Game of Pong

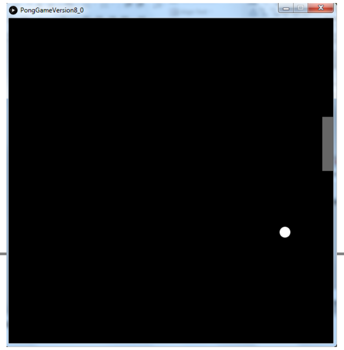
V5

Produced by: Dr. Siobhán Drohan
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 V5.0

PongGameV5.0 – sample output

```
   Score:  1
   Score:  2
♥  Lives lost:  1
   Score:  3
♥  Lives lost:  2
   Score:  4
♥  Lives lost:  3
➡  Game Over!
   Starting a new game...
♥  Lives lost:  1
♥  Lives lost:  2
♥  Lives lost:  3
➡  Game Over!
```

```
   Starting a new game...
   Score:  1
   Score:  2
♥  Lives lost:  1
   Score:  3
♥  Lives lost:  2
♥  Lives lost:  3
➡  Game Over!
   Starting a new game...
   Score:  1
♥  Lives lost:  1
   Score:  2
♥  Lives lost:  2
♥  Lives lost:  3
➡  Game Over!
```

```
   Starting a new game...
♥  Lives lost:  1
   Score:  1
   Score:  2
♥  Lives lost:  2
♥  Lives lost:  3
➡  Game Over!
   Tournament Over!
```

