Week 7 Presentation Team 3

By Emily, Carson, & Mat

Terrain Improvements

- Added the first map type
  - “Vanilla Random”
- Added random “noise” to the “Square” step
  - 60% - no change
  - 30% - add a random number between -0.5 and 0.5
  - 10% - add a random number between -1 and 1
- Still had the problem of models loading at a static height
  - Needed a way to place the models on top of the ground on load

My Raycaster implementation

- Create a raycaster that points straight down
- Place the raycaster at the same location of the new model
- Bump the model up until the raycaster intersects the ground
  - To track this, I check the size of the raycaster's intersectObjects array
  - Since there aren't any objects in the scene beneath the map, when the array size > 0 you've found the map.

```
var down = new THREE.Vector3(0,-1,0);
var caster = new THREE.Raycaster(new THREE.Vector3(0,0,0), down);
caster.far = .05;
var floorMesh = scene.getObjectByName(floorMesh);
```

```
//place the raycaster at the same location as the model
caster.set(root.position, down);
let intersects = caster.intersectObjects(scene.children);

//bump the model up until the raycaster intersects the ground
while(intersects.length < 1){
  caster.set(root.position, down);
  root.position.y += .05;
  intersects = caster.intersectObjects(scene.children);
}
```

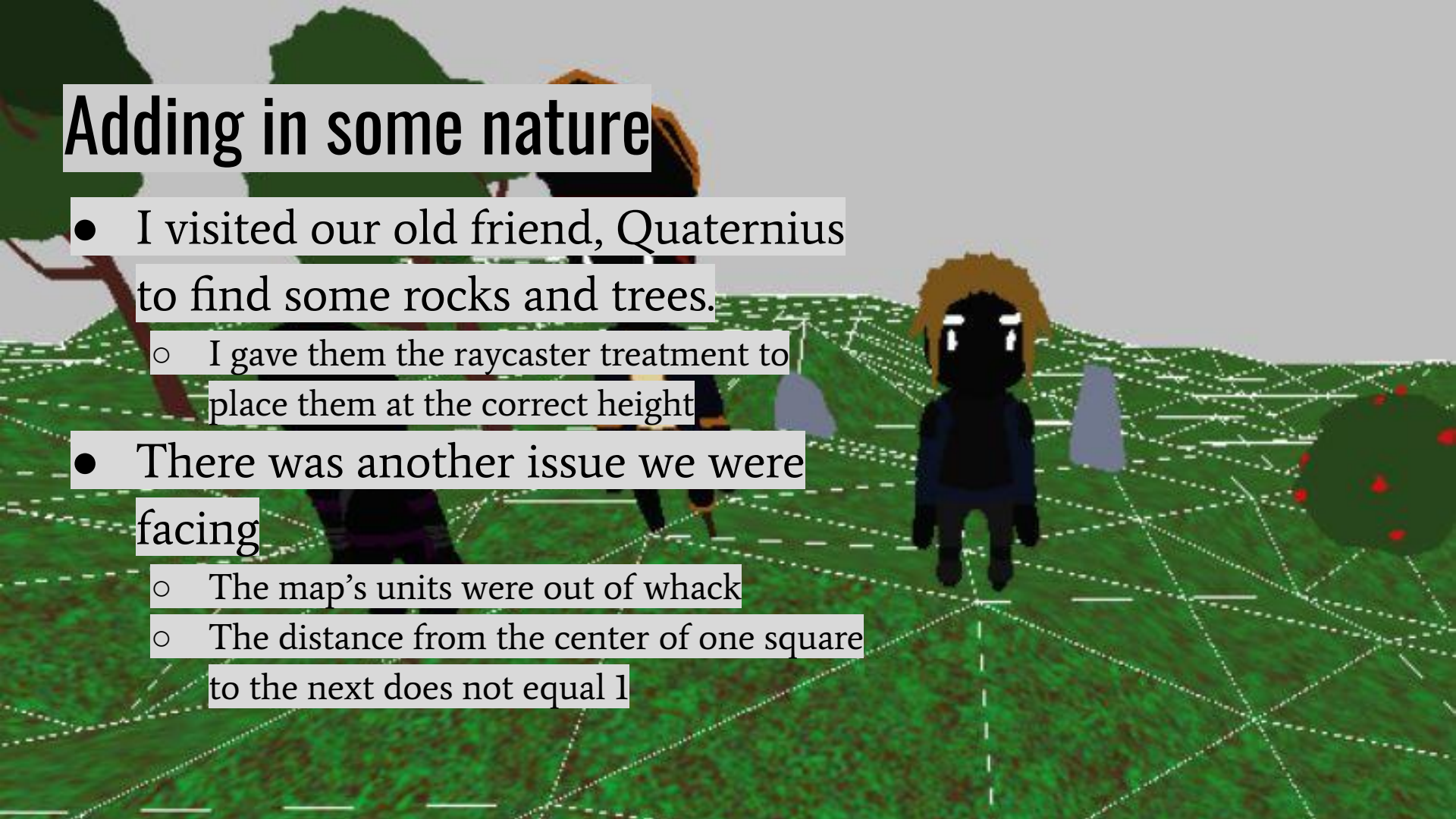
This gave us bashful models...

