

Desenvolvimento de jogos eletrônicos com Unity

Luiz Sergio de Britto Pinto¹, Jucimar Maia da Silva Junior¹

¹Escola Superior de Tecnologia – Universidade do Estado do Amazonas (UEA)
Av. Darcy Vargas, 1200, Parque Dez – 69.050-020 – Manaus – AM – Brasil

{lsergiobp, jucimar.jr}@gmail.com

Abstract. *This paper discusses in general the proposed project for the course-work. It will present the work goal, the methodology used and why this was the chosen theme. The game that will be developed is a 2D platform game and the tool chosen for the project was Unity for having lots of features for game development.*

Resumo. *Este documento discute de forma geral a proposta de projeto para o trabalho de conclusão de curso. Será apresentado o objetivo do trabalho, a metodologia que foi utilizada e o motivo pelo qual foi escolhido o tema. O jogo a ser desenvolvido é um jogo plataforma 2D e a ferramenta utilizada é o Unity por possuir vários recursos para desenvolvimento de jogos.*

1. Descrição do Problema

O desenvolvimento de jogos eletrônicos é uma área que cada vez cresce mais. Segundo o jornal da globo, o Brasil é o país onde o mercado de jogos eletrônicos mais cresceu no mundo em 2012. As vendas de jogos e equipamentos atingiram 1,6 bilhão de reais neste mesmo ano. Com essa ampliação do mercado de entretenimento virtual, cada vez mais, demanda-se profissionais nesta área que exige conhecimento técnico específico e diferente daquele encontrado em processos de engenharia bem difundidos.

O processo de desenvolvimento de jogos eletrônicos costuma ser bastante complicado. Além do conhecimento específico necessário, é preciso avaliar as tecnologias disponíveis, requisitos a serem satisfeitos e um conhecimento na área de design para manipular sprites e animações.

Existem vários frameworks disponíveis para desenvolvimento em várias plataformas diferentes como o Blender Game Engine, Cry Engine, UDK, entre outros e cada um, além do conhecimento geral sobre o processo de desenvolvimento de jogos, cobra um conhecimento específico na ferramenta para melhor uso da mesma. Cada um tem suas vantagens e desvantagens, mas, na maioria dos casos, o framework limita o desenvolvimento para uma plataforma.

A ferramenta escolhida para o desenvolvimento do jogo foi o Unity. Essa é uma ótima ferramenta para criação de jogos 3D, porém foi escolhido o desenvolvimento de um jogo plataforma 2D para o trabalho. Jogo plataforma é o nome dado a um gênero de jogos em que o jogador corre e pula entre plataformas e obstáculos, enfrentando inimigos e coletando objetos bônus. Alguns dos mais conhecidos e difundidos exemplos destes tipos de jogos são o Super Mario Bros e o Sonic the Hedgehog. Algumas ferramentas como o Game Maker e o Construct são mais fáceis para iniciantes criarem um jogo 2D, porém

não tem tanto poder quanto outras engines maiores. Como foi citado, existem várias engines poderosas para a criação de jogos, porém, algumas são bastante difíceis de serem utilizadas para quem nunca usou, como o caso da Blender Game Engine. Outras, como o UDK (Unreal Development Kit) e a Cry Engine, além de serem engines muito pesadas, possuem um custo muito alto para funcionalidades medianas e avançadas, diferente do Unity que em sua versão gratuita possui muitas funções avançadas.

Outros motivos pela escolha do Unity são:

- Mesmo utilizando a versão gratuita do Unity, o mesmo oferece muitos recursos para o desenvolvimento, criação de cenários, etc.
- Tem suporte a vários formatos de arquivos(Assets) como o .max(AutoDesk 3D Max), .blend(Blender), entre outros.
- Permite o desenvolvimento em várias linguagens como Javascript, C# e Boo. Para o projeto foi escolhido a linguagem C# por ter mais documentação e ser uma linguagem mais familiar.
- Permite a exportação do jogo para várias plataformas como Windows, Mac, Android, Iphone, Windows Phone, Xbox, Wii. Para algumas delas, é necessária uma versão específica e paga do Unity.
- É uma ferramenta que está crescendo muito no conceito dos desenvolvedores de jogos e essa seria uma ótima oportunidade de aprender a usar esse framework.

