Game of Pong

V8 Developing the game further

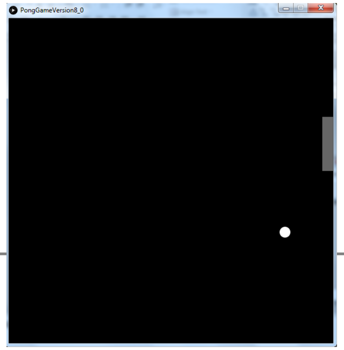
Produced by:

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)

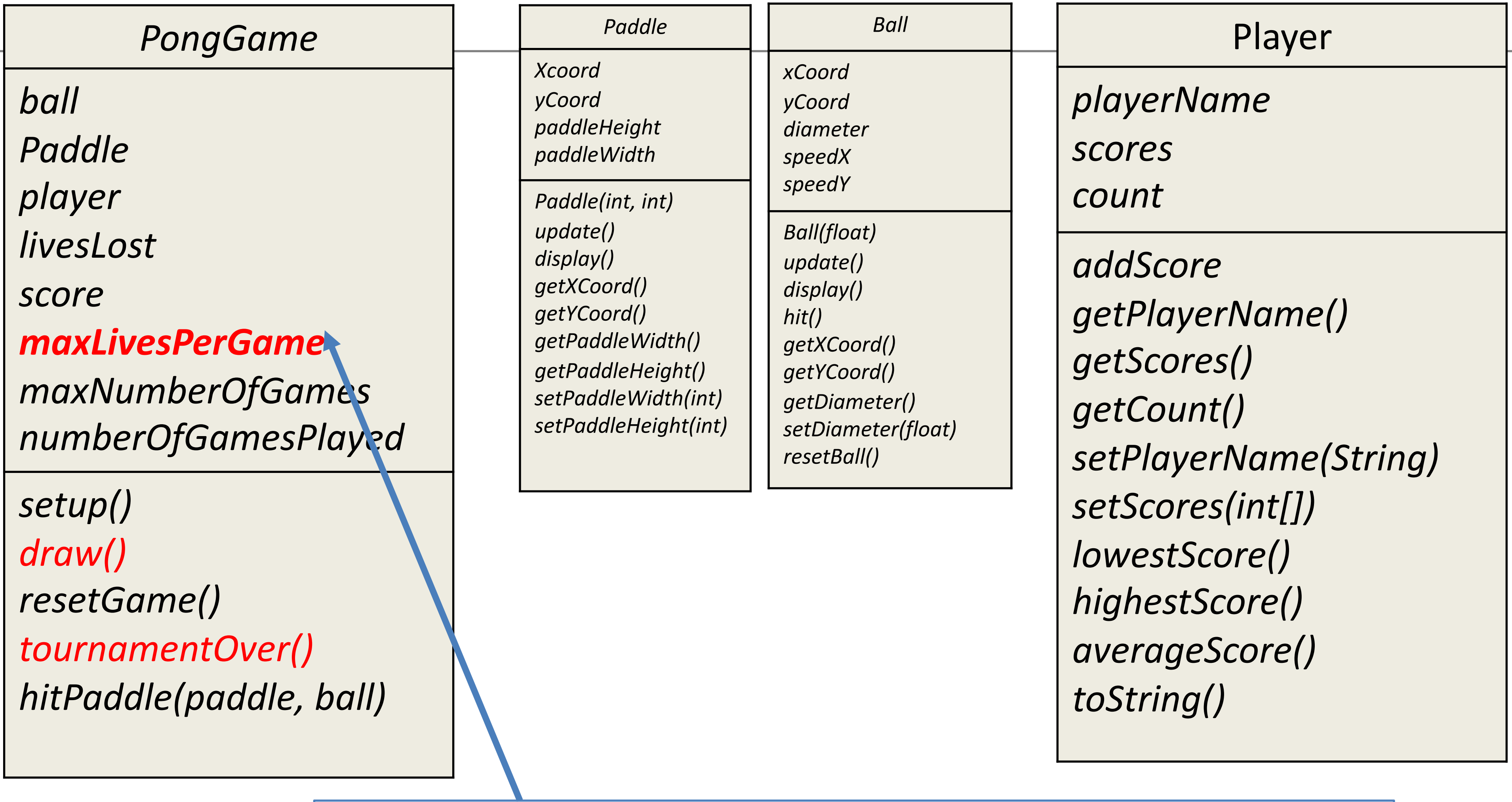
v7 - Player class using arrays (with **statistics** (Tournament Over - highest, lowest, average score))

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

v9 - alternative algorithm using **Pythagoras Theorem**

Demo of
Pong Game V8.0

Classes in the PongGame**V8.0**



PongGame

ball
Paddle
player
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()

Player

playerName
scores
count

addScore
getPlayerName()
getScores()
getCount()
setPlayerName(String)
setScores(int[])
lowestScore()
highestScore()
averageScore()
toString()

JOptionPane allows user input, during the running of the program.

