Proposal Authors

Nick Friedman
Zach Lazow
Ethan Jones

Statement

Our goal of the project is to create a game that models frisbee golf in a physically driven environment. As three avid frisbee golf players (Zach and Ethan are captains of the Ultimate Frisbee team) we are passionate about this project idea. Similar to regular golf, the player will be able to explore beautifully designed courses as they traverse through a hole. Additionally, we plan to have dynamic lighting and interactions between the disc and the environment - including wind. These objects will have a MeshPhongMaterial. We will add one light point in the scene to represent the sun.

Technical Outline

For the objects we are building in blender, we will place an image in the background and build it similar to how we built the submarine and bird.
We will manipulate the keyboard commands so that they will affect variable values.

- ```
  Var key = {
  _pressed:{},
  A:65
  W:87
  D:68
  S:83
  SPACE: 32,

  isDown:
  function(keyCode){
        Return
  this._pressed[keyCode];
  },
  onKeydown:
  function(event) {
  this._pressed[event.keyCode] = true;
  },
  onKeyup:
  function(event){
  Delete
  ```
  ```

```
this._pressed[event.keyCode];
}
},
```

We will need an algorithm to determine the location of the frisbee after it is thrown based on the variable values like power, direction, angle, and wind speed.

- Constant rate drop in Y Direction
- Constant speed determined by power bar. Affected by wind
- Change in X direction = rate of change from tilt + rate of change from wind
- Effect when frisbee hits an object (ie a tree or the ground)
    - Frisbee can slide when it hits the ground and bounce off when hits a tree

To cast shadows, we need to add code to the renderer, the lights, and the objects.

- `renderer.shadowMapEnabled = true;`
- `object3d.castShadow = true;`
  `object3d.receiveShadow = false;`
- `light.castShadow = true;`
- `light.shadowDarkness = 0.5;`
- `light.shadowCameraVisible = true;`
- `light.shadowCameraRight = 5;`
  `light.shadowCameraLeft = -5;`
  `light.shadowCameraTop = 5;`
  `light.shadowCameraBottom = -5;`

3D Text:

```
var materialFront = new THREE.MeshBasicMaterial( { color:
0xff0000 } );
var materialSide = new THREE.MeshBasicMaterial( { color: 0x000088
} );
var materialArray = [ materialFront, materialSide ];
var textGeom = new THREE.TextGeometry( "Score: " + Score + "
Par: "+Par,
{
      size: 10, height: 4, curveSegments: 3,
      font: "helvetiker", weight: "bold", style: "normal",
      bevelThickness: 1, bevelSize: 2, bevelEnabled: true,
      material: 0, extrudeMaterial: 1
});
// font: helvetiker, gentilis, droid sans, droid serif, optimer
// weight: normal, bold
```

```
	var textMaterial = new THREE.MeshFaceMaterial(materialArray);
	textMesh = new THREE.Mesh(textGeom, textMaterial );

	textGeom.computeBoundingBox();
	var textWidth = textGeom.boundingBox.max.x -
textGeom.boundingBox.min.x;

	textMesh.position.set( -0.5 * textWidth, 0, 1800 );
	console.log(-.5 * textWidth);
	textMesh.rotation.x = Math.PI / 8;
	scene.add(textMesh);
```

GUI

```
	var gui = new dat.GUI();
	var parameters =
	{
		m: false,
		a: 0, // numeric slider
		p: 25, //numeric slider
		b: true, //boolean (checkbox)
		c: false, // boolean (checkbox)
		d: "#ffff00", // color (hex)
		next: function(){ nextThrow()},
		h: function(){ change_hole(Hole_Num)},
		reset: function(){ resetDisc(Hole_Num)}
	};
```

Camera follow frisbee

