

Mobile Computing

Laboratory 3

1 Fire!

- In today's lab, we're going to create a fire effect similar to the one found at <https://www.youtube.com/watch?v=ErWOpKxBWZM>
- The intention is to show how something cool can be made with very little actual code.
- I've provided a skeleton which you can download from moodle.
- We're going to take it step by step.

2 The existing code

- Take a look at the provided code, making sure to read the comments in the file.
- Try changing the update method to have the following contents, and run your code.

```
putPixel(20, 5, 200);
```

- You should see a pixel appear on the screen in position x=20, y=5. Try changing it to:

```
putPixel(10, 5, 200);  
putPixel(11, 10, 150);
```

3 Draw a line

- Now, use a for loop to draw a solid line with intensity 255 across the bottom line (y=100). Your for loop should iterate over all x values (up to width), with y value height-1 and intensity 255.
- This line used a fixed intensity of 255. Now, try setting the intensity to a random number between 0 and 255, so that each pixel has a different intensity. Code to generate random numbers is shown below. The code creates a random number generator and then uses it to generate a random number. Note that in your code, you should just have one random number generator which you use over and over again to generate numbers. Note that you will need to import java.util.Random;

```
Random myRandom=new Random();  
int intensity = myRandom.nextInt(256);
```

- When you run your code, you should see the system continuously replacing the random numbers with new ones, so your line should be updating randomly every 50 milliseconds.

4 Fire!

- Weirdly, that random number generator plus a little trick is going to give us fire.
- The next thing you need to add to your update method is a double for loop that loops over every x and y value in the image except for the bottom row (every x up to width and every y up to height-1)
- It should set the value of every pixel to the average of it's own value and the three pixels below it.
- Then, run your code, and it should look like fire...

5 Next steps (just for fun)

- The fire is radiating without losing any energy, so the flame seems to carry on for too long. Try subtracting 1 from the calculated average to make the fire lose energy as it rises.
- Try messing around with the parameters in the putPixel and getPixel functions to see if you can change the colour of the fire. Try to see if you can make it grey. Or green, green looks insane!