

Desenvolvimento de um jogo educativo para o estudo de Física

Gabriel Rossi Zanini¹, Guilherme Chagas Kurtz¹, Ricardo Frohlich da Silva¹

¹Centro Universitário Franciscano – Santa Maria – RS – Brasil

gabrielroszan@gmail.com, guilhermekurtz@unifra.br,
ricardo.frohlich@unifra.br

Abstract. *This paper aims to describe the development of an electronic game that serves as an aid to educators through the use of the concept of Gamification. The development will use the Unity game engine and the programming language C#, being implemented using the FDD methodology. Upon completion of this work, it is expected to get a game prototype that offers an alternative to present concepts of physics to students in a context of easy understanding and enabling them to see the application of these concepts in the real world.*

Resumo. *Este artigo tem como objetivo apresentar o desenvolvimento de um jogo eletrônico que sirva como um auxílio aos educadores, por meio da utilização do conceito de Gamification. O desenvolvimento fará uso do motor de jogos Unity e a linguagem de programação C#, sendo implementado utilizando a metodologia FDD. Ao concluir este trabalho, espera-se obter um protótipo do jogo que ofereça uma alternativa para apresentar conceitos de Física aos alunos em um contexto de fácil entendimento e que lhes permita visualizar a aplicações destes conceitos no mundo real.*

1. Introdução

O avanço tecnológico nos faz pensar em como aplicar as novas tecnologias da forma mais produtiva possível. Entretanto, grande parte dos educadores ainda não está a favor de utilizar Tecnologias da Informação e Comunicação (TICs) nas escolas [Barroso e Antunes 2015].

Estes educadores precisam perceber que as mídias digitais podem ser utilizadas para apoiar as atividades do professor e do aluno por facilitarem, sobretudo, o intercâmbio de informações, a visualização de forma mais clara, utilizado recursos tecnológicos variados, tais como slides, exercícios virtuais, vídeos, plataformas de Ensino a Distância (EAD), lousas digitais, e-mails, armazenamento em nuvens, entre outros [Barroso e Antunes 2015].

Entre estes recursos, estão os jogos de computadores, que tem como intuito facilitar o trabalho dos professores, os auxiliando a passar conhecimento aos estudantes de forma mais inovadora e criativa uma vez que os jogos estão cada vez mais presentes no cotidiano dos alunos [Barroso e Antunes 2015].

Desenvolver jogos educativos que empolguem e ensinem o jogador sobre diferentes tipos de conteúdos educacionais, por meio da aplicação de conceitos como *Gamification*, desta forma, apresentando estes conceitos de uma maneira amigável e que

facilite a compreensão dos alunos, permite que os mesmos reconheçam e entendam certas regras e identifiquem os contextos em que elas estão sendo utilizadas, além de possibilitar a invenção de novos contextos para a modificação das mesmas [Tarouco et al. 2004].

1.1. Objetivos Gerais

O objetivo deste trabalho é desenvolver um jogo educativo para o estudo de Física, utilizando o motor de jogos *Unity*, que possa servir como auxílio aos educadores. No jogo o jogador controla um personagem principal, que pode ser masculino ou feminino. Neste jogo o personagem principal pode interagir com NPCs (*Non-player Characters*, ou Personagens Não-jogáveis). Ao interagir com um NPC, o mesmo lhe apresenta um desafio que precisaria ser resolvido utilizando conhecimentos de Física que devem ser apresentados ao jogador ao longo do jogo, por meio de um Glossário que pode ser acessado a qualquer momento do jogo.

1.2. Objetivos Específicos

1. Projetar o processo de desenvolvimento do jogo.
2. Estudar e aplicar os conceitos de *Gamification* no projeto do jogo, utilizando conteúdos de Física.
3. Estudar o motor de jogos *Unity* e as funcionalidades necessárias para implementação do jogo descrito.
4. Desenvolver um protótipo com um personagem andando em um pequeno cenário e interagindo com NPCs.
5. Elaborar diferentes desafios que abordem conteúdos de Física, em um contexto prático que facilite o entendimento do jogador.
6. Desenvolver um glossário com os conteúdos que serão utilizados nos desafios apresentados pelos NPCs.

2. Referencial Teórico

Nesta seção, serão apresentados os conceitos, metodologias e ferramentas que serão utilizados no desenvolvimento deste trabalho.

2.1. Jogos Eletrônicos (Videogames)

Os jogos eletrônicos marcaram a vida de muitas pessoas e hoje são um dos setores que mais cresce na indústria de mídia e entretenimento. Em menos de quatro décadas eles se tornaram uma indústria global de enormes proporções [Wolf 2008].

Um dos principais objetivos deste trabalho é atingir este nível de imersão em um ambiente educativo em que os jogadores se sintam confortáveis e se disponham a absorver o conhecimento teórico que lhes for passado de forma indireta por meio de um jogo educativo. Este tema será abordado na Seção 2.2.

2.2. Jogos na educação

A sociedade tem se tornado mais complexa à medida da que a área tecnológica cresce e que a educação adquire importância fundamental nesta mudança. Os alunos são capazes de demonstrar uma produtividade maior por meio da utilização de jogos e simulações que proporcionam uma integração entre o ambiente de aprendizagem e o mundo real, melhorando assim a qualidade de ensino [Rombaldi e Bilésimo 2015].

Durante vários anos discutiu-se a possibilidade de jogos eletrônicos terem uma influência negativa nos jogadores e se tornarem uma fonte de instigação para a violência em crianças e adolescentes. Porém, o interesse em pesquisar os aspectos positivos dos jogos eletrônicos aumentou nos últimos anos, onde os pesquisadores visam os benefícios para os jogadores junto com o seu potencial como recurso didático e o seu uso na área da educação [Savi e Ulbricht 2008].

Os professores que vem usando jogos em suas salas de aula estão em sua maioria empolgados, já que, ao utilizarem jogos educativos passam a ser mais diretores de pesquisa e treinadores do que disciplinadores, pois poupam tempo e energia, e os utilizam para treinar os estudantes, ou observar suas forças e fraquezas intelectuais [Rombaldi e Bilésimo 2015].

