Odin: Web-Based Crowd Simulation

WebGL 2.0
gpu.js

Pitch

- Cleaned up base code
- Include all appropriate files and project base ready for working

Milestone 1

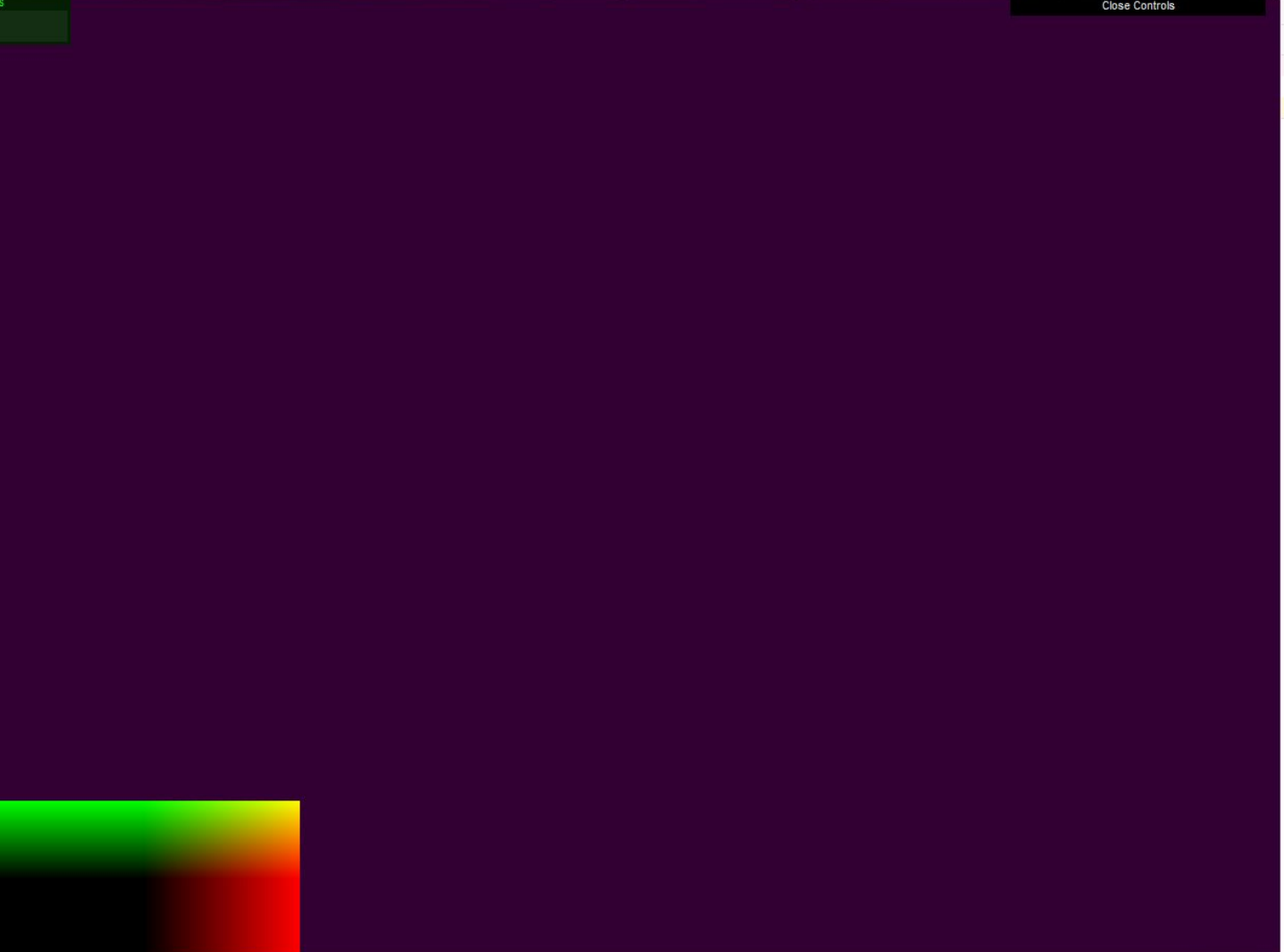
- Procedural marionette loading and working, beginning signed distance functions for meshing
- Basic setup of gpu based bio crowds (for initial gpu setup and for initial basic crowds setup)

