

CS1013 - Programming Project

Dr. Gavin Doherty

ORI LG.19

Gavin.Doherty@cs.tcd.ie

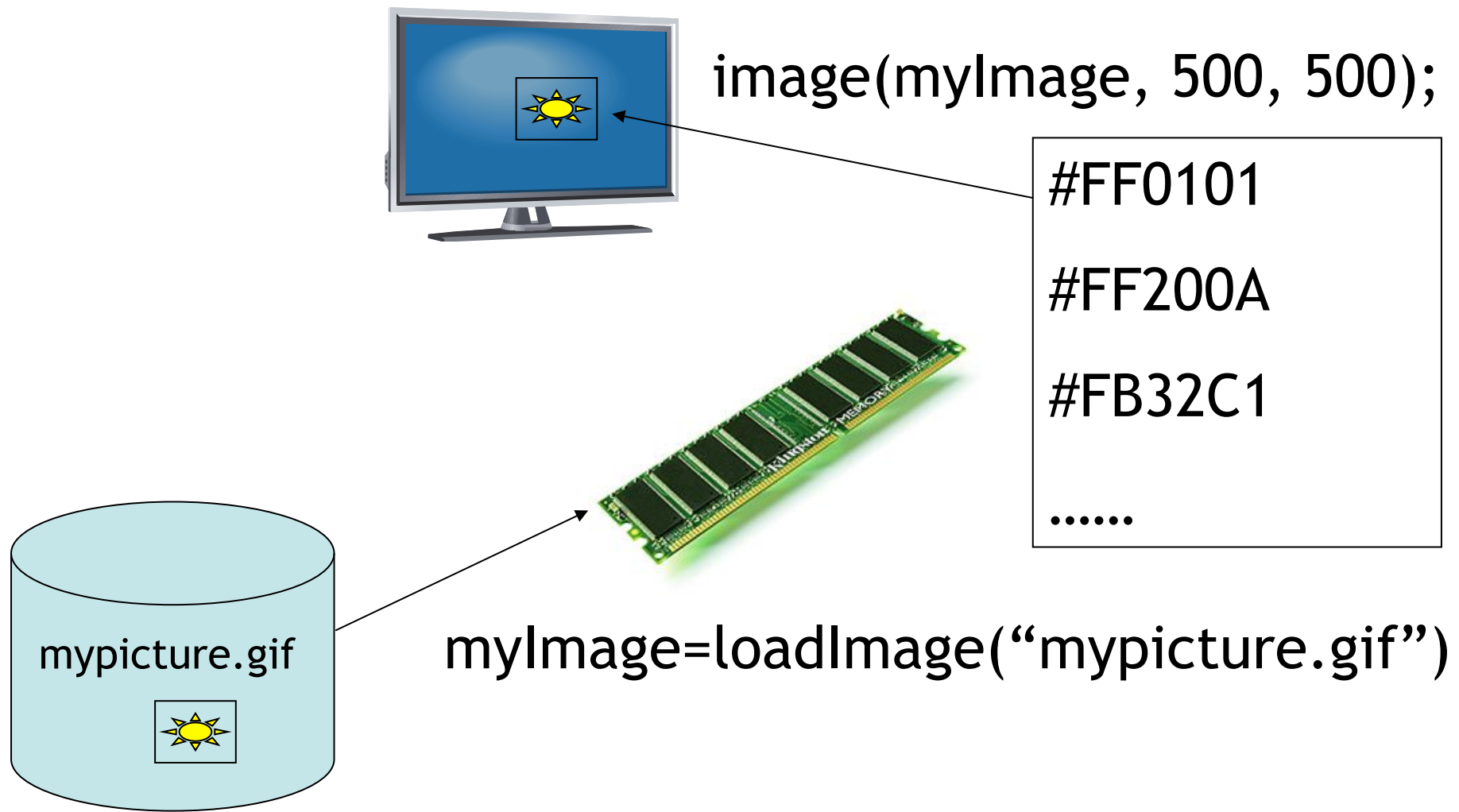
Review

- Move the center of the computer player towards the ball position. Define a new method eg. *auto_move* in *Player* class. Pass the ball object to this method.
- Reset/pause - use Boolean variable *paused*, set it to true when ball goes off the screen. Define *mousePressed* to make *paused* false.
- Add *lives* variable to *Player* class. Message display- use *if(paused)* statement in *draw()*;

Displaying Images

- An image file is a file on the hard disk.
- There are a lot of different formats, often compressed etc.
- To display an image, you would really want it available in memory, decoded etc.
- That's why we have a `loadImage` primitive, separate to displaying the image.
- The `PImage` class stores the image in a format which is convenient for rendering on the screen.