Assim sendo, não basta apenas tentar criar um jogo e esperar por resultados positivos, é necessário que o jogo seja capaz de entreter e ensinar o jogador ao mesmo tempo. Com isso em mente, o jogo desenvolvido neste trabalho fará uso de um conceito chamado *Gamification*.

2.3. Gamification

Gamification, também apresentada como *Gamefication* ou Gamificação, é a utilização de elementos de jogos e técnicas de game design para levar a aprendizagem de um determinado assunto, tendo a diversão com um dos elementos mais importantes do processo. A meta é fazer com que as pessoas queiram investir seu tempo, compartilhar conhecimento e contribuir com a sua energia para o alcance do resultado [Alves 2015].

Um jogo é uma atividade voluntária, fazemos porque queremos fazer, ou seja, espontaneamente. Isto traz à tona o conceito chave da *Gamification* que é a motivação do jogador, pois é a condição do organismo que influencia o comportamento de uma pessoa e está diretamente relacionada ao impulso que leva esta pessoa a tomar uma ação [Alves 2015].

2.4. Motores de jogos

Um motor de jogos, ou *Game Engine*, é um *software* que contém um conjunto de bibliotecas, que visam simplificar o processo de desenvolvimento de jogos eletrônicos e outras aplicações que utilizam gráficos em tempo real [Santos e Caruso 2011].

O motor escolhido para realizar este trabalho é o *Unity* que será apresentado na Seção 2.4.1, bem como a justificativa de sua escolha.

2.4.1. Unity e suas ferramentas

O *Unity* é um motor de jogos que permite construir jogos e aplicativos em 2D e 3D com alta qualidade. Além disso, a ferramenta possui alta tecnologia no que diz respeito a programação, renderização, áudio, terrenos, iluminação, Física, partículas e networking.

O *Unity* suporta vários efeitos diferentes de iluminação e texturas. A ferramenta possibilita a modelagem de ambientes como montanhas, lagos, praias, ilhas e vulcões.

Os jogos desenvolvidos com *Unity* variam desde os mais simples jogos de plataforma 2D até os mais complexos jogos de aventura 3D. O motor permite a produção de jogos para praticamente todas as plataformas disponíveis no mercado, desde

dispositivos móveis (Android, iOS) a consoles de mesa (Playstation, Xbox) e computadores pessoais (Windows, Linux e Mac).

Além de possibilitar o desenvolvimento para diversas plataformas diferentes, o *Unity* possui uma loja virtual (*Unity Asset Store*) onde os usuários podem divulgar suas criações e comercializá-las. As ferramentas disponíveis nesta loja são das mais variadas possíveis, incluindo modelos 3D, animações, cenários, armas, texturas, scripts prontos e muitas outras ferramentas que se mostram muito úteis a qualquer desenvolvedor.

Na versão atual da ferramenta, *Unity 5*, é possível programar em duas linguagens diferentes, sendo elas *C#* (C-Sharp) e *UnityScript* (uma linguagem desenhada para ser usada com o *Unity* e modelada com base no JavaScript).

O *Unity* possui uma versão paga chamado *Unity Pro* que custa US\$ 1.500,00, enquanto que a versão gratuita pode ser usada tanto para fins educacionais quanto para fins comerciais. Uma empresa ou indivíduo é livre para usar a versão gratuita do *Unity* enquanto obtiver um rendimento anual menor que US\$ 100.000,00.

Mas o fator mais relevante para este trabalho é que existe a possibilidade de atribuir Física a todos os objetos, onde é possível a modificar o valor de atributos como: peso, massa e gravidade [Santos e Caruso 2011]. A soma desta característica com as outras apresentadas anteriormente é o bastante para justificar a escolha do *Unity* como a principal ferramenta para ser utilizada no desenvolvimento do jogo.

2.5. Game Design

Os jogos são uma forma de objeto cultural constituído de um conjunto de regras, visuais e narrativa em um ambiente interativo com um participante (*player* ou jogador). *Game design* é o processo de criação das regras e do conteúdo encapsulados em um jogo, além de elaborar os objetivos que devem motivar o jogador. Envolve a descrição dos elementos e componentes que existirão no jogo e a comunicação com os profissionais de cada área do desenvolvimento [Mastrocola 2015].

As diferentes fases do desenvolvimento de jogos serão apresentadas nas seções posteriores. A primeira sendo o processo de *Game Design*, onde são concebidas as ideias de como o jogo de ser. A segunda é a metodologia de desenvolvimento, que diz respeito à implementação do jogo como um software. Por fim estão as ferramentas que serão utilizadas na etapa de desenvolvimento.

2.5.1. Processo de Game Design

Em seu livro *Game Design: Modelos de Negócio e Processos Criativos - Um Trajeto do Prototipo ao Jogo Produzido*, Mastrocola (2015) apresenta algumas das várias metodologias para se aplicar no processo de desenvolvimento de um jogo. Entre elas está o fluxo de trabalho proposto por Co (2006) em seu livro *Level Design for Games: Creating Compelling Game Experiences* o qual foi utilizado no desenvolvimento deste trabalho. Este processo se divide em nove etapas:

1. **Pré-Produção:** Montar uma equipe de profissionais para o de desenvolvimento do jogo e fazer uma junção das ideias a serem implementadas no jogo.
2. **High concept:** Filtrar as ideias da etapa anterior e utilizá-las na elaboração do projeto; estabelecer a direção de arte a ser tomada.

3. **Documentos de Design:** Elaborar os documentos que servirão como base para o desenvolvimento, contendo os conceitos que deverão estar presentes no jogo.
4. **Protótipo/Demo:** Criação de uma versão puramente demonstrativa para poder estabelecer quais ferramentas e tecnologias serão utilizadas no processo de desenvolvimento.
5. **Produção:** Nesta etapa são criados os personagens, os cenários e as principais mecânicas de jogo, visando prepara a estrutura básica jogo.
6. **Alfa:** Esta etapa serve para que os desenvolvedores possam se dedicar a melhorar o conteúdo produzido na etapa anterior e implementar a estética que será mantida até o produto final; também envolve a correção dos problemas encontrados no resultado da etapa anterior.
7. **Beta:** Seu foco está principalmente em corrigir os problemas encontrados na versão anterior.
8. **Versão final:** Nesta etapa o projeto é finalizado após a correção dos problemas mais complexos e persistentes e um polimento final nas mecânicas do jogo.
9. **Gold master:** É onde os envolvidos podem descansar e se preparar para o próximo projeto.

