**2023/11/07 00:00|Лекция. Практика. Часть 1.**

**Практика. Часть 1**

На данном занятие начинаем **серию практических занятий по разработке собственной игры на библиотеке arcade.**

В качестве примера разработаем **игру Пинг-Понг**, и у нас уже есть заготовок проекта. За кадром создали быстрый проект, создали файл game, в котором будет находиться логика игры, и сделали ракетку и мячик, то есть нашли картинки(рис.1). Они будут спрайтами в нашей игре.  
В игровом процессе **Sprite**(спрайт) – это любой подвижный объект.

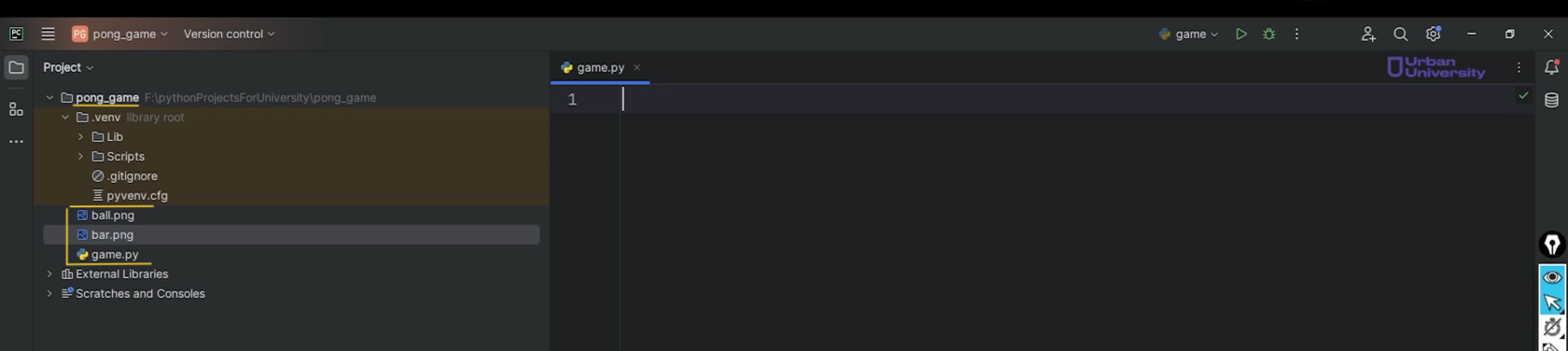


Рис.1

Также установили заранее **библиотеку arcade**, чтобы вы не ждали. Однако вам необходимо будет открыть терминал после создания нового проекта и скачивания исходных материалов, то есть это ракетки и мячика, обновить библиотеку, написав в терминале команду pip install arcade(рис.2). После окончания установки вам остаётся её импортировать(рис.3). Раз мы собираемся работать с библиотекой arcade, значит, она нам 100% понадобится.



