

# Notwen - Protótipo de jogo em realidade virtual sobre as leis de Newton

Felipe L. Arquino, André C. da Silva

Grupo de Pesquisa Mobilidade e Novas Tecnologias de Interação  
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas  
Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP)  
Campus Hortolândia – SP – Brasil

felipe.arquino@aluno.ifsp.edu.br, andre.constantino@ifsp.edu.br

**Abstract.** *This article shows the process of prototyping a Virtual Reality game, approaching the creation of characters, game mechanics, as well as the coding of the gameplay. The prototype proposes a new form of experimentation to aid in the teaching of Newton's three laws. The development process followed Fullerton's methodology, using the steps of brainstorming, physical prototype and digital prototype, reaching a prototype that brings an experience of experimentation with Newton's laws in an alternative and complementary way to that presented in the classroom.*

**Resumo.** *Este artigo mostra o processo de prototipação de um jogo em Realidade Virtual, abordando a criação dos personagens, mecânicas de jogo, bem como a codificação da jogabilidade. O protótipo propõe uma nova forma de experimentação para auxílio no ensino das três leis de Newton. O processo de desenvolvimento seguiu metodologia de Fullerton, usando os passos de brainstorming, protótipo físico e protótipo digital, alcançando um protótipo que possibilita a experimentação com as leis de Newton de uma forma alternativa e complementar à apresentada em sala de aula.*

## 1. Introdução

Refletindo sobre o motivo dos seres humanos gostarem de jogar, Huizinga (2001, p. 15-16) descreve que o jogo envolve experiências e significados que permanecem com o indivíduo por sua vez as carrega de volta para o cotidiano. Segundo Mastrocola (2012), “quando o indivíduo entra no círculo mágico e experimenta o que há lá dentro em termos de diversão, sonhos, narrativa, catarse, etc. carrega de volta consigo uma carga de experiências e significados”. Os jogos com propósitos educacionais ou de treinamento são chamados de jogos sérios (MICHAEL; CHEN, 2006). Savi e Ulbricht (2008) apontam diversos benefícios para o uso de jogos na educação, sendo alguns deles o efeito motivador (envolve diversão, entretenimento e incentivo ao aprendizado), facilitam o aprendizado em vários campos de conhecimento e desenvolvem habilidades cognitivas e coordenação motora, estimulam o aprendizado por descoberta e, possivelmente, colaboram e também socializam os conhecimentos, pois aproximam os alunos jogadores, competitivamente ou cooperativamente, dentro do mundo virtual ou no próprio ambiente físico. Já no âmbito da Realidade Virtual, Bricken (1993) diz que ela pode “Apoiar e motivar o aprendizado em excursões, fornecendo cenários virtuais semelhantes aos reais acrescidos de informações complementares”.

Neste trabalho será abordado o processo de desenvolvimento de um protótipo de jogo eletrônico sério de física em Realidade Virtual, mais especificamente sobre as três leis de Newton. A escolha dessa temática se dá ao fato de ser uma matéria de melhor representação gráfica e o desenvolvimento se aprofundará na criação artística, *gameplay* e na programação em si das mecânicas do jogo. O jogo apresentará um pequeno enredo sobre um extraterrestre

que caiu em um planeta desconhecido com leis físicas diferentes de sua terra natal. A nave em que veio foi despedaçada e sua missão é passar por ruínas desse planeta e procurar pelas peças da nave enquanto descobre sobre as leis físicas do local.

## 2. Fundamentação Teórica

Antes de apresentar o processo de desenvolvimento de um jogo, é fundamental conhecermos a definição de o que é um jogo, algumas características e como desenvolver um jogo, abrangendo documentações e ferramentas.

Segundo Adams & Rollings (2007), jogo é um tipo de atividade real onde os participantes (jogadores) tentam alcançar um objetivo não trivial agindo de acordo com as regras estabelecidas. De acordo com Crawford (1982), existem quatro elementos fundamentais em comum entre jogos:

- Representação: um jogo é um sistema fechado que representa uma porção da realidade, ou seja, uma estrutura completa e auto-suficiente.
- Interação: um jogo é uma forma de representar a realidade de maneira interativa, podendo provocar alterações e verificar suas consequências.
- Conflito: o conflito aparece naturalmente a partir da interação. O jogador está a todo momento tentando alcançar um objetivo e o jogo impõe obstáculos para que adicione um grau de dificuldade para chegar ao objetivo.
- Segurança: conflito traz risco de dano físico e isso é algo indesejável, mas jogo é uma ferramenta que provém experiências psicológicas de conflitos e perigos sem a realização física.

Mas nem todo jogo pode ser útil para ser usado em uma sala de aula, segundo Moratori (2003):

“Um jogo, para ser útil no processo educacional, deve promover situações interessantes e desafiadoras para a resolução de problemas, permitindo aos aprendizes uma auto-avaliação quanto aos seus desempenhos, além de fazer com que todos os jogadores participem ativamente de todas as etapas.”  
(Moratori, 2003)

Ainda sobre jogos educacionais, existem alguns desafios no desenvolvimento desse tipo de projeto. Valente, Freire e Arantes (2018) nos mostram alguns requisitos que o jogo educacional deve alcançar:

- Considerar os aspectos diversão e educação juntos, não focando somente em apenas um e mantendo o equilíbrio;
- Os elementos do jogo como a mecânica, a narrativa e a estética devem ser permeados pelos objetivos educacionais;
- Manter o equilíbrio entre os desafios e habilidades. A dificuldade do jogo deve aumentar gradativamente.

Com isso, precisamos entender também os passos para se desenvolver um jogo. Fullerton (2008, p. 15 e 16) defende sete passos:

1. *Brainstorming*;
2. Protótipo físico, podendo ser realizado a caneta e papel, ou algum outro material para apresentar as mecânicas de jogo;
3. Apresentação (Opcional);
4. Protótipo digital, que são modelos crus da mecânica de jogo;
5. Documentação de design, que é usado para determinar elementos como público-alvo, história e objetivos do jogo;
6. Produção do projeto;
7. QA, realizando testes de jogabilidade.

