МИНОБРНАУКИ РОССИИ

**Федеральное государственное автономное образовательное учреждение высшего образования**

**«Санкт-Петербургский политехнический университет Петра Великого»**

**Институт компьютерных наук и технологий**

**Высшая школа программной инженерии**

**К У Р С О В А Я Р А Б О Т А**

**Создание игры Arcanoid**

по дисциплине «Объектно-ориентированное программирование на Java»

Выполнили

студенты гр.23537/1 А.Ю.Ланцева

А.Д.Осипова

А.В.Хоменко

А.С.Щёголева

Руководитель

Доцент C.M.Бойко

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2017 г.

Санкт-Петербург

2017

Оглавление

[Введение 3](#_Toc501044106)

[1. Детальная постановка задачи 3](#_Toc501044107)

[2. Структура программы 3](#_Toc501044108)

[3. Обработка коллизий 4](#_Toc501044109)

[4. Интерфейс 5](#_Toc501044110)

[5. Тестирование и отладка 5](#_Toc501044111)

[Заключение 6](#_Toc501044112)

[Список литературы 6](#_Toc501044113)

# **Введение**

Arcanoid — это классическая компьютерная игра, в которой нужно разбить все кирпичи на поле при помощи шарика, который игрок может отбивать с помощью платформы. Если шарик проскочит мимо биты и уйдет за нижний край поля, игра проиграна.

В работе рассмотрено создание однопользовательской игры Arcanoid. Для реализации была использована графическая библиотека Swing для языка Java.

**Цель работы:**

Разработать однопользовательскую игру Arcanoid на языке программирования Java с использованием библиотеки Swing.

**Задачи:**

1) выбор инструментального программного обеспечения (язык Java),

2) проектирование игрового процесса игры,

3) проектирование интерфейса,

4) разработка игры Arcanoid.

5) отладка

**Актуальность:**

Актуальность разработки компьютерных игр очень велика, так как потребность в таких продуктах растет с каждым днем. Результатом технологического развития, а также массового распространения персональных компьютеров, планшетов и мобильных устройств, стал все более расширяющийся рынок развлечений, что является веским аргументом в пользу проектов по созданию игр. Таким образом, создание компьютерной игры является очень актуальным и выгодным проектом.

# **Детальная постановка задачи**

Реализуемая игра должна содержать в себе несколько объектов:

1. Двигающийся шарик;
2. Двигающаяся платформа;
3. Набор статичных блоков.

Пользователь с помощью клавиш управляет платформой. Его цель - отбивать шарик платформой и направлять его в блоки, разбивая их. Цель игры - разбить все блоки, не дав шарику упасть (коснуться нижней границы игрового поля).

Конец игры наступает в двух случаях:

1. Все блоки разбиты. Игра считается выигранной.
2. Шарик касается нижней границы игрового поля. Появляется сообщение о проигрыше.

# **Структура программы**

В программе реализовано 5 основных классов и 1 дополнительный:

1. класс AG\_ArcanoidGame:  
   Основной класс, в котором реализована функция main. В данном классе происходит создание графического окна, в котором реализуется вся графика.
2. класс AG\_GamePanel:  
   Класс игрового мира, в котором происходит обработка основных действий игры, таких как отрисовка, изменение состояния игровой системы и обработка смены масштаба.
3. класс AG\_Ball:  
   Класс имитирующий шарик. В нем хранится информация о местоположении шарика, направлении его движения и размеров. Также реализованы функции отрисовки шарика на экран, обработка поведения на следующем фрейме, коллизия со стенкой и поведение при масштабировании окна.
4. класс AG\_Platform:  
   Класс имитирующий платформу. В нем хранится информация о местоположении платформы и ее размерах. Реализованы функции отрисовки, обработки движения по нажатию соответствующих клавиш (влево – a или стрелка влево, вправо – d или стрелка вправо), поведение при масштабировании окна, а также обрабатывается коллизия платформы и шарика.
5. класс AG\_Cube:  
   Класс имитирующий кубик. В нем хранится информация о местоположении кубиков, их состоянии (выбит или нет) и его размере. Написаны функции, обрабатывающие отрисовку, масштабирования и коллизии с шариком(т.е. выбивание).
6. вспомогательный класс AG\_Vector:  
   Класс, хранящий в себе вектор длины 2. Используется в программе для хранения координат объектов и вектора скорости для шарика. Реализованы вспомогательные математические функции для работы с вектором, такие как: сложение, вычитание, скалярное и векторное умножения, нормализация, расчет угла между векторами и поворот на заданный угол.