Рис.2

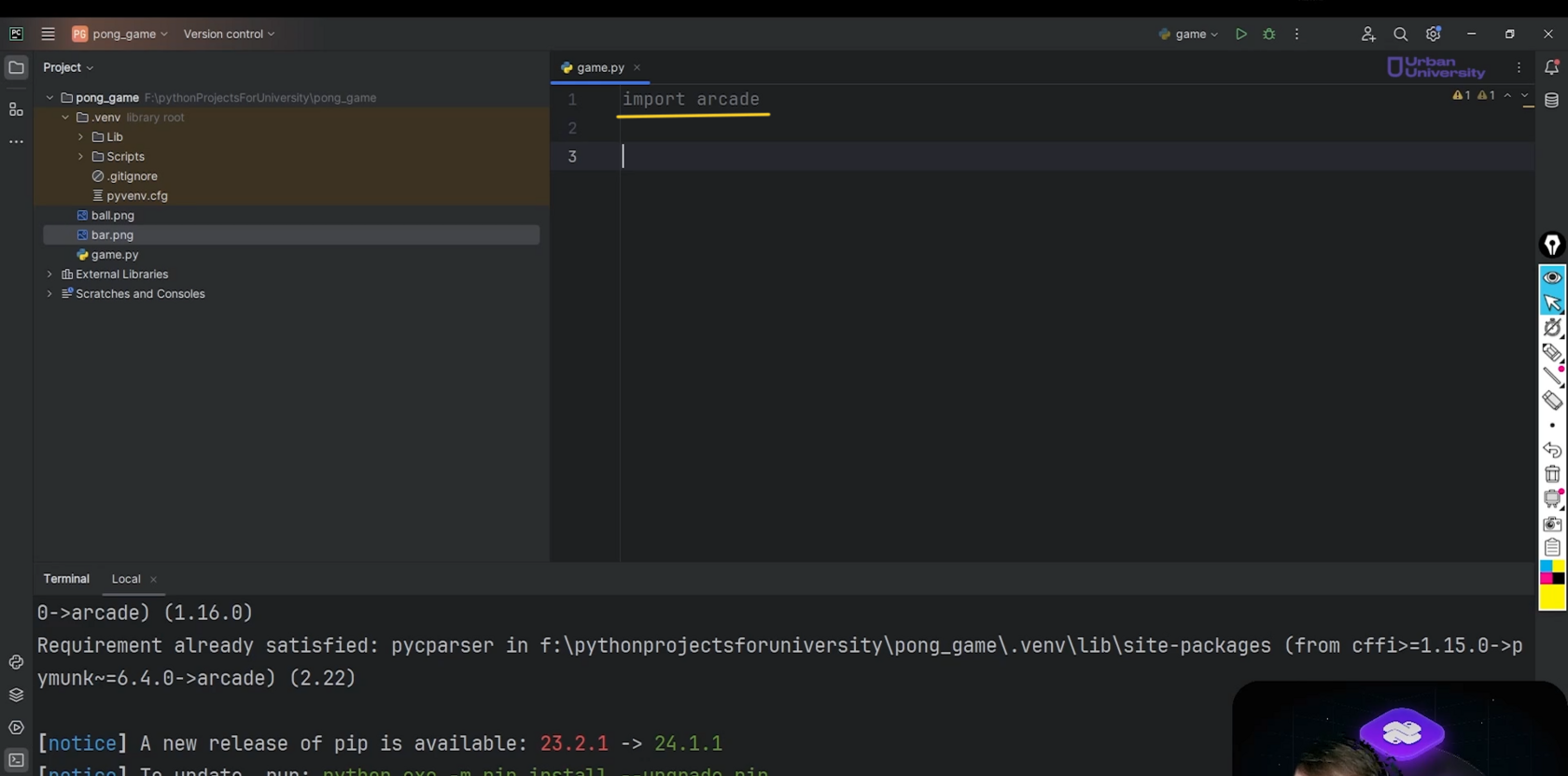


Рис.3

**Суть работы с библиотекой** будет заключаться в том, чтобы использовать классы, которые определены внутри этой библиотеки. При этом будем реализовывать свои классы и наследоваться от уже готовых классов.

Наша игра представляет собой графическое окно. Графическое окно тоже будет являться отдельным классом, а также можно сказать, что основным в нашем проекте.

**Начнём с создания графического окна.**Чтобы его создать, необходимо создать класс Game. Он будет наследоваться от класса arcade.Window(рис.4). По сути, данный класс будет представлять наше главное окно. Если посмотрим внутрь Window(рис.5), то увидим определённое количество методов, характеристик и так далее. Более подробно, более удобно, конечно, читать документацию на сайте библиотеки arcade. Но, пока что нам этого хватит.