- The models would just reach the map and stop.
  - Not entirely sure why, the ray should have been between the models feet
- Rather than troubleshoot for hours, I used a practical solution...

```
//move the model up so that it is above the ground
root.position.y += .95;
```

Adding in some nature

- I visited our old friend, Quaternius to find some rocks and trees.
  - I gave them the raycaster treatment to place them at the correct height
- There was another issue we were facing
  - The map's units were out of whack
  - The distance from the center of one square to the next does not equal 1

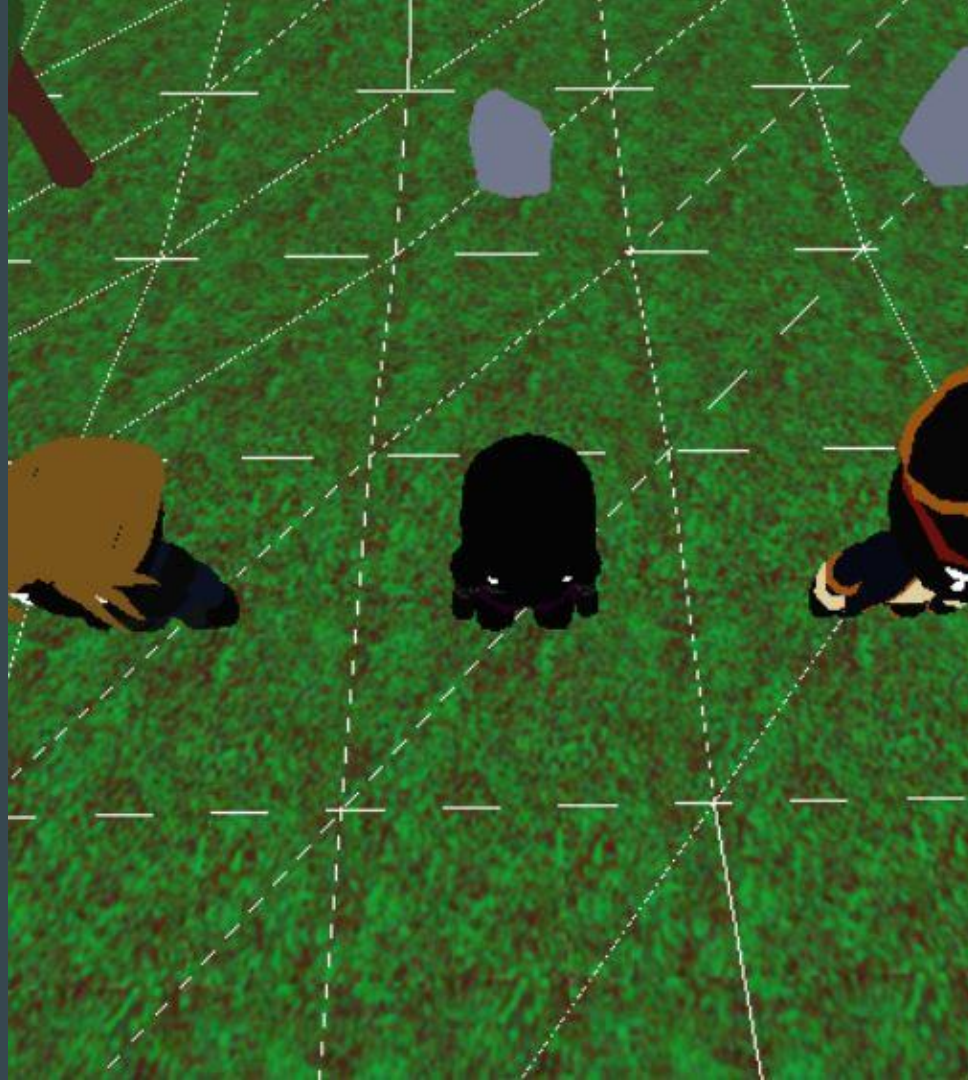


Creating the map “unit”