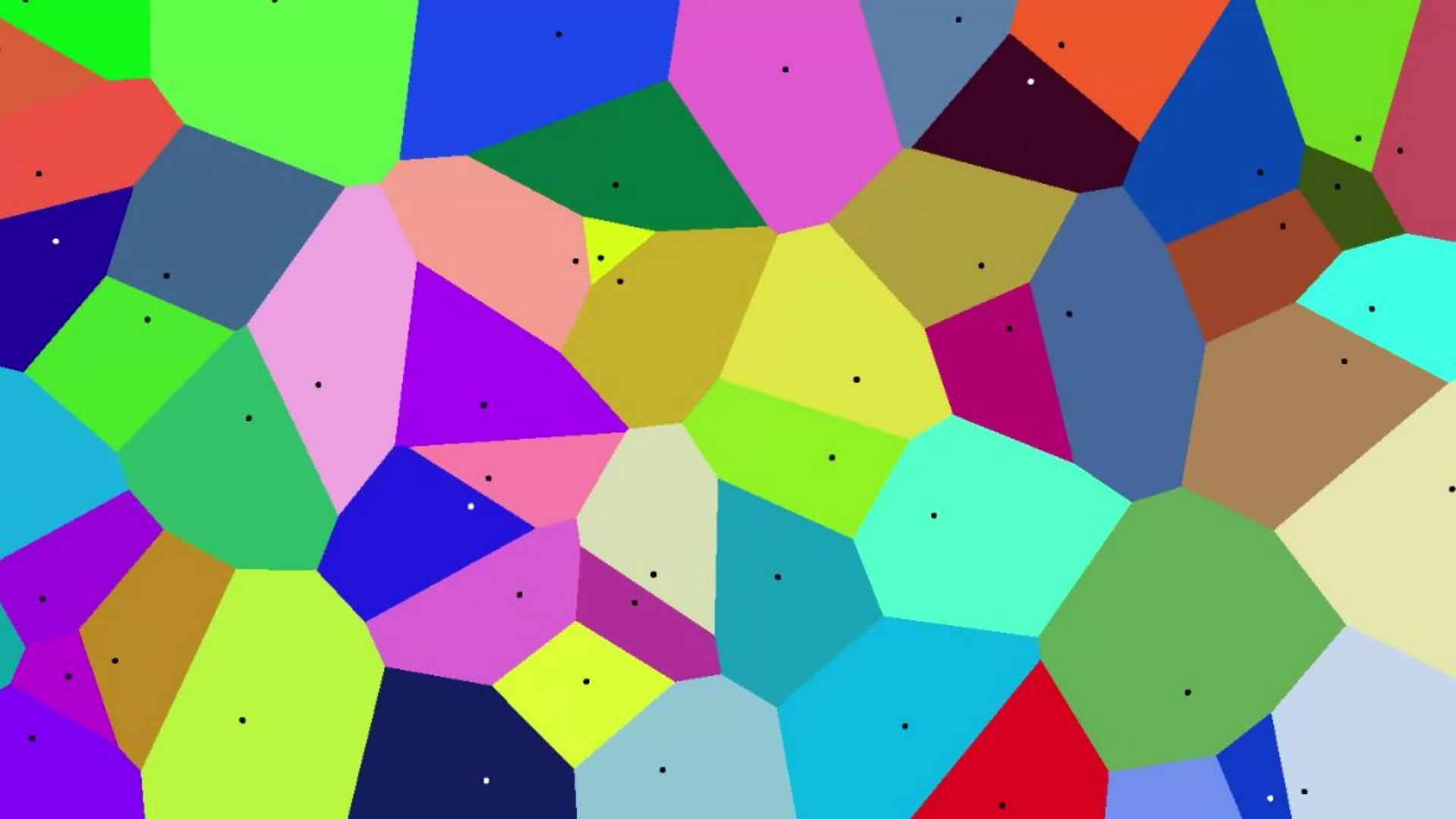
Milestone 2 - Nov 26

- Ik and scene graph gpu
- Optimized crowds based on the papers

Milestone 3 - Dec 3

- Debugging and code cleanup
- Stretch goals


Close Controls
Inspector
Console
Debugger
Style Editor
Filter output
[WDS] Disconnected!
WebGL 2 activated.
window.controllers/Controllers is deprecated. Do not use it



```
export const shadeScreen = gpu.createKernel(function(widthDim, heightDim, mode, inputImage) {
  var xLoc = this.thread.x / widthDim;
  var yLoc = this.thread.y / heightDim;

  var random2 = vec2(random(vec2(xLoc, yLoc)));

  this.color(vec3(random(vec2(xLoc, yLoc), 1, 1));
})
.setOutput([canvas.clientWidth, canvas.clientHeight])
.setGraphical(true);

shadeScreen.addNativeFunction('random', `highp float random(vec2 co)
{
    highp float a = 12.9898;
    highp float b = 78.233;
    highp float c = 43758.5453;
    highp float dt= dot(co.xy ,vec2(a,b));
    highp float sn= mod(dt,3.14);
    return fract(sin(sn) * c);
}`);
```

```
var random2 = vec2(random(vec2(xLoc, yLoc)));

this.color(vec3(random(vec2(xLoc, yLoc), 1, 1));