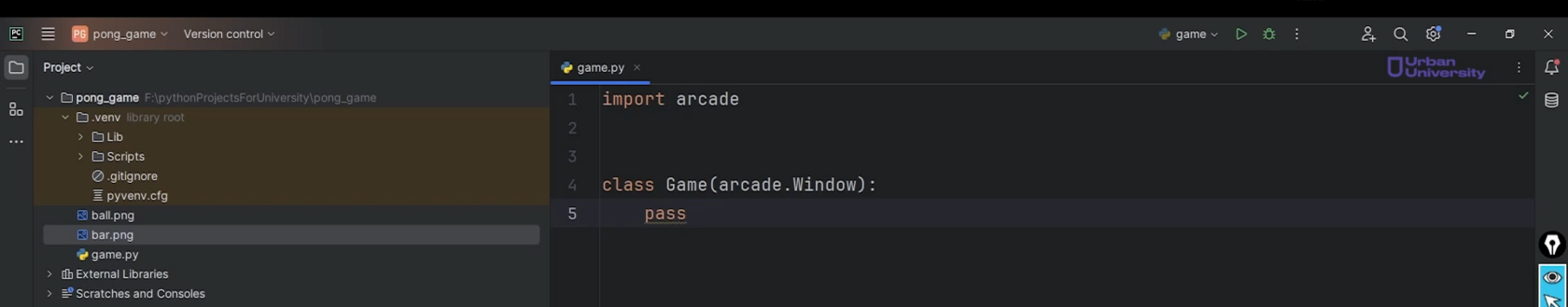


Рис.4

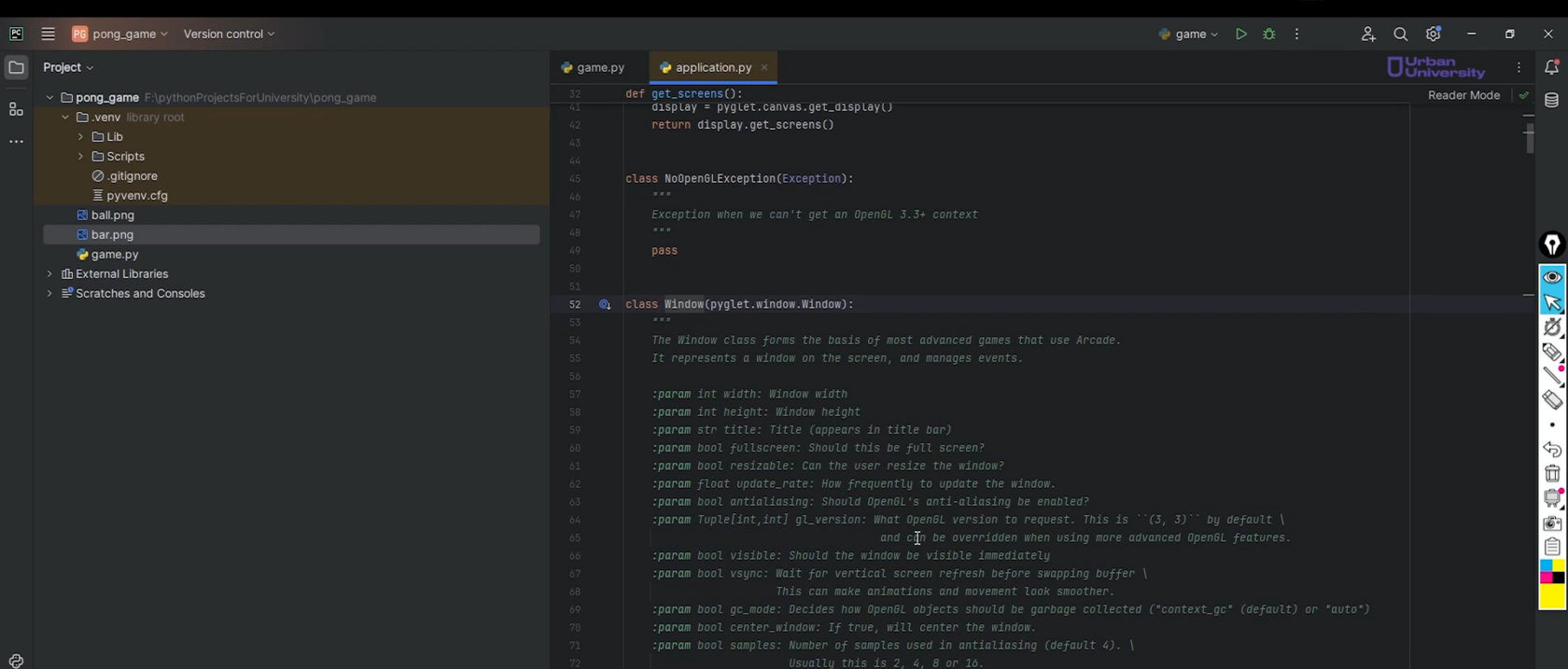


Рис.5

Создадим здесь конструкцию if name = ‘\_\_main\_\_’, где будем реализовывать создание экземпляра класса Game и запуск цикла обновлений arcade.run()(рис.6). После вызова метода run можем увидеть наше окошко(рис.7) непосредственно, которое сможем перемещать. Именно с этим окном мы в будущем будем работать. Однако сейчас оно обладает черным фоном, и нам необходимо сделать его белым, так как на чёрном фоне Пинг-Понг будет выглядеть крайне плохо.