Precisamos conhecer também a definição de Realidade Virtual, pois ela abrange muito mais do que se acredita. Segundo Kirner e Siscoutto (2007) “A Realidade Virtual (RV) é uma “interface avançada do usuário” para acessar aplicações executadas no computador, propiciando a visualização, movimentação e interação do usuário, em tempo real, em ambientes tridimensionais gerados por computador”. Diante deste contexto, um cenário em RV não é apenas o que visualizamos através de *Headsets* de Realidade Virtual, mas sim o que é transmitido por qualquer monitor. Segundo Kirner, Tori e Siscoutto (2007), temos dois tipos de Realidade Virtual:

- **Imersiva:** Usa dispositivos multissensoriais, que capturam seus movimentos e comportamento e reagem a eles, provocando uma sensação de presença dentro do mundo virtual;
- **Não Imersiva:** Observa o mundo virtual através de uma janela (monitor ou projeção, por exemplo), mas continua a sentir-se predominantemente no mundo real.

### 3. Trabalhos correlatos

Estes são alguns jogos educacionais que também usufruem da tecnologia da Realidade Virtual imersiva:

#### 3.1. Nanome (Vive/Rift/WMR)

Produzido pela *Nanome Inc*, *Nanome* permite o jogador a interagir com inúmeros tipos de elementos químicos, dando a oportunidade de ver de perto moléculas de uma forma nunca antes vista (Figura 1). Neste é possível visualizar, construir e simular compostos químicos, proteínas e ácidos nucleicos, dando ainda a opção de conectar com outro usuário e colaborar remotamente em tempo real.



**Figura 1 – Usuário interagindo no jogo Nanome**

### 3.2. Hold The World (Rift/WMR)

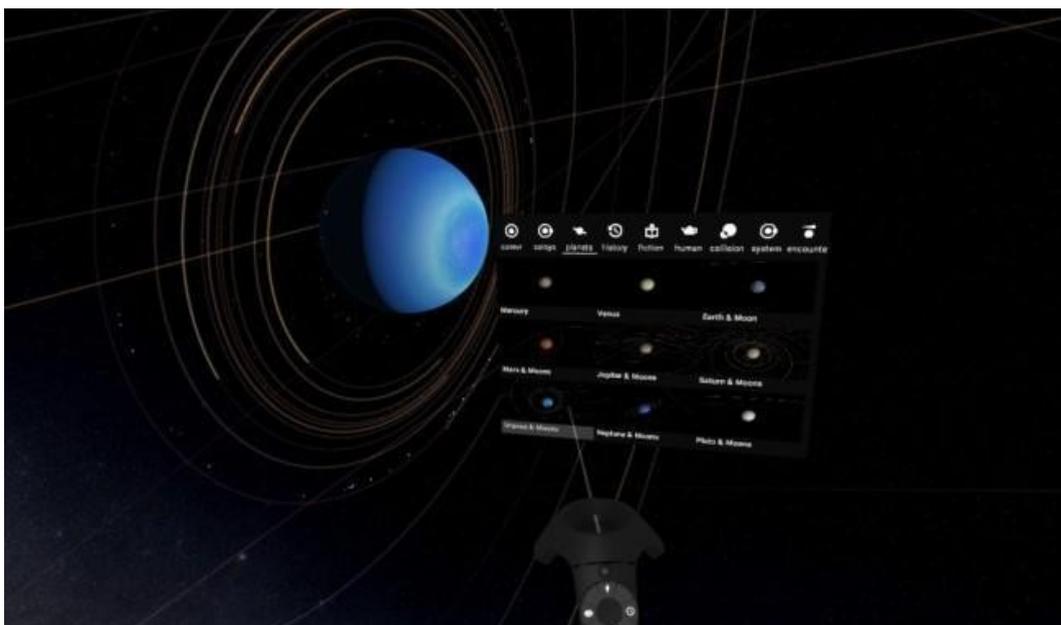
Produzido por *Sky*, *Hold The World* é um museu virtual que foca na interação apresentando objetos selecionados e lhe mostrando a história por trás deles de uma forma apenas possível pela realidade virtual (Figura 2).



**Figura 2 – Interação com esqueleto de baleia azul virtual do aplicativo *Hold The World*.**

### 3.3. Universe Sandbox (Vive/Rift/WMR)

Produzido pela *Giant Army*, *Universe Sandbox* é uma experiência que te permite criar, interagir e destruir planetas com a palma da sua mão (Figura 3). O jogo também conta com a possibilidade de alterar a massa, densidade, raio e muitas outras coisas de um planeta, e também aumentar ou diminuir o tempo passado, permitindo realizar diversos experimentos.



**Figura 3 – Seleção de planetas do jogo *Universe Sandbox***

#### 4. Métodos e ferramentas

Utilizou-se o desenvolvimento incremental que, segundo Sommerville (2011), é baseado na ideia de desenvolver uma implementação inicial, expô-la aos comentários dos usuários e continuar por meio da criação de várias versões até que um sistema adequado seja desenvolvido. A escolha desse modelo se deu ao fato de ser necessário evoluir várias etapas do projeto a partir de uma implementação inicial, como por exemplo a criação do personagem e dos objetos a partir de uma *Concept Art*. Segundo Sommerville (2011) existem três tipos de atividades no modelo em questão, de especificação, desenvolvimento e validação, e essas atividades são intercaladas, e não separadas, com rápido *feedback* entre todas as atividades.

Neste artigo vamos passar pelas atividades de especificação e de desenvolvimento, como são abordadas abaixo:

Primeira atividade: será especificada o que vai ser feito em geral, como o enredo e o cenário em que o jogo se passará. Também será determinada a *gameplay* do jogo, que segundo Vannucchi (2009) emerge das interações do jogador com o ambiente, a partir da manipulação das regras e mecânicas do jogo, pela criação de estratégias e táticas que tornam interessante e divertida a experiência de jogar.

Segunda atividade: será abordada a criação dos *Concept Arts*, os modelos 3D e a importação desses modelos na ferramenta Unity3D.

Já a terceira atividade citada por Sommerville não foi realizada, pois o protótipo elaborado não está funcional o suficiente para realização de testes em massa.

