**Пояснительная записка**

**Название проекта:** «Самое ценное сокровище»  
**Исполнитель:**  
**Цель проекта:** Создать игру-платформер на 2 персонажей.  
**Используемые библиотеки:**  
– pygame  
– sqlite3 3.3.3  
– pip 24.3.1

**План проекта:**

1. Создание приветственного экрана

1.1. Создание экрана с уровнями

1. Создание уровней для игры

2.1. Создание сюжета и концепции игры

2.2. Создание Персонажей

2.3. Создание уровней

1. Создание конечного экрана

3.1. Создание логики игры

1. Скрины приложения
2. База данных
3. Используемые функции и классы

**Необходимые таблицы в базе данных:**

1. **users** (Поля: levels\_complite)  
   Хранение количества пройденных уровней пользователем.
2. **levelOne** (Поля: time, coins)  
   Хранение статистики о прохождении уровня 1
3. **levelTwo** (Поля: time, coins)  
   Хранение статистики о прохождении уровня 2
4. **levelThree** (Поля: time, coins)  
   Хранение статистики о прохождении уровня 3
5. **levelFour** (Поля: time, coins)  
   Хранение статистики о прохождении уровня 4

**1. Создание приветственного экрана**

Приветственный экран включает в себя: кнопку для запуска игры. (рисунок 1)

**1.1. Создание экрана с уровнями**

Экран с уровнями включает в себя 5 кнопок. Четыре кнопки для входа на уровни (пока не пройден предыдущий уровень, следующий не открывается) и кнопка exit для выхода из игры. (рисунок 2)

**2. Создание уровней для игры**

**2.1. Создание сюжета и концепции игры**

Описание:  
Маг и разбойник находят карту сокровищ и отправляются на их поиски. Игра выполнена в формате платформера и рассчитана на двух игроков. У каждого из двух персонажей есть свои особенности:

* **Маг:** – Способен открывать магические двери – Низкого роста – Высоко прыгает – Не может убивать монстров
* **Разбойник:** – Высокого роста – Низко прыгает – Может убивать монстров – Может двигать коробки

Для того чтобы пройти игру, игрокам необходимо работать вместе. (концепты персонажей и уровней представлены в приложении)

**2.2. Создание Персонажей**

Каждый спрайт в игре является наследником от класса GameSprite. Полями этого класса являются: положение спрайта по x и y, группа спрайтов, к которой спрайт принадлежит, картинка, ширина и высота спрайта. Все спрайты, которые могут двигаться, представлены классом MovableGameSprite, который является наследником класса GameSprite.

**Маг:**  
Персонаж маг представлен классом Mag. Этот класс является наследником класса MovableGameSprite. У класса мага есть три функции:

* can\_move (Проверяет, может ли персонаж пройти дальше)
* can\_stay (Проверяет, может ли персонаж стоять на данном блоке)
* do\_update (Отвечает за обработку всех событий, связанных с персонажем)

**Разбойник**:  
Персонаж разбойник представлен классом Robber. Этот класс является наследником класса MovableGameSprite. У класса разбойника есть несколько функций:  
• **can\_move** (Проверяет, может ли персонаж пройти дальше, проверяя содержимое блока)  
• **can\_stay** (Проверяет, может ли персонаж стоять на данном блоке)  
• **find\_box** (Находит коробку рядом с персонажем)  
• **chek\_colide\_with\_box\_right**, **chek\_colide\_with\_box\_left**, **chek\_colide\_with\_box\_top** (Проверяют столкновения с коробкой в разных направлениях)  
• **do\_update** (Обрабатывает движения персонажа, учитывая столкновения и нажатые клавиши)  
• **update** (Обновляет состояние персонажа, включая падение и прыжки)

**2.3. Создание уровней**

Каждый уровень представлен в виде текстового файла. При запуске программа читает из него расположение каждого спрайта и подрисовывает их. На уровне есть монетки, их необходимо собирать. Для того чтобы уровень засчитался пройденным, необходимо, чтобы оба персонажа достигли дверей, расположенных на уровне. В базу данных будет заноситься информация о каждом прохождении (сколько времени затрачено, сколько монет собрано).

**3. Описание игровых объектов**

**3.1. Block**

Объект, представляющий собой блок на игровом поле. Он используется для создания препятствий. Класс Block наследует от базового класса GameSprite, который предоставляет функциональность для работы с изображениями и размещением объектов на экране.

**3.2. Coin**

Монета, которую игрок может собирать. Когда маг или вор соприкасаются с монетой, она исчезает, а счетчик собранных монет увеличивается.

**3.3. Door**

Обычная дверь, которая открывается, если персонажи (маг или вор) находятся рядом с ней. Завершение уровня.

**3.4. MagicDoor**

Магическая дверь, которую может открыть только Маг.

**3.5. Button**

Кнопка, которая активирует определенные механизмы в игре, такие как открытие дверей или активизация других объектов.

**3.6. Lever**

Рычаг, который может переключать состояние активации. Он использует событие нажатия клавиши для проверки активации.

**3.7. GorizontalDoor**

Дверь, которая открывается, если определенные кнопки активированы.

**3.8. Spike**

Шипы, которые убивают игроков, если они с ними сталкиваются.

**3.9. VerticalDoor**

Вертикальная дверь, работающая по аналогии с горизонтальной, но ориентированная вертикально.

**3.10. Monsters**

Монстры, которые перемещаются по уровню и могут взаимодействовать с персонажами, что может привести к их смерти.

**4. Механика взаимодействия объектов**

* **Персонажи** могут взаимодействовать с различными игровыми объектами.
* **Кнопки и рычаги** активируют двери и другие механизмы в игре.
* **Двери** открываются при наличии нужных условий.
* **Магические двери** открываются только магом.
* **Монстры** могут столкнуться с персонажами и убить их. Только Разбойник может дать отпор.

**5. Алгоритм работы игры**

* Все объекты в игре обновляются в методе update, где проверяется их состояние, а также выполняются проверки на столкновение с другими объектами.
* Обновление карты уровня происходит при активации двери или другого механизма.

**6. Заключение**

Данный код представляет собой основу для создания игры с разнообразными объектами, которые взаимодействуют друг с другом. Для расширения проекта можно добавить дополнительные механизмы, такие как новые уровни, улучшение логики искусственного интеллекта монстров, а также новые виды объектов для более сложных уровней. Интеграция магических элементов и адаптация карт в зависимости от действий игрока могут значительно усложнить игровой процесс и добавить элементы стратегии.