

Bacteria Project Two

COMP-4471, Winter 2021

<https://github.com/Soupy1184/COMP-4471-Project2>

Christopher Campbell and Ben Puhalski

Table of Contents

- [Description](#)
- [Bacteria Rendering](#)
- [Growth](#)
- [Spawning](#)
- [Poison](#)
- [Screenshots](#)

Description

This is a JavaScript program that implements a bacteria game using WebGL.

Contains the following mechanics:

- Bacteria spawn around sphere randomly
- Bacteria grow out uniformly from it's origin
- Eliminate bacteria by clicking them to administer 'poison'
- Score System
- Adds score depending on how large the bacteria eliminated was
- Bacteria spawn rate
- Random bacteria color

Bacteria Rendering

The bacteria are spheres that are rendered by creating a larger sphere around the play surface, rotating the model matrix to the position the bacteria should be rendered at, then using a subset of the total indices to create a sphere that looks like it is topologically set onto the play sphere.

Growth

```
grow() {  
    this.growth++;  
    var sect = (this.object.indices.length / this.sphereDiv);  
    for (var i = sect * this.growth; i < sect * (this.growth+1); i++) {  
        this.indices.push(this.object.indices[i]);  
    }  
}
```

The above function is used inside the bacteria class to grow the bacteria by pushing more indices according to the bacteria's size to give the look of a bacteria growing

Spawning

The bacteria spawn every couple of seconds. The growth rate can be easily increased to give a version of increasing difficulty as the game is played.

```
Time();
//every few seconds
if(Math.floor(time / growth_rate) > growthCounter) {
  growthCounter++;
  for(var b = 0; b < bacteria.length; b++) {
    bacteria[b].grow();
  }
}
//spawn bacteria every few seconds
if(Math.floor(time / spawn_rate) > spawnCounter) {
  spawnCounter++;
  if(bacteria.length < max_bacteria) {
    bacteria.push(new Bacteria(bacteriaSize, SPHERE_DIV));
    console.log("spawned bacteria");
  }
}
```

The above is used for rates that need to be triggered every so often at different timings. This takes advantage of the Date.now() to get elapsed time and uses counters to confirm if this frame is one that should be used to either grow the bacteria or spawn a bacteria.

Poison

The poison is implemented by detecting the color of the pixel at the position of the mouse when the mouse is clicked then the program checks all bacteria for a matching color. If there is a matching color the corresponding bacteria is destroyed.

```
function onClick(ev, gl, bacteria){
  //get canvas x and y position
  var x = ev.clientX;
  var y = ev.clientY;
  var rect = ev.target.getBoundingClientRect();
  var x_canvas = x - rect.left;
  var y_canvas = rect.bottom - y;

  //get color at mouse position
  var color = checkPointColor(gl, Math.round(x_canvas), Math.round(y_canvas));

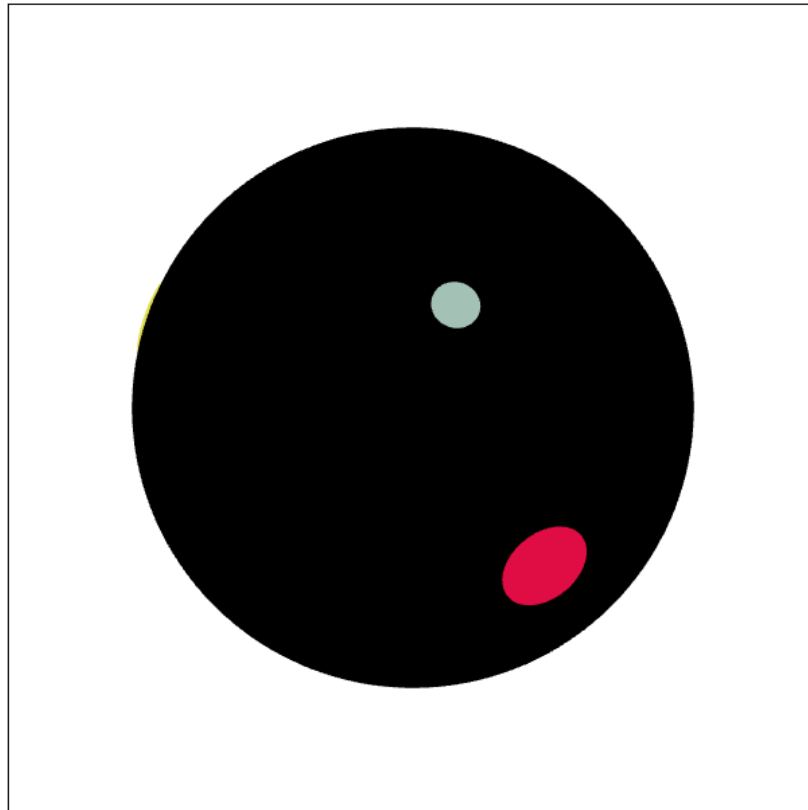
  //delete bacteria with the same color
  for(i = 0; i < bacteria.length; i++) {
    bcolor = bacteria[i].getColor();
    if(bcolor.r == color.r && bcolor.g == color.g && bcolor.b == color.b) {
      //delete bacteria
      bacteria.splice(i, 1);
      break;
    }
  }
}
```