2. Trabalho Relacionados

Já existem vários jogos desenvolvidos utilizando o Unity. Dentre eles estão:

- Monkey Quest, desenvolvido pela Nick Games. Site do jogo: <http://www.monkeyquest.com/>
- Bad Piggies, desenvolvido pela Rovio, mesma marca que criou o famoso Angry Birds. Site do jogo: <http://www.badpiggies.com/>
- Harold, jogo 2D desenvolvido pela Moon Spider Studio. Site do jogo: <http://www.moonspiderstudio.com/games.php>
- Slingshot Racing, desenvolvido por Crescent Moon. Site do jogo: <http://crescentmoongames.com/wordpress/slingshot-racing/>
- Super Crossfire, por Radian Games. Site para download do jogo: <http://www.desura.com/games/super-crossfire>
- Temple Run 2, continuac, ao do jogo de sucesso Tempe Run, desenvolvida por Imani Studios. Site do jogo: <http://www.templerun2.com>
- Ballistic, primeiro FPS nacional criado para o facebook por Aquiris.
- Dungeonland, desenvolvido em parceria de Critical Studios e Paradox Interactive. Site do jogo: <http://www.dungeonlandgame.com/>
- Ravensword: Shadowlands, desenvolvido por Crescent Moon Games. Site do jogo: <http://crescentmoongames.com/wordpress/ravensword-shadowlands/>

3. Objetivo

O trabalho tem por objetivo portar um jogo eletrônico de plataforma 2D feito em flash(NutMeg) para Unity e verificar as vantagens e desvantagens do framework. No jogo, o jogador irá controlar um pássaro que poderá virar e andar a direita, à esquerda e pular sobre as plataformas do jogo. Haverá plataformas fixas e móveis. Terá distribuído

pelo cenário, estrelas que deverão ser coletadas pelo jogador. Se o número de estrelas coletadas for igual ao número total de estrelas na fase, o jogador vence e passa para a próxima fase, caso tenha. Se for a última fase e o jogador coletar todas as estrelas, é mostrada uma tela de vitória. Caso o jogador caia na água ou, seja tocado pelo inimigo que será um gato, o jogador perde e a fase é reiniciada.

4. Objetivos Específicos

O jogo terá três fases, sendo a primeira uma fase fácil para o jogador se acostumar com os comandos e o objetivo do jogo, não tendo inimigos para atrapalhar e um mapa pequeno. A segunda fase já terá um mapa maior e alguns inimigos para atrapalhar e plataformas que se movem verticalmente. Já a terceira e última fase terá mais inimigos, mais estrelas para serem coletadas em vários lugares e plataformas se movendo horizontalmente e verticalmente. O jogo será exportado para um arquivo executável.

5. Metodologia

Para o jogo ser concluído, deve ser feito:

- Manipular sprites do personagem: O personagem deve obedecer aos comandos do jogador, então, por exemplo, caso o jogador aperte para virar para a esquerda, a sprite do personagem virado para a esquerda deve aparecer.
- Manipular comandos do personagem. O que deve ser feito quando o jogador apertar as teclas de comando.
- Criar controle para a câmera acompanhar o personagem.
- Criação do cenário: Criar as fases do jogo, plataformas, inimigos, plano de fundo, etc. manipulando os sprites.
- Criar inimigos que, ao serem tocados, causam derrota no jogo.
- Adicionar som ao jogo.
- Disponibilizar visualização da pontuação do jogo (quantidade de estrelas que foram coletadas e quantidade total).
- Permitir ao jogador pausar o jogo e voltar do mesmo lugar na hora que quiser.
- Criar tela para iniciar e finalizar o jogo.
- Exportar o jogo para um arquivo executável no Windows.

5.1. Sprites

Em computação gráfica, um sprite (do latim spiritus, significando "duende", "fada") é um objeto gráfico bi ou tridimensional que se move numa tela sem deixar traços de sua passagem (como se fosse um "espírito").

Os sprites foram inventados originalmente como um método rápido de animação de várias imagens agrupadas numa tela, em jogos de computador bidimensionais, usando hardware especial. A medida que a performance dos computadores melhorou, esta otimização tornou-se desnecessária e o termo evoluiu para referir-se especificamente às imagens bidimensionais que eram integradas numa determinada cena, isto é, figuras geradas por hardware ou software eram todas referenciadas como sprites.

