menu включает в себя звуковое оформление пускового окна, связь с дальнейшей игрой или выходом (в зависимости от действий пользователя), в основном отвечает за отрисовку пускового окна.
- В классе Next_figure происходит формирование следующей фигуры – с помощью модуля random моментально выбирается случайная форма и цвет будущей фигуры, и происходит её отрисовка в специально отведённой под это области экрана, то есть пользователь во время игры видит, какая фигура выпадет следующей.
- Класс Piece используется для поворота фигур, то есть когда пользователь нажимает на клавишу, отвечающую за поворот фигуры, происходит ссылка на данный класс и в нём происходит переворот модели фигуры (все модели фигур лежат в файле shapes.py, где фигуры представляют собой двумерные массивы, состоящие из 0 и 1, где единицы – это контур фигуры) и её отрисовка на поле.
- Класс Sprite полностью отвечает за работу анимированного спрайта, смену его действия и смену кадров для создания эффекта анимации.
- Класс Tetris – самый большой и важный класс игры, отвечает за все процессы, происходящие во время работы самого тетриса, отрисовывает фигуры, проверяет корректность хода, удаляет строки и так далее. В этом классе импортируются почти

все остальные классы, то есть фактически управление всей игрой идёт именно из этого класса.

Выводы

После долгой работы нам удалось добиться поставленной цели: изучив огромное количество новой информации и прочитав множество книг, преодолев массу трудностей мы создали собственную версию классической игры Тетрис. У нас получилось реализовать придуманный алгоритм и добавить собственный дизайн и музыкальное сопровождение. Наша игра отвечает всем поставленным требованиям, и по качеству не уступает многим ее коммерческим версиям. Мы довольны проделанной работой и достигнутым результатом: ведь в конечном счете мы получили не только готовый продукт, но и ценный опыт и полезные знания в области программирования.

Также хотим отметить, что у нашего проекта есть будущее: нас очень заинтересовала многопользовательская игра Тетрис, отечественная версия которой так и не была опубликована. Мы бы хотели попробовать реализовать ее самостоятельно: для этого нам придется научиться работать с WebSockets и API, и приобрести еще много новых навыков. Но мы уверены, что результат оправдает себя, а полученные знания окажутся полезными в будущем.

Литература

1. Beginning Python Games Development, With PyGame, 2-е издание (2015)
2. Program Arcade Games: With Python and Pygame, 4-е издание (2016)
3. Making Games with Python & Pygame, Al Sweigart (2012)
4. Изучаем Python. Программирование игр, визуализация данных, веб-приложения, Эрик Мэтиз.

Использованные сайты:

5. <https://ru.wikipedia.org/wiki/Тетрис>
6. <https://pythonru.com/uroki/biblioteka-pygame-chast-1-vvedenie>