For the above function, 'gl' is the WebGL context, 'bColor' is the bacteria's color, 'x_canvas' and 'y_canvas' are the x and y positions within the WebGL canvas. The function splices the bacteria out of the array so that when the program decides to render the deleted bacteria will not exist within the array.

Screenshots

First Display

Score: 0

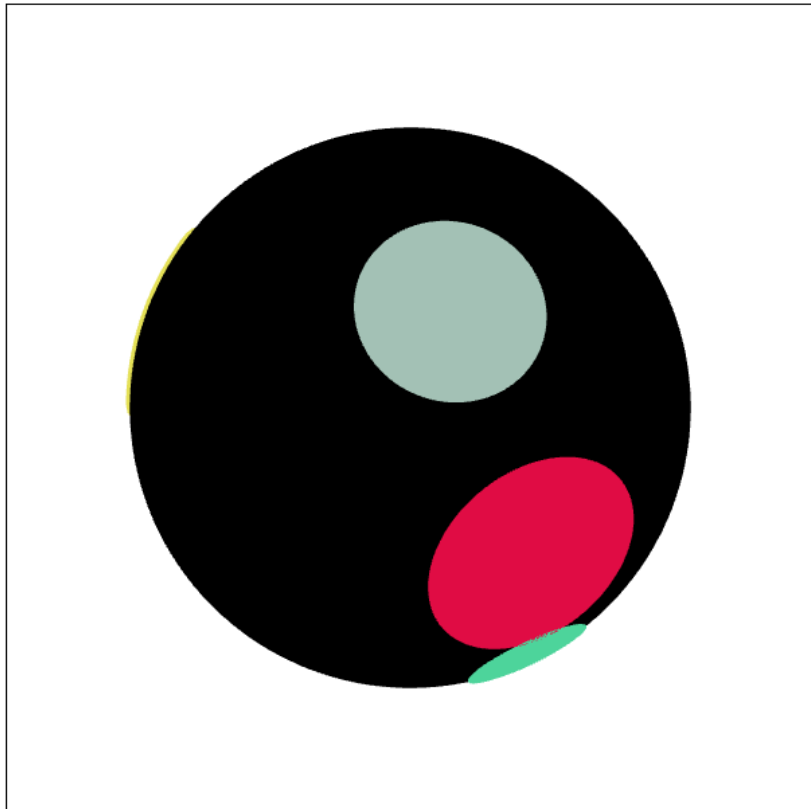


Click on the bacteria to administer poison

When the program is launched a score counter and canvas will be displayed. The program spawns in a few starting bacteria around the sphere.

Bacteria growth and spawn

Score: 0



Click on the bacteria to administer poison

Here the bacteria grows every few seconds and spawns in a new bacteria every few seconds at a different rate.