Computer Games: ARGame

James Rogers, 100062949

School of Computing Sciences, University of East Anglia, UK

Abstract

Abstract.

1 Introduction

2 Requirements

2.1 Specification

2.2 MoSCoW

3 Design

3.1 IID

3.1.1 Version 0.1

3.2 Technologies

3.3 Inital Architecture

3.4 Final Architecture

3.5 Finite State Machine

4 Implementation

4.1 Graphics

4.1.1 Drawing

4.1.2 Model Loading

Model loading is performed using the `ModeLoader` class. It provides an interface of static methods which take a file path to the model file as their parameter, then return an instance of the `Model` class with the data loaded inside it.

The Assimp[1] model loading library was used to import 3D mesh and material data from Wavefront .obj formats. The Assimp library is written in C++, therefore a class, `AssimpModelLoader` was implemented in C++ to interact with the Assimp library, interpret and temporarily store it's output. The Swift `ModelLoader` class interacts with `AssimpModelLoader` through a C bridge header interface; which consists of functions that Swift can interact with to retrieve data from `AssimpModelLoader`.

[1] `www.assimp.org`

4.1.3 Photogrametery

4.2 Collisions

4.2.1 Collision Detection

4.2.2 Collision Resolution

4.3 Augmented Reality

4.3.1 Marker Tracking

4.3.2 Smoothing

4.3.3 Illumination Model

5 Testing

5.1 Unit Testing

5.2 Intergration Testing

6 Conclusion