Platformer 2

Due: April 4, 2022

Introduction

Now that your engine supports movement around an arbitrary environment mesh, let's optimize how it handles collisions!

Directions

For this checkpoint, you will implement a spatial organization data structure. You may choose which data structure you implement (BVH, Uniform Grid, Hierarchical Grid, Octree, K-D Tree, BSP Tree). Then, you will use this data structure to speed up your collision checking between the player and the static environment mesh. After that, you will use a data structure to speed up your collision checking between dynamic objects. You may use different data structures to handle static and dynamic objects.

Design Check

- What data structure(s) will you implement? How will your data structure(s) handle static and dynamic objects?
- What will be the member variables of the nodes of your data structure?
- How will you build your data structure?
- How will you traverse your data structure?
- Will you need to update your data structure as the game state changes?

Engine Requirements

- Spatial organization data structure that handles collisions with static objects
- Spatial organization data structure that handles collisions between dynamic objects

Game Requirements

- Screen populated with enough dynamic objects (e.g. collidable cylinders) to demonstrate benefit of spatial organization data structure
- Screen with a large enough environment to demonstrate benefit of spatial organization data structure