Para a criação das *Concept Arts*, que segundo Senna (2013) são a representação visual de personagens, ambientes e objetos, ou simplesmente a criação de uma atmosfera visual para uso em filmes, videogames, cinema de animação e histórias em quadrinhos, será utilizado a Mesa Digitalizadora One by WACOM CTL-472. O modelo permite a criação de esboços no computador com uma tela de 210 x 146 x 8,7 mm e uma caneta sem bateria, o suficiente para a criação das *Concept Arts* deste projeto. Em conjunto com a Mesa Digitalizadora, será usado o Krita, que é uma ferramenta de pintura grátis e de código aberto designada para artistas conceituais, ilustradores, artistas de *matte* e texturas, e a indústria *VFX* (Krita, 2018).

O desenvolvimento de jogos pode ser apoiado por uma *Engine*, também conhecido como motor de jogo, que será o integrador de diferentes componentes, que vão dos gráficos 2D, modelos 3D e animações, áudio, interfaces com dispositivos de E/S até a parte referente aos recursos de rede (BITTENCOURT e OSÓRIO, 2006). A ferramenta Unity3D foi escolhida como *Engine* do jogo, já que oferece a integração de objetos modelados em 3D, físicas pré-estabelecidas e a programação em cima desses dois fatores. O Unity3D usa as linguagens de programação *C#* e *Javascript*, e entre elas a escolhida foi *C#* por já ter um conhecimento básico sobre a linguagem.

Para a criação dos elementos em 3D foi escolhido o Blender, por ser uma opção gratuita de modelagem em 3D. A ferramenta se torna muito ágil de usar depois de se acostumar com suas teclas de atalho e além disso conta com um enorme acervo de plug-ins, que auxiliam na criação de texturas, formas humanas e animações.

A disponibilização ocorrerá de forma gratuita pelo site *Game Jolt*, conhecido por

hospedar inúmeros jogos independentes de várias categorias. Segundo o próprio site (*Game Jolt*) a ferramenta conta com várias opções de interação e *feedback*, como uma área para comentários e discussões, um sistema de avaliação e fóruns. O site auxiliará em versões futuras do projeto por tornar possível o teste com vários usuários, retornando assim *feedback* de bugs e melhorias.

O Gênero do jogo é de *Puzzle*, que segundo Bakie (2010) e Toshi(2012), citado por Hax e Filho (2015) são jogos eletrônicos que usualmente apresentam quebra-cabeças para serem resolvidos com a combinação de estratégia, combinação de padrões, lógica e sorte, frequentemente com o elemento tempo presente.

## 5. Desenvolvimento

### 5.1. Atividade de Especificação

Nesta Seção, será abordado a seção de *brainstorming* realizada para definir o jogo, os esboços e os protótipos iniciais para alcançar um protótipo aceitável de um jogo.

O primeiro passo feito foi a história do jogo, que serviu de base para a criação do personagem e da ambientação do universo. A história é sobre um alienígena chamado Notwen (Figura 4) que partiu de seu planeta natal em busca de respostas sobre o universo. Ele está em sua nave a caminho do centro do universo, mas a força gravitacional de um planeta atrapalhou a estabilidade do veículo e o puxou para a superfície do planeta, causando uma grande queda. Quando Notwen acorda, percebe que está em um planeta curioso e o admira por um tempo, até perceber sua nave quebrada a uns metros dali. Notwen chega para verificar e observa que tem pedaços de sua nave faltando, e ao procurar percebeu fumaça saindo de uma ruína e resolve averiguar. Ao chegar, percebe que a ruína contém objetos estranhos que querem lhe ensinar algo.

O nome Notwen foi escolhido em homenagem ao próprio descobridor das leis que serão abordadas no jogo, já que “Notwen” é “Newton” ao contrário. Apesar do nome ser uma homenagem ao físico, a escolha do estilo do personagem foi uma mais atual para que chame mais atenção das pessoas. O estilo do cabelo mistura um corte moderno com um visual alienígena, com mechas mais grossas e da mesma cor que a pele. As roupas dele são simples e a forma do corpo mais humanizada para parecer mais carismático.



**Figura 4 – Concept art do personagem principal Notwen**

O gênero *Puzzle* foi escolhido para este projeto pois obriga o jogador a resolver um quebra-cabeça de forma lógica. O jogo usará da física e das três leis de Newton para criar desafios, e mostrará em fórmulas matemáticas o que acontece quando você as usa, como demonstrado na Figura 5, que apresenta o *Concept Art* do jogo.



**Figura 5 – Concept art da gameplay**

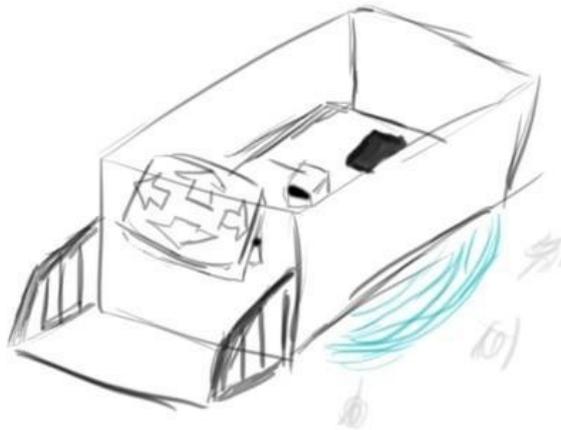
Antes de transformar os conceitos das leis de Newton em uma *gameplay*, é necessária uma breve explicação sobre as mesmas. A primeira lei de Newton diz que um corpo tende a permanecer em repouso ou em movimento retilíneo uniforme desde que nenhuma força atue sobre ele (Oliveira, 2017). A segunda lei de Newton diz que a facilidade em colocar um corpo em movimento depende de sua massa e aceleração, isto é, depende da interação entre os corpos (Oliveira, 2017). A terceira lei de Newton diz que quando um corpo A aplica uma força sobre o corpo B, este também aplica uma força no corpo A, ao mesmo tempo, com a mesma intensidade e na mesma direção, mas em sentido contrário (Oliveira, 2017).

