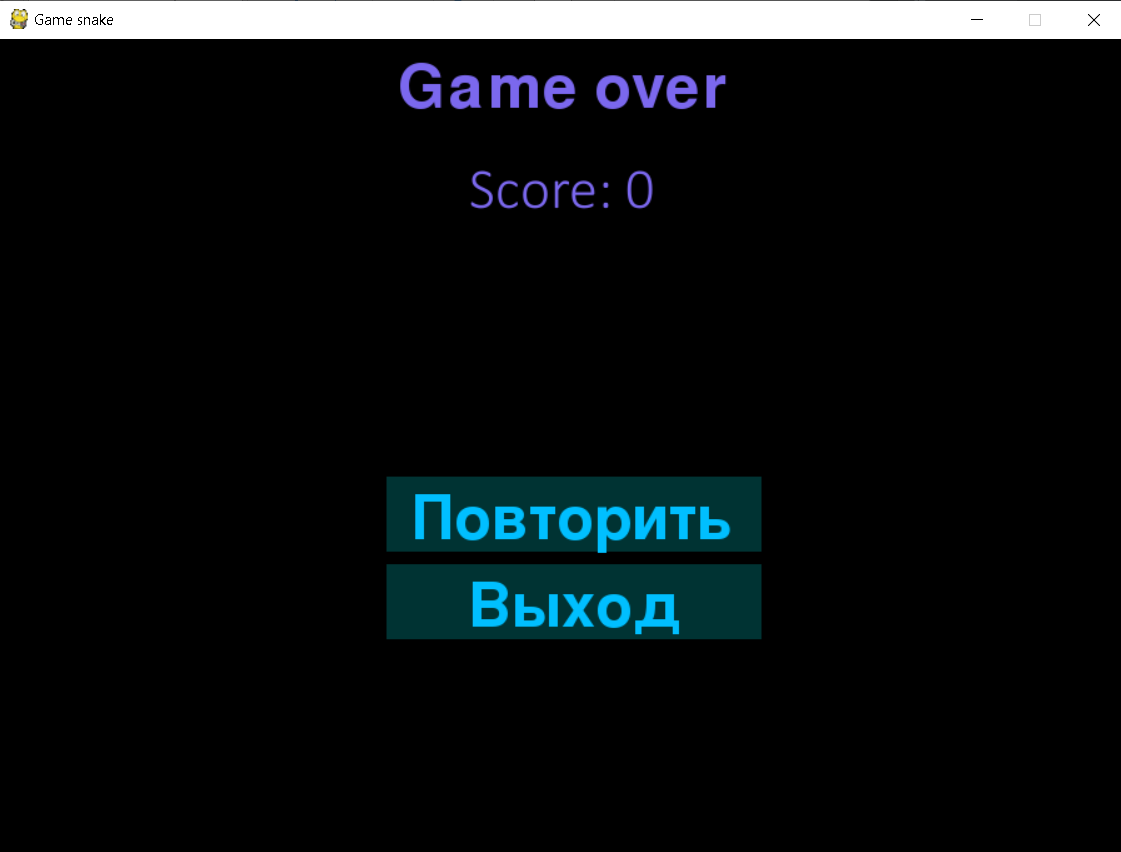
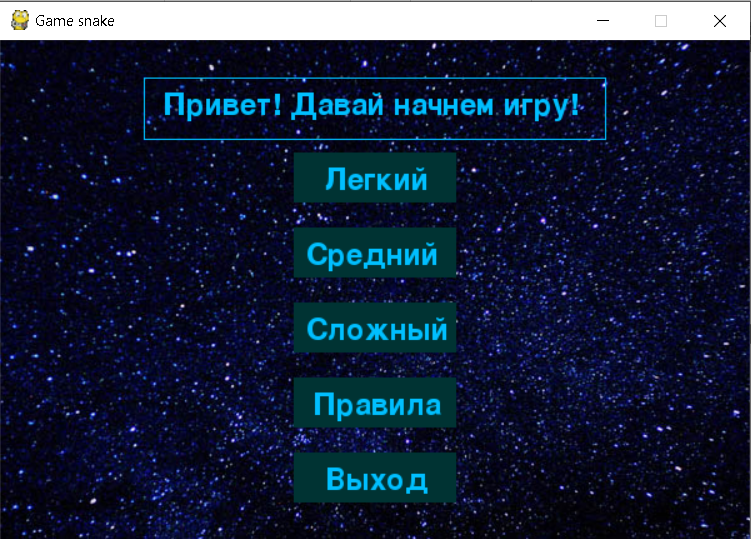
1. **Название проекта:** Игра «Змейка»
2. **Автор проекта:** Босачева Татьяна Владиславовна
3. **Описание игры:** Игрок управляет длинным, тонким существом, напоминающим змею, которое ползает по плоскости, ограниченной стенками собирая еду, избегая столкновения с собственным хвостом и краями игрового поля. В самом сложном уровне на поле присутствуют дополнительные препятствия. Каждый раз, когда змея съедает кусок пищи, она становится длиннее что постепенно усложняет игру. Игрок управляет направлением движения головы змеи (есть 4 направления: вверх, вниз, влево вправо), а хвост змеи движется следом. Игрок не может остановить движение змеи.
4. **Описание реализации**:

