Lời Cảm Ơn

Lời đầu tiên tôi xin bày tỏ lòng biết ơn chân thành tới các thầy cô giáo Trường Đại Học Bách Khoa – Đại Học Đà Nẵng nói chung và các thầy cô bộ môn công nghệ phần mềm nói riêng. Trong suốt thời gian tôi học tại trường, các thầy cô đã luôn tận tình dạy dỗ, chỉ bảo để tôi có được kết quả như ngày hôm nay.

Đặc biệt, tôi xin chân thành cảm ơn giảng viên, tiến sĩ Lê Thị Mỹ Hạnh. Cảm ơn cô vì những định hướng, nhận xét quý báu và động viên kịp thời để tôi hoàn thành khóa luận này. Tôi cũng cảm ơn Gameloft – Đà Nẵng đã tạo điều kiện cho tôi có môi trường làm việc trong quá trình hoàn thành khóa luận này.

Cuối cùng tôi xin bày tỏ lòng biết ơn sâu sắc tới gia đình, bạn bè, những người đã luôn động viên, giúp đỡ tôi cả về vật chất lẫn tinh thần trong suốt những năm tháng qua.

Đà nẵng, ngày …. Tháng … năm…

Sinh viên

Tóm tắt

Sự phát triển của công nghệ mô phỏng và giải trí, ngày càng có nhiều ứng dụng đồ họa và game 3D được xậy dựng nhằm phục vụ mục đích nghiên cứu và mô phỏng và giải trí… Cùng với đó, công nghê phần cứng cũng phát triển rất nhanh, đặc biệt là các phần cứng xử lý đồ họa có thể lập trình được và các ngôn ngữ lập trình trên các phần cứng này.

Nói đến game thì chắc phần lớn mọi người đều đồng ý đó là một loại hình giải trí có sức thu hút rất lơn, nhất là với giới trẻ hiện nay. HIện nay game được phát triển mạnh trên các thiết bị di động.

Xuất phát từ thực tế đó, nội dung của đồ án tập trung nghiên cứu xây dựng một hệ thống game engine để build game trên các thiết bị di động sử dụng Opengl ES 2.0. Khi phát triển game 3D, người lập trình không cần phải quan tâm đến quá trình quản lý tài nguyên, cũng như xủ lý đồ họa ở mức thấp bên dưới. Thay vào đó, họ chỉ cần tập trung vào quản lý ở mức logic các thành phần của game, hoặc thêm mới các thành phần cơ sở do hệ thống cung cấp.

1. Mục Tiêu

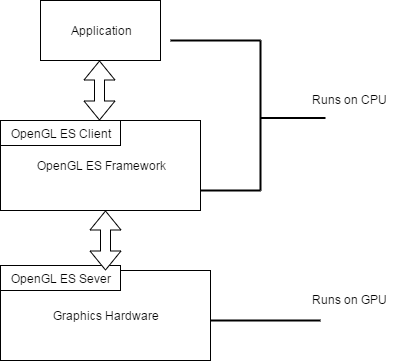
* Tìm hiều lập trình OpenGL ES 2.0
* Tìm hiểu cấu trúc 1 game
* Thiết kế mô hình hệ thống game
* Xây dựng game engine
* Xậy dựng 1 game 3D và chuyển sang di động ( Android )
* Thiết kế game Kill The Rat

1. Giới thiệu chung

Chương 1: Tổng quan về Game Engine

1. Khái niệm Game Engine

Game Engine là một công cụ hỗ trợ, một lớp trung gian ở giữa các ứng dụng game và nền tảng bên dưới , các thư viện lập trình cấp thấp. Game Enigine giúp phát triển ứng dụng game một cách nhanh chóng và đơn giản, đồng thời cung cấp khả năng tái sử dụng mã nguồn cao do có thể phát triển nhiều ứng dụng game khác nhau từ một game Engine.



Hình 1: Kiến trúc tổng quan của Game Engine

1. Mô hình Game Engine

Game Engine hỗ trợ đa nền (Window PC và Mobile Android). Và nhắm đến thị trường Game Mobile đang rất phổ biến trên các thiết bị di động.

Đầu tiên là khả năng chạy đa nền. Để đạt được điều này, Engine sử dụng các bản build khác nhau trên các nền tảng khác nhau (chứ không phải sử dụng thông dịch). Để giải quyết vấn đề này, đầu tiên em chọn build game bằng Visual Studio 2013 trên nền tảng window sử dụng C++. Sau đó sử dụng NDK trên android để sử dụng code C++ và build game ra Mobilde Android.