We use this input to make changes in the game.

A few things to note

- We only use data input or data output in the PongGame(Driver) class.

- This is to ensure that the ‘user of classes’ (PongGame) gets to decide how the data is input and output.

- This is why **toString**() is useful
  - it returns a string version of an object of a class
  - then the user can decide how to show it e.g. on the console or via JOptionPane.

import JOptionPane

In order to use **JOptionPane**,
we must **import swing** at the top of the file.

```
import javax.swing.*;

//Objects required in the program
Ball ball;
Paddle paddle;
Player player;
:
```

PongGame

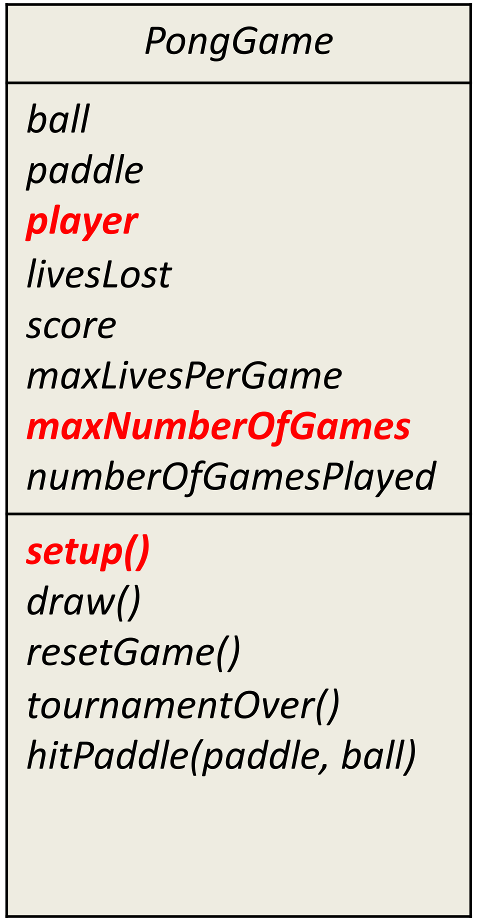
ball
paddle
player
livesLost
score
maxLivesPerGame
maxNumberOfGames
numberOfGamesPlayed

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

Reading in maxNumberofGames

```
int maxNumberOfGames;
//code omitted
maxNumberOfGames =
    Integer.parseInt (JOptionPane.showInputDialog(
          "Welcome to the Pong Tournament\n\n
          Please enter the number of games you would like to play:",
          "3"));

 player = new Player (JOptionPane.showInputDialog(
    "Enter the player name (max 6 chars: "), maxNumberOfGames);
```


PongGame

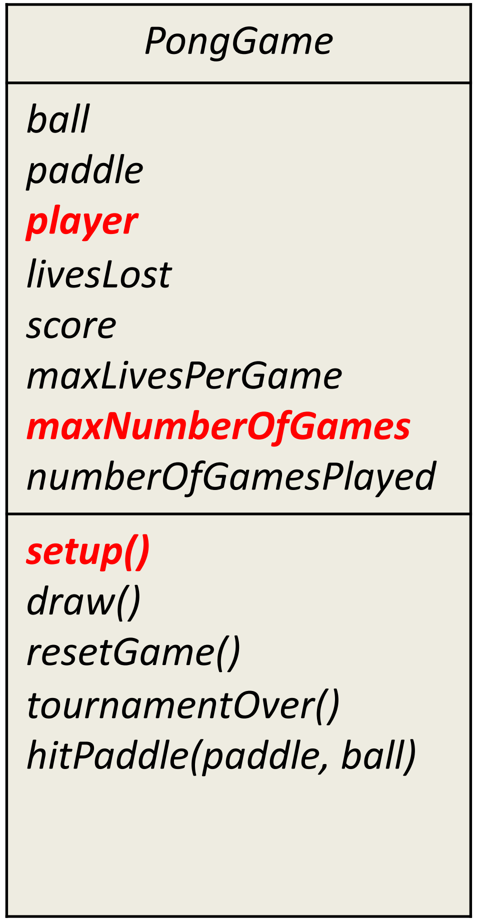
ball
paddle
player
livesLost
score
maxLivesPerGame
maxNumberOfGames
numberOfGamesPlayed