***Класс Game()*** необходим для запуска игры. Сначала мы задаем параметры экрана и необходимые цвета. С помощью функции ***init\_and\_check\_for\_errors()*** можно проверить код на наличие ошибок. С помощью функции ***set\_surface\_and\_title()*** создаем холст для рисования и заголовок окна. С помощью функции ***event\_loop()*** можно отслеживать нажатие клавиш, в этой игре можно использовать сочетание букв «daws» или стрелки. С помощью функции ***show\_score()*** отображается набранный результат: во время игры результат отображается слева вверху, а после завершения попытки результат отображается по центру окна. Функция ***game\_over()*** сообщает о завершении игры и выводит набранный результат, а так же появляется возможность начать игру сначала или выйти из игры.

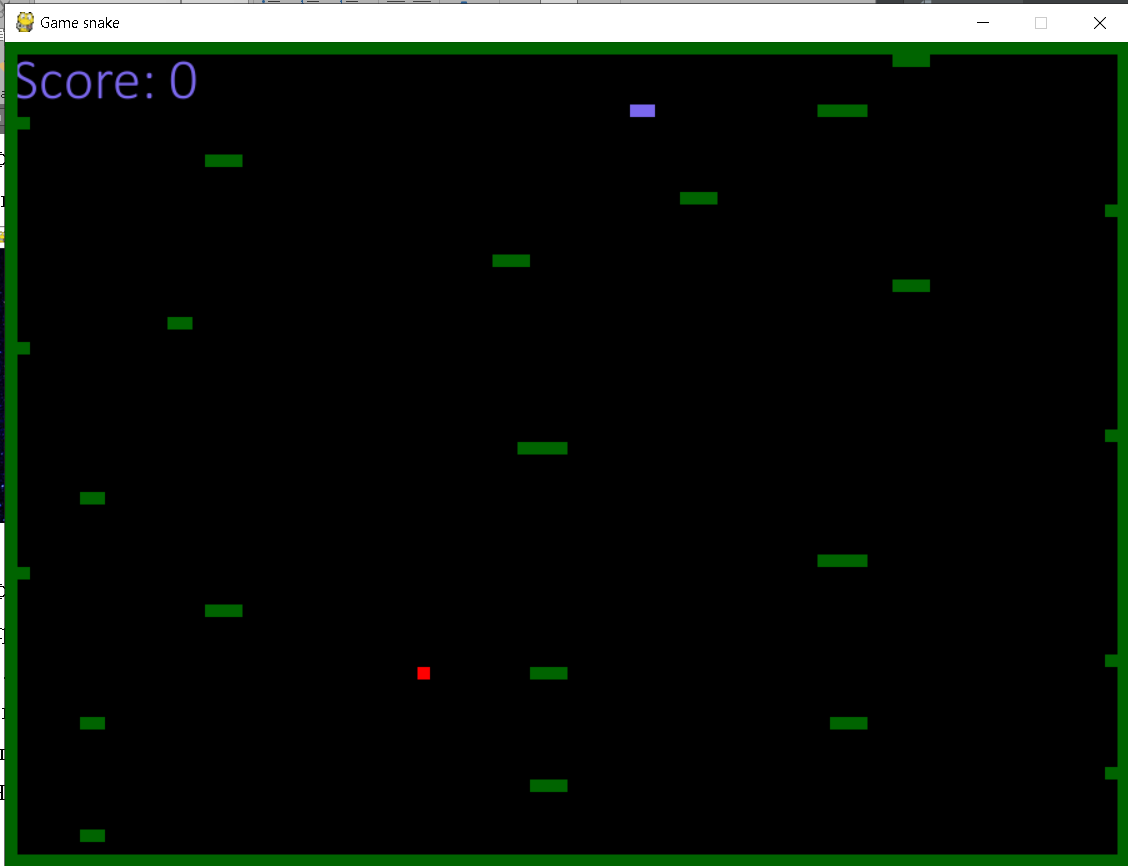


***Класс Snake()*** необходим для создания змейки. Для начала мы создаем по определенным координатам змейку, задаем начальное направление. С помощью функции ***validate\_direction\_and\_change()*** мы можем изменять направление движения змеи только в том случае, если оно не противоположно текущему. Функция ***change\_head\_position()*** изменяет