```
	camera.position.z -= 0.75 * z;
	camera.position.y += y;
	camera.position.x += x;
	camera.lookAt(disc.position)
```

Bibliography

Three.js tutorial (All Objectives):
https://aerotwist.com/tutorials/getting-started-with-three-js/

Extension for grass (Objective 7): https://github.com/jeromeetienne/threex.grass

Mario Golf, Camelot Software Planning, Nintendo.

3D Physics (Objective 3 and 5):
http://learningthreejs.com/blog/2012/06/05/3d-physics-with-three-js-and-physijs/

Particle Systems (Objective 1):
http://solutiondesign.com/blog/-/blogs/webgl-and-three-js-particles

Casting Shadows (Objective 2):
http://learningthreejs.com/blog/2012/01/20/casting-shadows/

SkyBox (Objective 8): http://stemkoski.github.io/Three.js/Skybox.html

Camera follow frisbee (Objective 4):
http://stemkoski.github.io/Three.js/Chase-Camera.html

Glow in the dark frisbee (Objective 1):
http://stemkoski.github.io/Three.js/Shader-Glow.html

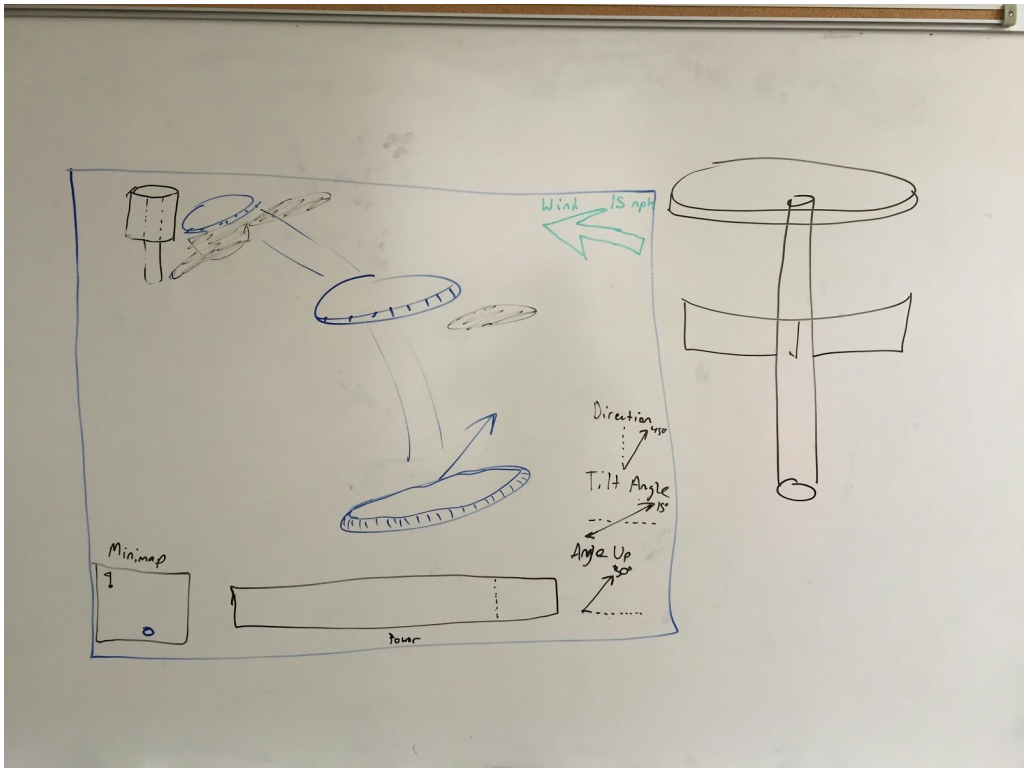
GUI (Objective 3 and 10): http://stemkoski.github.io/Three.js/GUI.html

3D text (Objective 3): http://stemkoski.github.io/Three.js/Text3D.html

Objectives

1) Model a frisbee in Three.js and frisbee-golf hole in blender. The frisbee will have a lambert material to look plastic. In the dark model, the frisbee will glow.
2) Implement Shadow Mapping. We will make a lighting model to display shadows coming from all the objects including the frisbee, hole, and any objects in the scene like trees. The source of the shadows will be from the lighting source representing the sun.
3) Graphic UI: Animate the frisbee to move towards hole by incorporating a physics engine. The physics of where the frisbee will be thrown to will be determined by the wind, gravity, power, direction, and angle.
4) Manipulate the Camera. We want to be able to adjust the camera so after the frisbee is thrown, it moves to the new spot. The camera will also follow the frisbee as it is moving during the throw.
5) Adjust the wind in the scene. Either have the wind constantly change power and direction, or have widgets on the screen that allows the user to choose between no wind, low wind, and high winds. Conversely, the wind amount can increase with each hole, making each hole harder to be accurate.
6) Create widgets on the screen to allow the user to manipulate the power, direction and angle variables using keyboard and gui features.
7) Model the grass throughout the entire scene.
8) Create lighting models for night vs day. The night will have a light source to represent the moon. The day will have a light source to represent the sun.
9) Model multiple holes in the scene. Each hole will be unique.
10) Create a minimap that displays the position of the thrower, the hole, and any obstacles in the course. For the minimap, the camera will have an aerial view of the course.

Photos


Wind
15 mph
Direction
Tilt Angle
15°
Angle Up
Minimap
Power


Hole 1
PAR 4
PIN 401y
3y
11mph
±0
0
1W
Normal
212y
A
GAMESPOT

Plan

Week 1

1) Work on Camera movement (Nick)
2) Add grass to the scene (Nick)
3) Finish objects in blender and export to three.js (Ethan)
4) Work on movement of frisbee (Ethan and Zach)
5) Create textures for the objects (Zach)

Week 2

1) Create widgets on the screen (Nick)
2) Get wind working (Ethan)
3) Implement Shadow Mapping (Zach and Nick)
4) Create Lighting Model (Zach and Nick)

Week 3

1) Add obstacles to the scene (Ethan)
2) Finish the Camera movement for one hole (Ethan, Zach, and Nick)
3) Add multiple holes (Ethan)
4) Create minimap (Nick)
5) Have the camera movement work on all holes