As etapas 1, 2 e 3 englobam o planejamento do jogo, enquanto as outras representam o processo de desenvolvimento de software, que foi realizada utilizando a metodologia ágil FDD.

2.5.2. Metodologias para o desenvolvimento de jogos

Hoje em dia, o desenvolvimento de jogos se depara com requisitos de tempo cada vez menores, ao ponto que muitas vezes é inviável utilizar metodologias clássicas de Engenharia de Software. Metodologias de desenvolvimento ágil surgiram para combater este problema [Kasperavicius et al. 2008].

Em sua maioria, os métodos ágeis tentam diminuir ao máximo o número de riscos do processo de desenvolvimento de software, fazendo uso interações periódicas, de curto intervalo, para reanalisar e incrementar as funcionalidades do projeto [Kasperavicius et al. 2008].

Estes métodos se provaram eficientes em projetos de software com prazos curtos, pequenas equipes e baixa complexidade [Sutherland 2004]. Por estes motivos, a metodologia ágil Feature Driven Development (FDD) foi escolhida para o desenvolvimento do jogo proposto neste trabalho.

2.5.2.1. Feature Driven Development (FDD)

Features Driven Development, ou Desenvolvimento Guiado por Funcionalidades, visa combinar os conceitos de metodologias ágeis com uma abordagem para Engenharia de Software orientada a objetos. Assim, cobre os três principais grupos envolvidos em um projeto de software: os clientes, gerentes e desenvolvedores [FDD 2015].

Esta metodologia foi criada entre no final dos anos 90, em Cingapura, por Jeff De Luca e Peter Coad e inclui alguns benefícios de processos rigorosos, como modelagem, planejamento prévio e controle do projeto, possuindo também características de processos ágeis, dos quais se pode citar: o destaque na área da programação, uma interação constante com o cliente e entregas frequentes de versões atualizadas do produto. A metodologia prevê apenas as práticas para o desenvolvimento de software, assim sendo,

não tem foco em quais serão as tecnologias e ferramentas serão utilizadas na sua implementação [Filho 2012].

De acordo com Barbosa et al. (2006) o FDD se divide em cinco processos:

1. **Desenvolver um modelo geral:** Confeccionar a estrutura do sistema utilizando as funcionalidades e os requisitos pedidos pelo cliente.
2. **Construir por lista de funcionalidades:** Listar as funcionalidades do sistema.
3. **Planejar por funcionalidade:** é criado um planejamento de como cada funcionalidade que será desenvolvida.
4. **Arquitetar (ou projetar) por funcionalidade:** É feita uma análise e um estudo de cada funcionalidade e, por meio disso constrói-se um Diagrama Sequencial ou um Diagrama de Classes, que será utilizado na construção de uma funcionalidade.
5. **Construir por funcionalidade:** Tem como objetivo produzir (ou implementar) as funcionalidades planejadas e arquitetadas nos processos anteriores.

Por terem uma visão geral e inicial de todo projeto, os processos Desenvolver Modelo Inicial, Criar Lista de Funcionalidades e Planejar por Funcionalidade podem sofrer alterações no decorrer do projeto para se manterem consistentes até o final do desenvolvimento do mesmo. Os outros processos são executados de maneira iterativa, uma vez para cada funcionalidade documentada [FDD 2015]. A Figura 1 ilustra a interação dos cinco processos do FDD.



Figura 1. Interação entre processos no FDD [Palmer e Felsin 2002].

2.6. Física aplicada ao desenvolvimento do projeto

Nesta Seção serão apresentados os conceitos de Física que estarão presentes no jogo, por meio do glossário que os disponibilizará a qualquer momento que o jogador desejar. Neste trabalho, serão abordados dois tópicos principais: cinemática e dinâmica.

O tópico de cinemática irá abordar os conteúdos de:

1. Velocidade média
2. Movimento retilíneo uniforme
3. Movimento circular

O tópico de dinâmica irá abordar os conteúdos de :

1. Leis de Newton
2. Força peso
3. Força de atrito
4. Força elástica

Os tópicos acima foram escolhidos pois eles podem ser facilmente encontrados em situações do mundo real, propiciando aos alunos um melhor entendimento dos mesmos por meio das suas relações com os eventos do dia a dia.

É importante ressaltar que nem todos estes conteúdos estarão necessariamente presentes nos desafios que o jogo irá oferecer, mas serão mantidos no glossário para que o jogador possa estudá-los mesmo que o jogo não exija.

3. Trabalhos Correlatos

Nesta seção serão apresentados alguns trabalhos que serviram de auxílio no desenvolvimento do jogo por abordarem temas semelhantes e por fazerem uso do motor de jogos *Unity*.

3.1. Vlad the Platformer

No trabalho de Brasil e Prass (2015), foi desenvolvido o jogo *Vlad the Platformer*, um jogo de plataforma 2D utilizando o motor de jogos *Unity*, com o objetivo de realizar um estudo das funcionalidades necessárias do *Unity* para o desenvolvimento do jogo, bem como desenvolver um protótipo do jogo composto de no mínimo uma fase.

A metodologia de desenvolvimento escolhida para este trabalho foi uma adaptação do Scrum para um trabalho individual, visto que o Scrum é voltado para o desenvolvimento por equipes.

O jogo *Vlad the Platformer* não possui muitas semelhanças com o jogo proposto neste trabalho, entretanto serviu como referência para a utilização do *Unity*, além de enfatizar que o entretenimento de jogador é o objetivo principal de um jogo.

3.2. Little Newton

Neste trabalho de Lyon et al. (2015), foi desenvolvido o jogo *Little Newton*, um jogo de defesa 3D utilizando o *Unity*, com o objetivo de ensinar o jogador sobre os conceitos básicos de lançamento o Movimento Oblíquo por meio do controle dos atributos físicos do arremesso de um projétil.

O jogo foi criado com a ideia de fazer com que o jogador aprenda sobre a influência desses atributos em tempo real para poder avançar para a próxima fase.