# **Обработка коллизий**

Данная часть работы включает в себя дополнительные математические расчеты, нужные для правильной обработки отражения шарика от препятствий. В игре рассматриваются три вида коллизий:

1. от стен игрового поля;
2. от платформы;
3. от кубиков.

Рассмотрим каждый вид коллизии отдельно. Все расчеты проводились для системы координат, начало которой находилось в левом верхнем углу, ось х направлена вправо, а ось у – вниз.

* Отражение от стен игрового поля.

Самый простой вид коллизии для реализации. На каждом кадре происходит сравнение координат крайних границ шарика с координатами стенки. При вылете шарика за границу происходит смена направления скорости на противоположную.

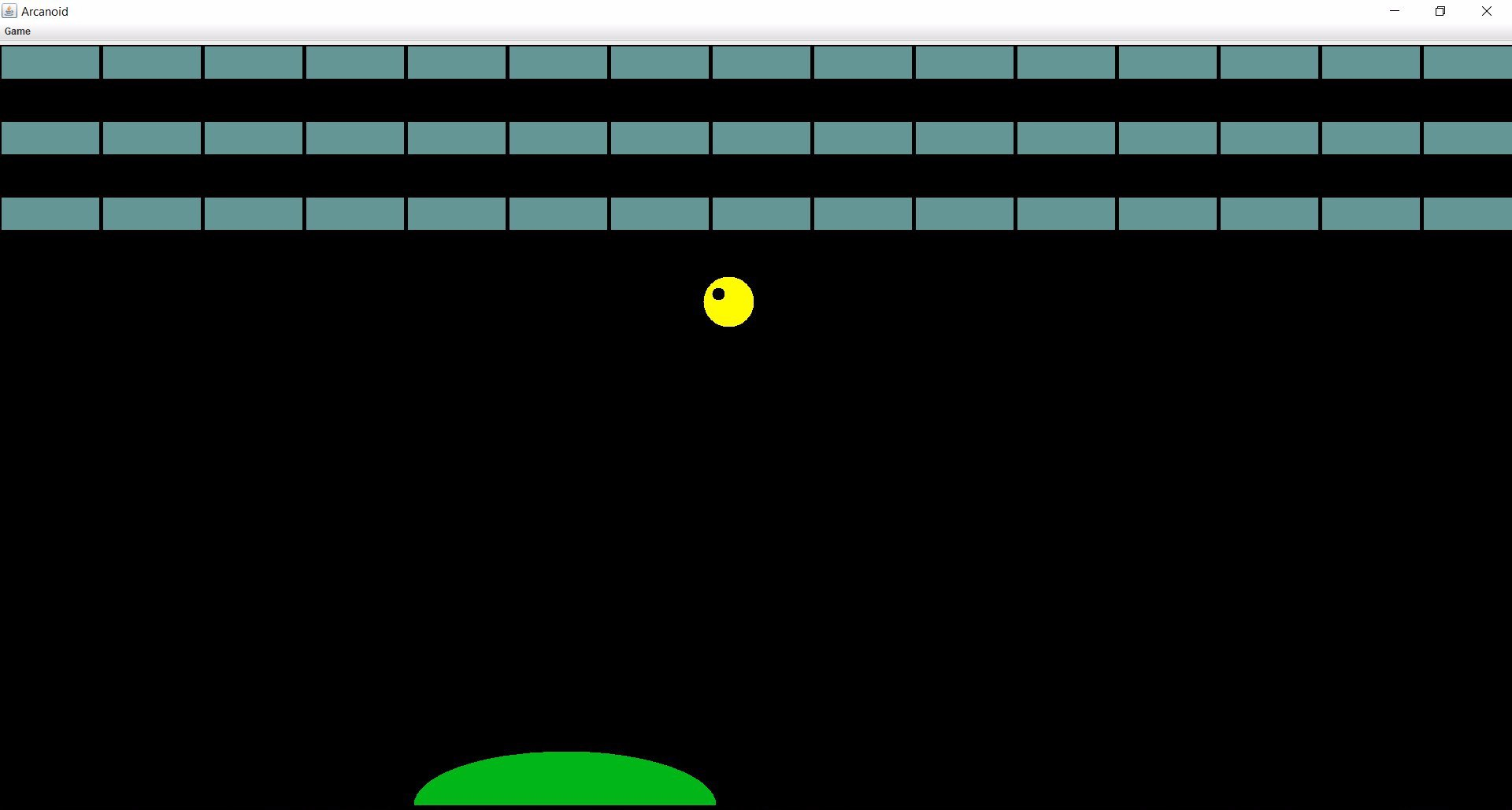
* Отражение от платформы/кубиков

Расчет отражения от платформы и от кубиков практически одинаков, за исключением определения граничной точки платформы/кубика. Далее будем рассматривать «объект», под которым понимаем платформу либо кубик.