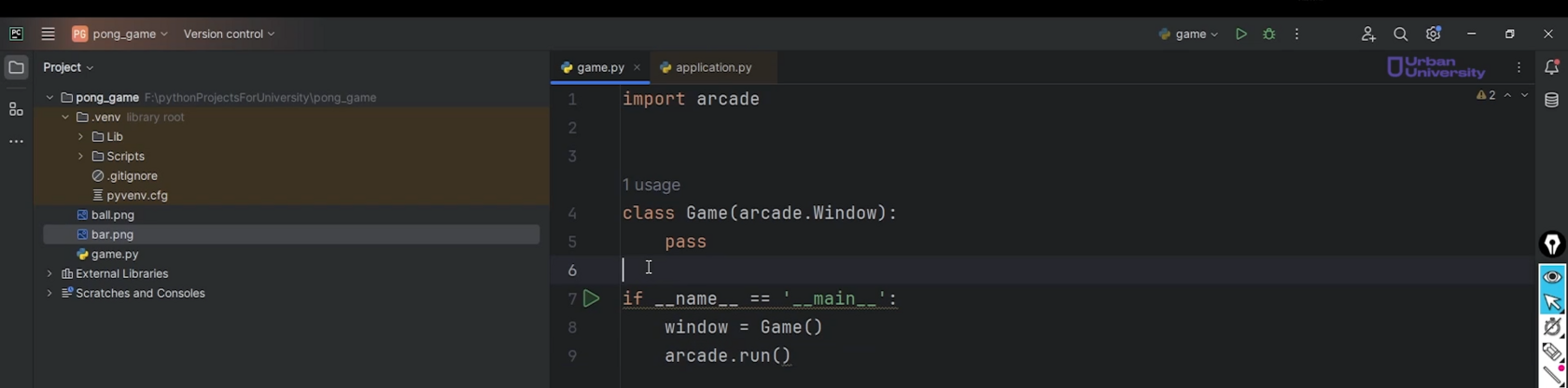


Рис.6

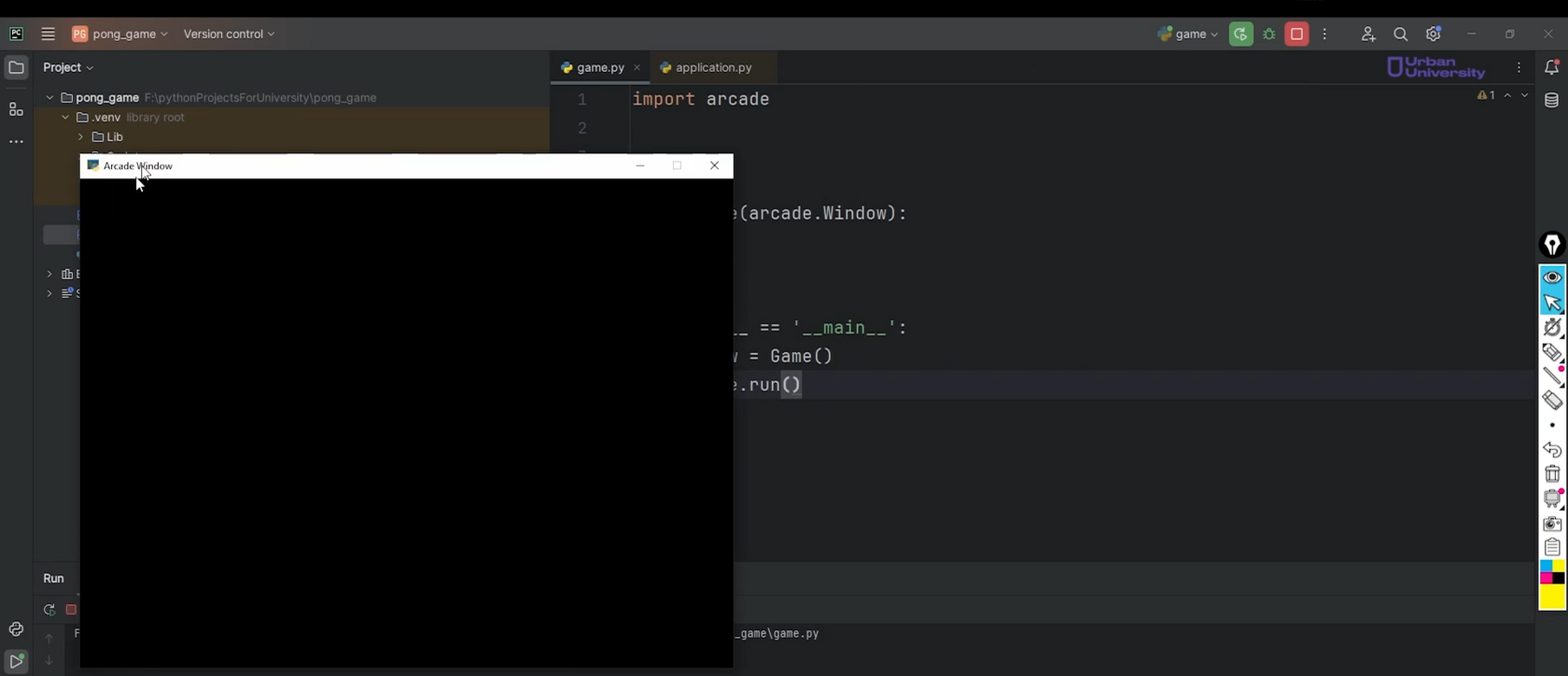


Рис.7

**Для того чтобы сделать окошко белым**, нам необходимо будет отрисовать белый цвет на заднем фоне. За отрисовку элементов на нашем окне отвечает метод on\_draw(), поэтому переопределим его и в этом методе вызовем метод clear()(рис.8). Данный метод отвечает за очистку нашего экрана. По сути, в 2D очистка представляет собой изменение одного цвета на другой, то есть, грубо говоря, сплошняком закрашиваем наш задний фон. Видим, что первый параметр — это кортеж из 3 значений типа int(рис.9). Эти значения у нас **RGB -**сочетание цветов интенсивностью от 0 до 255. Если передадим кортеж, который будет состоять из 3 чисел 255(рис.10), по итогу получим белый задний фон.

