Odd Semester (2021)



**BINUS UNIVERSITY**

**BINUS INTERNATIONAL**

**Student Information**

**First Name:** Titan **Last Name:** Russo **Student ID:** 2101693624

**Course Name:** Introduction to Programming **Course Code:** COMP6502

**Major:** Computer Science **Lecture(s):** Jude Martinez

Minaldi Loeis

**Title of Assignment:** 2 Player Pong

**Type of Assignment:** Final Project **Due Date:** 7-11-2017

# Plagiarism/Cheating

Binus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

# Declaration of Originality

By signing this assignment, I understand, accept and consent to Binus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student: (Name of Student)

1. Titan Russo

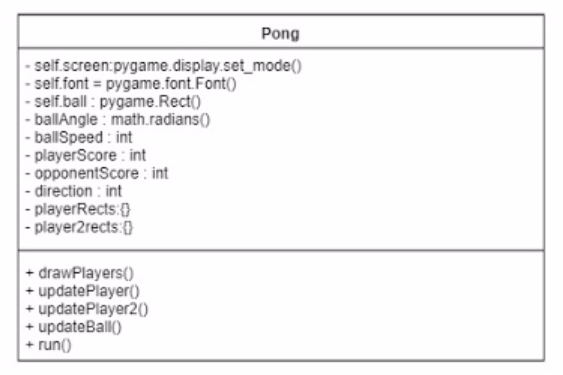
**“2 Player Pong”**

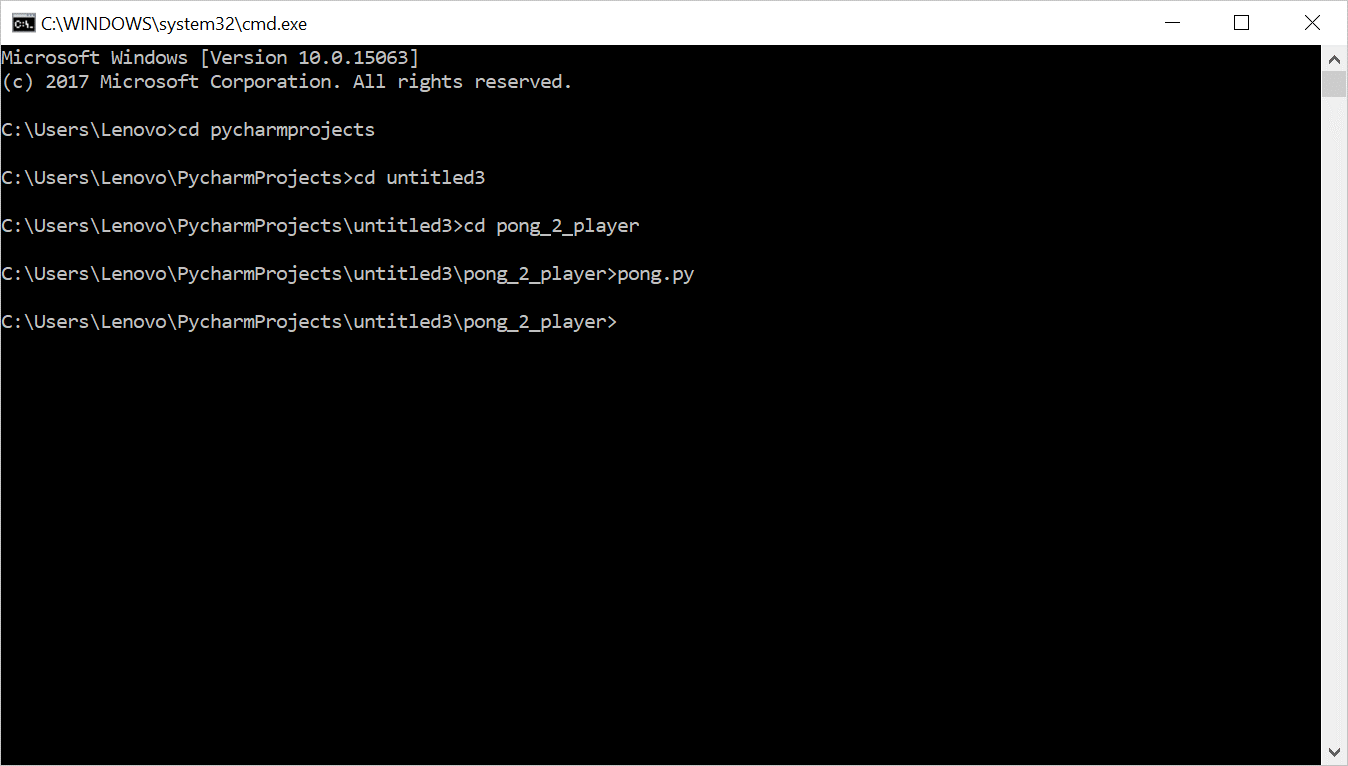
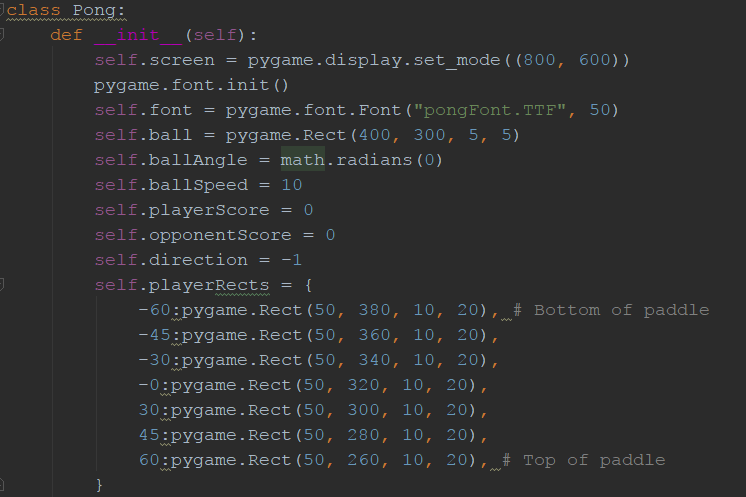
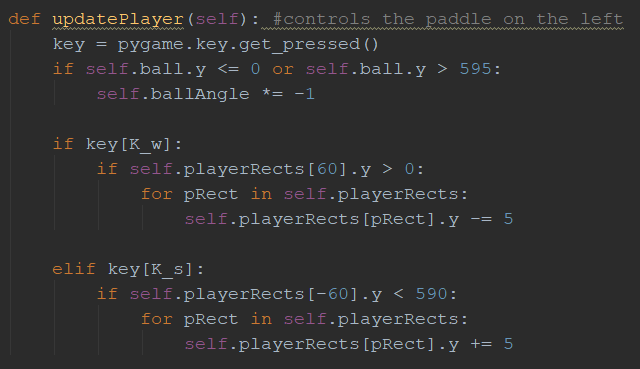
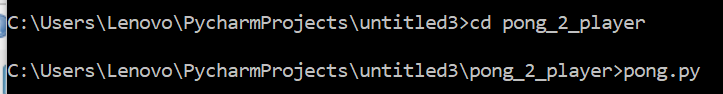
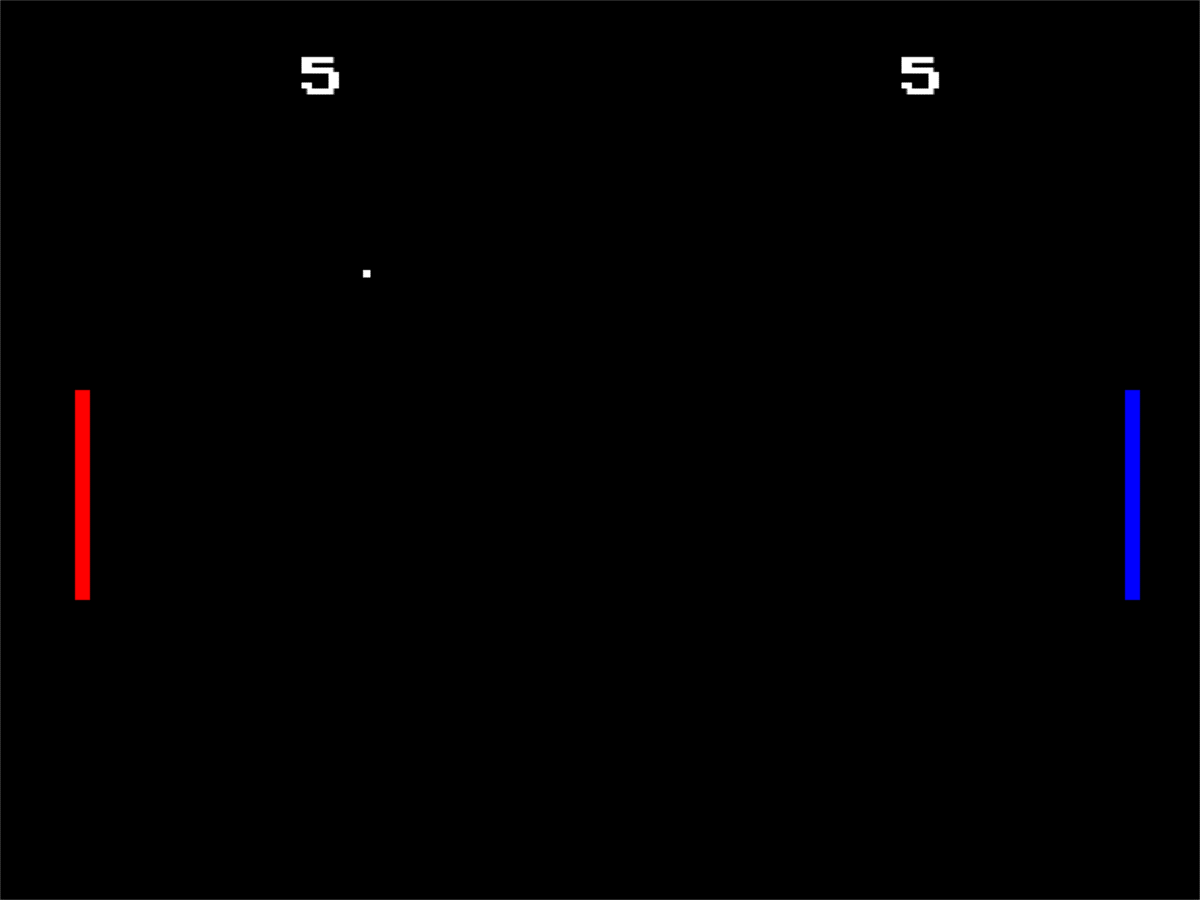
1. **Description**

