

Game of Pong

V4

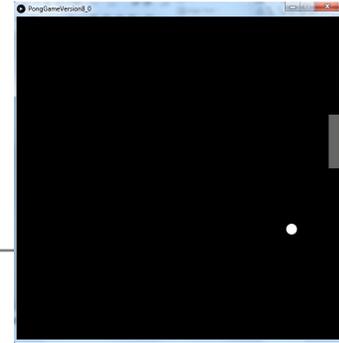
Produced Dr. Siobhán Drohan
by: Mr. Colm Dunphy
 Mr. Diarmuid O'Connor



Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
<http://www.wit.ie/>

Pong Versions - introduction



v1 - **Ball moving** from left to right of screen. Can bounce off top or bottom

v2 - **Mouse controlling the Paddle**

v3 - **Collision detection** (ball bounces back). Changes made only to PongGame

v4 - **Game Over** (when 3 lives gone), Score (lives Lost). Output to Console. Changes made only to PongGame.

v5 - **Tournament** (no of games per tournament default is 5). Changes made only to PongGame.

v6 - new **Player class using arrays** (no statistics)

v7 - Player class using arrays (with **statistics** (Tournament Over - highest, lowest, average score))

v8 - **JOptionPane for I/O** instead of console

v9 - alternative algorithm using **Pythagoras Theorem**



Demo of Pong Game V4.0

PongGameV4.0

- This version **stores game information**:
 - The number of **lives lost**
 - The **maximum lives** allowed per game
 - The **score** of the game
- Game Over
 - when user loses the number of lives allowed per game.
- Changes
 - None in the Ball and Paddle class
 - All changes in PongGameV4.0 class.

Classes in the PongGameV4.0

| PongGame |
|--------------------------------|
| <i>ball</i> |
| <i>Paddle</i> |
| <i>livesLost</i> |
| <i>score</i> |
| <i>maxLivesPerGame</i> |
| <i>setup()</i> |
| <i>draw()</i> |
| <i>hitPaddle(paddle, ball)</i> |

| <i>Paddle</i> |
|-----------------------------|
| <i>Xcoord</i> |
| <i>yCoord</i> |
| <i>paddleHeight</i> |
| <i>paddleWidth</i> |
| <i>Paddle(int, int)</i> |
| <i>update()</i> |
| <i>display()</i> |
| <i>getXCoord()</i> |
| <i>getYCoord()</i> |
| <i>getPaddleWidth()</i> |
| <i>getPaddleHeight()</i> |
| <i>setPaddleWidth(int)</i> |
| <i>setPaddleHeight(int)</i> |

| <i>Ball</i> |
|---------------------------|
| <i>xCoord</i> |
| <i>yCoord</i> |
| <i>diameter</i> |
| <i>speedX</i> |
| <i>speedY</i> |
| <i>Ball(float)</i> |
| <i>update()</i> |
| <i>display()</i> |
| <i>hit()</i> |
| <i>getXCoord()</i> |
| <i>getYCoord()</i> |
| <i>getDiameter()</i> |
| <i>setDiameter(float)</i> |
| <i>resetBall()</i> |

PongGameV4.0 class – global fields

```
//Current game data  
int livesLost = 0;           //keeps track of number of lives lost in current game  
int score = 0;             //high score of the current game  
int maxLivesPerGame = 3;   //maximum number of lives that can be lost  
                           //before the game ends
```

PongGameV4.0 class – draw()

Version 3.0

```
// Update the ball position.  
ball.update();
```



Version 4.0

```
// Update the ball position. If true is returned, the ball has left the display window  
// i.e. a life is lost  
if (ball.update() == true){  
    livesLost++;  
    println("Lives lost: " + livesLost);  
}
```

PongGameV4.0 class – draw()

Version 3.0

```
//Draw the ball at its new location and check for a collision with the paddle
ball.display();

//Set variable to true if ball and paddle are overlapping, false if not
boolean collision = hitPaddle (paddle, ball);

if (collision == true){
    ball.hit();    //the ball is hit i.e. reverses direction.
}
}
```

PongGameV4.0 class – draw()

Version 4.0

```
//If the player still has a life left in the current game,  
//draw the ball at its new location and check for a collision with the paddle  
if (livesLost < maxLivesPerGame){  
    ball.display();  
  
    //Set variable to true if ball and paddle are overlapping, false if not  
    boolean collision = hitPaddle(paddle, ball);  
    if (collision == true){  
        ball.hit(); //the ball is hit i.e. reverses direction.  
        score++; //increase score in the current game by 1, if the player hit the ball.  
        println("Score: " + score);  
    }  
}  
  
//The player has no lives left so the game ends  
else{  
    println("Game Over!");  
    println("You have lost all of your lives: " + livesLost);  
    println("Your final score is: " + score);  
    exit();  
}
```

```
Lives lost: 1  
Score: 1  
Score: 2  
Score: 3  
Score: 4  
Lives lost: 2  
Lives lost: 3
```

```
Lives lost: 1  
Score: 1  
Score: 2  
Score: 3  
Score: 4  
Lives lost: 2  
Lives lost: 3  
Game Over!  
You have lost all of your lives: 3  
Your final score is: 4
```

PongGameV4.0 – sample output

```
Lives lost: 1  
Score: 1  
Score: 2  
Score: 3  
Score: 4  
Lives lost: 2  
Lives lost: 3  
Game Over!  
You have lost all of your lives: 3  
Your final score is: 4
```

Questions?



References

- Reas, C. & Fry, B. (2014) Processing – A Programming Handbook for Visual Designers and Artists, 2nd Edition, MIT Press, London.