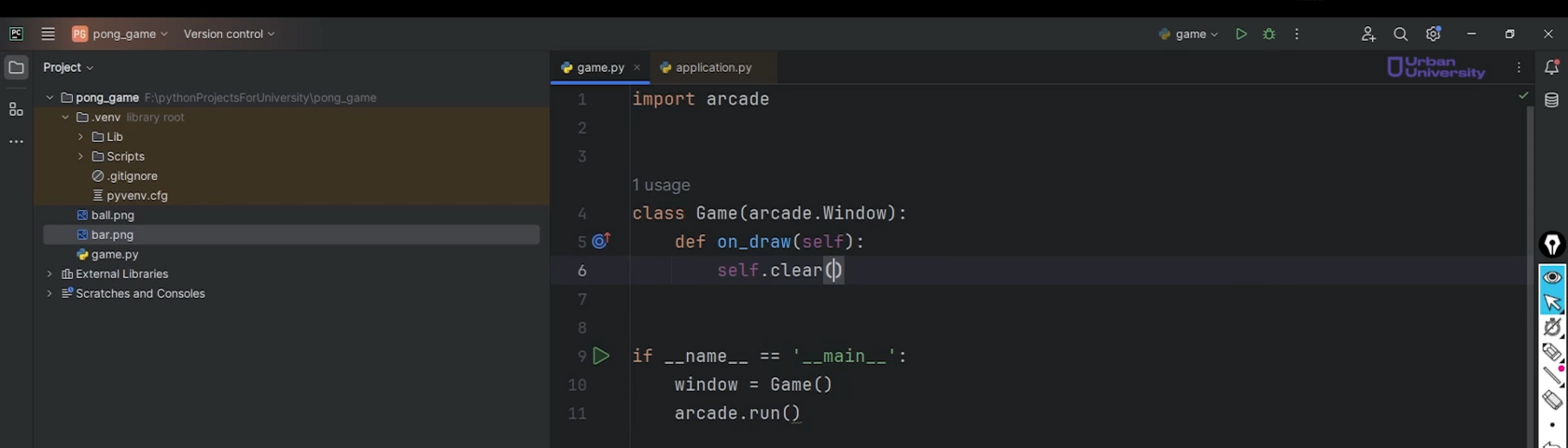


Рис.8

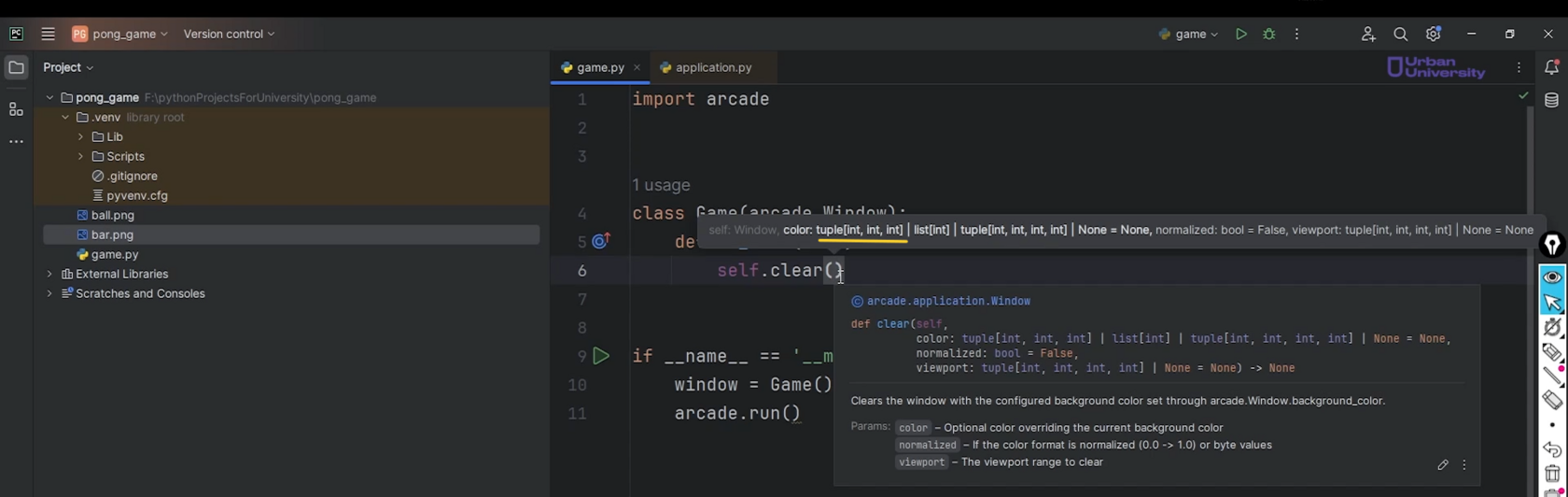


Рис.9

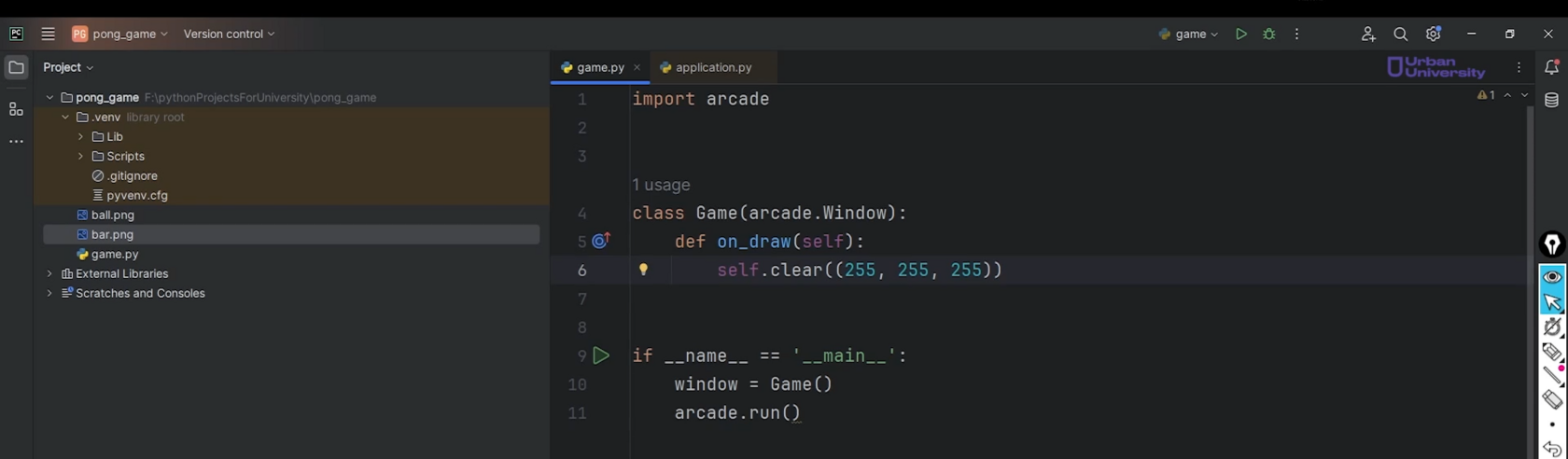