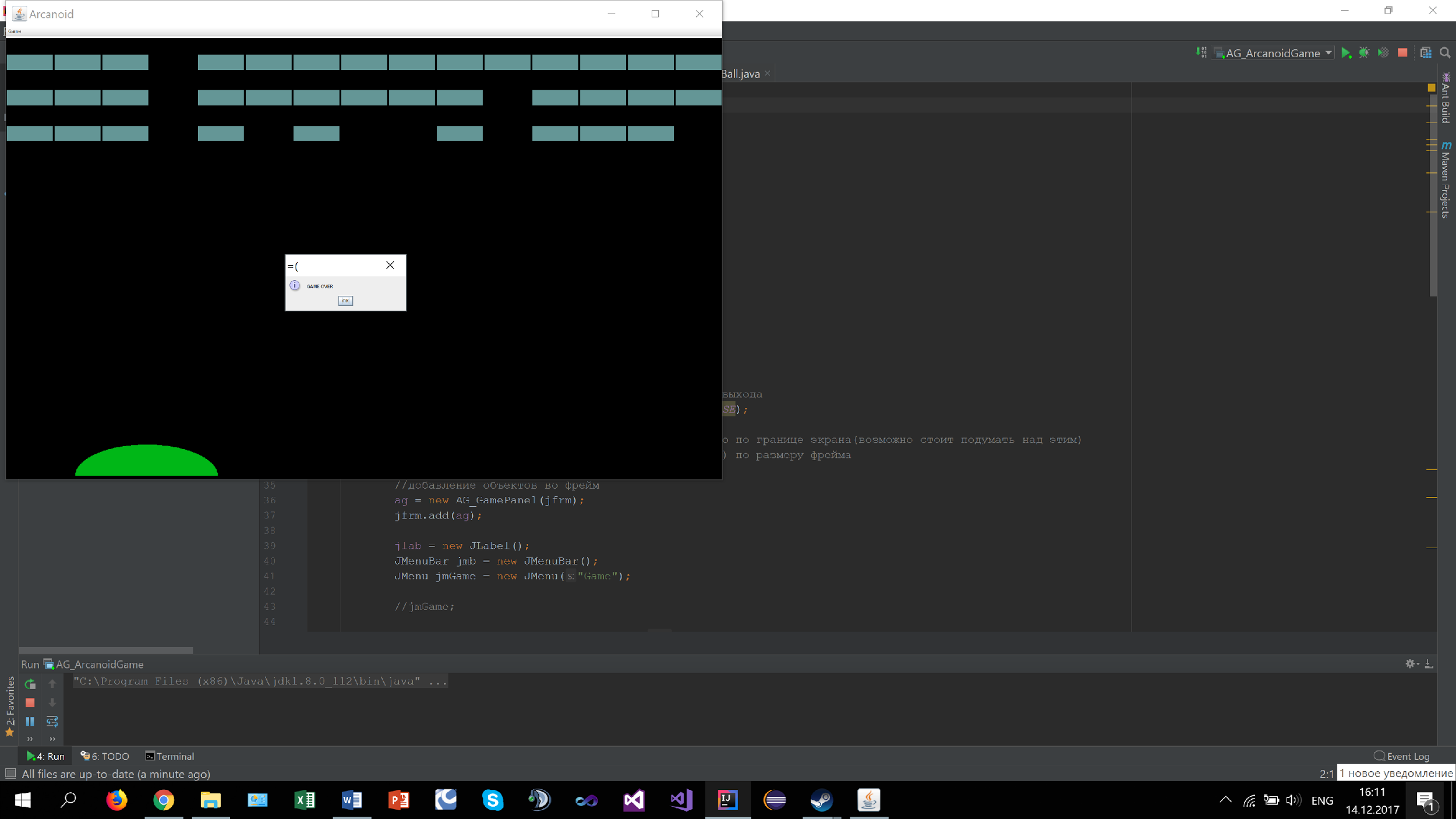
Для расчета коллизии нам не важно местоположение шарика и объекта в пространстве, необходимо лишь знать их местоположение друг относительно друга, поэтому, для удобства, рассмотрим новую систему координат, начало которой совпадает с центром объекта. Для определения момента столкновения, проводим расчет уравнения прямой, проходящей через центры объекта и шарика. Затем подсчитываем длину отрезка этой прямой, ограниченной двумя центрами. Назовем эту длину D. Затем считаем сумму длин двух отрезков, лежащих на данной прямой и ограниченных центром шара/объекта и его границей (расчет граничной точки является тем самым отличием обработки коллизии кубика и платформы). Назовем эту величину Dc. Итого очевидно, что столкновение произошло, если D <= Dc. Если происходит столкновение, то вектор скорости шарика поворачивается на угол, равный углу падения относительно нормали, построенной в точке столкновения.

# **Интерфейс**

Для того, чтобы игра была более понятной и удобной для восприятия, мы решили оставить ее в стиле “минимализм”. Желтый шарик оформлен как пэкмен из одноименной игры. Форма блоков прямоугольная, чтобы облегчить попадание в них.

