Отчет по лабораторной работе 1

Студента группы №3331506/00401 Орехова Алексея Михайловича

Код работы выложен на [гитхаб](https://github.com/alexeyorehoff/cv-lessons)

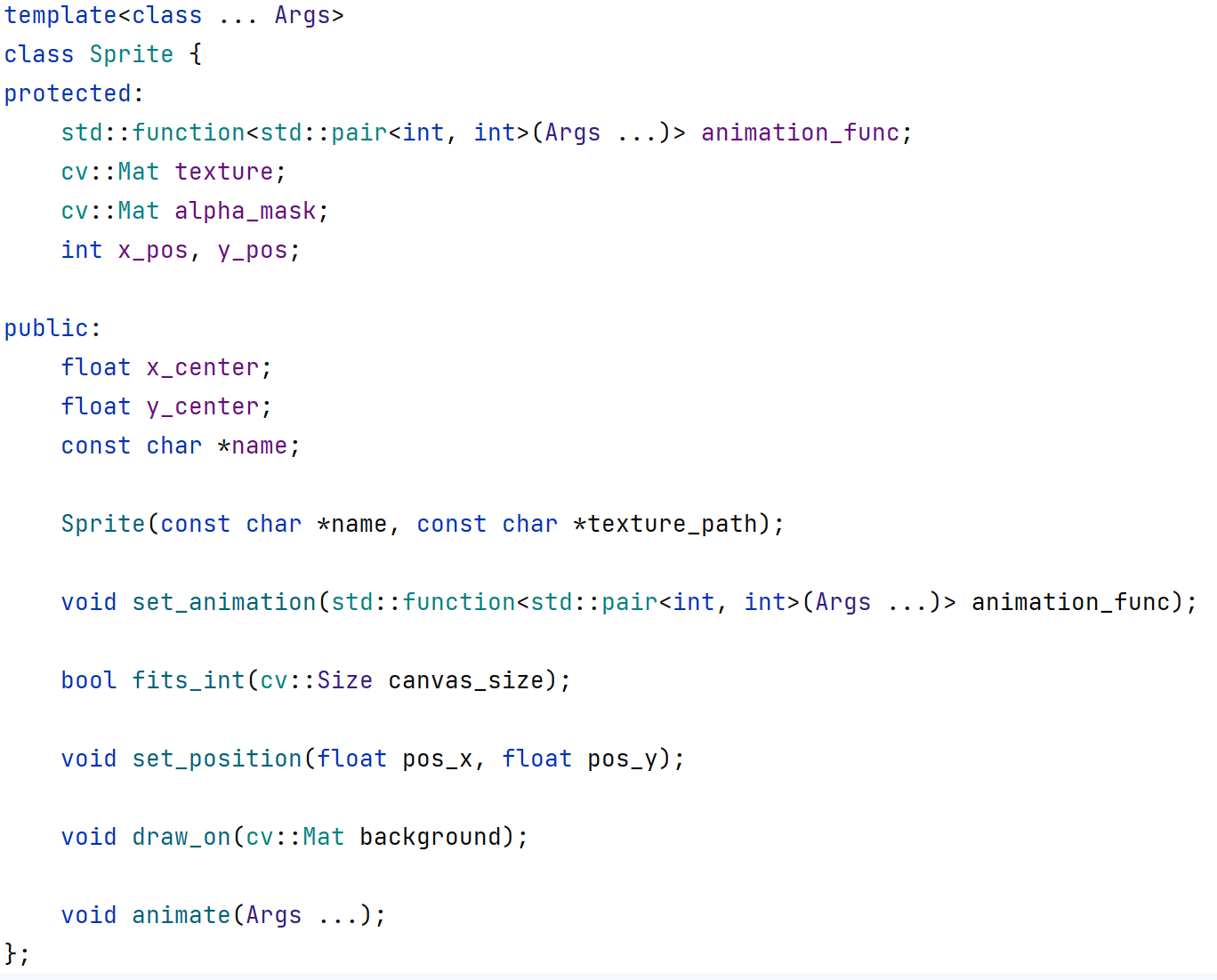
В рамках работы №1 было выполнено два задания – базовое и дополнительное задание №2 – Задача трех тел. С целью унификации задания 1 и 2 были созданы два вспомогательных класса:

* Класс App – объединяющий в себе создание окна, хранение, анимацию движения и отрисовку спрайтов.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

