Graphics and Computational Programming Assignment 1

Graham Rigler  
i7465070

# Part 1

## Brute-Force Ray-Triangle Intersection

### Theory

* Ray-Plane testing
* Plane contact point – Triangle testing

### Implementation

* Reason for ordering the way it is
* Pre-computation

## Bounding Volumes

### General Theory

A bounding volume (BV) is a representation of an object in world-space using some primitive shape, such as a cube or a sphere. This allows for more complicated geometry to be represented in a simplistic, if not completely accurate, form. A popular implementation of BVs is the Axis-Aligned Bounding Box (AABB), this encompasses an object in a box that is relative to the X, Y and Z axis as opposed to being oriented to the object (although this implementation exists as the Oriented Bounding Box). The purpose of this in the context of this assignment is that it allows for each ray to be tested against the bounding volume before being tested against triangle, this is a computationally cheaper test and has been explained further below.

* TODO – Find a good example of an AABB’s components being projected onto each axis – it’ll help massively with the explanation and stuff

### Per-Object

* Working out a bounding volume for an object and what that looks like / the significance

### Per-Triangle

#### Differences

#### Memory Overhead vs. Performance

### Bounding Volume Hierarchy

* What it is, the benefits and why I’ve not done it (laziness in truth, but easy marks in the report)

## Algorithm Analysis

# Part 2