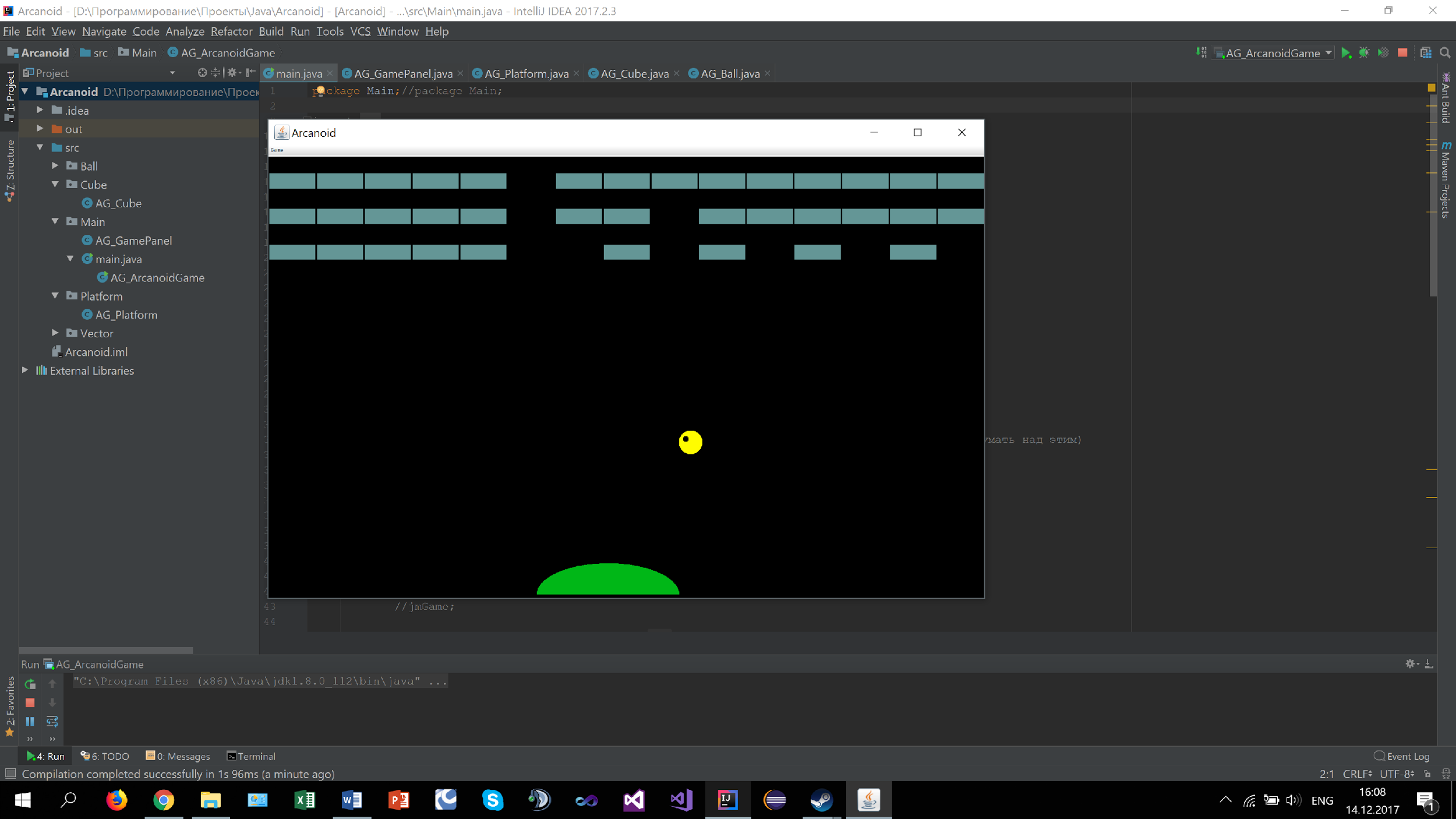