```

```
var random2 = vec2(random(vec2(xLoc, yLoc)));

this.color(vec3(random(vec2(xLoc, yLoc), 1, 1));


float random2 = vec2(float, float);


const random2 = this.vec2(random(vec2(xLoc, yLoc)));
```

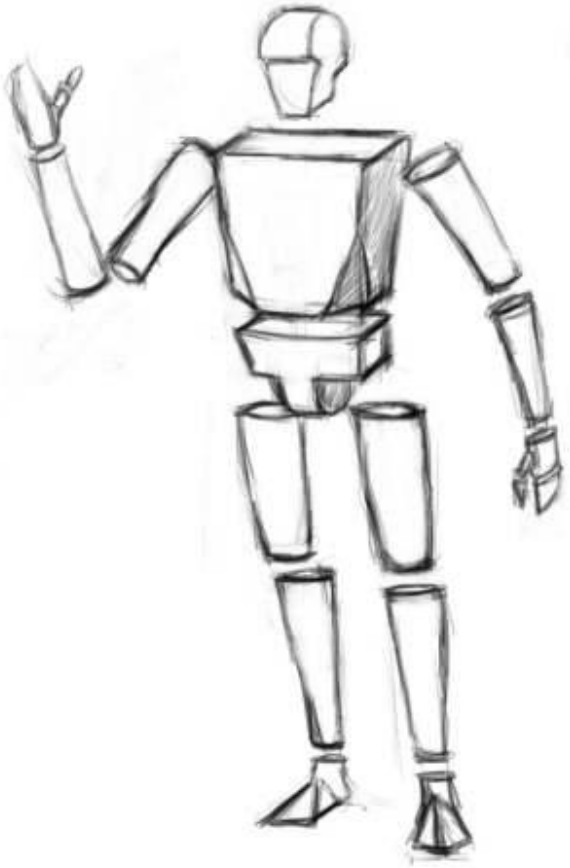
```
var random2 = vec2(random(vec2(xLoc, yLoc)));

this.color(vec3(random(vec2(xLoc, yLoc), 1, 1));
```

```
float random2 = vec2(float, float);
```

```
const random2 = this.vec2(random(vec2(xLoc, yLoc)));
```

```
var red_channel = 0;
var green_channel = 0;
var blue_channel = 0;
var alpha_channel = 1;
```



Milestone 1

- Procedural marionette loading and working, beginning signed distance functions for meshing
    a. Don’t have randomization yet for large scale creation
- Basic setup of gpu based bio crowds (for initial gpu setup and for initial basic crowds setup)
    a. Code is there, update based on image passing (like compute), still debugging gpu.js

Milestone 2 (Updated)

- Add randomization yet for large scale creation
- Finish crowd update method, debug gpu.js
- Ik and scene graph gpu
- Optimized crowds based on the papers