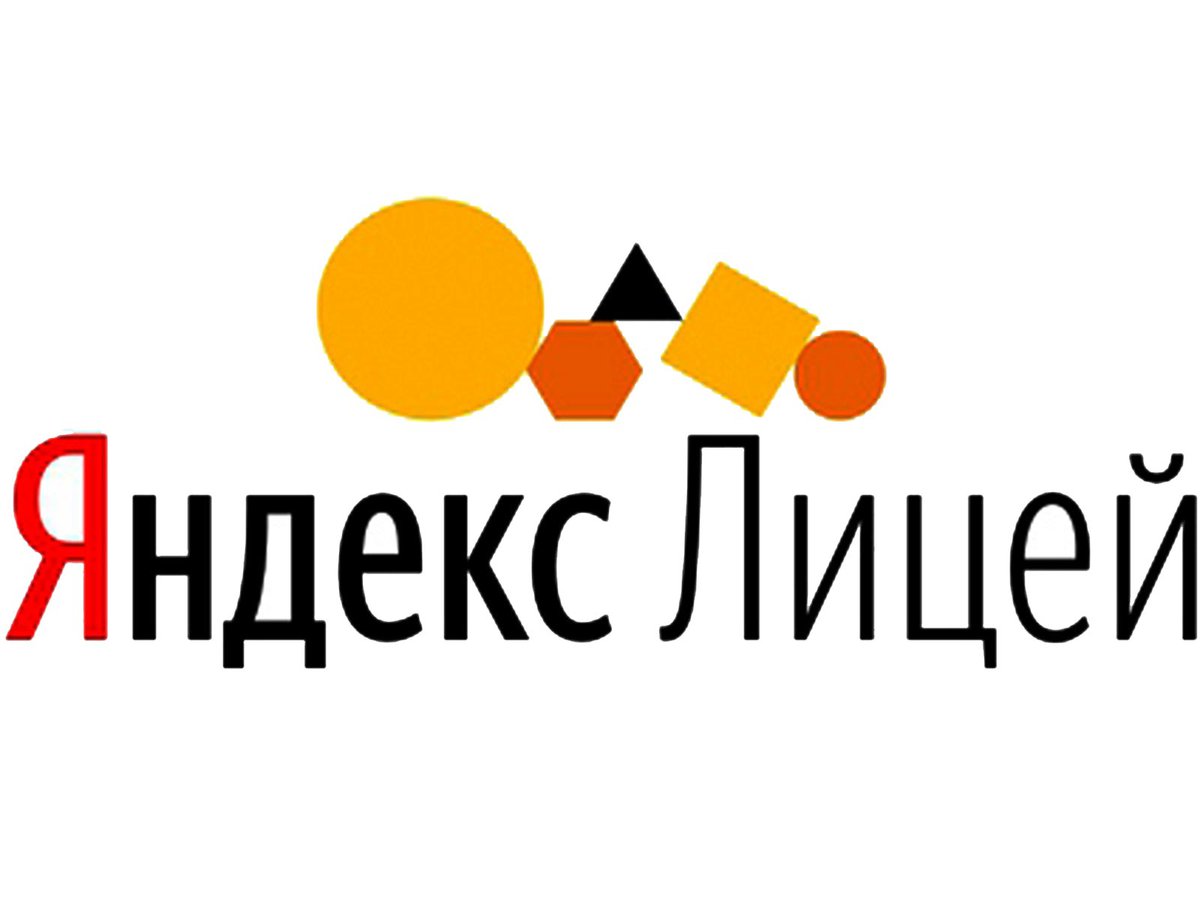
*Яндекс Лицей*

Проект по Pygame

*Тема: «Сборник мини-игр с сюжетом»*



*Работу выполнили:*

*Студенты II курса Яндекс Лицея*

*ГБОУ Школы №185*

*Опарин Григорий*

*Элькинсон Максим*

*Руководитель проекта:*

*Галганова Дарья Анатольевна*

Москва 2021

Оглавление

[Введение 3](#_Toc56114494)

[Реализация 3](#_Toc56114495)

[Классы: 3](#_Toc56114496)

[Библиотеки(модули): 5](#_Toc56114497)

[Вывод 6](#_Toc56114498)

# Введение

***Идея проекта***: Создать компьютерную игру из множества мелких мини-игр, связанных между собой сюжетом. Игроку предстоит играть за ИИ, который был создан человечеством для решения глобальных проблем на планете, но помочь человечеству или предать его – это выбор , который будет зависеть от игрока.

***Принцип работы***: С начала игрок увидит стартовое меню из которого он может попасть ко всем мини-играм(по цепочке). Сами мини-игры устроены по принципу КАТ-СЦЕНА-ОПИСАНИЕ > ИГРА > КАТ-СЦЕНА-ИТОГ ИГРЫ. Если по итогам мини-игры игрок проиграл, то мини-игра перезапускается или это приводит к проигрышу во всей игре.

# Реализация

## Классы:

По назначению, классы можно распределить на группы: Вспомогательные – вспомогательные классы , созданные для наследования своих методов классам других групп, графические – классы для отображения графических элементов, которых нет в стандартных библиотеках, игровые-технические – классы нацеленные на работу с пользователем.

* Вспомогательные:
* GameStage
* CellGame
* Графические:
* Button(pygame.sprite.Sprite)
* Timer(pygame.sprite.Sprite)
* ReakTile(pygame.sprite.Sprite)
* Saveload(GameStage)
* Choice(pygame.sprite.Sprite)
* Help(GameStage)
* Counter(pygame.sprite.Sprite)
* Speech(pygame.sprite.Sprite)
* Tile(pygame.sprite.Sprite)
* Игровые-технические:
* Reakcia(GameStage)
* Adaptation(GameStage)
* GameOver(GameStage)
* MainMenu(GameStage)
* MainSettings(GameStage)
* Intro(GameStage)

В данной программе создано 17 классов, которые выполняют различные функции(от простого дополнения других классов до отрисовки и обработки отдельных мини-игр). Однако, для ознакомления с принципом работы программ, можно выделить 3 ключевых класса (GameStage, Speech, Button)

***GameStage*** – вспомогательный класс отвечающий за сменяемость фаз игры и регулирование необходимых спрайтов.

**Методы**:

***update()*** – изначально пустой метод для обновления спрайтов.

***transform(stage=None)*** – метод для перехода с одной стадии игры на другую. При смене стадий удоляет все текущие спрайты , в целях оптимизации.

***Speech*** – класс для отрисовки кат-сцен.

**Методы**:

***update()*** – метод для пошаговой или единовременной отрисовки всех фраз(в зависимости от действий пользователя).

***set\_text(text)*** – метод для изменения текста кат-сцены.

***next\_phrase()*** – метод для перехода с к следующей вразе.

***is\_complete()*** – метод для проверки наличия текста, который может быть отображон .

***Button*** - класс отвечающий за отрисовку кнопок.

**Методы**:

***update()*** – метод проверяющий наличие нажатого курсора в зоне действия кнопки.

***set\_text(text)*** – метод для установки текста на кнопке.

## 

## Библиотеки(модули):

***Pygame –*** *для визуализации игровых объектов.*

***Random*** *– для генерации случайных элементов мини-игр.*

***Sys, os*** *– вспомогательные модули, для корректной работы функций и методов.*

# 

# Вывод

В ходе реализации проекта, были достигнуты все заявленные цели, а мы получил ценный опыт работы в команде и создания видео-игр. Далее мы планируем добавлять новые мини-игры и исправлять недочёты в уже имеющихся мини-играх.