Images



Displaying Images

```
PImage myimage;  
void setup() {  
    size(400, 400); background(255);  
    myimage= loadImage("invader.GIF");  
}  
void draw(){  
    image(myimage, 100, 100);  
}
```

- Try it:
 - Load new Sketch, Save blank Sketch.
 - Add it to your program with <sketch menu>-><add file>
 - Copy above code
 - Copy image name and extension -case Sensitive!
 - Replace invader.GIF above with your image name... RUN CODE!

Arrays

- Suppose we want to have lots of instances of the same shape in our sketch.
- Could draw them all individually - messy code, how do you keep track of them?
- Could store them as variables, don't want to have to repeat the same code for each one.
- Could store them in an array and process the whole array.

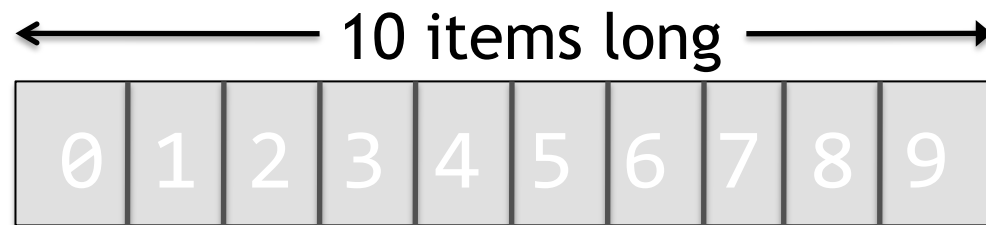
Loops

- In a way, our processing programs already contain one (infinite) loop, as `draw()` is called over and over again.
- We can have loops inside `draw()`, but these **must** terminate or our programs will hang and never display anything.
- In the `draw()` method, we have a loop to draw each of the individual objects.

The for loop

```
for (int i= 1; i< 10; i++)  
{  
    println("The number is: " + i);  
}
```


Arrays of objects



- Let's produce a sketch in which we have lots of circles which move towards one point.
- Use `ellipse(x, y, width, height)` ;
- `width=height`, gives us a circle.
- Use **length** attribute of array to tell us how many elements are in it.

Input

- One way of dealing with user input is to use the current value of the mouse.
- For the circles program, we can use the current location of the mouse in calculating the movement of the circles.

Arrays of objects

- General pattern will be:
- Declare the array - this creates a placeholder for it.
- Create the empty array with a particular number of positions.
- Fill in the array
 - For each position in the array, create a new object.
- To draw the array
 - For each position in the array, call the draw method for the object in that position.

Code outline

```
Circle myCircles[];
```

```
void setup() {  
    myCircles = new Circle[10];  
    init_array(myCircles);  
}
```

```
void draw(){  
    draw_array(myCircles);  
}
```

```
void draw_array(Circle theArray[]){  
    for(int i=0; i<theArray.length; i++)  
        theArray[i].draw();  
}
```

Circle class

```
class Circle {
    int x, y; int radius=10;
    color circleColor;

    Circle (int xpos, int ypos) {
        x=xpos; y=ypos;
        circleColor = color(int(random(0,255)),
                            int(random(0,255)),int(random(0,255)));
    }
    void draw(){
        fill(circleColor);
        ellipse(x, y, radius, radius);
    }
    void move(int targetX, int targetY){
        if(x<targetX) x++; else x--;
        if(y<targetY) y++; else y--;
    }
}
```

CS1013 Programming Project

```
Circle myCircles[];
void setup() {
    myCircles = new Circle[10];
    size(SCREENX, SCREENY);
    noStroke();
    init_array(myCircles);
}
void draw(){
    move_array(myCircles);
    draw_array(myCircles);
}
void init_array(Circle theArray[]){
    for(int i=0; i<theArray.length; i++){
        theArray[i] = new Circle(int(random(0, SCREENX)), int(random(0,
            SCREENY)));
    }
}
void draw_array(Circle theArray[]){
    for(int i=0; i<theArray.length; i++){
        theArray[i].draw();
    }
}
void move_array(Circle theArray[]){
    for(int i=0; i<theArray.length; i++){
        theArray[i].move(mouseX, mouseY);
    }
}
```