Agora aplicando estes conceitos em uma *gameplay*, foram determinados 3 tipos diferentes de fases, uma para cada lei, dessa forma elas serão melhores representadas. Para a primeira lei de Newton, Notwen terá que movimentar um objeto flutuante que se movimenta aplicando forças em quatro direções, e através desse movimento guiar um cubo ativador até um buraco, ambos situados dentro do objeto flutuador. Para a segunda lei de Newton, Notwen precisará aplicar uma certa quantidade de força, dependendo do peso do cubo ativador para que esse cubo chegue no destino e desbloqueie o próximo nível. A terceira lei de Newton no jogo se assemelha muito a segunda, só que ao invés de uma força ser aplicada, Notwen deverá escolher um objeto a partir do seu peso e acertá-lo em outro para chegar ao destino. Nos três tipos, a força e a direção serão representados por uma seta.

## 5.2. Atividade de Desenvolvimento

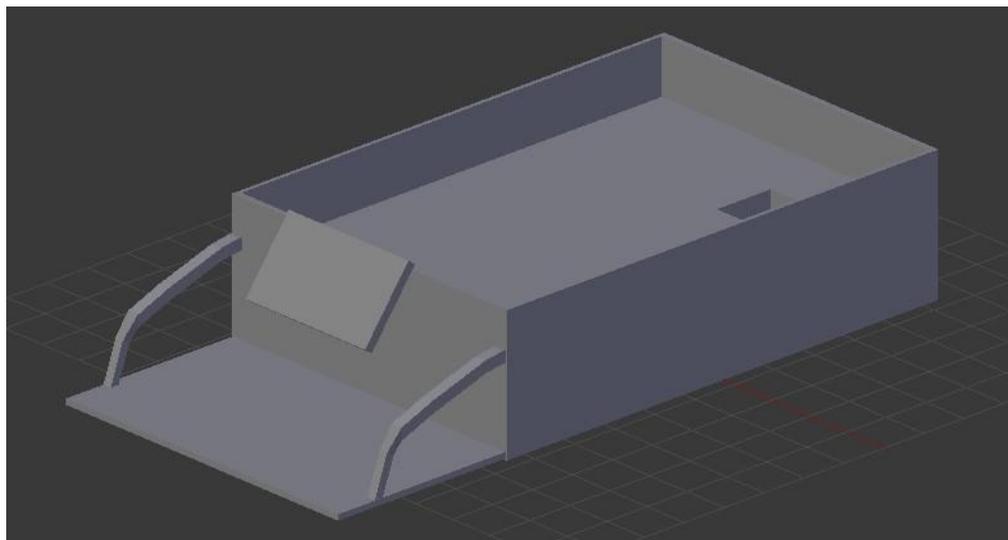
Para começar a evoluir os esboços, foi criado uma *Concept art* do objeto flutuante para servir como base da criação do modelo 3D (Figura 6). Nele foi colocado uma plataforma para o personagem subir, uma série de botões ilustrativos e o compartimento onde o cubo ativador

precisará estar.



**Figura 6 – Concept art do objeto flutuante.**

A partir deste *Concept Art*, foi utilizado o Blender para fazer a modelagem 3D do objeto flutuante.



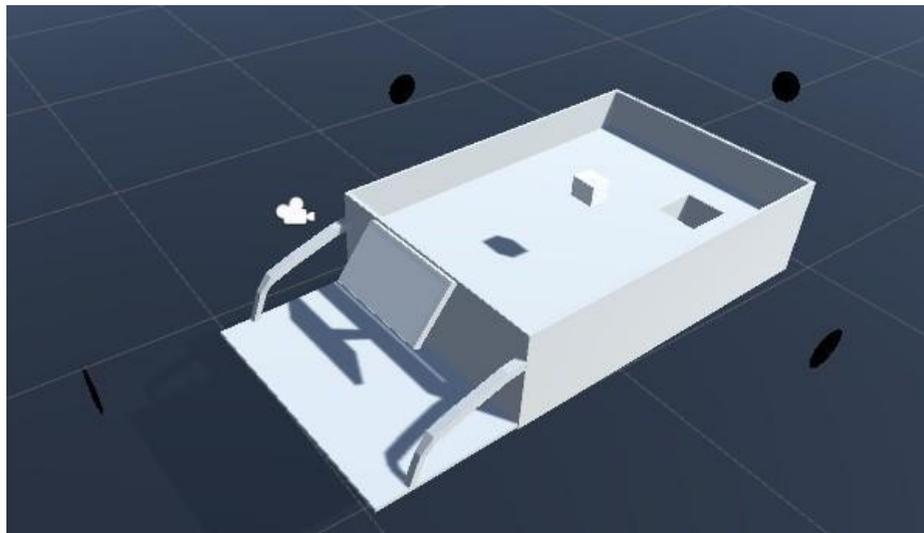
**Figura 7 – Modelo 3D do Objeto flutuante**

Com este modelo 3D, já é possível criar o protótipo pelo Unity3D. Ao importar o objeto será necessário a criação de *colliders*, que segundo a própria documentação do Unity3D (2018) são componentes que definem a forma de um objeto para propósitos de colisões físicas. Após isso, podemos criar a física dos objetos, criar indicadores de força e o cubo ativador (Figura 8).

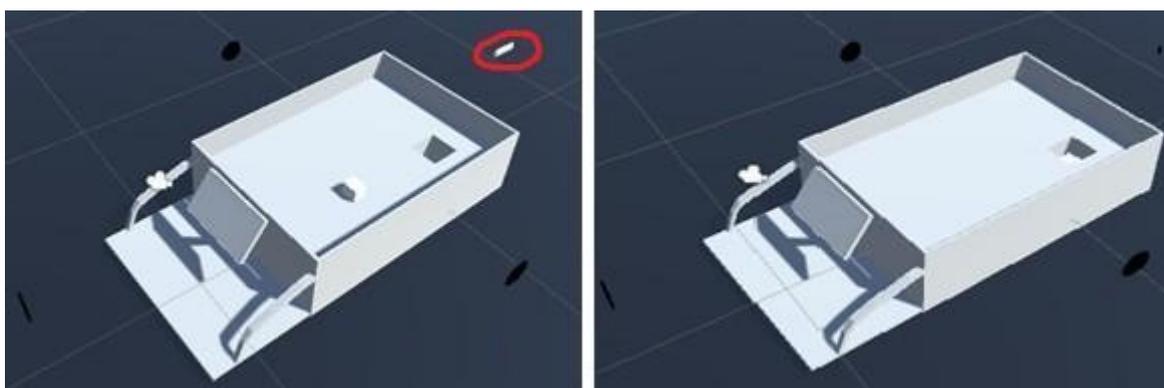
Agora, para criar o movimento do objeto flutuante, iremos adicionar uma força em uma certa direção, dependendo da tecla segurada. A partir do movimento do objeto, o cubo poderá se mover para o buraco (Figura 9).

