

UNIP - UNIVERSIDADE PAULISTA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
CAMPUS ARARAQUARA

PROPOSTA DE TRABALHO DE CONCLUSÃO DE CURSO:



1. Dados Gerais

Aluno: Paulo Maria Neto

Título provisório: Desenvolvimento de um motor para criação de jogos em java

Área(s): Computação gráfica, Desenvolvimento de jogos, Game engine

Professor-orientador: Daniele Colturato

Data: 15 de Março de 2016

2. INTRODUÇÃO

Os primeiros jogos de video game foram produzidos por equipes pequenas com gráficos minimalistas devido as limitações do hardware da época, com os avanços da tecnologia equipes cada vez maiores foram se formando e a criação de jogos totalmente do zero se tornou algo inviável para os desenvolvedores, surgiu então a necessidade de criar um software que pudesse reaproveitar os elementos básicos dos jogos como capturar as entradas do usuário ou renderizar imagens na tela, esses softwares são chamados de Motores de jogos ou Game engines.

As primeiras engines foram desenvolvidas em linguagens de baixo nível como C e eram construídas visando apenas um tipo de jogo, hoje em dia os desenvolvedores optam por utilizar mais de uma linguagem no desenvolvimento de jogos, dividindo as engines em duas partes, a parte do núcleo (geralmente desenvolvida em C++ e que é responsável por renderizar imagens, capturar

entradas do usuário, reproduzir audio, etc) e a parte da lógica de jogo (geralmente desenvolvida em uma linguagem de alto nível interpretada através de uma maquina virtual) afim de facilitar a reutilização da engine para mais de um tipo de jogo.

3. PROBLEMA

Hoje em dia é possível desenvolver jogos em praticamente qualquer linguagem, entretanto desenvolver um jogo do zero é uma tarefa extremamente difícil e trabalhosa, o que faz com que vários desenvolvedores recorram a ferramentas existentes no mercado. Apesar existirem ferramentas gratuitas para o desenvolvimento de jogos, a grande maioria possui o código-fonte fechado, complexo ou é desenvolvida utilizando linguagens de baixo nível, o que obriga os desenvolvedores a investir horas (e até mesmo dinheiro) em treinamentos para aprender a utilizar essas ferramentas ou desenvolver alguma extensão para atender as suas necessidades.

4. OBJETIVOS

A ideia é desenvolver um conjunto de ferramentas de código aberto que facilite o desenvolvimento de jogos independente do seu estilo e ao mesmo tempo ofereça flexibilidade e facilidade para que o usuário desenvolva extensões da ferramenta afim de atender as suas necessidades.

4.1. OBJETIVOS GERAIS

Desenvolver um motor que auxilie no desenvolvimento de jogos utilizando Java e JavaScript.

4.2. OBJETIVOS ESPECIFICOS

- Desenvolver sistema de renderização e animação 2D com suporte a OpenGL 3.0 ou superior
- Implementar suporte a reprodução de audio e efeitos sonoros utilizando OpenAL
- Implementar suporte a simulação de fisica 2D utilizando a biblioteca Box2D
- Implementar controle sobre as entradas do usuario via teclado, mouse e controles USB

- Implementar suporte a scripts desenvolvidos em JavaScript para o controle de lógica em tempo de jogo
- Desenvolver interface para criação, edição e teste de jogos

5. JUSTIFICATIVA

Dentro da comunidade de desenvolvimento de jogos é possível encontrar varias ferramentas para auxiliar no desenvolvimento amador de jogos, no entanto a grande maioria dessas ferramentas não possui um editor gráfico, o que dificulta o trabalho com designers que não possuem conhecimento sobre programação.

Esse projeto visa facilitar esta interação oferecendo uma interface para que sejam criados os níveis, dessa forma designers podem criar os níveis e os programadores podem focar unicamente na lógica aplicada sobre os objetos do jogo.

6. LEVANTAMENTO DE LITERATURA

6.1. GAME ENGINE

A engine ou motor de jogo é responsável por simular a parte física do mundo real dentro do ambiente do jogo. O nível de complexidade da engine define também o nível de realidade que a mesma proporciona ao usuário. (BRITO, 2011).

Uma game engine é um software extensível que pode ser usado de base para vários jogos com poucas modificações. (GREGORY, 2009 e EBERLY 2001).

6.2. OPENGL

OpenGL é uma biblioteca desenvolvida em C que fornece acesso aos recursos gráficos de vários tipos de dispositivos, podendo ser implementado em hardware (como placas de video) ou em software caso não exista um hardware gráfico. (SHREINER, 2013).

Segundo (SELLERS, 2011) a grande vantagem em usar OpenGL ao invés de outras API é o fato de ela ser uma biblioteca de baixo nível, o que permite não apenas reimplementar as técnicas de renderização padrões, mas também inventar e testar novas técnicas.

6.3. SHADERS

Shader são como programas que são compilados especificamente para a Unidade de Processamento Gráfico (GPU). OpenGL possui todas as ferramentas necessárias para compilar o código fonte e gerar o código necessário para que a GPU o execute. (SHREINER, 2013).

6.4. OPENAL

OpenAL é uma biblioteca de audio multi-plataforma que possibilita a reprodução de audio em três dimensões, o que a torna ideal para varias aplicações de audio, principalmente para jogos. (HIEBERT, 2007)

7. METODOLOGIA

O projeto será desenvolvido utilizando a metodologia ágil de desenvolvimento SCRUM, utilizando a linguagem de programação Java (para desenvolver o núcleo da aplicação e interação com JavaScript), as bibliotecas LWJGL (Lightweight Java Game Library) que oferece suporte a chamadas OpenGL/OpenAL e controle sobre as entradas do usuário e Box2D para simulação de física 2D. A aplicação será desenvolvida utilizando a IDE IntelliJ IDEA 15 CE desenvolvida pela empresa JetBrains.

Para a conclusão do projeto serão realizadas as seguintes etapas:

7.1. FRAMEWORK

- Pesquisa sobre as tecnologias utilizadas atualmente e leitura da documentação das ferramentas utilizadas.
- Definição da arquitetura do software e diagramação de seu funcionamento.
- Implementação das funcionalidades básicas como captura de entradas do usuário, estrutura de dados, criação de janelas e buffers.
- Implementação do sistema de renderização utilizando a técnica de Forward Rendering.
- Implementação do sistema de scripting e controle sobre a lógica de jogo em tempo de execução utilizando JavaScript.

9. REFERÊNCIAS

GREGORY, JASON (2009). Game engine architecture 1st edition. Wellesley, Massachusetts: A. K. Peters, Ltd

EBERLY, DAVID H. (2001). 3D game engine design: a practical approach to real-time computer graphics. São Francisco: Morgan Kaufmann.

BRITO, ALLAN (2011). Blender 3D: jogos e animações interativas. São Paulo: Novatec.

SHREINER, DAVE; SELLERS, GRAHAM; KESSENICH, JOHN (2013). OpenGL Programming Guide 8th Edition: The Official Guide to Learning OpenGL, Version 4.3. New Jersey: Addison-Wesley

SELLERS, GRAHAM; JR. W. S. RICHARD; HAEMEL, NICHOLAS (2011). OpenGL SuperBible 5th Edition: Comprehensive Tutorial and Reference. New Jersey: Addison-Wesley

HIEBERT GARIN (2007). OpenAL Programmer's Guide.