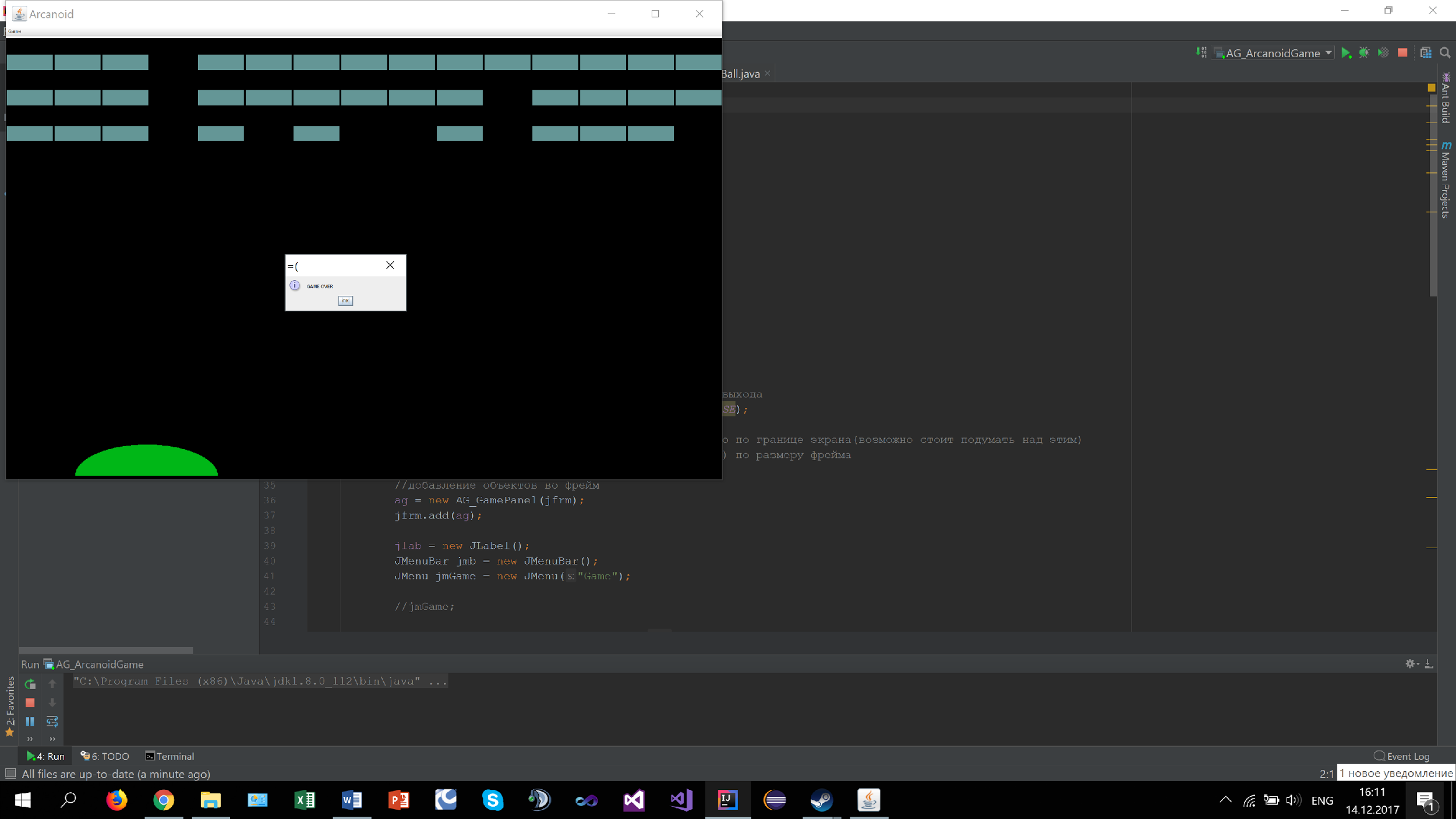