Após esse pequeno protótipo, foi realizado mais um teste de conceito, só que desta vez mais focado na jogabilidade, procurando uma boa forma de apresentar o conceito de inércia para o usuário em modo de jogo. Surgiu então este protótipo em 2D, onde o objetivo é aplicar uma certa força ao alterar a aceleração e peso do carrinho, fazendo com que o carrinho e a

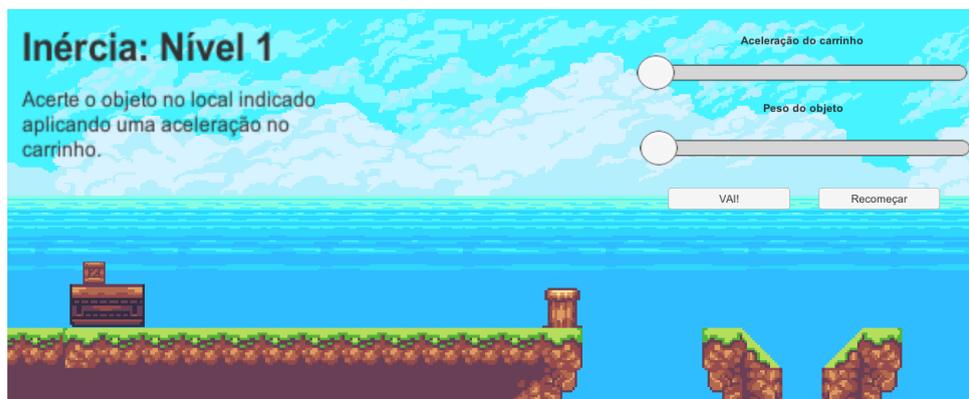
caixa sejam jogadas para frente e com que o carrinho bate no muro, enquanto a caixa continua seu caminho graças a inércia, fazendo com que a caixa acerte o buraco (Figura 10).



**Figura 8 – Posicionamento dos objetos**



**Figura 9 – Indicador de força e o cubo movido**



**Figura 10 – Protótipo 2D da primeira Lei de Newton**

Também foi feita uma segunda fase para esse protótipo, onde o objetivo é alcançar uma certa distância pré estabelecida, aplicando força da mesma forma que a fase anterior (Figura 11).



**Figura 11 – Protótipo 2D da primeira Lei de Newton**

Para acrescentar ao protótipo, foi feito um sprite do personagem Notwen em 2D, junto com uma animação para corrida, ficando como apresentado na Figura 12.



**Figura 12 – Protótipo 2D da primeira Lei de Newton**

Após esse *brainstorming*, surgiu a ideia de desenvolver o jogo para uma plataforma de realidade virtual, dessa forma a interação com o usuário e a imersão são muito altas, o que facilita a conexão com o jogador e o jogo. Os primeiros passos foram adaptar as ideias mostradas anteriormente para um jogo em realidade virtual. Alguns conceitos não funcionam muito bem com realidade virtual, como o personagem estar em uma plataforma móvel. Isso pode causar algo que chamam de “*Motion Sickness*”, pois sua visão diz que você está em movimento mas seu corpo diz o contrário. Com isso em mente, mais um protótipo foi feito aproveitando as ideias anteriores.

Nesta versão, um cubo é posto na base de um estilingue gigante, no qual o jogador pode puxar essa base e lançá-la a uma velocidade junto com o cubo. Ambos objetos então vão para frente até uma barreira que trava a base do estilingue, fazendo com que apenas o cubo continue e mostrando assim a inércia. O objetivo desta fase é acertar o cubo no recipiente (Figura 13).



**Figura 13 - Estilingue gigante e recipiente**

Para a lógica de programação do lançamento do estilingue, o código checa continuamente se o estilingue foi segurado pelo usuário, e caso tenha é alterado o status do estilingue para “segurado” (Figura 14). Depois disso o sistema continua checando se o estilingue foi solto, caso tenha é adicionado uma força no estilingue e no cubo em direção ao objetivo.

```
void LateUpdate()
{
    if (grab.isGrabbed == true)
    {
        estilingue.transform.rotation = initialRot;
        grabou = true;
    }
    if (grab.isGrabbed == false)
    {
        if (grabou == true) {
            estilingue.transform.rotation = initialRot;
            rbEstilingue.AddForce(-transform.forward * 1500);
            rbEstilingue.constraints = RigidbodyConstraints.None;
            grabou = false;
        }
    }
}
```

**Figura 14 - Codificação do lançamento do estilingue**

Ao cubo ser lançado, o sistema verifica se o mesmo entrou em colisão com o fundo da cesta, e caso tenha ocorrido, trará a tela de vitória para o usuário (Figuras 15 e 16).

```
void Update()
{
    if(ganhou == true){
        if (subiu == false)
        {
            vitoria.transform.position = new Vector3(vitPosX, vitPosY, vitPosZ);
            if (vitPosY < 12f)
            {
                vitPosY = vitPosY + 0.3f;
            }
            else
            {
                subiu = true;
            }
        }
    }
}

private void OnTriggerEnter(Collider other)
{
    ganhou = true;
}
```

**Figura 15 - Trecho do código que verifica a vitória da primeira fase**



**Figura 16 - Vitória apresentada ao usuário**

Após esse protótipo, houve a preocupação do conceito de inércia não estar muito claro desta maneira, então a ideia de um mundo absurdo surgiu, onde as leis de Newton não se aplicam. Este mundo absurdo (Figuras 17 e 18) pode ser acessado a qualquer momento no pulso, através de uma esfera que pode ser segurada pelo usuário e soltada, expandindo assim o mundo absurdo para a realidade.



