ChangeEngine Game Engine

By Cedric Wienold,
California Polytechnic Unversity,
San Luis Obispo, CA

Advised by Dr. Michael Haungs

April 28, 2011

Date Submitted:_____

Advisor:_____

Contents

Abstract

The goal of this project is to design a game engine which will provide the framework for fast, simple deployment of games. This project allows me to learn from several different game engine designs in order to create an engine concrete enough to allow programmers to work on game design rather than programming the structure of windows, graphics, sounds, et cetra, but abstract enough to give the programmer freedom to program a wide range of 2D games. The engine will be available as a static library programmed in C++, which gives the user the power to program in any language that can implement such a library. A major feature of this engine will be the pluggable functions which can be implemented by the user, for functions such as drawing, sound, and artificial intelligence.

1 Introduction

Introduction goes here. Forget latin. It bogs down the spell checker.

2 Requirements

ChangeEngine will fulfill the following software requirements:

1. General Requirements
   (a) Single library implementing all functions for a rudimentary game
      i. Engine Class
      ii. Level Container
      iii. Game Object Container
      iv. Input Listener
      v. Collision Listener
   (b) Extensible classes for programmers to "plug in" their own logic
      i. Artificial Intelligence
      ii. Input Handler
      iii. Collision Handler

3 Tools and Teamwork

This project will primarily be programmed by Cedric Wienold using various programming environments supporting C++ code. Teamwork will be limited to interviews with industry professionals concerning algorithms and class organization advice.

4 Research

Research will be conducted on the following areas, and methods and results will be included in the appenix.

1. SDL Usage

2. Collision Detection

3. Artificial Intelligence

5 Future Improvements

This project will remain open source for any programmer to extend and improve upon it. ChangeEngine will maintain a pluggable interface in many ways so that programmers need only design their own functions and pass them into the engine if they wish to use them. Artificial intelligence, collision detection, and input detection are all areas which programmers can design their own plug-ins for.

6 Conclusions

A Research

A.1 Collision Detection

In an interview with Cal Poly Alumni Computer Science alumni Daniel Nutting, methods of collision detection employing best possible complexity in average cases were outlined, with worst possible complexity in edge cases being sacraficed. This method will be called the "Grid Collision Detection" method.

A.1.1 Grid Collision Detection

Theory: Truly basic collision detection entails each object checking against all other objects for collision, and doing the same for every object. This leads to an average $O(n^2)$.

The most apparent problem is repeated checks. One solution can be a stack method. For each object, push onto stack. Check collision with everything behind it. INCLUDE PSEUDOCODE HERE.

The next issue is restricting the number of objects to check on. If objects are not close to each other, there is no point in checking. This is where the grid comes in. For a set of objects, calculate the average X and Y coordinates, and split the field there into 4 quadrants. Do this more as is necessary for the number of objects. INCLUDE PSEUDOCODE HERE.

B Source Code

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla ac felis nibh. Maecenas vel risus erat, non convallis risus. Vivamus congue suscipit porta. Fusce cursus lobortis metus et rhoncus. Nulla in ligula eget felis vestibulum iaculis. Nullam ac enim at lorem iaculis accumsan. Nullam at diam a turpis blandit tincidunt ut vitae ipsum. Integer quis enim eget est porta rhoncus. Vivamus commodo, lacus nec laoreet pretium, nunc lectus tincidunt mauris, blandit lobortis erat ligula pretium quam. Mauris et libero nec felis tempor aliquam. Maecenas lacus ante, fringilla eget convallis et, fringilla a dui. Nam bibendum, elit porttitor blandit imperdiet, est magna fringilla diam, eu accumsan ante nisl at metus. Cras non gravida quam. Nulla nec lorem lorem. Proin et massa mauris. Aenean non metus orci. Maecenas porta purus et ligula consequat tempus.