O jogo *Little Newton* aplica de conceitos físicos em um cenário desafiador que estimula o aprendizado do jogador, tais métodos serviram de referência na elaboração das atividades que o jogador exercerá no jogo proposto nesse trabalho.

3.3. EducaCorpoHumano3D

No trabalho de Rombaldi e Bilésimo (2015), foi desenvolvido o jogo *EducaCorpoHumano3D*, um jogo de perguntas e respostas utilizando o motor de jogos *Unity*, com o objetivo de abordar o estudo da área da anatomia para demonstrar que é possível inserir tais tecnologias em todas as outras áreas educacionais. A metodologia de Aprendizagem Baseada em Jogo Digital (DBGL) foi utilizada para desenvolver o projeto.

As descrições das técnicas e conceitos utilizados no desenvolvimento do jogo *EducaCorpoHumano3D* servem de referência para a utilização do motor gráfico *Unity*.

4. Metodologia

Nesta seção, serão descritos a proposta do jogo a ser desenvolvido e o projeto do mesmo, aplicando as etapas da metodologia FDD, comentando sobre a sua implementação e os recursos utilizados na construção do jogo.

4.1. Proposta

O objetivo deste trabalho é utilizar o motor gráfico *Unity* para desenvolver um jogo em mundo aberto (uma pequena cidade) onde um dos dois personagens principais é escolhido para ser controlado pelo jogador, e é capaz de andar livremente e interagir com NPCs que apresentam desafios na forma de mini games educativos, os quais abordam um determinado assunto dos conteúdos de Física ensinados no ensino médio e fundamental, de uma forma que permita os alunos visualizarem estes conceitos sendo aplicados em situações do mundo real.

O jogador também terá a liberdade de trocar de personagem principal em qualquer momento do jogo, desde que não esteja resolvendo um desafio. O jogador também tem o direito de acessar um glossário a qualquer instante do jogo para poder pesquisar sobre um assunto de Física que lhe interesse, novamente, desde que não esteja no meio de um desafio.

O jogo apresenta uma história simples, ao ponto que não será necessário fazer uso de qualquer estrutura de enredo específica, pois ela apenas serve como justificativa para o que o jogador irá presenciar, sendo ela: “Dois amigos acabaram de se mudar para uma cidade nova que não conhecem e decidem explorá-la, em sua jornada encontrarão várias pessoas que necessitam de ajuda. Para ajudá-las será necessário resolver diversos desafios utilizando conhecimento de Física”. Esta narrativa é apresentada no início do jogo, quando o jogador selecionar o personagem que ele deseja jogar.

Para o áudio do jogo, será utilizada uma coleção de bibliotecas de músicas e efeitos sonoros que estão disponibilizadas gratuitamente na loja virtual do *Unity*.

É necessário frisar que o jogo proposto neste trabalho tem como objetivo ser um auxílio aos educadores ao invés de um substituto. A ideia principal é apresentar os conteúdos contidos no jogo de forma que facilite o entendimento dos mesmos quando forem apresentados em salas de aula.

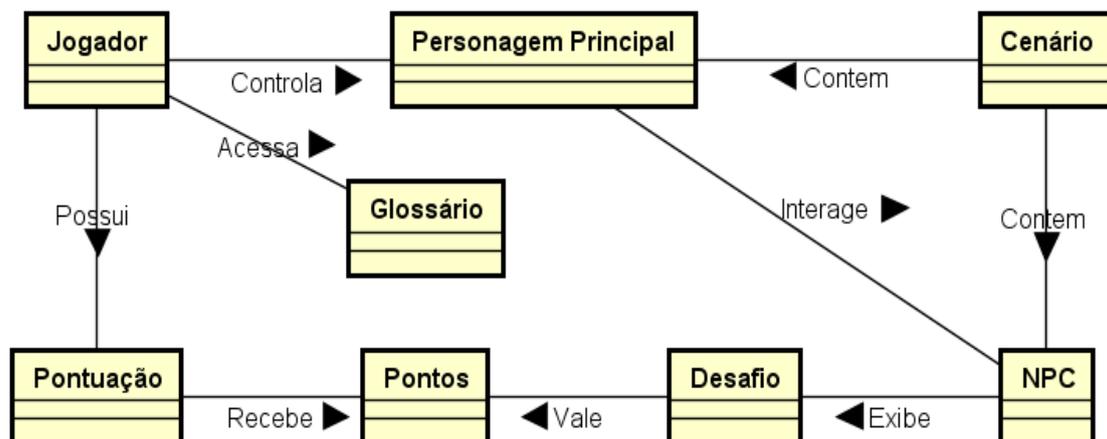
4.2. Projeto do jogo

Nesta seção estarão presentes as etapas da metodologia FDD sendo usadas para planejar a implementação do jogo.

4.2.1. Desenvolver Modelo Inicial

A Figura 2 apresenta o Diagrama de Domínio do jogo, onde o jogador controla uma personagem principal que pode interagir com um NPC, estando ambos contidos em um cenário 3D.

Ao interagir com o personagem principal, o NPC exibe um desafio para o jogador. Este desafio tem um determinado valor de pontos que é acrescentado a pontuação do jogador, caso ele consiga resolver o desafio. Além disso, o jogador também tem a opção de acessar um glossário que lhe permite estudar os conteúdos de Física de sua escolha.



powered by Astah

Figura 2. Diagrama de Domínio.

4.2.2. Criar Lista de Funcionalidades

Uma Lista de Funcionalidades pode ser representada por um conjunto de Requisitos Funcionais e Requisitos Não Funcionais que descrevem as verdadeiras necessidades do negócio do ponto de vista do cliente [FDD 2015].

Além dos Requisitos Funcionais e Requisitos Não Funcionais, também é apresentado o Diagrama de Caso de Uso, por representar as funcionalidades do sistema que podem ser observadas externamente, junto com os elementos externos ao sistema que interagem com ele.

Os Requisitos Funcionais e Requisitos Não Funcionais podem ser encontrados no Apêndice A, seguidos do Diagrama de Caso de Uso e seus Descritivos no Apêndice B.

Também foi modelado um Diagrama de Atividades, descrito na Seção 4.2.2.1, que apresenta o fluxo das atividades e quais delas são exercidas pelo Sistema e quais são exercidas pelo jogador.