Рис.10

Давайте запустим проект и убедимся в этом. Все работает, наше окошко стало полностью белым(рис.11). Таким образом, мы подготовили скелет для нашей будущей игры.

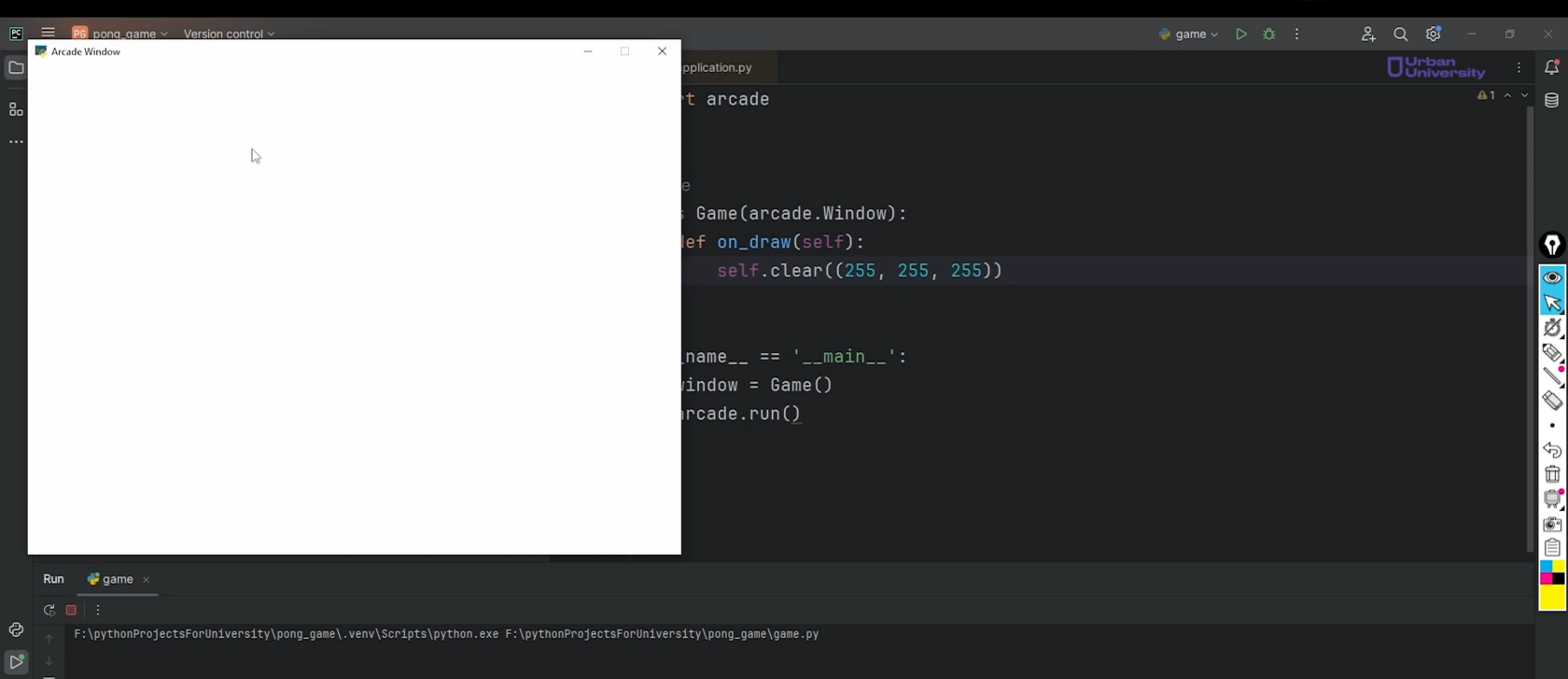


Рис.11