**Figuras 17 e 18 - Mundo absurdo no pulso e expandido**

O jogador pode notar que o cubo e a base não respeitam mais a primeira lei de Newton, já que não mantém velocidade e também não mantém o estado de repouso, ficando quase impossível de cumprir o objetivo. O usuário consegue também voltar para o mundo normal, com mais uma esfera em seu pulso que pode ser segurada e ao ser solta ela suga o mundo absurdo (Figura 19).



**Figura 19 - Esfera usada para voltar ao mundo normal.**

O mundo absurdo foi programado se baseando em dois estados, checando se o mesmo está expandido ou não (Figura 20). Caso não esteja expandido, é verificado se a esfera foi solta para realizar a expansão do mundo absurdo, fazendo assim a animação da expansão da esfera em volta do usuário.

```
if (extendido == false)
{
    if (grab.IsGrabbed == true)
    {
        grabbed = true;
        selecMundo.transform.parent = null;
    }
    else
    {
        if (grabbed == true)
        {
            if (grabconfirm == false)
            {
                this.transform.position = selecMundo.transform.position;
                selecMundo.transform.position = new Vector3(3.61299992f, -8.82999992f, -0.041000011f);
                absorveMundo.transform.position = anchorMundo.transform.position;
                absorveMundo.transform.parent = anchorMundo.transform.parent;
                absorveMundo.GetComponent<Rigidbody>().velocity = Vector3.zero;
                grabconfirm = true;
            }
            if ((this.gameObject.transform.localScale.x <= 54.76326f) && (extendido == false))
            {
                this.gameObject.transform.localScale += scaleChange;
                scaleChange += scaleChange;
            }
            else
            {
                this.gameObject.GetComponent<SphereCollider>().enabled = false;
                scriptAbsorve.setAbsorvido(false);
                scriptAbsorve.setGrabbed(false);
                extendido = true;
            }
        }
    }
}
```

**Figura 20 - Código do mundo absurdo a se estender.**

Agora, caso a esfera já esteja no modo expandido, os objetos alterados pela física irão receber alguns valores aleatórios, e dependendo deste valor irá receber uma aplicação de força diferente (Figura 21).

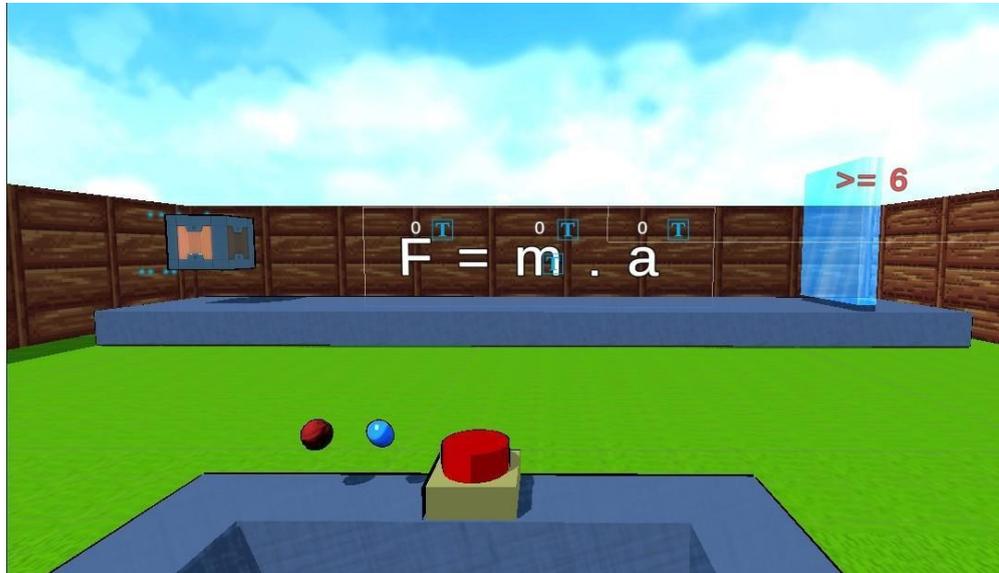
```
x += Time.deltaTime * 10;
this.transform.rotation = Quaternion.Euler(x, 0, 0);

countC += Time.deltaTime;

if (countC >= 0.5)
{
    int rand = Random.Range(0, 10);
    int randpos = Random.Range(0, 3);
    if (rand <= 7)
    {
        int randposneg = Random.Range(-1, 2);
        switch (randpos)
        {
            case 0:
                cubinho.GetComponent<Rigidbody>().AddForce(new Vector3(50f * randposneg, 0, 0));
                break;
            case 1:
                cubinho.GetComponent<Rigidbody>().AddForce(new Vector3(0, 500f * randposneg, 0));
                break;
            case 2:
                cubinho.GetComponent<Rigidbody>().AddForce(new Vector3(0, 0, 50f * randposneg));
                break;
        }
    }
    else
    {
        cubinho.GetComponent<Rigidbody>().velocity = Vector3.zero;
        countC = -2;
    }
    countC = 0;
}
```

**Figuras 21 - Código do mundo absurdo expandido.**

Como mecânica de jogo para a segunda lei de Newton, o usuário deverá escolher entre duas orbes, a Orbe Vermelha que aumenta a massa do cubo e a Orbe Azul que aumenta a aceleração que o cubo terá ao ser lançado (Figura 22). Com essas duas variáveis, o usuário terá que chegar em uma força resultante maior do que o indicado para quebrar o vidro e conquistar a vitória. Dessa forma, o usuário conseguirá entender de maneira visual que a força resultante de um objeto é igual ao valor da massa multiplicado pela aceleração.