4.2.2.1. Diagrama de Atividade

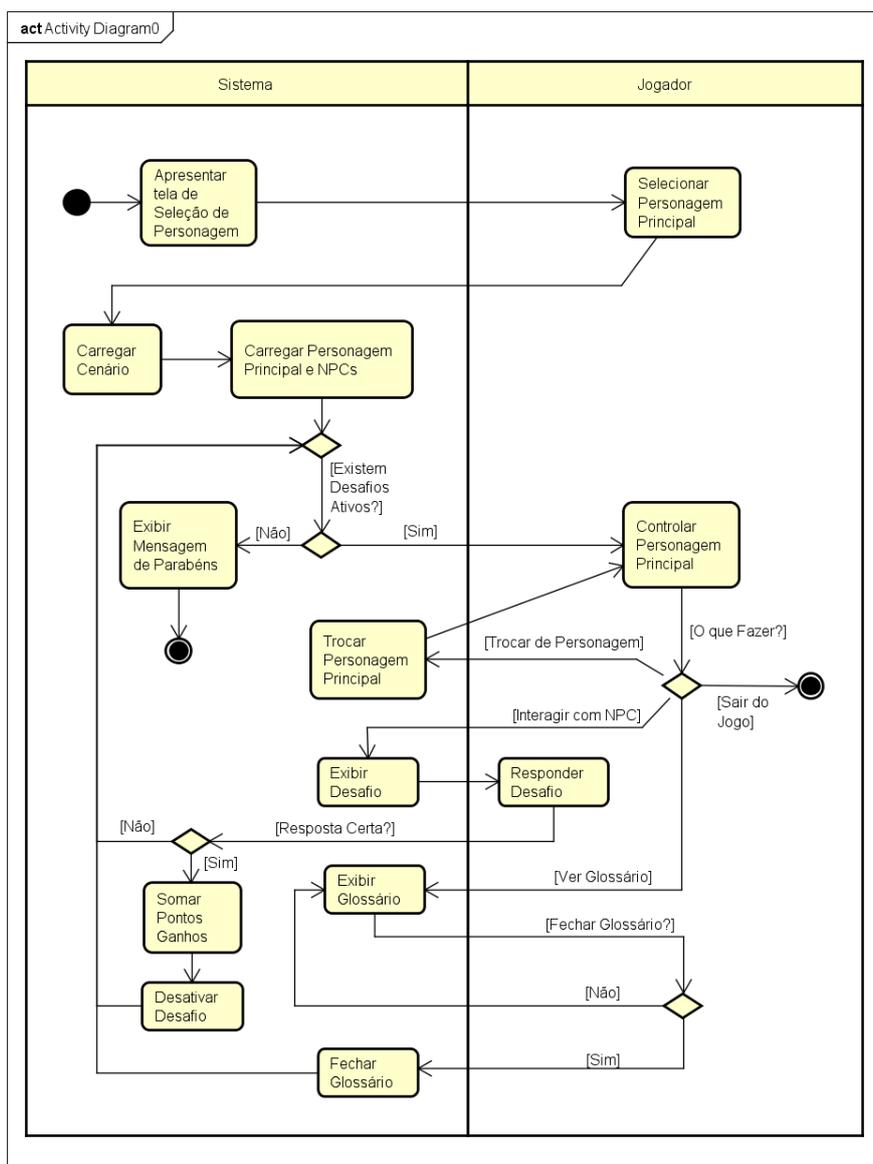
A Figura 3 representa o Diagrama de Atividade, demonstrando a relação entre o jogador e o sistema do jogo. Inicialmente o sistema apresenta uma tela onde o jogador deve escolher com qual dos personagens principais ele deseja jogar, e após isso, o sistema carrega o cenário do jogo e os personagens dele, incluindo o personagem principal e os NPCs.

A cada nova interação do jogador, o sistema irá verificar se existem desafios ativos. Se não existirem, o jogo irá apresentar uma mensagem de parabéns ao jogador por ter resolvido todos os desafios e irá se encerrar. Se existirem, o jogador assume o controle do personagem escolhido e se depara com as seguintes opções: “Trocar de Personagem”, “Interagir com um NPC”, “Ver Glossário” e “Sair do Jogo”.

No caso de o Jogador escolher “Trocar de Personagem”, o sistema do jogo irá substituir o personagem principal escolhido no início pelo outro. Porém, se o jogador escolher sair do jogo, o sistema irá se encerrar.

Caso o jogador decida “Interagir com um NPC”, o mesmo irá lhe apresentar um desafio para ser solucionado. Se o jogador conseguir resolver o desafio, o sistema irá somar os pontos daquele desafio e desativa-lo. Se o jogador não conseguir resolver o desafio ele não receberá nenhum ponto e o desafio continuará lá para o jogador tentar novamente.

Entretanto, se a opção escolhida for “Ver Glossário”, o sistema irá exibir um glossário que lhe apresentará os conteúdos de Física que estão englobados no jogo, até que o jogador decida fechá-lo. Após escolha ser feita, caso o jogador não tenha escolhido “Sair do Jogo”, o sistema irá retornar ao ponto de verificar se ainda existem desafios ativos no sistema.



powered by Astah

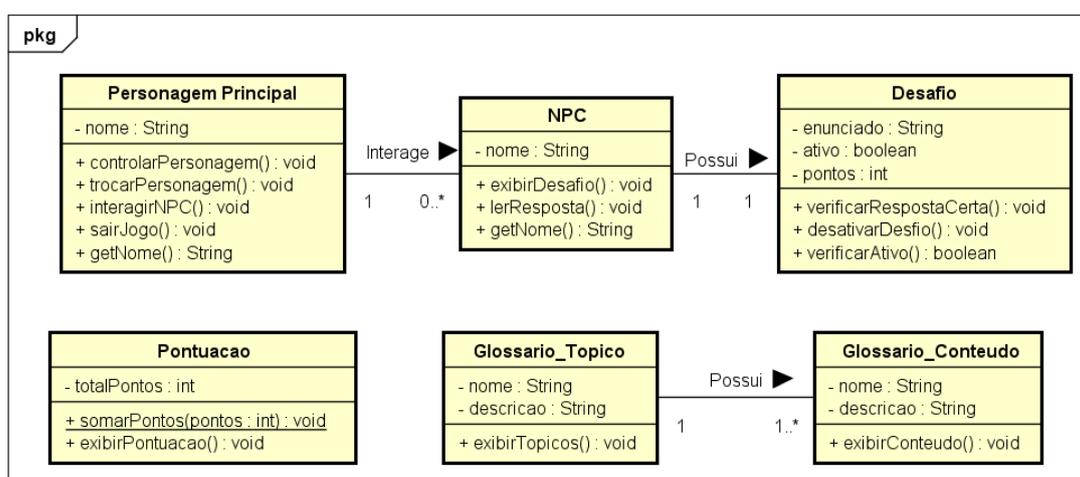
Figura 3. Diagrama de Atividade

4.2.3. Planejar por Funcionalidade

Nesta etapa, foi desenvolvida uma tabela com a ordem em que cada funcionalidade do jogo foi desenvolvida e, a quantidade de tempo necessária para desenvolvê-la, de forma que a construção de cada funcionalidade tivesse tempo o bastante dentro do período de desenvolvimento proposto. Esta tabela está disponível no Apêndice C.

4.2.4. Arquitetar por Funcionalidade

A Figura 4 apresenta o Diagrama de Classes do jogo, que permite a visualização das classes, das quais o sistema do jogo é composto. Nestas classes, também é possível observar os seus respectivos métodos e atributos, além de representar o relacionamento entre as mesmas.



powered by Astah

Figura 4. Diagrama de Classes

4.2.5. Construir por Funcionalidade

Nesta seção, serão citados os recursos utilizados no processo de implementação do jogo e as tarefas nas quais esses recursos foram utilizados.

4.2.5.1. Recursos Utilizados no Desenvolvimento

Para a construção do jogo, foram utilizados vários recursos diferentes. Dentre eles pode-se citar:

Programa Mixamo Fuse: este programa gratuito foi utilizado para modelar os personagens que estão presentes no jogo [Mixamo 2016].

Animações dos Personagens: os personagens modelados no *Fuse* foram animados através do site da empresa Mixamo que possui uma biblioteca de animações gratuitas.

Fungus: é um *Plug-in* para o *Unity*, desenvolvido pela *Fungus Games*, e foi utilizado para desenvolver o sistema de interação com os NPCs.

Outros recursos utilizados no desenvolvimento são apresentados no Apêndice D.

4.2.5.2. Atividades Realizadas no Desenvolvimento

Nessa seção serão apresentadas as atividades que foram realizados no processo de desenvolvimento do jogo.

Foi iniciada a modelagem dos personagens do jogo (Figura 5), desde sua fisionomia até as roupas que eles estarão utilizando, através do programa *Mixamo Fuse*. Após os personagens estarem prontos, foi preciso pesquisar e baixar as animações para eles através do site da empresa Mixamo.



Figura 5. Os dois personagens principais do jogo.

Foi necessário ajustar certas animações para que elas funcionassem corretamente no *Unity*, configurando a transição entre elas para o momento que o jogador estivesse controlando os personagens.

Os controles dos personagens são *scripts* escritos em C# no *Visual Studio Community 2015* e são responsáveis pela movimentação e animação dos personagens conforme os comandos do jogador.

Outros *scripts* também foram utilizados para controlar as câmeras do jogo, tanto a principal quanto a do mini mapa (Figura 6), para que elas sigam os personagens principais ao se moverem e girarem ao seu redor conforme a movimentação do *mouse*.



Figura 6. Mini mapa do jogo.

Após estudar os componentes de Interface de Usuário do Unity, esses foram utilizados na construção do Menu de Pausa (Figura 7) e do Glossário do Jogo (Figura 8). No Menu de Pausa, foi implementada a opção de o jogador trocar de personagem.



Figura 7. Menu de Pausa do jogo.

Outra opção que foi adicionada ao Menu de Pausa é a de acessar o Glossário do jogo, onde os conteúdos inclusos nele estão separados em diferentes páginas, permitindo a navegação entre elas.



Figura 8. Telas do Glossário. Direita: Tela Inicial, Esquerda: Tela de Conteúdo.

Cada página do Glossário possui um sistema de navegação que permite voltar a página inicial, avançar para o próximo conteúdo ou regressar para o conteúdo anterior. Essa navegação foi construída através de botões localizados no cabeçalho e no rodapé de cada página, como pode ser visto na Figura 8 onde um botão vermelho indica que é possível acessar a próxima página do Glossário.

Para criar o sistema de interação com os NPCs foi necessário utilizar o *Plug-in Fungus*, apresentado na Figura 9, com o qual foi possível criar os diálogos presentes no jogo e exibir os desafios ao jogador junto com as diversas alternativas de resposta e a soma de pontos caso a alternativa selecionada seja a correta.



Figura 9. Personagem Principal interagindo com o NPC Cientista por meio do Fungus.

O cenário principal do jogo foi construído em um projeto separado e foi posteriormente incorporado ao projeto principal do jogo, gerando assim uma paisagem de uma cidade localizada em uma pequena ilha no meio do mar, como é apresentado na Figura 10.



Figura 10. Cenário principal do jogo.

Por fim, foi criado um cenário que simula a Lua (Figura 11), que fica disponível para ser acessado após o jogador resolver o desafio que um NPC específico lhe apresentar. Nele o jogador pode visualizar como a diferença de gravidade faz com que o pulo do personagem principal seja muito mais alto que na Terra.



Figura 11. Cenário da Lua.

5. Conclusões

Este trabalho apresentou o projeto e as etapas de desenvolvimento de um jogo educativo com temática de conteúdos de Física, com o intuito de auxiliar os educadores em salas de aula, incorporando o conceito de *Gamification* e utilizando o motor de jogos Unity.

Neste trabalho foi abordado sobre como os jogos são uma fonte de entretenimento com uma presença forte e crescente no cenário mundial, além de como a aplicação do conceito de *Gamification* pode trazer resultados positivos a cenários normalmente desinteressantes.

No decorrer do desenvolvimento do jogo, houve um grande foco na criação do sistema de interação com os NPCs e controle dos personagens principais. Apesar da maioria dos recursos utilizados já existirem e não terem sido criados do zero, foi possível notar a complexidade de fazer com que esses recursos trabalhem em conjunto para moldar uma experiência sólida do início ao fim.