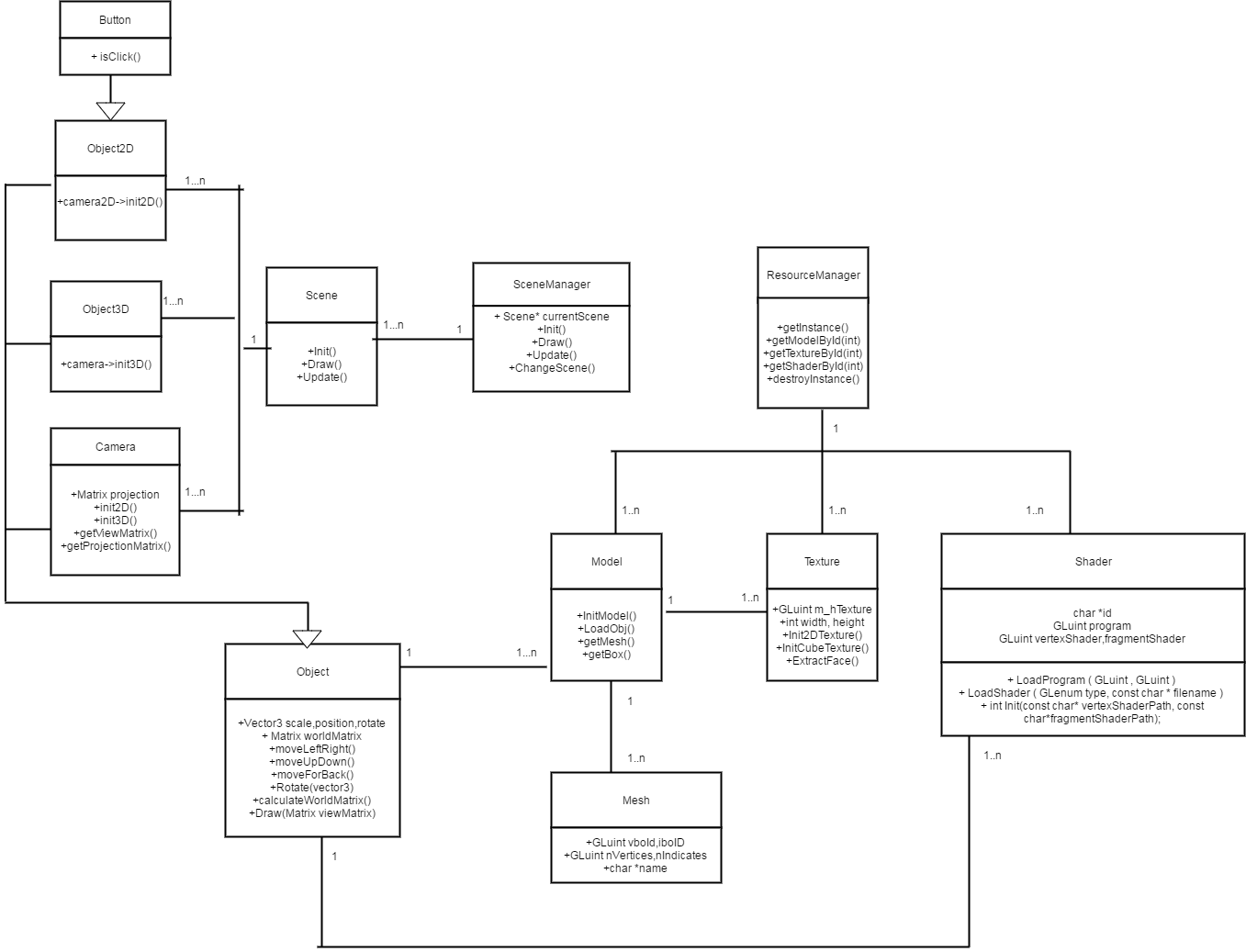
Engine được thiết kế hướng đối tượng và yêu cầu hiệu năng chạy cao nên việc sử dụng ngôn ngữ C++ ngôn ngữ đáp ứng điều kiện trên.

1. Các thành phần chính của Engine

Engine hỗ trợ người dùng các công việc thiết yếu để tao cảnh 3D và các mô hình 3D. Người dùng sẽ không cần biết nhiều đến những công việc tầng thấp như quá trình đọc file tài nguyên, sử dụng các API đồ họa 3D, tạo các hiệu ứng… mà quản lý logic các đối tượng bằng các giao diện do Engine cung cấp.

Engine giao tiếp với phần cứng đồ họa thông qua thư viện đồ họa 3D cấp thấp, cụ thể ở đây là OpenGL .

Sau đây là biểu đồ lớp của engine



Hình 1.2 : Game Engine Class Diagram

* Resource Manager là mô đun quản lý tài nguyên cần thiết để xây dựng nên các Object và cảnh 3D như texture 2D , cubemap… Mô đun này giúp cho việc dễ dàng quản lí tài nguyên , tăng tốc độ game và tránh việc sử dụng các model , texture trùng nhau
* Scene Manager là mô đun quản lý các màn chơi trong game (Scene), bao gồm các hàm chuyển màn chơi, pause ..
* Scene là một cảnh game level, chẳng hạn như các cảnh trong 1 game thường có chia làm Start Scene, Play Scene , GameOver Scene. Việc chia làm các Scene và thông quan Scene Manager giúp cho việc quản lý game trở nên dễ dàng hơn, dễ control hơn trong quá trình phát triển game.
* Object chia làm 2 loại là Object 2D và Object3D. Việc mô phỏng chuyển động của các Object này sẽ thông qua một ma trận worldMatrix , dựa vào các vector postion, scale, rotate mà ta sẽ mô phỏng chuyển động cho các Object này. Chúng ta sẽ bàn kỹ hơn về vấn đề này ở chương sau.
* Camera trong engine giống như góc như ngoài đời thật. Ở các vị trí khác nhau sẽ cho ra các kết quả khác nhau khi nhìn các Object. Camera cũng là một object tren engine , và bằng vào việc di chuyển camera trong môi trường 3D sẽ cho ra các kết quả khác nhau khi nhìn vào một cảnh game.
* Một Object sẽ bao gồm nhiều model và texture và mỗi model sẽ được tạo ra từ nhiều mesh.
* Shader sẽ giúp chúng ta load các file shader tạo hiệu ứng màu sắc cho object. Bằng việc này chúng ta chỉ việc thao tác với các file shader để tạo hiệu ứng cho object một cách đơn giản hơn.

Chương 3: Cơ sở lý thuyết

3.1 Vòng lặp game

Phần cốt lõi của hầu hết các game chính là vòng lặp được dùng để cập nhật và hiển thị trạng thái của game.

Một ví dụ đơn giản:

|  |  |
| --- | --- |
|  | while(gameRunning)  {      processInput(); // keyboard, mouse,...      updateGame();      draw();      // checkGameOver();  } |

Minh họa:

