**ВВЕДЕНИЕ**

В условиях быстро развивающейся современной высокотехнологичной экономики актуальной задачей является разработка различных автоматизированных систем. Они позволяют наиболее эффективно использовать различные современные средства информационной вычислительной техники и решать прикладные задачи. Для успешного решения таких задач нужно уметь использовать высокоуровневые средства разработки и декомпозировать задачу.

В данной работе решаются задачи высокоуровневого программирования, согласно индивидуальному заданию.

Актуальность курсовой работы связана с наличием большого количества прикладных задач, которые необходимо эффективно решать с помощью инструментов высокоуровневых методов программирования.

* **ПОСТАНОВКА ЗАДАЧИ КУРСОВОЙ РАБОТЫ**
* Исходные данные к задаче курсовой работы.
* Формулировка: Создание 2D игры на языке программирования Python:
* Язык программирования python
* Основные библиотеки pygame, time, os, numpy, sys, math

Игра сделана в формате 2D. Игра имеет функциональное меню с кнопками 'Start', 'Continue', 'Options', 'About', 'Quit', из которых все функциональны и стабильно работают. Функция 'Options' включает в себя настройку полноэкранного режима и возможность вкл/выкл 'Показать ФПС'. Кнопки Start, Continue, Quit - работают соответственно. Сама игра сделана в стиле известной игры 'Mario' : ГГ(Главный герой) может передвигатьсяь влево/вправо, прыгать/падать. Задача игрока добраться до финиша на определённом уровне. Когда игрок пройдёт все уровни - он выиграет.

* Анализ исходных данных курсовой работы

Для выполнения задания нужно проделать следующие этапы:

* Грамотно распределить время работы в команде для более координированной работы с кодом.
* Для более удобной работы игры необходимо создать главное меню игры: присваиваем кажой кнопке определённый индекс, добавляем обновление кнопки при нажатии(анимация),добавляем проверку события для нажатой и отпущенной кнопки, добавляем задний фон и анимацию фона и прочее.
* Для удобства пользователя необходимо добавить функцию полного экрана и возможность убрать/показать FPS в меню 'Options'
* При написании кода понадобится импортировать библиотеки: Pygame, sys, os, time, numpy(Некоторые будут убраны после оптимизации кода)
* Будет добавлен файл data/engine.py - в котором будет прописаны: базовая физика и механика игры, прорисовка и отображение игрового мира, подгрузка файлов. А так же это файл отвечает за полную настройку игры. - Ядро игры.
* Необходимо нарисовать все необходимые изображения и включить их в код, а именно: кнопки(в нажатом и отпущенном виде), игрок(в статичном положении и в движении), фон игры(в игре и в состоянии паузы), эффекты частиц и прочего, игровые объекты(платформы, цепи, стены)
* Выбор и обоснование необходимых библиотек и среды разработки.
* Поскольку в методических указаниях к выполнению курсовой работы указано использование языка программирования Python, среды разработки PyCharm, а также стандартных библиотек и модулей стандартной библиотеки, то для решения заданий курсовой работы были выбраны следующие средства: Язык программирования Python3 – согласно рекомендациям; Интегрированная среда разработки PyCharm – согласно рекомендациям;
* Pygame— библиотека Python для упрощения написания игрового кода на движке Python
* Sys— библиотека Python для прямого взаимодействия с интерпритатором.
* Os— библиотека Python для получения наглядной информации о работоспособности кода
* Time— библиотека Python для более удобной работы с секундами игры, FPS, выводом его на экран и прочее.
* Math— библиотека Python для более удобной работы с математическими функциями кода, и для составления более сложных вычислений.
* Numpy— библиотека Python необходима для составления высокоуровневых математических функций, составления массивов и пр.
* Выводы по 1 главе.

Изучив полученный материал из главы 1, можно сделать вывод, что задание включает наиболее часто используемые прикладные задачи высокоуровневого программирования – это построение приложений с графическим пользовательским интерфейсом, использование алгоритмов, работа в наиболее популярной среде разработки, ставшей де-факто стандартом для разработчиков на языке программирования Python.

Успешное выполнения задания будет свидетельствовать о полноте усвоенного материала в рамках дисциплины «Высокоуровневые методы программирования» и умение решать задачи низкой и средней сложности.

* **РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА**
* Создание движка(ядра) игры. Для этого импортируем библиотеки: pygame, math, os, sys, в файл data/engine а так же из библиотеки pygame импортируем locals.
* Добавляем класс Font - Массив, который содержит в себе названия переменных, символов , которыми пользуется игра для корректного чтения кода.
* Добавляем и инициализируем(\_\_init\_\_) объекты физики и задаём функцию движения отдельно от игрока(камера центрирована на персонаже)
* Так же добавляем в код функции для очищения объектов, если персонаж их не видит.
* Добавляем отрисовку спрайтов и прочих изображений
* Создание главного меню игры. Для этого создаём файл menu.py, в который мы поместили классы: MenuScreen, OptionsMenu, VideoMenu - которые отрисовывают меню и проверяют события стартового экрана игры.(нажатие кнопок) В файл main.py мы поместили класс MainMenu - который является связующим классов выше, который инициализирует, проверяет на события и обнавляет экран главного меню.
* Добавляем файл Settings.py - в который мы добавили такие функции, как длинна и высота окна, размер окна, обозначения цветов, скорость движения игрока - для поддержания работоспособности и читабельности кода.
* Создание игры. Для начала мы добавили файл objects.py, в который поместили класс MapLevel - в котором происходит загрузка изображений для игры(карты, объекты игрового окружения), задаётся размер объектов(изображений),отрисовка игровых плит, стен колючек и тд. А так же обновление карты.