

Department of Computer Science & CINSAM

NKU Summer Programming Workshop 2015



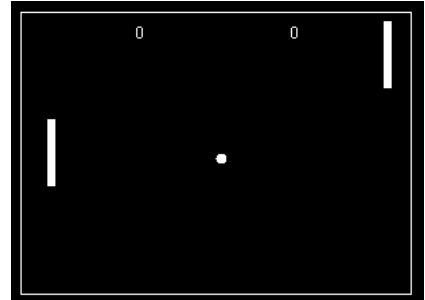
Project 16: Pong!

Implement the game of Pong using the pong.java skeleton on the website.

Pong was one of the earliest arcade games and one of the first home video games as well. The idea is somewhat similar to tennis. The game contains a playing area, a ball and two paddles. The paddles move up and down but not left or right. The ball bounces off the top and bottom of the playing area and off the paddles, but if the ball gets passed a paddle, the other player gets a point. Use the keyboard to control the paddle's up and down motion (for instance, 'a' and 'z' will move the left paddle and 'k' and 'm' move the right paddle).

Implement this game as follows:

1. Implement KeyListener and ActionListener (for the Timer).
2. Use variables y1 and dy1 for the left paddle's current position and its current motion (how fast it is going up or down). You do not need x1 or dx1 because the paddles do not move left or right. Have y2 and dy2 for the right paddle.
3. Have variables for the ball: bx, by, bdx, bdy. At the start of each new ball, bx will be the middle of the playing area and by can either be the middle of the playing area or a random spot. Use random values for bdx and bdy but make sure that bdx is never 0 (or the ball would only move up and down, not left or right). bdy can be 0, this would move the ball horizontally.
4. For actionPerformed, move the ball and each paddle, see if the ball hit a paddle, the top or bottom border, or has gone passed a paddle. If the ball hits a paddle, compute a new bdx and bdy value (see below). If the ball hits the top or bottom border, keep bdx the same but change bdy to have it move in the opposite direction. If the ball gets passed a paddle, add one to the other player's score and start a new ball (or, if the score indicates that a player has won, stop the timer and display a message to indicate who won and that the game is over).
5. When the ball hits a paddle, it should reverse direction (change bdx) and also possibly change bdy based on the location of where it hit the paddle (top or bottom part of the paddle makes the ball travel more sharply). This means you not only have to determine if the ball hits the paddle but where it hits the paddle.
6. In your paintComponent method, display the background (playing area), score, two paddles and ball.



Enhancements:

1. As the score increases, to make the game harder, make the ball faster (this can be done by increasing the range of bdx, bdy values, or by lowering the Timer's value).
2. As the score increases, reduce the size of the paddles (or of the paddle of the player who just scored the point). This is a real challenge because it requires also modifying the code for detecting if the ball hit the paddle because now the size of the paddle changes.
3. Add a "gravity well" somewhere near the center of the playing field that alters the course of the ball.