Group 1: Week 6

Merritt Hancock, Ryan Trull, Alan Bettis, Kenda Blair

Slime Puzzle Game

Tasks

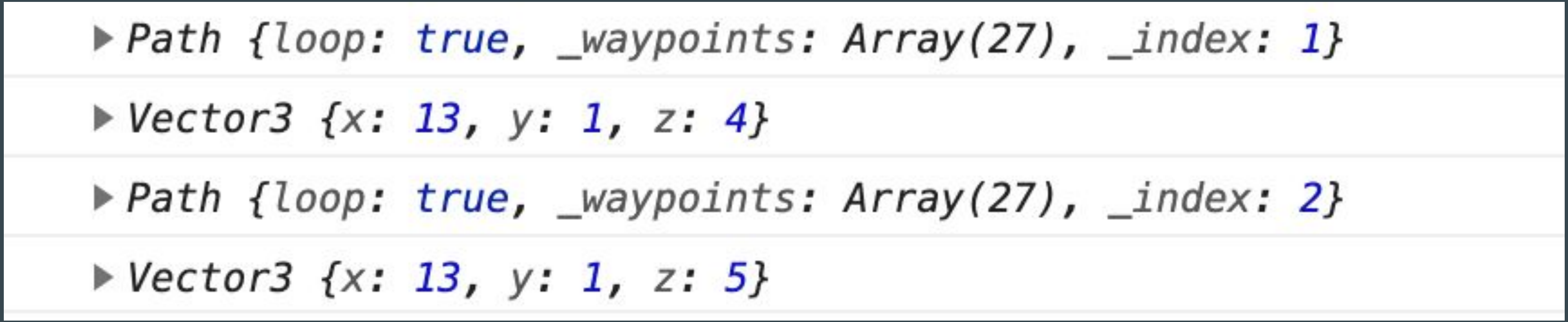
- Enemy AI Based Movement
- Entity Selection
- Began implementation of Turn-based System
- UV Mapping Models
- Refactoring

Enemies and AI

- Researched Yuka AI Library
- Specifically, looked into pathing and path following
- Yuka has two useful classes: Path and FollowPathBehavior
- FollowPathBehavior is more for smooth pathing, while Path can define a series of waypoints that can loop when the enemy follows it.

YUKA

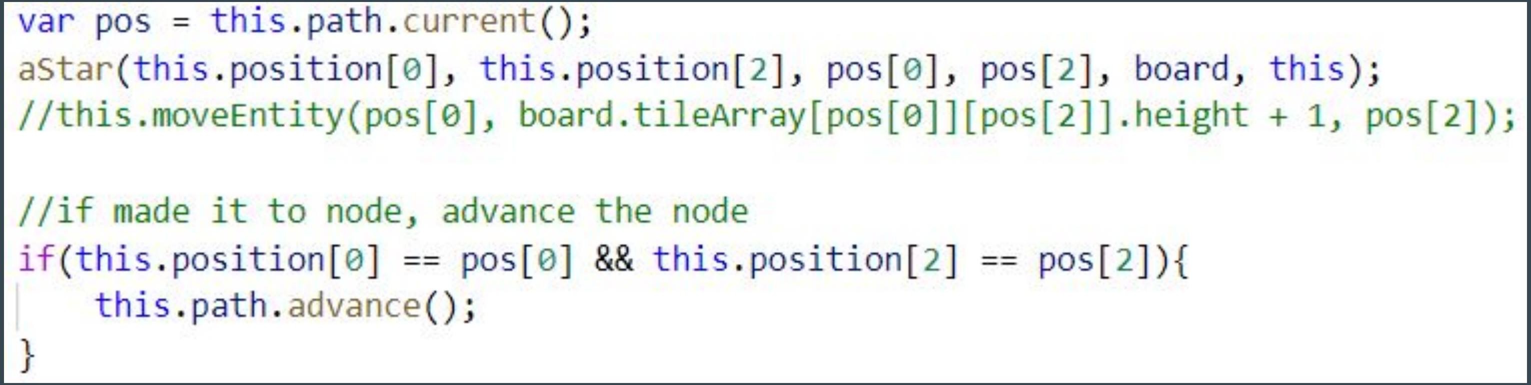
v0.3.5



```
▶ Path {loop: true, _waypoints: Array(27), _index: 1}
▶ Vector3 {x: 13, y: 1, z: 4}
▶ Path {loop: true, _waypoints: Array(27), _index: 2}
▶ Vector3 {x: 13, y: 1, z: 5}
```