**The function of this program:**

This Program is a simple 2- player version of th classic arcade game Pong played locally on the same computer. This program’s main focus is to give gamers an opportunity to play the first arcade game ever made.

1. **UML diagram**



1. **Rundown of the program**
2. **pong.py**
   1. simply run the program using a command prompt or terminal to enjoy.
3. **Basic parameters**
   1. These values can be changed to suit the user:
      1. self.screen controls he size of the game window (changes to this parameter will need to be carried over several functions later on)
      2. self.ball determines the starting position and size of the ball in use
      3. self.ballAngle is the ball’s initial trajectory
      4. self.ballSpeed is the ball’s initial speed
      5. self.playerRects is a set of sprites that together form the left/red paddle. (another set of sprites called self.player2rects forms the right/blue paddle)
4. **Various Functions**
   1. **def updatePlayer:**
      1. This function moves the left paddle. The first and last key-value pairs of playerRects are used to set boundaries for the paddle so it wont go beyond the screen. This function is later mirrored for the right paddle with the up and down keys used to control the paddle
      2. The first if statement is what causes the ball tto bounce off the paddle by reversing the angle
   2. **updateBall**
      1. this function is what keeps the ball moving and bouncing off the paddles
      2. it also keeps track of the score by figuring out if the ball passes either paddle or not
   3. **run**
      1. this function simply runs the program
5. **User Manual**
   1. **Simply locate and run the file in command prompt/terminal**
   2. **The “w” and “s” keys on your keyboard move the red paddle up and down, respectively. The blue paddle is controlled using the up and down arrow keys.**