**Figura 22 - Estágio da segunda lei de Newton**

Para movimentar a Orbe de grandeza vermelha ou azul para dentro do cubo, o sistema verifica se a Orbe está agarrada pelo usuário, e caso esteja, altera o status dessa orbe para agarrada e fazendo com que o sistema espere o usuário soltar essa orbe para adicionar uma força em direção ao cubo (Figura 23).

```

Vector3 relativePosition = cuboDestino.transform.InverseTransformPoint(ameba.transform.position);
if (grab.isGrabbed == true)
{
    grabou = true;
}
if (grab.isGrabbed == false)
{
    if (grabou == true)
    {
        soltou = true;
        grabou = false;
    }
    if (soltou == true) {
        aceleracao = aceleracao + 0.2f;
        amebaRB.AddForce(-((relativePosition * aceleracao) * 2));
    }
}

```

**Figura 23 - Codificação do movimento da Orbe de massa ou aceleração**

Quando a orbe entrar em colisão com o cubo, o sistema verifica a que grandeza essa orbe pertence e altera o valor de massa ou aceleração para que o usuário consiga verificar a força que será aplicada (Figura 24).

```

private void OnTriggerEnter(Collider other)
{
    GameObject obj = other.gameObject;
    if (obj.name == "add_aceleracao 2" || obj.name == "add_massa 2" || obj.name == "add_aceleracao" || obj.name == "add_massa" ){
        MoveObjeto move = other.GetComponent<MoveObjeto>();
        move.soltou = false;
        MoveObjeto.aceleracao = 0.0f;
        other.attachedRigidbody.velocity = new Vector3(0,0,0);
        other.gameObject.transform.position = move.posicaoInicial;
        LancaCubo lanca = cubo.GetComponent<LancaCubo>();
        other.GetComponent<TrailRenderer>().Clear();
        if (other.gameObject.name == "add_massa" || other.gameObject.name == "add_massa 2")
        {
            lanca.massa = lanca.massa + 0.2f;
        }
        else if (other.gameObject.name == "add_aceleracao" || other.gameObject.name == "add_aceleracao 2")
        {
            lanca.aceleracao = lanca.aceleracao + 0.2f;
        }
    }
}

```

**Figura 24 - Codificação da adição de massa ou aceleração**

Depois que o usuário configurar as grandezas, ele poderá lançar o cubo. O sistema verifica que o botão foi apertado, e caso seja, ele adiciona a força ao cubo dependendo do valor que foi colocado (Figura 25).

```

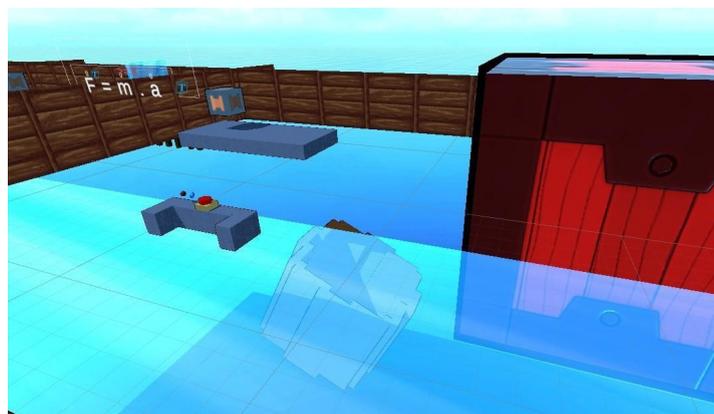
if (prontoLanca == true) {
    objRB.AddForce(new Vector3(0f, 0f, ((massa * 5) * aceleracao) * 100));
    prontoLanca = false;
}

obj.transform.localScale = new Vector3(massa, massa, massa);
ParticleSystem particleSystem = obj.GetComponent<ParticleSystem>();
if (particleSystem != null)
{
    var emission = particleSystem.emission;
    emission.rateOverTime = aceleracao * 10;
}

```

**Figura 25 - Código do lançamento do cubo**

Para a terceira lei de Newton foi usado uma mecânica parecida com a lei anterior, com as orbes usadas para a alteração das propriedades do cubo e um botão de lançamento, a diferença dessa vez é que o cubo colidirá com uma parede, fazendo que o mesmo quique de volta dependendo de sua força, mostrando assim que o cubo recebe a mesma força que está aplicando, só que em sentido oposto (Figura 26). O objetivo desta vez é acertar uma cesta que fica logo embaixo da plataforma de lançamento, fazendo assim o cubo acertá-la ou não dependendo da força que foi aplicada. A codificação para esta fase é praticamente a mesma que a anterior no quesito de adição dos valores de massa e aceleração. E para a verificação da vitória, a lógica segue a mesma da primeira fase, que é a verificação do cubo dentro da cesta.



**Figura 26 - Estágio da terceira lei de Newton**

## 6. Coleta de opiniões

Foi realizada uma coleta de opiniões com dois professores formados na área de Física e que atuam como professores (que chamaremos de Professor 1 e Professor 2), para validar a ideia, ou seja, verificar se o protótipo pode evoluir para um jogo em que sirva de apoio no ensino das leis de Newton.

Segundo o Professor 1, o protótipo é uma ideia legal e no ensino de ciências falta muito material interativo, já que o padrão é o conteúdo sair da lousa pelo professor e diretamente para o aluno. Também comentou que o aluno ao interagir com o jogo, vai naturalizando aquele conteúdo diferente de uma forma tradicional, e que gosta muito de trabalhar com tecnologia, mas nunca tinha visto um jogo de Realidade Virtual que conversasse com as três leis de Newton. Como alteração à proposta apresentada, o Professor 1 sugeriu liberar a alteração da superfície, para que o aluno consiga experimentar o atrito de formas diferentes.

O Professor 2 teve as mesmas opiniões sobre o uso do protótipo em sala de aula, mas propôs algumas alterações em prol da clareza do projeto. Teve dificuldades em entender a mecânica da primeira lei de Newton no protótipo, dado a visão em primeira pessoa do usuário não conseguir enxergar a colisão da base do estilingue no final da plataforma, e sugeriu a inserção de uma câmera em frente ao usuário, mostrando o movimento da plataforma e do cubo em uma perspectiva lateral.

## 7. Conclusão

Para esse projeto foram empregados conhecimentos assimilados em diversas disciplinas, e como a principal devo citar a de Desenvolvimento de Jogos (DJOI6), que ensinou o conceito de jogos, jogos sérios e auxiliou muito na utilização do Unity3D. Também foram de grande importância o conhecimento obtido em disciplinas como Programação Orientada a Objetos (POOI4), Linguagem de Programação III (LP3I3) e Inglês Técnico Avançado (IGTI2) para a leitura de artigos e materiais suplementares.