No jogo desenvolvido, para os vários movimentos do personagem principal e movimentos dos inimigos, foi utilizada uma folha de sprites(sprite sheet). Essa folha de sprites, contém as várias sprites que podem ser utilizadas. Cada sprite tem exatamente

as mesmas dimensões uma da outra, sendo divididas por retângulos. A figura 1 mostra a folha de sprite do personagem principal e a figura 2 ilustra a folha de sprite do inimigo. Cada comando chamado pelo jogador, deve fazer mudar a sprite.



Figura 1. Folha de sprites do personagem principal



Figura 2. Folha de sprite do inimigo

5.1.1. Sistema de controle de sprites

Como o foco do Unity, na versão utilizada para o projeto, não é o desenvolvimento de jogos 2D, a engine não fornecia um sistema para controle das sprites. Ao fazer pesquisa, foi verificado que existiam vários plugins que faziam essa funcionalidade, mas em sua maioria eram pagos.

Após muitas pesquisas sobre como fazer esse controle, foi encontrada algumas video-aulas que davam uma base para fazer o seu próprio sistema de controle de sprites via código. O sistema final foi dividido em classes da seguinte forma:

- **SpriteRect:** Classe que representa cada retângulo de uma folha de sprites. Essa classe possui as informações sobre as dimensões de cada retângulo dentro da folha de sprites.
- **SpriteSheetInfo:** Classe que guarda informações como nome da sprite, textura a ser utilizada, entre outras informações.
- **Sprite:** Classe responsável pela mudança de uma sprite para outra. Essa classe que controla o tempo que cada sprite deve aparecer ou ser trocada.
- **SpriteSet:** Classe responsável por chamar o carregamento das sprites. Essa classe é chamada antes de iniciar o jogo e já carrega todas as sprites do jogo.
- **XMLParser:** Classe que realmente interpreta e carrega as sprites. Todas as informações de sprites são guardadas em um arquivo XML chamado `Sprite-Data.xml`. Essa classe de parser, interpreta o xml e todas as informações contidas nele e carrega os sprites passados pelo mesmo.

5.2. Movimento do jogador

O personagem pode se movimentar para direita, para a esquerda e pular para qualquer um dos lados. A cada comando desses, foi necessário implementar uma lógica para, além de mudar as sprites, fazer o jogador realmente se mover. Foi utilizado um módulo oferecida pelo próprio Unity, chamado `Character Controller`. Esse controller, te dá várias funcionalidades para controle de movimento e colisões, principalmente em um cenário 2D.

5.3. Controle de câmera

A câmera deve sempre seguir o personagem durante seu movimento pelo cenário. Existe um jeito bastante simples de fazer isso que a engine proporciona. Apenas botando o objeto da câmera principal como um objeto filho do personagem, a câmera iria seguir o movimento do personagem.

Esse é um jeito simples porém daria alguns problemas durante o jogo, como o caso que a câmera acabaria mostrando a parte que não existe do cenário. Para resolver esse problema, foi feito um controle simples pelo código que faz com que a câmera acompanhe o personagem até o limite especificado para cada fase.

5.4. Criação do cenário

Na maioria dos jogos, os cenários são criados por outras ferramentas especializadas e importadas para o unity. No caso desse projeto, nenhuma ferramenta especializada foi utilizada.

A partir de uma página de sprites com várias partes possíveis do cenário, foram criados sprites específicos. Com esses sprites específicos, foi utilizado de outra ferramenta do unity, chamado prefab. Um prefab é apenas um modelo de objeto que pode ser instanciado ou destruído em qualquer momento do jogo. Tendo esse modelo, ao iniciar o jogo, é montado o cenário inteiro a partir desses prefabs.

Utilizar um prefab é um jeito bem fácil e prático para instanciar objetos que se repetem várias vezes no cenário durante o jogo. Todos eles possuem sempre as mesmas propriedades.

5.5. Inimigos

Para a criação dos inimigos, foi utilizado do mesmo sistema de controle de sprites utilizado no controle do personagem principal. A diferença é que foi criada uma lógica diferente para a movimentação. Cada inimigo já tem uma rota certa no mapa com um limite máximo a ser percorrido, podendo começar a rota para esquerda ou para a direita, dependendo da configuração. Do mesmo jeito que o cenário, a criação dos inimigos é feita por prefabs.