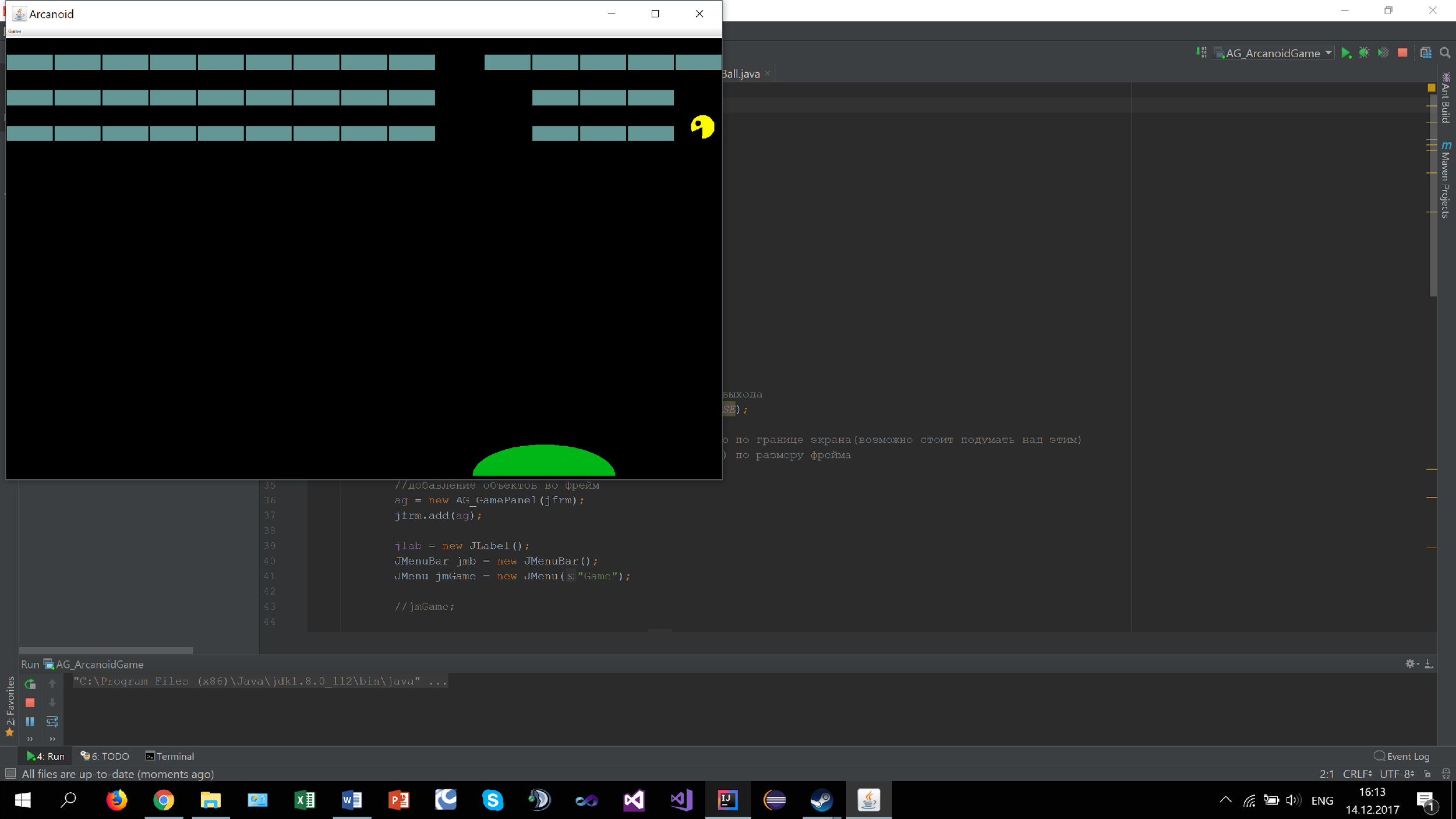
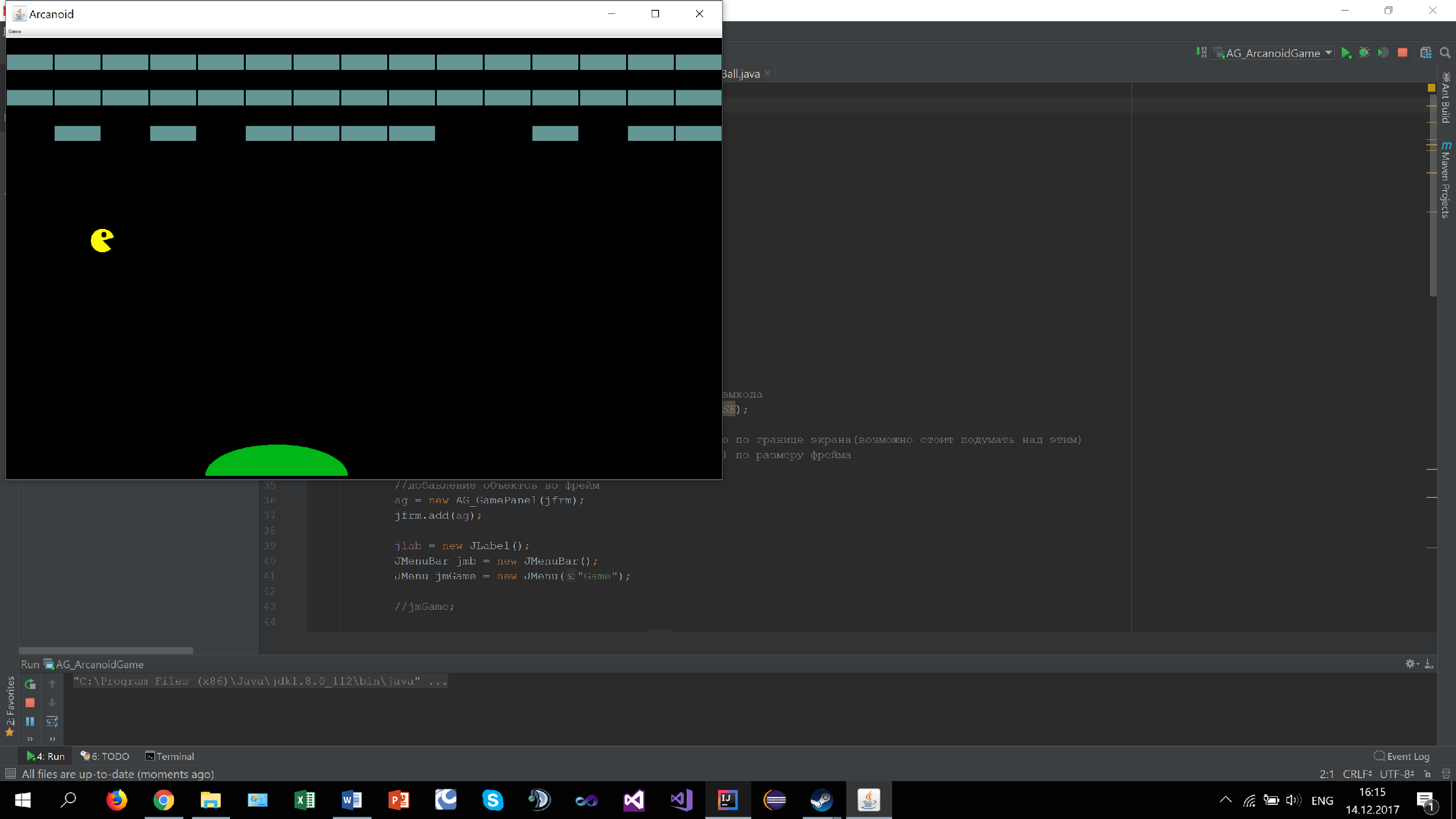
Платформа выбрана овальной, благодаря чему угол отражения шарика не так просто вычислить в уме. По окончании игры появляется всплывающее окно с сообщением.



# **Тестирование и отладка**

Отладка проводилась многократным запуском программы в пошаговом режиме и проверкой каждой функции на правильность обработки. Далее приведены скриншоты работы программы.

# **Заключение**

Была реализована однопользовательская игра Arcanoid на языке программирования Java с использованием библиотеки Swing. Во время выполнения этой работы нами были изучены основы работы с библиотекой Swing. Для некоторых из нас это была первая проба работы в команде, а это является очень важным в современных реалиях опытом. Также хотелось бы отметить, что изучение библиотек для работы с графическими интерфейсами необходимо для тех, кто хочет в дальнейшем заниматься разработкой видеоигр.

# **Список литературы**

1. Г. Шилдт, Java. Руководство для начинающих, пятое издание – М, СПб, Киев: Вильямс, 2012.
2. Г. Шилдт, Swing. Руководство для начинающих – М, СПб, Киев: Вильямс, 2007.