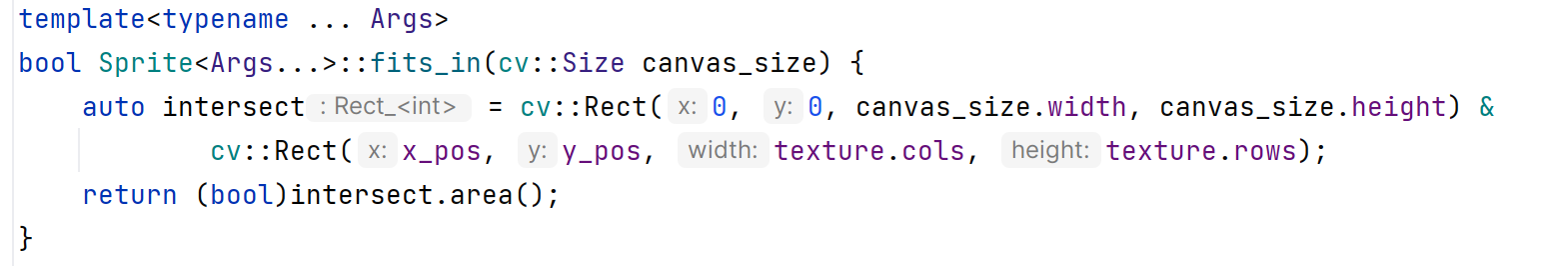
* Класс Sprite – класс, хранящий в себе текстуру и альфа-маску для вывода изображений с прозрачным фоном. Также, класс позволяет задавать положение спрайта на экране как через прямой ввод координат, так и через функцию анимации параметрически.



Аргументы, которые принимает функция анимации вводятся как шаблон. В конструкторе базового класса Sprite он принимает путь к текстуре в формате .png и производит разложение картинки на альфа маску и текстуру.

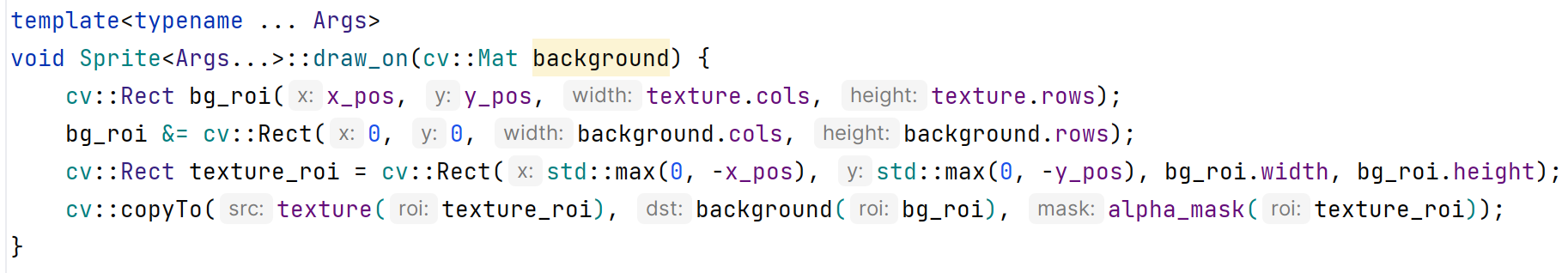
Из соображений удобства, со стороны пользователя ноль координат спрайта находится не в левом верхнем углу, а в центре. Для этого введен метод set\_position, производящий преобразование.

В случае задания позиции через функцию анимации пользователь задает анимирующую функцию и после в каждой итерации цикла отрисовки с помощью метода animate передает параметры, после чего положение спрайта обновляется.

Для отрисовки спрайта на фоне используются два метода. Метод fits\_in проверяет, попадают ли хоть какие-то пиксели спрайта в пределы экрана:  


С его помощью метод отрисовки спрайтов класса App, который будет рассмотрен позже, выбирает, попадают ли спрайты на экран.

Метод draw\_on копирует текстуру спрайта на фон, при этом, из-за того, что выбор области интереса за пределами изображения вызовет ошибку, этот метод обрезает ROI по границам экрана, после чего формирует ROI для текстуры и маски чтобы размер ROI на фоне и на текстуре были равны.



Класс App содержит в себе хранилище для хранения спрайтов в виде std::vector и матрицу для хранения оригинала фонового изображения. По умолчанию, пользователь в главном цикле отрисовки может вызвать функцию render, которая пройдется по всем спрайтам хранилища и вернет фоновое изображение с отображенными на нем спрайтами в соответствии с правилом, что отрисовываются только спрайты, где хоть один пиксель попадает на область экрана.