- I made a guess that the spacing issue had to do with that distance being slightly larger or smaller than 1
  - I played with some guesses and got:

```
//setup units for setting positions
let mapVerts = heightMap.length;
let unit = mapVerts/(mapVerts - 1);
let x = unit/2;
```

```
//Place the model, "root" at the proper coordi
root.position.set(x, 0.01, -3*(unit/2));
x += unit;
```

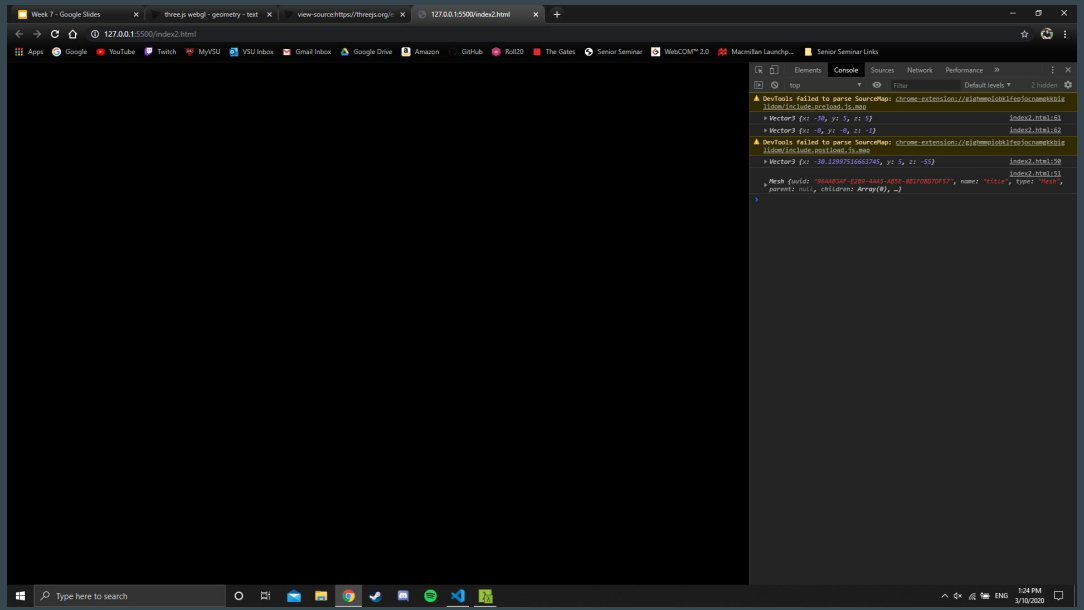


Talks on Yuka Implementation

- Multiple groups coming together to discuss the topic of implementing AI (Yuka.js) into our games.
- What we learned during this:
  - What functionalities Yuka has
  - Which ones we can implement into our games
  - Which ones may be useful to groups that could not make the talk
- Benefits:
  - Allows multiple groups working on the same part of their game to work together to implement their work faster
  - Makes it easier to understand certain components or to see new ways of utilizing them
- We may want to look into doing stuff like this more often if the opportunity presents itself

My Struggles of making a Title Screen

- I’ve spent the last week focusing on creating a title screen for our game
- My issue:
  - My TextGeometry would not want to show up on my screen



```
var fontLoader = new THREE.FontLoader();

fontLoader.load('./font/Abril.json', function (font) {
    let title = new THREE.TextBufferGeometry("Tacticool", {
        font: font,
        size: 10,
        height: 1,
        curveSegments: 12,
        bevelThickness: 1,
        bevelSize: .5,
        bevelEnabled: true
    });

    title.computeBoundingBox();

    let center = -0.5 * (title.boundingBox.max.x - title.boundingBox.min.x);

    let titleMaterial = new THREE.MeshPhongMaterial({
        color: 0xff0000,
        specular: 0xffffff
    });

    var titleMesh = new THREE.Mesh(title, titleMaterial);
    titleMesh.name = "title";

    titleMesh.position.set(center, 5, -55);

    console.log(titleMesh.position);
    console.log(titleMesh);

    scene.add(titleMesh);
});
```

```
⚠ DevTools failed to parse SourceMap: chrome-extension://gighmmpiobklfepjocnamgkkbiglidom/include.preload.js.map
▸ Vector3 {x: -30, y: 5, z: 5}                                  index2.html:61
▸ Vector3 {x: -0, y: -0, z: -1}                                 index2.html:62
⚠ DevTools failed to parse SourceMap: chrome-extension://gighmmpiobklfepjocnamgkkbiglidom/include.postload.js.map
▸ Vector3 {x: -30.12997516663745, y: 5, z: -55}                 index2.html:50
                                                                index2.html:51
▸ Mesh {uuid: "96AAB3AF-E2B9-4AA5-AB5E-8B1FDBD7DF57", name: "title", type: "Mesh", parent: null, children: Array(0), …}
>
```

