Пояснительная записка к проекту PyGame “PyBall” Погонцева Данила

Содержание

1. Введение
   1. Идея проекта
   2. Игровой процесс
2. Описание реализации
   1. Структура игры
   2. Особенности реализации
   3. Используемые технологии
3. Планы на будущее
   1. Идея проекта

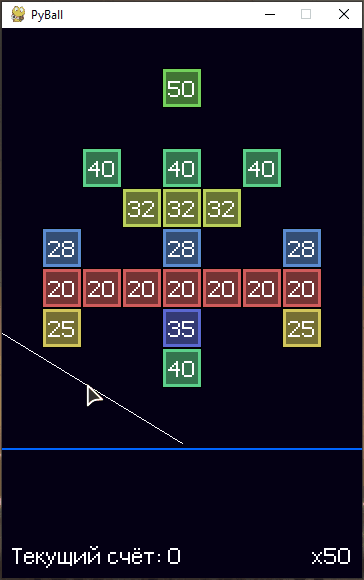
**PyBall** – небольшая, но увлекательная игра, игровой процесс которой заключается в прохождении уровней, разрушая блоки различной прочности.

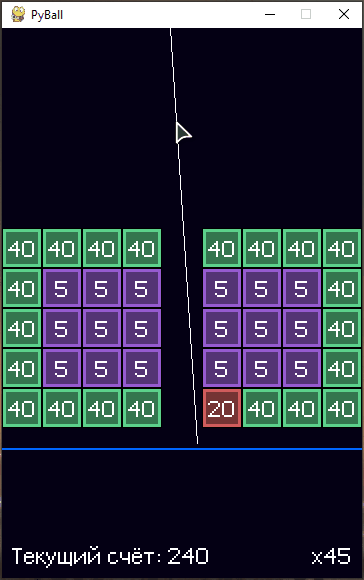
**Примечание**: на анимации здесь и далее могут наблюдаться артефакты, искажённые цвета и низкая частота кадров из-за особенностей формата gif. В игре таких проблем нет.

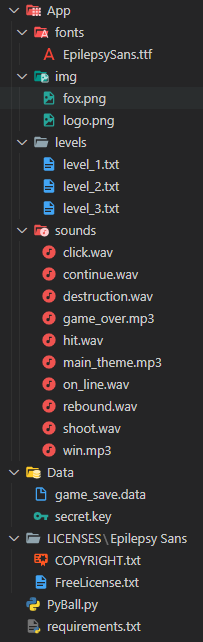


1.2 Игровой процесс

Игрок выстреливает n количеством шариков, n определяется конкретным уровнем. Направление выстрела игрок задаёт с помощью мыши. Шарики, попадая в блоки, снимают определённое количество прочности блока и отскакивают от него, соблюдая закон отражения. Блоки, когда теряют прочность, меняют свой цвет, а при достижении прочности 0 – разрушаются. После каждого хода блоки опускаются к ограничивающей линии. Игрок побеждает, если все блоки были разрушены, до того, как они достигнут ограничивающей линии. Пока шарики находятся в полёте игрок не может выстелить ещё раз. Особое внимание было уделено звуковым и визуальным эффектам. Присутствует возможность ускорять время.





2.1 Структура игры

1. PyBall.py – главный файл игры

2. requirements.txt – файл с перечнем зависимостей

3. fonts – папка с используемыми в игре шрифтами, на данных момент необходим только один

4. img – папка, содержащая изображения для спрайтов

5. levels – папка, в которой находятся уровни игры

6. sounds – содержит все звуки и музыку

7. Data – папка с пользовательскими данными

8. game\_save.data – файл сохранения игрового прогресса, в зашифрованном виде

9. secret.key – файл, содержащий уникальный ключ для расшифровки game\_save.data

10. LICENSES – папка, содержащая лицензии используемых ресурсов

2.2 Особенности реализации

* Простое добавление новых уровней
* Шифрование файла сохранения игрового процесса
* Простое добавление новых механик в будущем
* Все магические данные вынесены в виде констант
* Использование ООП и спрайтов
  + GameMap – основной класс, управляющий игровым процессом
  + Block и Ball – классы соответствующих названию спрайтов, описывающие их поведение
  + TextRender – класс для более удобного размещения текста на экране
  + BottomLine – класс, описывающий ограничивающую линию, является спрайтом
  + AnimatedSprite – класс анимированного спрайта для добавления милой лисы на главный экран
  + Button – класс кнопки, описывает её поведение и определяет логику взаимодействия, является спрайтом
  + SimpleBall – класс, описывающий поведение декоративных шариков на главном экране
  + Particle – класс для создания эффекта разрушения блока
  + Image – класс, для добавления картинки на экран в виде спрайта

2.3 Используемые технологии

* Python 3.9.13
* PyGame 2.1.2
* Cryptography 39.0.1
* PhotoShop 2020

3 Планы на будущее

* Добавление новых уровней и механик, к примеру подбор новых шариков, бустеры и т.п.
* Добавление валюты и скинов для блоков и шариков
* Добавление настроек приложения прямо в его интерфейсе
* Мелкие улучшения