Foi implementada uma lógica que ao ser detectada uma colisão entre um objeto instanciado de um inimigo e o personagem principal, o jogo considera como derrota e a fase reinicia.

5.6. Som de fundo

Para o som de fundo, a unity nos proporciona uma funcionalidade chamada de Audio Source, a qual você pode escolher um som e mudar algumas configurações do mesmo.

Ao criar essa funcionalidade em um objeto vazio, o objeto foi incluído como um objeto filho do personagem principal, desse modo o som iria sempre estar ativo não importando se o player se movesse.

5.7. Pontuação do jogo

Durante a criação do cenário, também são criados vários objetos, utilizando do prefab, que são estrelas a ser capturadas pelo personagem. Para pontuação do jogo, foi implementado

uma lógica na qual a cada colisão do personagem com um objeto estrela, a pontuação aumenta. Caso a quantidade de estrelas capturadas fosse o mesmo da quantidade de estrelas totais criadas no cenário, significa que o jogador passou da fase.

5.8. Pausar o jogo

Foi criada uma lógica para toda vez que o jogador apertar a tecla ESC, todos os objetos do jogo são congelados e nenhum comando pode ser executado até a próxima vez que o jogador apertar novamente a tecla ESC e o jogo voltaria a rodar de onde parou.

5.9. Tela inicial e final

Ao começar o jogo, foi criada uma tela contendo o nome do jogo e informações para iniciar o mesmo. Ao apertar a tecla enter, essa tela some e o jogo inicia realmente.

Caso o jogador vença a terceira e última fase, uma tela que informa a vitória do jogador aparece. Nessa última tela a única opção do jogador é apertar a tecla enter e sair do jogo.

5.10. Exportação do jogo

Como foi dito, o Unity proporciona um jeito fácil para exportar o jogo para várias plataformas. No caso desse jogo, foi realizada apenas a exportação para um arquivo executável(.exe) para PC.

6. Conclusão

As maiores dificuldades encontradas foi exatamente pelo fato de a engine não ter um foco muito grande na criação de jogos 2D, desse jeito, não oferecendo a quantidade de recursos que é disponibilizado para um jogo 3D. Outra dificuldade grande foi a parte de criação de cenários. Como não conhecia nenhuma ferramenta especializada na criação do cenário e das sprites, muito tempo foi gasto na implementação da lógica para montar o cenário.

O Unity é uma ferramenta muito poderosa para qualquer tipo de jogo, porém existem outras ferramentas especializadas que não possui a quantidade de funcionalidades que o Unity proporciona, porém, para um jogo de porte pequeno, acabaria facilitando muito.

O objetivo do projeto era explorar e aprender a usar essa ferramenta que está cada vez mais difundida entre os criadores de jogos do mundo inteiro. É uma ferramenta com muitas opções mesmo na versão gratuita e muito poderosa para todo tipo de jogo.

7. Cronograma

As atividades realizadas no processo de desenvolvimento deste trabalho se encontram detalhadas abaixo.

Etapa	Ago	Set	Out	Nov	Dez
Estudo do Unity	26-31	01-05			
Proposta do TCC		19-26			
Estudo do sistema de Sprites		06-09			
Criação do sistema de sprites		10-18			
Criar personagem com sprites		26-30	01-05		
Movimentar Personagem(Pular, Andar)			06-15		
Criação de um cenário básico com sprites			16-20		
Melhorar cenário utilizando outros objetos			21-23		
Implementar movimentação nas plataformas			24-31		
Instanciar estrelas no mapa				01-03	
Utilizar sistema de colisão para capturar estrelas				04-06	
Exibir pontuação				07-08	
Criar tela inicial				09-10	
Criar segunda fase				11-15	
Criar terceira fase				15-20	
Instanciar inimigos no mapa				21-25	
Logica para derrota do jogador				26-28	
Criar funcionalidade de pause				29	
Criar tela de vitória				30	
Correção de erros					01-10
Exportar jogo para executável					11-15

8. Referências

Carlos de Lannoy. Disponível em <http://g1.globo.com/jornal-daglobo/noticia/2013/05/brasil-lidera-crescimento-do-mercado-de-jogos-eletronicosem-2012.html> . Acessado em: 24/09/2013.

Unity. Disponível em <http://portuguese.unity3d.com/gallery/made-withunity/game-list> . Acessado em: 25/09/2013.