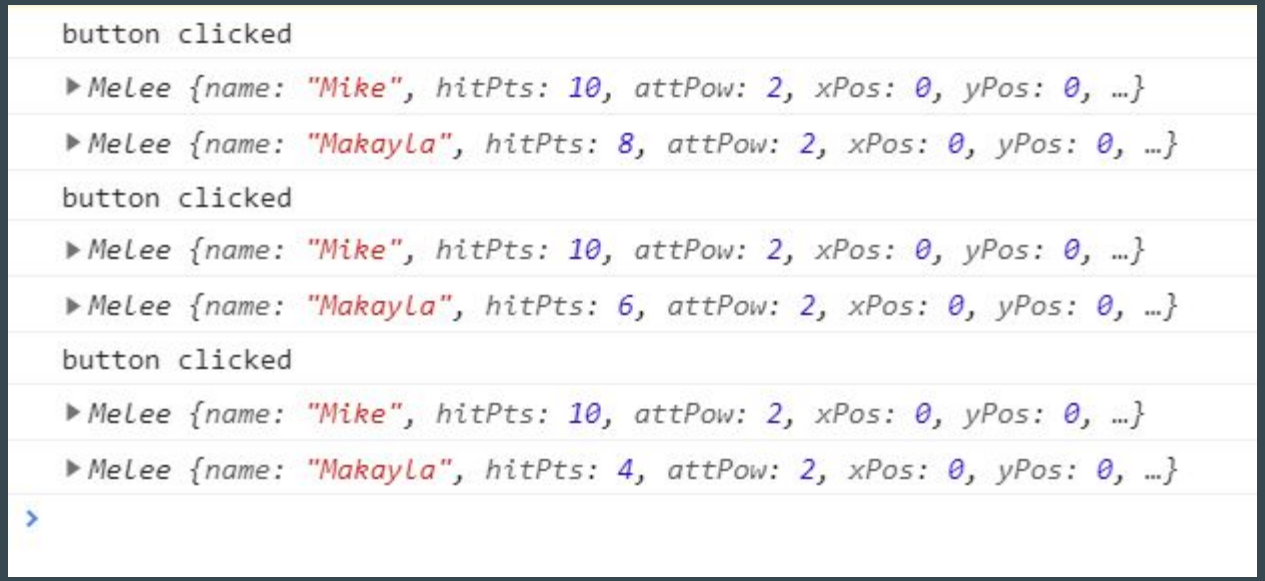
Attack Implementation

- Each Actor (class) has certains strengths and weaknesses.
- The attack() function takes these into consideration when decreasing HP
  - *attacker.attack(target)*
  - If the attacker's attack type is the target's weakness, then the damage is doubled
  - If the attacker's attack type is the target's resistance, then the damage is halved
- The overall attack power of the character also determines how much HP is removed

```
attack(actor, array){
    var attMod = 1;                                    //attack modifier

    // if(!this.inRange(actor)) //in progress
    //     return;

    if(this.attType != null && actor.weakness != null){
        for(let i = 0; i < this.attType.length; i++){
            if(actor.weakness.includes(this.attType[i])){ //if the arg actor's weakness includes the type, attack mod is doubled
                attMod *=2;
            }
        }       Carson Davis, a month ago • Fixed conflicting issues between all branches in testing
    }
    if(actor.resist != null){
        for(let i = 0; i < this.attType.length; i++){      //second verse, same as the first (but for resistance)
            if(actor.resist.includes(this.attType[i])){    //if the arg actor is resitant, attack mod is halved
                attMod /= 2;
            }
        }
    }
    if(!(actor.hitPts - (this.attPow * attMod) <= 0)){
        actor.hitPts -= this.attPow * attMod;              //reduce the arg actor's HP
    }else{
        //they ded
        actor.hitPts = 0;

    }
}
```

Attack Implementation

- Determining whether a character is in range is in progress
- We plan on adding a “dummy” value to the appropriate array (character or enemy) element when a character dies



This week's useful findings...

- Me, Mat, and Wallace are super cool coders

Next big hurdle: Alpha build

- An alpha build of a game is the first playable iteration where most of the features are in place and the systems work (almost) properly.
  - We are hoping to begin the process of integrating our work together over spring break.
  - We are shooting for around a 2 week deadline for this
- Things we will include in the alpha build:
  - Procedurally generated terrain with obstacles
  - Randomly assembled team
  - Teams are able move and attack
    - Enemy AI may not be in place yet but we will have ways to test enemy attacking / movement