

# Project 1

*Due: 9 Feb 2015*

This is the basic assignment: animate a bouncing ball with horizontal stripes, which responds to (user-controlled) gravity and mouse clicks that “hit” the ball to change its direction, and has some additional element of your own.

Now let’s break down what that means and clarify the constraints.

## Elements

**The ball** should be a circle that has at least three horizontal stripes (rather than radial coloration like in the “beachball” example). It should be less than half the size of the canvas in each dimension so that it can visibly bounce around. The vertices of the ball should be generated programmatically (i.e. don’t compute a bunch of points by hand and type them all into the program).

**The animation** should be motion that happens as time passes, such that the ball continues in the direction it was previously moving. Keeping track of the movement is usually done with a vector named “velocity” (for, I hope, obvious reasons), whose components are small enough that they can be simply added to the current location coordinates to compute the new ones. The motion should be accomplished by updating a single offset coordinate that gets pushed out to the shaders, not through recomputing the circle’s vertex coordinates every time tick.

**The bouncing** should have the ball reversing when it hits a “wall” (the boundary of the canvas). This can be accomplished by negating the  $x$  component of the velocity when the side walls are touched, and negating the  $y$  component when the top or bottom are touched. The ball should not get “stuck” in a wall due to errant computation, but it’s not a problem if the edge of the ball goes slightly out-of-frame during a bounce.

**The gravity** should be controlled by the user with a button that toggles it on and off. When gravity is “on”, that means that every time tick, a small negative amount is added to the  $y$  component of the velocity.

**Direction changes** can be caused by the user by clicking inside the canvas.

At each such click, the ball's velocity should change such that it is heading more in the direction opposite the click. There are multiple ways to interpret the phrase "the direction opposite the click", some more complicated than others; any of them should be fine as long as you explain them in your documentation.

**The additional element** (Marmorstein would probably call this "glitter") should be something that uses some aspect of WebGL or the HTML/canvas interface that we haven't really talked about. The easiest way to do this would probably be to look through the event types in Section 3.6 that we skipped and think about how to use them; but feel free to be more creative. Your addition shouldn't substantially interfere with the core functionalities described in the other paragraphs unless you talk to me about it first.

## Working on it

You'll be doing this in (some subdirectory of) your `public_html` directory, which has certain implications about the visibility of your code. This is "collaborative" in the sense that it is not locked down like an exam is, but it should be your own work: although you can talk it over generally with the other students and provide (and receive) a limited amount of debugging help, you should not be spending a lot of time reading the code of other students in the class.

If you have never read <http://cs.longwood.edu/~dblaheta/collab.html>, or haven't read it recently, now would be a good time to look at the program-based collaboration examples.

Note that you *can*, and are encouraged to, look at the files we made in class.

## Handing it in

I do want you to hand in all the files for this, including a readme; but the readme should also include the URL that I can use to run/test your code. The handin command is

```
handin cmsc381 proj1 dirname/
```

where “dirname” would be whatever directory you’re trying to hand in. (You can also hand in individual files—after the `proj1`, the rest of the line is all filenames to be handed in, and directory names will hand in the full contents of that directory.)

## Rubric

The first twenty points (of 100) are for the very basics: do you include documentation that tells me where to go and what it does, and does your program successfully display a canvas that has something on it.

The rest of the points are for the following items, at 5 points each:

Ball: successfully draw a circle

Ball: has multiple colours

Ball: colours are in horizontal stripes

Animation: drawing on canvas changes as time ticks

Animation: ball moves on canvas

Animation: ball moves in  $\geq 2$  directions, not parallel to axes

Bouncing: ball reacts to hitting any wall (even if it just stops)

Bouncing: all four walls

Bouncing: negates one dimension, preserves other (i.e. true bounce)

Gravity: button click toggles some movement behaviour

Gravity: turning gravity on starts ball moving downward

Gravity: when on, gravity is acceleration

Clicking: click in canvas triggers ball movement

Clicking: ball changes velocity

Clicking: ball goes more in the direction opposite the click

Additional: you add something and it works