Chegamos ao final deste projeto com um protótipo digital funcional em Realidade Virtual, que consegue dar a oportunidade de experimentação ao usuário de forma que seria impossível replicar no mundo real, graças a ajuda do Mundo Absurdo, que também auxilia na clareza da proposta da fase. Vale salientar que existe um grande espaço para melhoria não só em clareza, mas também robustez do protótipo, como as alterações sugeridas pelos professores. Essas alterações em prol da clareza da proposta foram sem dúvida a maior dificuldade deste projeto, já que sem a opinião de terceiros é muito difícil enxergar que certa fase ou jogabilidade não está fazendo sentido; algo que a coleta de opiniões ajudou muito e que, com mais informações desse tipo, o protótipo pode alcançar um resultado ainda melhor. Mas com isso em mente vejo que o protótipo alcançou o resultado desejado, pois mostra que com a evolução do mesmo é possível chegar em um jogo funcional para ser usado como complemento no ensino das Leis de Newton.

Como trabalhos futuros para esse projeto, sugere-se a realização das próximas etapas de Fullerton, como a produção do projeto com o intuito de transformar este protótipo em um jogo completo, adicionando músicas e indicações de áudio para facilitar na clareza dos conceitos sendo entregues, um menu principal para ajustes e início do jogo, uma HUD ou *Heads-up Display* também melhorando a clareza do jogo. Após a produção, sugere-se

realização dos testes do jogo ao distribuir uma versão Alpha para pessoas interessadas para estas darem suas sugestões ou reportar *bugs* encontrados.

A evolução do alcance da internet trouxe muito mais formas de estudo e também a oportunidade para muitas pessoas interessadas em publicar suas formas de passar conhecimento das maneiras mais criativas possíveis objetivando atrair todos os tipos de pessoas.

## Referências Bibliográficas

ADAMS, E.T.; ROLLINGS, A. **Fundamentals of Game Design**. Universidade da Califórnia: Pearson Prentice Hall, 2007.

ASSIS, W. S. de; BITTENCOURT, T. N. **UTILIZAÇÃO DE RECURSOS MULTIMÍDIA NO ENSINO DE CONCRETO ARMADO E PROTENDIDO**. São Paulo: Departamento de Engenharia de Estruturas e Fundações, Escola Politécnica da Universidade de São Paulo.

BITTENCOURT, J. R.; OSÓRIO, F. S. **Motores para Criação de Jogos Digitais: Gráficos, Áudio, Interação, Rede, Inteligência Artificial e Física**. UNISINOS. 2006.

Blender3D. Disponível em: <http://www.blender3d.com>. Acesso em: 20 de novembro de 2018.

FIALHO, N. N.. **OS JOGOS PEDAGÓGICOS COMO FERRAMENTAS DE ENSINO**. Disponível em: <http://quimimoreira.net/Jogos%20Pedagogicos.pdf>. Acesso em: 24 de novembro de 2018.

GAME JOLT. Disponível em: <https://gamejolt.com/learn>. Acesso em: 12 de dezembro de 2018.

HAX, F. C.; FILHO, R. **Jogos Eletrônicos e Educação**. Rio Grande do Sul: Instituto Federal de Educação Ciência e Tecnologia Sul-rio-grandense. 2015.

HERBST, A. M. N. **PRODUÇÃO DIDÁTICO PEDAGÓGICA O USO DOS JOGOS ELETRÔNICOS EDUCACIONAIS PARA O PROCESSO DE ENSINO E APRENDIZAGEM DA MATEMÁTICA**. Disponível em: [http://www.diaadiaeducacao.pr.gov.br/portals/cadernospede/pdebusca/producoes\\_pde/2013/2013\\_uenp\\_mat\\_pdp\\_angela\\_maria\\_nogueira\\_herbst.pdf](http://www.diaadiaeducacao.pr.gov.br/portals/cadernospede/pdebusca/producoes_pde/2013/2013_uenp_mat_pdp_angela_maria_nogueira_herbst.pdf). Acesso em: 20 de novembro de 2018.

IBGE – INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA, 2018. Pesquisa Nacional por Amostra de Domicílios Contínua: **Acesso à Internet e à televisão e posse de telefone móvel celular para uso pessoal 2016**. Rio de Janeiro: IBGE. IBGE – Instituto Brasileiro de Geografia e Estatística, 1998.

KIRNER, C.; SISCOOTTO, R. **Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações**. Petrópolis: Editora SBC, 2007.

KRITA. Disponível em: <https://krita.org/en/features/highlights/>. Acesso em: 13 de dezembro de 2018.

OLIVEIRA, Carlos A.G. **Física – Educação de jovens e adultos (EJA)**. Curitiba:

InterSaberes, 2017.

SENNA, M. G. S. de. **Concept Art: Design e Narrativa em Animação**. Rio de Janeiro: PUC - RJ. 2013.

SOMMERVILLE, I. **Engenharia de Software**. 9. Ed. São Paulo: Pearson Education do Brasil, 2011.

Unity3D. **Unity User Manual**. v. 2018. 3 beta. Disponível em: <https://docs.unity3d.com/2018.3/Documentation/Manual/UnityManual.html> Acesso em: 12 de dezembro de 2018.

VALENTE, J. A.; FREIRE, F. M.; ARANTES, F. L. **Tecnologia e Educação: passado, presente e o que está por vir**. Campinas, SP : NIED/UNICAMP, 2018

ZILIO, S. C.; BAGNATO, V. S. **Mecânica, calor e ondas**. Disponível em: <http://www.fotonica.ifsc.usp.br/ebook/book3/Capitulo4.pdf>. Acesso em: 20 de novembro de 2018.

# Documento Digitalizado Público

## TCC - Artigo

**Assunto:** TCC - Artigo  
**Assinado por:** Andre Constantino  
**Tipo do Documento:** Relatório Externo  
**Situação:** Finalizado  
**Nível de Acesso:** Público  
**Tipo do Conferência:** Documento Digital

Documento assinado eletronicamente por:

- **Andre Constantino da Silva, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 02/10/2022 22:04:02.

Este documento foi armazenado no SUAP em 02/10/2022. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifsp.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

**Código Verificador:** 1117078

**Código de Autenticação:** 1ad1cace0a