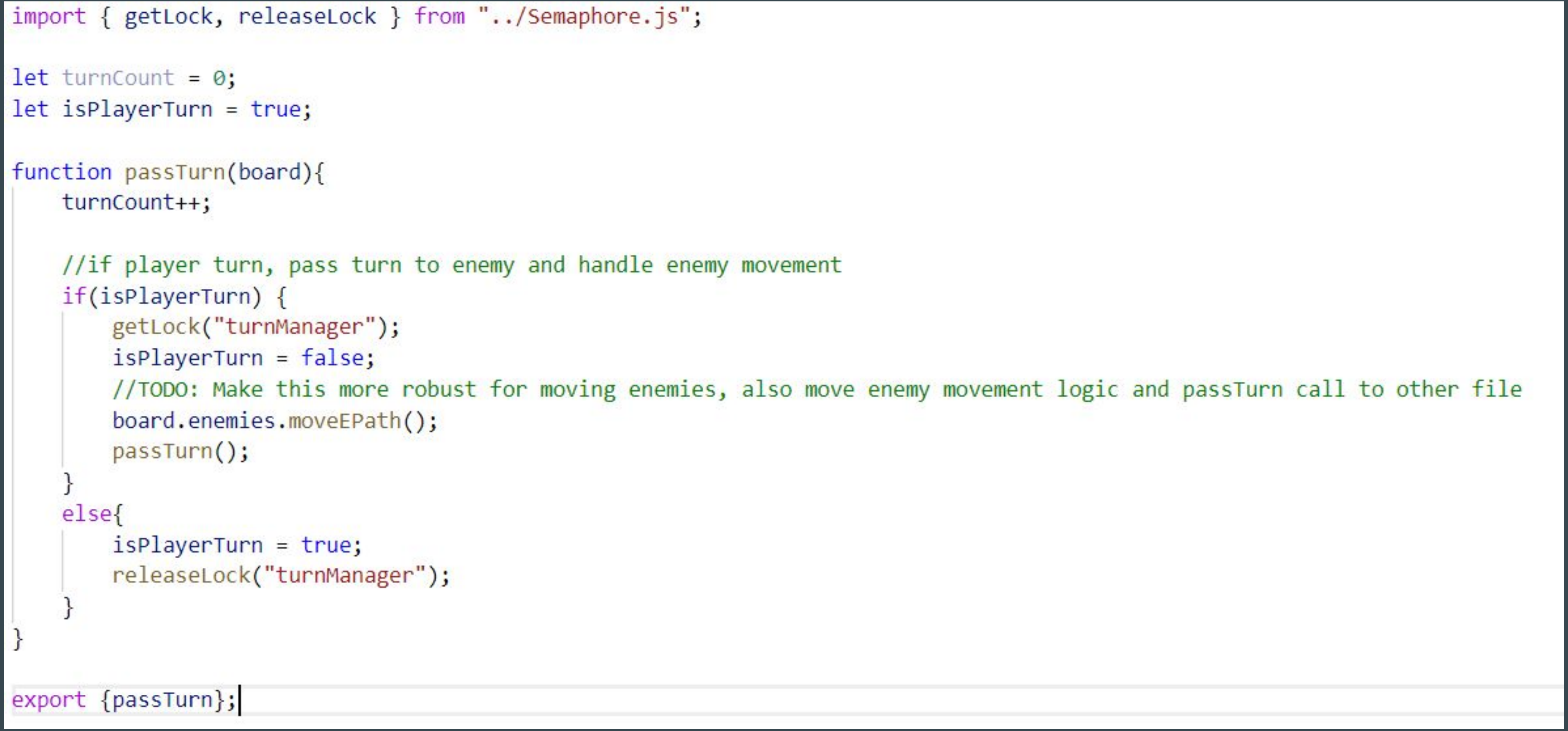
Yuka + A*

- Yuka’s Path sets up a linked list of coordinates for waypoints.
- Enemy uses A* to navigate towards a waypoint.
- Once a waypoint is reached, it advances to the next waypoint.



```
var pos = this.path.current();
aStar(this.position[0], this.position[2], pos[0], pos[2], board, this);
//this.moveEntity(pos[0], board.tileArray[pos[0]][pos[2]].height + 1, pos[2]);

//if made it to node, advance the node
if(this.position[0] == pos[0] && this.position[2] == pos[2]){
    this.path.advance();
}
```

