Pier436

Software Requirement Specification

Common Infrastructure Team

6 March 2007

Matthew Bensley
Zachary Patterson
Troy Ruths
Jared Stein

Table of Contents

Table of Figures

The common infrastructure team seeks to specify a type of application, and to provide a platform on which an application of this type could be built. The platform is motivated by three needed applications: the Building Security Simulation, the Island/Planet Exploration, and the Traffic Flow Simulation. This initial specification for the common infrastructure is initially conceived from the common aspects and possible dependencies of these applications.

Objective

Provide a platform implementing the most common features of the three applications, and a framework that can be used in developing applications that will use the platform.

Requirements

1 Game Engine

Pier436 will include a game engine which must support the following:

1.1 2D Support

The game engine must support the following common 2D graphics libraries:

1.1.1 Camera draw canvas - Embeddable UI component that provides the viewpoint from a specified camera, or 'eye', in the game world. The default camera will be orthographic and look down on the XY plane.

1.1.2 Drawing utilities - A framework for adding custom vector draw and/or raster draw to the camera view specified in 1.1.1

1.1.3 Sprite rendering - Ability to draw rasterized images with alpha transparency

1.1.3.1 Sprite library - Collection of often used images and objects

1.1.3.2 User specified sprites - Ability to load custom made sprites

1.1.4 2D Sound - Playback ability for sound formats wav and mp3

1.1.5 Collision Detection - Collision between regions will be supported. Sprites with alpha transparency will utilize whatever region method such that their transparent regions are not represented by a collision region.

1.1.6 Entity Picking—ability to select entities in the game world by clicking on the canvas (1.1.1). These regions will be defined as above.

1.2 Display Controller

The game engine will support 3D objects and scenes but must maintain:

1.2.1 3D view of 2D game world - The game world can be defined in 2D yet viewed in 3D.

1.2.2 2D Libraries Crossover - All 2D libraries required in 1.1 must be supported in 3D.

1.2.3 Object Loading - 3D Object loading and rendering from popular formats (to be decided, but probably .obj)

1.2.4 User Interface - Common user interface commands that will affect the camera canvas, which will be consumed by keyboard and mouse commands, the default commands are yet to be determined but will be customizable.

1.2.4.1 Scrolling - Translate the map and all its entities.

1.2.4.2 Scaling - Adjust the relative size of the map and all its entities.

1.2.4.3 Rotation - Spin the orientation of the map and all its entities.