положение головы змеи после изменения направления. Функция ***change\_head\_position3()*** так же отслеживает столкновения с препятствиями. Функция ***snake\_body\_mechanism()*** необходима для увеличения змейки, если она съела еду, и назначение нового рандомного положения еды, а так же для увеличения счетчика. Функция ***check\_for\_boundaries()*** проверяет на столкновения с границами окна и на то, столкнулась ли змея сама с собой.

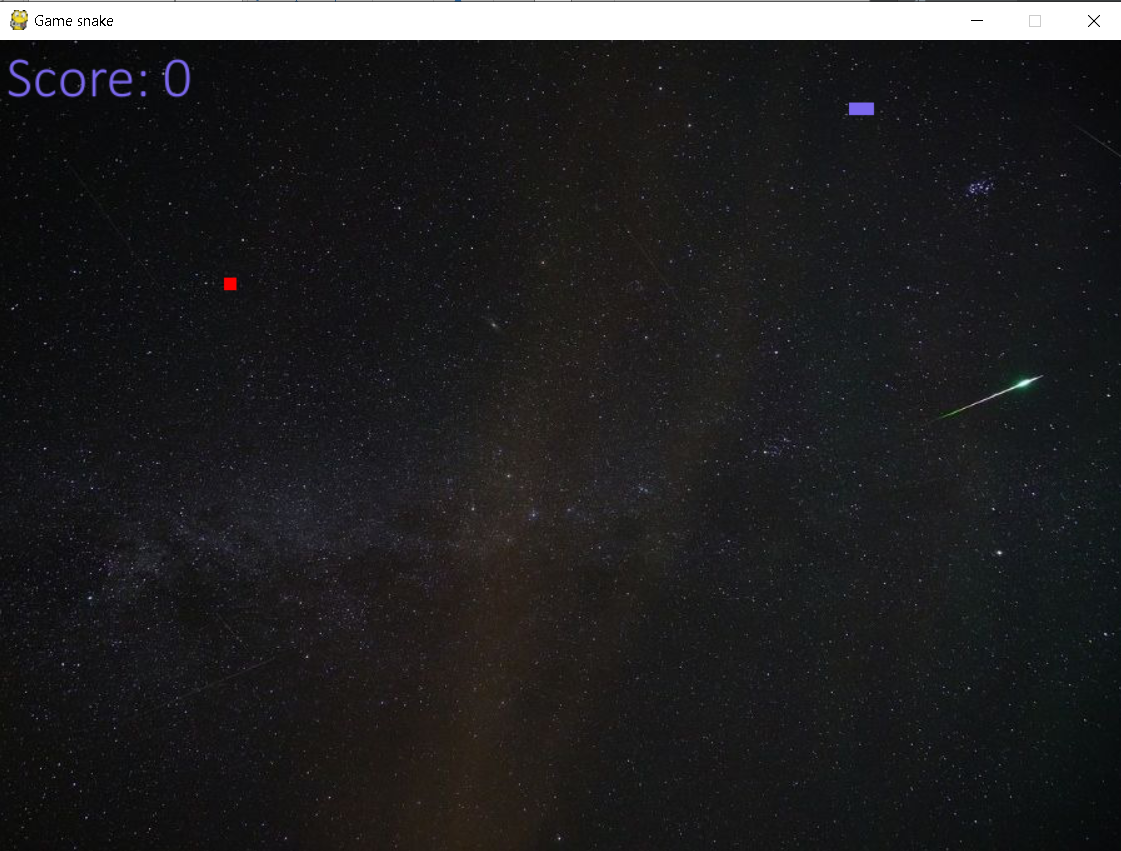
Функция ***start\_screen()*** рисует заставку игры, на которой можно выбрать уровень сложности игры, почитать правила игры или выйти.

****

Функция ***prep()*** необходима для рисования самого сложного уровня с препятствиями. И используется для запуска функции ***uroven3().***



Функции ***uroven1()*** и ***uroven2()*** запускают игровой процесс для игры без препятствий с различной скоростью перемещения змейки.



1. **Описание технологий:** В данном проекте использовался pygame — набор модулей (библиотек) языка программирования Python, предназначенный для написания компьютерных игр и мультимедиа-приложений.

Необходимые для запуска библиотеки: pygame и встроенные sys, random, time.