A utilização e interação desses recursos é o que diferencia o desenvolvimento de jogos do desenvolvimento de um software convencional, uma vez que é necessário conhecimento de diversos conceitos diferentes (animação, modelagem 3D, design de layouts, narrativas, construção de cenários, programação, computação gráfica e outros) para planejar e construir a melhor experiência possível para conquistar o interesse do jogador.

5.1. Trabalhos Futuros

Para trabalhos futuros pode-se aumentar o número de NPCs com que o jogador pode interagir, ou acrescentar mini games interativos durante os desafios para tornar a aprendizagem mais dinâmica. Também é possível otimizar alguns recursos do jogo como resolução de texturas e o número de polígonos de modelos 3D para que o jogo execute em máquinas de menor desempenho.

Outra possibilidade é a de incorporar um sistema de *Multiplayer* em rede, para que mais de um jogador possa jogar o jogo ao mesmo tempo e interagir um com o outro aumentando assim a imersão e possivelmente o nível de aprendizagem.

Referências

- Alves, F. (2015) *Gamification - Como criar experiências de aprendizagem engajadoras. Um guia completo: do conceito à prática*, 2ª Edição, DVS Editora.
- Barbosa, A., Azevedo, B., Pereira, B., Campos, P. e Santos, P. (2006) "Metodologia Ágil: Feature Driven Development", http://paginas.fe.up.pt/~aaguiar/es/artigos%20finais/es_final_22.pdf, Abril.
- Barroso, F. e Antunes, M. (2015) "Tecnologia na educação: ferramentas digitais facilitadoras da prática docente", *Revista do Programa de Pós-Graduação Profissional em Gestão e Avaliação da Educação Pública*.
- Brasil, M. R. e Prass, F. S. (2015) "Desenvolvimento de Jogo Platformer em Unity: Vlad the Platformer", Trabalho Final de Graduação, Centro Universitário Franciscano.

- Co, P. (2006) Level Design for Games. Creating Compelling Game Experiences.
- FDD. Features Driven Development site oficial (2015). Disponível em: <http://www.featuredrivendevelopment.com/>, Maio.
- Filho, Nivaldo T. (2012) “Estudo do Impacto do Uso das Metodologias Ágeis na Melhoria do Planejamento e Acompanhamento do Processo de Ensino e Aprendizagem em Sala de Aula”, Fortaleza: Universidade Estadual do Ceará.
- Kasperavicius, L., Bezerra, L., Silva, L. e Silveira, I. (2008) “Ensino de Desenvolvimento de Jogos Digitais Baseado em Metodologias Ágeis: o Projeto Primeira Habilitação”, Anais do XXVIII Congresso da SBC.
- Lyon, N., Shao, N., Valls, J., Zhu, J., Guevara, C. e Zhu, J. (2014) “Little Newton: An Educational Physics Game”, Publicado no simpósio anual CHI PLAY em New York.
- Mastrocola, V. (2015) “Game Design: Modelos de Negócio e Processos Criativos - Um Trajeto do Protótipo ao Jogo Produzido”, 2ª Edição, Editora Cengage Learning.
- Mixamo (2016). Disponível em: <https://www.mixamo.com>, Maio.
- Palmer, Stephen R.; Felsin, John. M. (2002) “A Practical Guide to Feature-Driven Development.” Prentice Hall, Maio.
- Rombaldi, B. e Bilésimo, P. (2015) “Desenvolvimento de Um Jogo Sério Utilizando Unity 3d Para o Estudo de Anatomia no Ensino Fundamental”, Trabalho Final de Graduação submetido à Universidade Federal de Santa Catarina.
- Santos, M. e Caruso, A. (2011) “Estudo E Desenvolvimento De Jogos Para Internet Utilizando Unity 3D”, Monografia apresentada ao Curso de Tecnologia em Sistemas para Internet do Instituto Federal Sul-Rio-Grandense.
- Savi, R. e Ulbricht, V (2008) “Jogos Digitais Educacionais: Benefícios E Desafios” Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento / UFSC.
- Sutherland, J. (2004) Agile Development: Lessons Learned From the First Scrum. Cutter Agile Project Management Advisory Service – Executive Update, v. 5, n. 20.
- Tarouco, L., Roland, L., Fabre, M. e Konrath, M. (2004) “Jogos educacionais” Revista Novas Tecnologias na Educação CINTED-UFRGS.
- Unity Technologies. (2015) “Unity – Game Engine”, <http://unity3d.com/>.
- Wolf, M. (2008) The Video Game Explosion: A History From Pong To Playstation And Beyond.

Apêndice A

Requisitos Funcionais

RF1 – Selecionar Personagem Principal.

RF2 – Mostrar Cenário.

RF3 – Mostrar Personagens.

RF3.1 – Mostrar Personagem Principal.

RF3.2 – Mostrar NPCs.

RF4 – Controlar Personagem Principal.

RF4.1 – Fazer o Personagem Principal Andar (Frente, Trás, Esquerda e Direita).

RF4.2 – Fazer o Personagem Principal Pular.

RF4.3 – Fazer o Personagem Principal interagir com os NPCs.

RF5 – Trocar de Personagem.

RF6 – Interação com NPC.

RF6.1 – Exibir Desafio.

RF6.2 – Ler Resposta.

RF6.3 – Somar Pontos.

RF7 – Exibir Glossário

RF8 – Exibir Pontuação.

Requisitos Não Funcionais

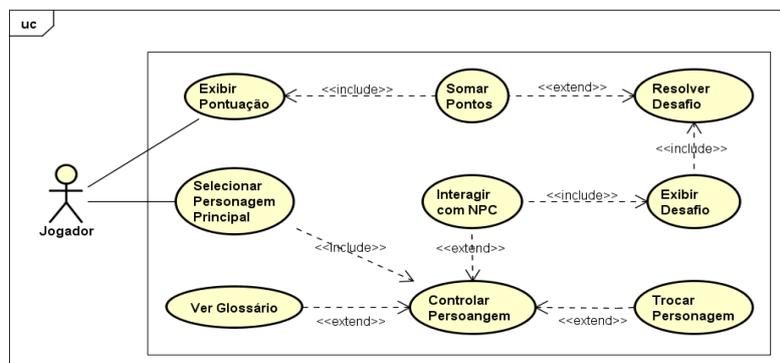
RNF1 – Desenvolver o jogo no *Unity*.