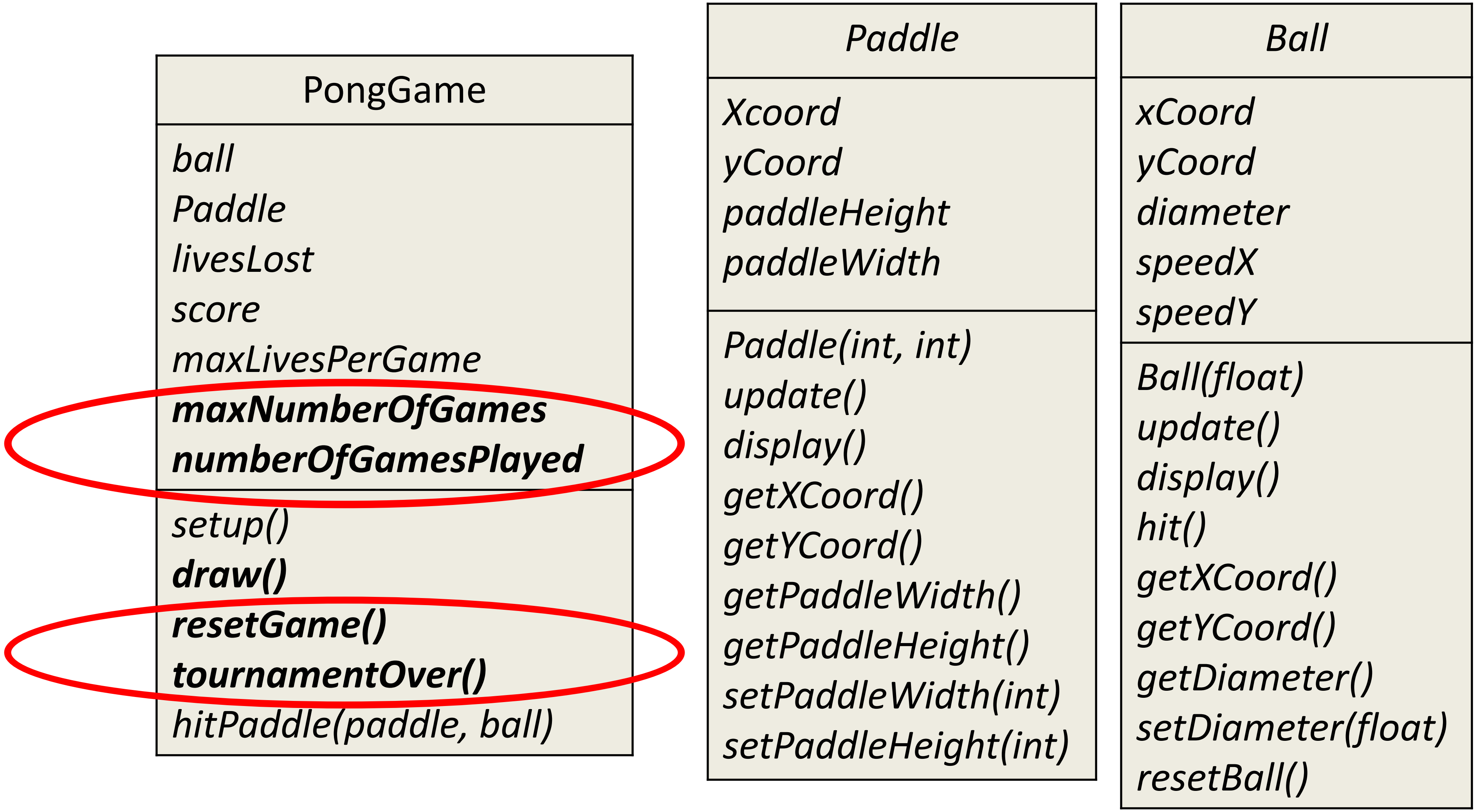
5 games in tournament

3 lives in a game

PongGame**V5.0**

- This version **stores tournament information**:
  - The number of **games in a tournament**.
  - The number of **games played** so far.
- If the number of games in the tournament is over, **end the program**.
- Changes
  - None in the Ball and Paddle class
  - All changes in PongGameV5.0 class.

Classes in the PongGameV5.0



PongGame

ball
Paddle
livesLost
score
maxLivesPerGame
maxNumberOfGames
numberOfGamesPlayed

setup()
draw()
resetGame()
tournamentOver()
hitPaddle(paddle, ball)

Paddle

Xcoord
yCoord
paddleHeight
paddleWidth

Paddle(int, int)
update()
display()
getXCoord()
getYCoord()
getPaddleWidth()
getPaddleHeight()
setPaddleWidth(int)
setPaddleHeight(int)

Ball

xCoord
yCoord
diameter
speedX
speedY

Ball(float)
update()
display()
hit()
getXCoord()
getYCoord()
getDiameter()
setDiameter(float)
resetBall()

PongGameV5.0 class – **global fields**

```
//Tournament data
int maxNumberOfGames = 5;       //maximum number of games in a tournament
int numberOfGamesPlayed = 0;  //num of games played, so far, in a tournament
```

PongGameV5.0 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){
    //displays the ball code
    //if the ball and paddle are overlapping, hit the ball and increase the score by 1
}
//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();
}
```

PongGameV5.0 class – draw

Version 5.0

```
//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){
    //displays the ball code
    //if the ball and paddle are overlapping, hit the ball and increase the score by 1
}
//The player has no lives left so the game ends
else{
    numberOfGamesPlayed++;
    //If the player has more games left in the tournament,
    //display their score and ask them if they want to continue with tournament.
    if (numberOfGamesPlayed < maxNumberOfGames)
        resetGame();
    else
        //the player has no more games left in the tournament
        tournamentOver();
}
```

PongGameV5.0 class – **resetGame()**

```
// method prepares for the next game by resetting the variables //
that store the current game information.
void resetGame()
{
    println("Game Over!");
    println("Starting a new game...");
    livesLost = 0;     //resets the lives lost in the current game to zero
    score = 0;         //resets the score of the current game to zero
}
```

PongGameV5.0 class – **tournamentOver** ()

```
// method displays the player information, before exiting
// the program.
void tournamentOver ()
{
    println("Game Over!");
    println("Tournament Over!");
    exit();
}
```

PongGameV5.0 – sample output

```
Score:  1
Score:  2
Lives lost:  1
Score:  3
Lives lost:  2
Score:  4
Lives lost:  3
Game Over!
Starting a new game...
Lives lost:  1
Lives lost:  2
Lives lost:  3
Game Over!
```

```
Starting a new game...
Score:  1
Score:  2
Lives lost:  1
Score:  3
Lives lost:  2
Lives lost:  3
Game Over!
Starting a new game...
Score:  1
Lives lost:  1
Score:  2
Lives lost:  2
Lives lost:  3
Game Over!
```

```
Starting a new game...
Lives lost:  1
Score:  1
Score:  2
Lives lost:  2
Lives lost:  3
Game Over!
Tournament Over!
```

5 games in tournament

3 lives in a game

Questions?

References

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