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

Reading in maxNumberofGames

maxNumberOfGames is read in

```
int maxNumberOfGames;
//code omitted
maxNumberOfGames =
      Integer.parseInt(JOptionPane.showInputDialog(
            "Welcome to the Pong Tournament\n\n
            Please enter the number of games you would like to play:",
            "3"));
 player = new Player (JOptionPane.showInputDialog("Enter the player
                     name (max 6 chars: "), maxNumberOfGames);
```



PongGame

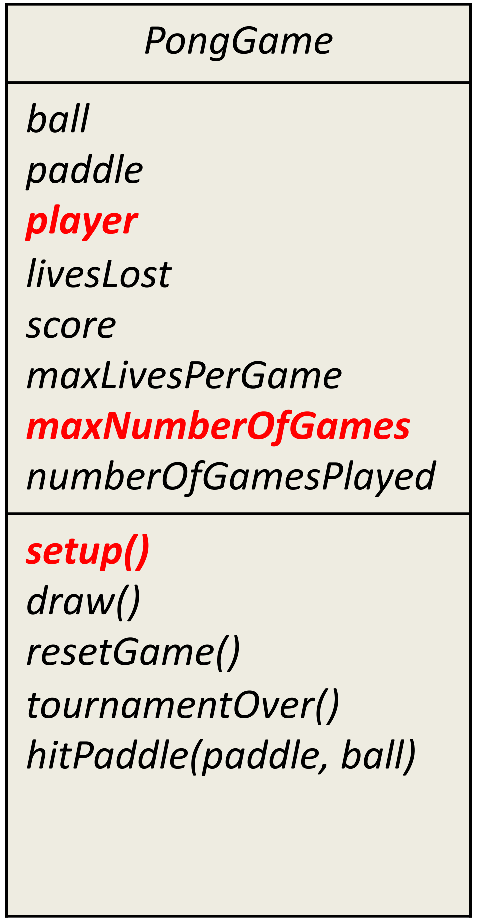
ball
paddle
player
livesLost
score
maxLivesPerGame
maxNumberOfGames
numberOfGamesPlayed

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

Reading in maxNumberofGames

```
int maxNumberOfGames;
//code omitted
maxNumberOfGames =
    Integer.parseInt(JOptionPane.showInputDialog(
          "Welcome to the Pong Tournament\n\n
          Please enter the number of games you would like to play:",
          "3"));
player = new Player (JOptionPane.showInputDialog("Enter the player
                   name (max 6 chars: "), maxNumberOfGames);
```

The Player constructor is called and the JOptionPane input is passed into the constructor.



PongGame

ball
paddle
player
livesLost
score
maxLivesPerGame
maxNumberOfGames
numberOfGamesPlayed

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

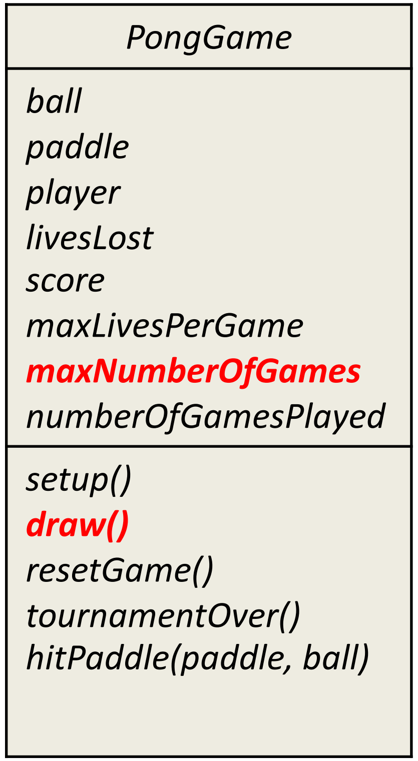
Adding choice during the game

- Having read in the maximum number of games a player can have,
  the **player is asked at the end of each game**
  if they wish to continue.
  - If they choose to end, their tournament is over.
- When max number of games as read in, is reached
  - they will finish without being asked.

