Platformer 1

Due: Mar 14 2022

Introduction

In this assignment you will create your own 3D platformer from scratch. You will start out by implementing collision detection and response for a player moving in a world made of triangles.

Directions

You get to choose whether you implement ellipsoid-triangle collisions or the GJK algorithm and the Expanding Polytope Algorithm. Then, you will load obj files into your engine. There are obj files and corresponding texture maps in `/course/cs195u/asgn/platformer`. Finally, you will create smooth player movement across the environment.

Design Check for Ellipsoid-Triangle Collisions

- What transformations do we apply to collide an ellipsoid with a triangle?
- Give analogies for the three cases required to collide a sphere with a triangle. For example, "sphere-plane collision is the same thing as [blank]."
- What are the relevant pieces of information to return for when an ellipse collides with a triangle?
- What is the "mtv slide", and why is it necessary for proper movement? Give a demonstration of a few iterations of the slide.
- What is the purpose of the "nudging" hack?

Design Check for GJK Algorithm

- What is a support function for a convex shape?
- Explain what a Minkowski Difference is and how it relates to the GJK algorithm.
- Why do we need to use the Expanding Polytope Algorithm after detecting a collision with the GJK algorithm?
- How do we expand the polytope in EPA?

Requirements for Ellipsoid-Triangle Collisions

- Analytic ellipsoid-triangle collision detection
  - Ellipsoid-triangle collision
  - Ellipsoid-edge collision
  - Ellipsoid-vertex collision
  - All collision tests return the correct parametric t value
  - Collision routines return the contact point
- Loading in environment obj file
- Smooth player movement across environment

Requirements for GJK

- GJK implementation
- EPA implementation
- Loading in environment obj file
- Smooth player movement across environment