PAT 204/504: Creative Coding (Fall 2024)

Homework 1: Bouncing “Hello, World!”

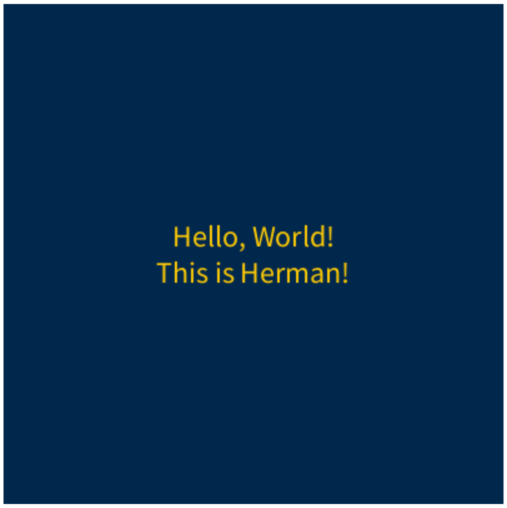
Due at 11:59pm ET on September 6

1 Bouncing “Hello, World!” (5 points)

“Hello, World!” is usually the first program we write when learning a new programming language. Although that’s not quite the case for Processing, we can do something cooler in Processing. You will start from writing a Processing sketch that

- renders “Hello, World!” in the first line and “This is {YOUR NAME HERE}!” in the second
- with a Michigan maize (#FFCB05) font color
- of a font size of 24 pixels (or smaller if your name doesn’t fit)
- with the text centrally aligned
- at the center (both horizontally and *vertically*) of a 400×400 canvas
- with a Michigan blue (#00274C) background

This should look something like:



Hello, World!
This is Herman!

Now, you want to make it a *bouncing* “Hello World!” That is, the text box should

- move at a preset speed of `speedX` (defaults to 1) along the x-axis and
- move at a preset speed of `speedY` (defaults to 2) along the y-axis and
- bounce back when it hit the walls

Here is an example of what it should look like and a template sketch for you to start from:

```
String msg = ”Hello, World!\nThis is {YOUR NAME HERE}!”;
float fontSize = 24;  // font size
boolean saveFrames = false;  // whether to save the frames
float speedX = 1;  // speed along the x-axis
float speedY = 2;  // speed along the y-axis

// DEFINE OTHER GLOBAL VARIABLES YOU NEED HERE

void setup() {
  // YOUR CODE GOES HERE
}

void draw() {
  // YOUR CODE GOES HERE

  // Save frames for making a GIF
  if (saveFrames) {
    saveFrame(”frames/###.png”);
  }
}
```

Please submit both your code and your final rendering as an animated GIF. You will receive zero credit if the code is missing. The template includes a boolean switch `saveFrames` that controls if the `saveFrame()` function at the end of the `draw()` function will be executed. When `saveFrames` is set to `true`, each frame generated by a call of the `draw()` function will be saved to the folder `frames/` in the project directory. You can then use the built-in movie maker (“Tools” → “Movie Maker”) to create an animated GIF. Also, the GIF standard does not support 60 fps, so you might want to add `frameRate(30)` to your `setup()` function to set the frame rate to 30 fps.

Hints

- You might want to search in the documentation to find the functions that you need.
- You can use `\n` to start a new line when calling `text()`. For example,

  ```
  text(”Hello, World!\nThis is ______!”, height / 2, width / 2)
  ```

  will be rendered as

```
Hello, World!
This is ______!
```

- You can pass a second argument to `textAlign()` to vertically center the text.
- You might find `textWidth()`, `textAscent()` and `textDecent()` useful in estimating the size of the text box. Note that these functions must run after you set the desired font size. Also note that sometimes it can be tricky to get the exact size of a text box in Processing, so feel free to make manual adjustments to the estimated values.

2 Bonus: Controlling Bouncing “Hello World!” (1 extra point)

Let’s add some *controls* and *randomness* at the same time! Make the bouncing “Hello World!”

- move toward a *random* direction of a preset speed of `speed` when the program starts
- and, when the mouse is clicked, move toward the direction where the cursor is at a preset speed of `speed`

Here is an example of what it should look like and a template sketch for you to start from:

```
String msg = ”Hello, World!\nThis is {YOUR NAME HERE}!”;
float fontSize = 24;  // font size
boolean saveFrames = false;  // whether to save the frames
float speed = 5;  // speed of the ball

// DEFINE OTHER GLOBAL VARIABLES YOU NEED HERE

void setup() {
  // YOUR CODE GOES HERE
}

void draw() {
  // YOUR CODE GOES HERE

  // Save frames for making a GIF
  if (saveFrames) {
    saveFrame(”frames/###.png”);
  }
}

// Move the text box toward the cursor when the mouse is clicked
void mouseClicked() {
  // YOUR CODE GOES HERE
}
```

Please submit both your code and your final rendering as an animated GIF. You will receive zero credit if the code is missing. Click on several different spots to demonstrate the mouse controls.

Hints

- You might find atan() useful for finding the direction (with a tiny bit of trigonometry!).
- Which of the following code snippets gives a random velocity (i.e., speed and direction) with a fixed speed?

```
speedX = speed * random(0, 1);
speedY = speed * random(0, 1);
```

```
r = random(0, 1);
speedX = speed * r;
speedY = speed * sqrt(1 - r * r);
```

```
theta = random(0, TWO_PI);
speedX = cos(theta);
speedY = sin(theta);
```

3 Submission

- All assignments must be completed on your own. You are welcome to exchange ideas with your peers, but this should be in the form of concepts and discussion, not in the form of writing and code.
- Please provide proper citations/references for any external resources you use in your writing and code.
- Please submit your work to Gradescope.
- Late submissions will be deducted by 1 point per day.