

Lab Week9. Handling Multiple Classes and Objects

Last Week

- Using PImage
- Designing and implementing a Class
- Writing methods
- Writing constructor

Learning Objectives

- Designing & Implementing Multiple Classes for a more complex problem
 - Constructors & methods
- Creating instance objects and using them for a simple game
- Detecting a crash using pixel colours over an area

Today's **portfolio** exercise is formative **CHECKPOINT 1** for your assignment. It contains many elements which can be reused (with a little tweaking) when developing your assignment. Ask for feedback on your code and explanations for anything developed during the lab that you are unsure about

Please open any unsigned portfolio exercises – please ensure all exercises so far are complete. We will be chasing you next week if you are behind.

Note : If you see a nested Class error – you've probably named a Class within your code with the same name as the sketch file (change one of them).

Resources

- Lecture Notes – Multiple Classes
- Tutorial <https://processing.org/tutorials/objects/>
- Processing website – reference
- Background image on moodle

Save your code after each exercise in a new week 9 directory

Ex1. Open a new Processing sketch, paste in the code below and save it as week 9/**Defenderz**. On moodle (week 9) you will find a background **gif** file. Download the file to your Defenderz directory. The code below will give you a horizontal scrolling background image which wraps around.

```
PImage background;
int x=0; //global variable background location

void setup() {
  size(800,400);
  background = loadImage("spaceBackground.jpg");
  background.resize(width,height);
}
```

```

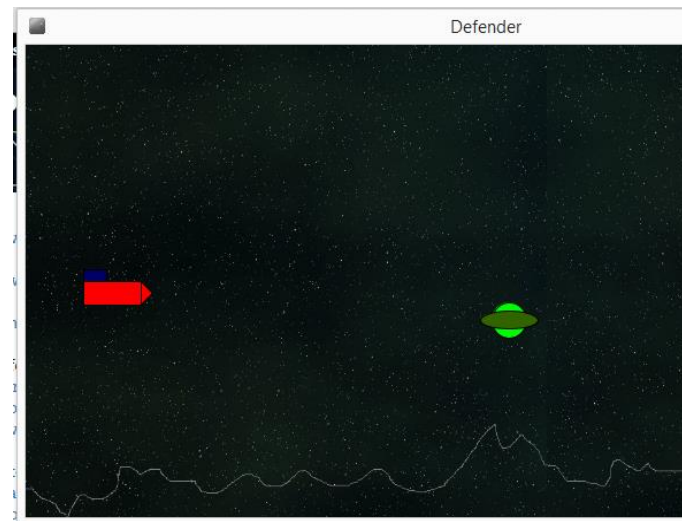
void draw ()
{
    image(background, x, 0); //draw background twice adjacent
    image(background, x+background.width, 0);
    x -=4;
    if(x == -background.width)
        x=0; //wrap background
}

```

Defenderz Game

1st prototype has

- alien (moving left, and perhaps up and down).
 - color ALIEN1 = color(0,255,0);
 - color ALIEN2 = color(50,100,0);
- A defender
 - Moved up and down by player
 - Can crash into alien
 - Detect by alien colour (area just in front of the defender)
 - color ALIEN1 = color(0,255,0);
 - color ALIEN2 = color(50,100,0);



What Classes do we require?

What constructors & methods for each?

Add the Defender class (do not use the same name as the sketch! – **nested class error**)

```

//draw a defender
fill(255,0,0);
rect(x,y,50,20);
triangle(x+50,y,x+50,y+20,x+60,y+10);
fill(0,0,100);
rect(x,y-10,20,10);

```

Add a defender instance with user interaction (key press move up and down) and test it

Add an Alien class

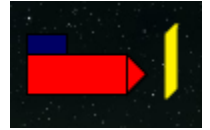
```

//draw an alien
fill(ALIEN1);
ellipse(x,y,30,30);
fill(ALIEN2);
ellipse(x,y,50,15);

```

Add an instance of an Alien and test it

Add a **crash Boolean method** to Defender. We could use a distance measure as we have previously (asserting that a crash has taken place if the two objects are within a certain distance) but here we'll use a different technique instead.



A crash has taken place if we detect the alien's colours just in front of the defender ship. To do this we need to check a range of pixels (e.g. yellow area in image) and test whether ALIEN1 colour OR ALIEN2 colour is present – how?

FOR LOOP – repeat a block of actions a set number of times

Ensure you have drawn the alien on the screen (in draw event) before checking for a crash!!

Portfolio Exercise – complete the 1st prototype of the game

Use a **gameMode** variable to stop the game if a crash occurs

Allow the alien to move up,down in sequence (as well as left

If the alien goes off the screen a new alien should appear on the right

Extension exercise – 2nd Prototype game (very useful for assignment)

Allow the defender to fire a bullet

Add **isShot** method to the alien class

Open up the background image, make a copy and draw on some objects (e.g. asteroids) in a specific (R,G,B) colour. You can then add this colour to your Defender crash detection method to make the game harder.

Add multiple aliens

Add scoring or an explosion.