Adding choice during the game

V7

```
//If the player has no lives left in the current game
else{
    //add the score of the current game to the array in player
    player.addScore(score);
    numberOfGamesPlayed++;
    //If the player has more games left in the tournament,
    //display their score and ask them if they want to
    //continue with the tournament.
    if (numberOfGamesPlayed < maxNumberOfGames){
      resetGame();
    }
    else{
      //the player has no more games left in the tournament
      tournamentOver();
    }
```



PongGame

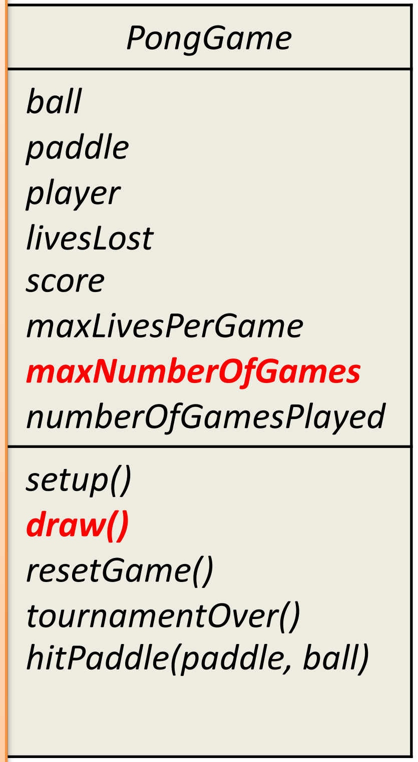
ball
paddle
player
livesLost
score
maxLivesPerGame
maxNumberOfGames
numberOfGamesPlayed

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

Adding choice during the game

V8

```
//If the player has no lives left in the current game
else{
    player.addScore(score);
    numberOfGamesPlayed++;
    if (numberOfGamesPlayed < maxNumberOfGames){
        int reply = JOptionPane.showConfirmDialog(null,
            "Game Over! You scored " + score +
            ".\nWould you like to play the next game in your  tournament?",
            "Play next game?", JOptionPane.YES_NO_OPTION);
        if (reply == JOptionPane.YES_OPTION){
            resetGame();
        }
        else{
            tournamentOver();
        }
    }
}
```



PongGame

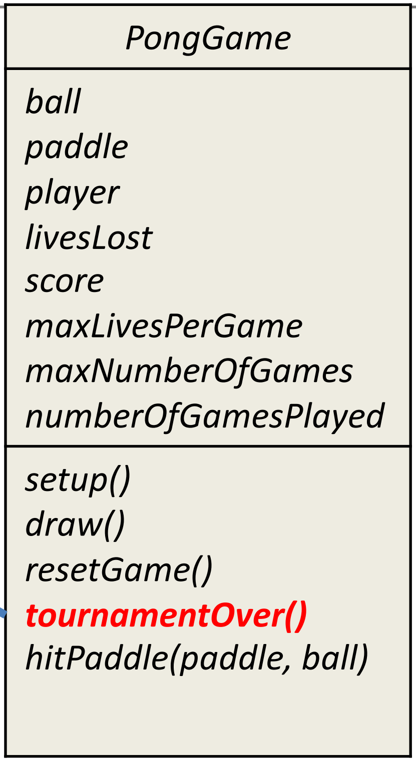
ball
paddle
player
livesLost
score
maxLivesPerGame
maxNumberOfGames
numberOfGamesPlayed

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

We added extra functionality here, based on our new field **maxNumberOfGames** and **JOptionPane**.

JOptionPane for output

```
void tournamentOver ()
{
      JOptionPane.showMessageDialog(null,
              player.getPlayerName() +
              ", your tournament is over! \n\n"     +
               "Number of games played: " +
              numberOfGamesPlayed    + "\n\n”+
               player.toString()  +
              "\n\nHighest Score: ” + player.highestScore()  +
              "\nLowest Score:  "      + player.lowestScore()    +
              "\nAverage Score: "      + player.averageScore());
      exit();
 }
```



PongGame

ball
paddle
player
livesLost
score
maxLivesPerGame
maxNumberOfGames
numberOfGamesPlayed

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

The same data is being output, just in a better way… we are using **JOptionPane** instead of the console.

Questions?

References

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