RNF2 – Utilizar o *Visual Studio Community 2015*.

RNF3 – Modelar os Personagens no programa *Mixamo Fuse*.

Apêndice B

Diagrama de Caso de Uso



powered by Astah

Figura 1. Diagrama de Caso de Uso.

Para um melhor entendimento do Diagrama de Caso de Uso, foram elaborados os descritivos de dois casos de uso, sendo eles: Controlar Personagem e Ver Glossário. Esses casos de uso foram escolhidos por representarem funções essenciais para o funcionamento do jogo e para cumprir o seu objetivo que é ensinar conteúdos de Física.

A seguir, é apresentado o Descritivo de Caso de Uso De Controlar Personagem (Tabela 1), que está explicando de forma detalhada como deve funcionar a parte do sistema do jogo que é responsável pelo controle dos personagens principais.

Tabela 1. Descritivo de Controlar Personagem

Identificação	UC002
Caso de Uso	Controlar Personagem
Descrição	Este Caso de Uso descreve o funcionamento do sistema de controle do personagem principal.
Atores Principais	Jogador
Atores Secundários	N/A
Pré-Condições	O jogador deve ter escolhido um dos dois personagens principais para começara a jogar no UC001.
Pós-Condições	O jogador optou por trocar de personagem principal, visualizar o glossário do jogo ou uma interação com um NPC foi iniciada.
Fluxo Principal	P01. Seguir o UC005 para Interagir com NPC, Trocar de Personagem [A01] ou Ver Glossário [A02]. P02. Fim do Caso de Uso
Fluxo Alternativo	A01. Trocar De Personagem Principal
	1. Segue para o UC003 para Trocar de Personagem.
	2. O sistema redireciona para P01.
	A02. Ver Glossário
	1. Segue para o UC004 para Ver o Glossário do Jogo.
	2. O sistema redireciona para P01
Fluxo de Exceções	N/A
Regras de Negócio	RN01. O jogo deve permitir que o jogador faça o personagem principal andar para frente, para traz, para a direita e para a esquerda, por meio de entradas de dados pelo teclado. RN02. O jogo de permitir que o jogador faça o personagem principal pular, por meio de entradas de dados pelo teclado. RN03. O jogo deve permitir que o jogador controle o ângulo com da câmara do jogo, por meio da movimentação do <i>mouse</i> . RN 04. O jogo deve permitir que o jogador inicie uma interação com um NPC, por meio de entradas de dados pelo teclado.

Casos de Uso ou Cenários Incluídos	N/A
Pontos de Extensão	PE01. Interagir com NPC para poder resolver o desafio que ele propõe. PE02. Trocar o Personagem principal escolhido pelo outro. PE03. Ver Glossário para que o jogador possa estudar os conteúdos e se prepara para os desafios seguintes.
Observações	N/A

Também é apresentado o Descritivo De Caso De Uso De Ver Glossário (Tabela 2), que elucida, em detalhes, o funcionamento do sistema do Glossário do jogo, de forma que fique clara a ordem de seleção e exibição da lista de tópicos, da lista de conteúdos e suas respectivas descrições.

Tabela 2. Descritivo de Ver Glossário

Identificação	UC004
Caso de Uso	Ver Glossário
Descrição	Este Caso de Uso descreve o funcionamento do sistema de glossário do jogo.
Atores Principais	Jogador
Atores Secundários	N/A
Pré-Condições	O jogador deve ter assumido o controle de um dos personagens principais.
Pós-Condições	O jogador voltou a ter o controle do personagem principal.
Fluxo Principal	P01. Seguir o UC002 para voltar a Controlar o Personagem Principal escolhido. P02. Fim do Caso de Uso.
Fluxo Alternativo	N/A
Fluxo de Exceções	N/A
Regras de Negócio	RN01. O jogo deve listar os tópicos para que o jogador escolha o que quer estudar. RN02. Ao selecionar um tópico, o jogo deve exibir os conteúdos que estão englobados no mesmo. RN03. Ao selecionar um conteúdo, o jogo deve apresentar uma descrição do mesmo, de forma que o jogador possa visualizá-lo e estudá-lo.
Casos de Uso ou Cenários Incluídos	N/A
Pontos de Extensão	N/A

Observações	N/A
-------------	-----

Apêndice C

Tabela 3. Planejamento por Funcionalidade

Ordem	Funcionalidade	Dias para desenvolver
1	RF1 – Selecionar Personagem Principal.	5
2	RF2 – Mostrar Cenário.	20
3	RF3.1 – Mostrar Personagem Principal.	5
4	RF3.1 – Mostrar NPCs.	10
5	RF4.1 – Fazer o Personagem Principal Andar.	10
6	RF4.2 – Fazer o Personagem Principal Pular.	5
7	RF4.3 – Fazer o Personagem Principal interagir com os NPCs.	10
8	RF5 – Trocar de Personagem.	15
9	RF6.1 – Exibir Pergunta.	20
10	RF6.2 – Ler Resposta.	10
11	RF6.3 – Somar Pontos.	20
12	RF7 – Exibir Glossário	10
13	RF8 – Exibir desempenho.	10
Total		150

Apêndice D

Lista de recursos utilizados na construção do jogo:

1. **Programa Mixamo Fuse:** desenvolvido pela empresa Mixamo.
2. **Animações Mixamo:** do site “<https://www.mixamo.com/>”.
3. **Fungus:** desenvolvido pela Fungus Games.
4. **Moon Landscape:** modelado por Black Sun.
5. **Earth Planet:** modelado por Digital Ruby (Jeff Johnson).
6. **Game Ready Boats:** modelado por Gabro Media.
7. **10 Skyboxes Pack Day - Night:** desenvolvido por Wello Soft.
8. **AQUAS Water LITE:** desenvolvido por Dogmatic.
9. **Low Poly Starship SS01:** modelado por Art Force.
10. **Unique Statue Model:** modelado por CJ Bou.
11. **Tents:** modelado por VIS Games.