Turn order



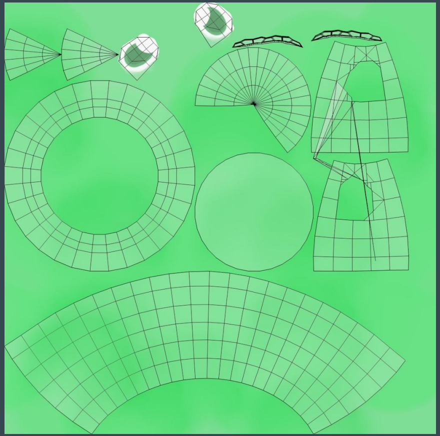
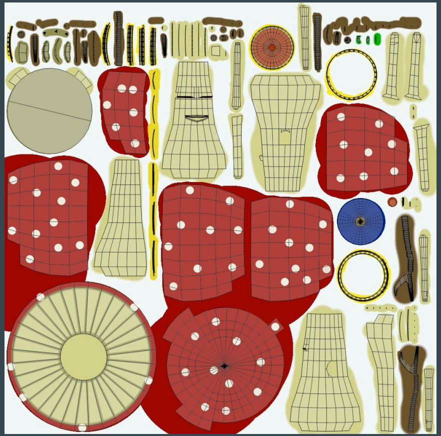
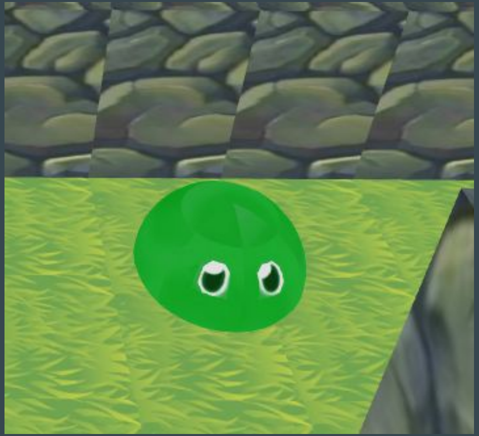
```
import { getLock, releaseLock } from "../Semaphore.js";

let turnCount = 0;
let isPlayerTurn = true;

function passTurn(board){
    turnCount++;

    //if player turn, pass turn to enemy and handle enemy movement
    if(isPlayerTurn) {
        getLock("turnManager");
        isPlayerTurn = false;
        //TODO: Make this more robust for moving enemies, also move enemy movement logic and passTurn call to other file
        board.enemies.moveEPath();
        passTurn();
    }
    else{
        isPlayerTurn = true;
        releaseLock("turnManager");
    }
}

export {passTurn};
```

Other useful Yuka AI Classes

- FleeBehavior
  - Flees from a target if the target enters its defined radius
- PursuitBehavior
  - Pursues a target and aims to predict ways to counter target movement
- ObstacleAvoidanceBehavior
  - Determines ways for object to avoid an obstacle in its path
- WanderBehavior
  - Adds an element of randomness to a radius for an enemy to move
  - Ex: A fish-type enemy wouldn’t follow a set path, but swim around

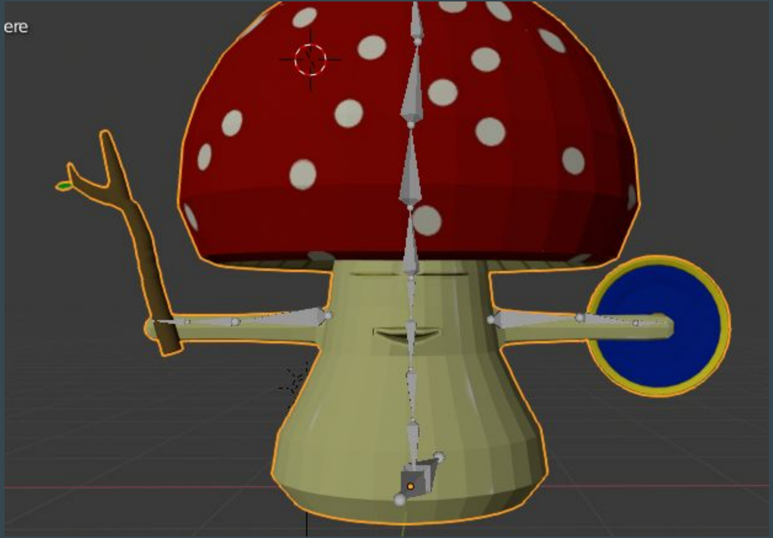
Models

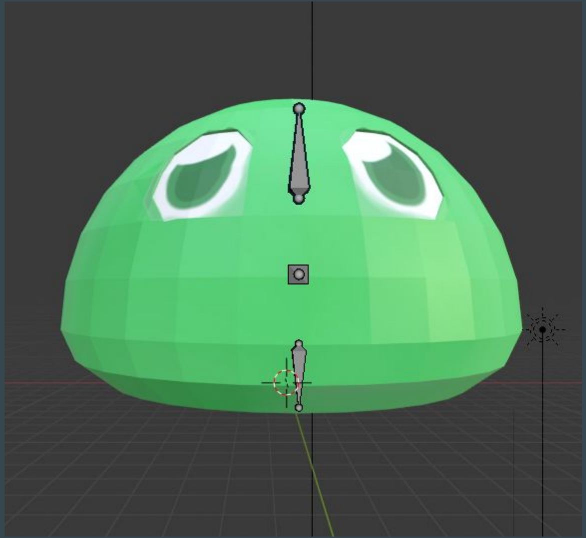
- Textures applied and baked into models using Principled BDSF shader node.

Model Rigs

- Both models are rigged.





Future Tasks/Levels

Storage of level objects

JSON

Level Editing

Working level after Spring Break

Next Week Tasks

<u>Modify Controller</u>

- Level Objects
- Handles multiple levels
- Optimize camera

<u>Models</u>

Working and moveable
Cursor model

<u>Enemy</u>

Absorption
Pursuit
Health