

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS CORNÉLIO PROCÓPIO
DIRETORIA DE GRADUAÇÃO E EDUCAÇÃO PROFISSIONAL
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

ELIAS DE MORAES FERNANDES

**NONDA: SERIOUS GAME NA EDUCAÇÃO DE RESÍDUOS SÓLIDOS URBANOS
ATRAVÉS DA VERMITECNOLOGIA**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO
2016

ELIAS DE MORAES FERNANDES

**NONDA: SERIOUS GAME NA EDUCAÇÃO DE RESÍDUOS SÓLIDOS URBANOS
ATRAVÉS DA VERMITECNOLOGIA**

Trabalho de Conclusão de Curso de graduação, do curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Coordenação de Informática – TADS – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para a obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Paulo Augusto Nardi

CORNÉLIO PROCÓPIO
2016



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Cornélio Procópio
Diretoria de Graduação e Educação Profissional
Sistemas da Coordenação de Informática
Tecnologia em Análise e Desenvolvimento de Sistemas



TERMO DE APROVAÇÃO

NONDA: SERIOUS GAME NA EDUCAÇÃO DE RESÍDUOS SÓLIDOS URBANOS ATRAVÉS DA VERMITECNOLOGIA

por

ELIAS DE MORAES FERNANDES

Este Trabalho de Conclusão de Curso de graduação foi julgado adequado para obtenção do Título de “Técnologo em Análise e Desenvolvimento de Sistemas” e aprovado em sua forma final pelo Programa de Graduação em Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná.

Cornélio Procópio, 10/11/2016

Prof. Dr. Paulo Augusto Nardi

Prof. Dr. Alexandre Rômulo Moreira Feitosa

Prof. Dr. Eduardo Filgueiras Damasceno

Dedico esse trabalho à minha família e à minha noiva.
Também dedico ao Departamento Acadêmico de Química e
Biologia da UTFPR-CT.

AGRADECIMENTOS

Agradeço primeiramente à Deus, por ter me dado inteligência e conhecimento para realização desse TCC.

Ao meu orientador Prof. Dr. Paulo Augusto Nardi, pela sabedoria com que me guiou nesta trajetória.

Ao co-orientador Prof. Dr. Alexandre Rossi Paschoal, por ajudar nos passos iniciais deste projeto.

A Secretaria do Curso, pela cooperação.

Gostaria de deixar registrado também o meu reconhecimento à minha família, pois acredito que sem o apoio deles seria muito difícil vencer esse desafio. Agradeço também à minha noiva Priscila, pela ajuda e companherismo em todas etapas da minha graduação.

Agradeço a equipe da empresa 2DVerse e também da empresa KAISE Entertainment, por apoiar durante o processo de desenvolvimento e teste.

Enfim, a todos os que por algum motivo contribuíram para a realização desse projeto como um todo.

“Não eduques as crianças nas várias disciplinas recorrendo à força, mas como se fosse um jogo, para que também possas observar melhor qual a disposição natural de cada um.”

Platão

LISTA DE FIGURAS

FIGURA 1 – TRADUÇÃO DE: VARIÁVEIS TECNOLÓGICAS INFLUENCIANDO TELEPRESENÇA (STEUER, 1993). FATORES QUE INFLUENCIAM A INTERATIVIDADE: JUNÇÃO DA TECNOLOGIA COM A EXPERIÊNCIA HUMANA	17
FIGURA 2 – ESTRUTURA DO TRELLO USANDO SCRUM SOLO	21
FIGURA 3 – FLUXO DE UMA SPRINT ESPECÍFICA DE UMA ENTREGA PARCIAL DO PROJETO.....	23
FIGURA 4 – FLUXOGRAMA DE CAMADAS DO MVC	24
FIGURA 5 – NONDA: SKETCH DO STORYBOARD REPRESENTADO ATRAVÉS DE TELAS	27
FIGURA 6 – TELA DE GAMEPLAY	28
FIGURA 7 – PERSONAGEM NONDA: SPRITESHEET DA ANIMAÇÃO "PULAR"	30
FIGURA 8 – NONDA: HUD – POSICIONAMENTO DO UI.....	30
FIGURA 9 – JOGO NONDA: PREDADOR PÁSSAROS	31
FIGURA 10 – JOGO NONDA: PREDADOR SANGUESSUGA	31
FIGURA 11 – JOGO NONDA: PREDADOR FORMIGA	31
FIGURA 12 – DESENHO PARA COLORIR.....	32
FIGURA 13 – DIAGRAMA DE SEQUÊNCIA.....	34
FIGURA 14 – DIAGRAMA DE CLASSE SEGUINDO O CONCEITO MVC	36
FIGURA 16 – CÓDIGO USANDO COROUTINES	39
FIGURA 17 – TRECHO DE CÓDIGO DA CLASSE ITEMSPAWNER	40
FIGURA 18 – CÓDIGO QUE ESTRUTURA CRIAÇÃO DE WAVES.....	41
FIGURA 19 – TELA SPLASHSCREEN	42
FIGURA 20 – TELA PRINCIPAL: MENU DO JOGO.....	43
FIGURA 21 – TELA DO QUIZ	43
FIGURA 22 – GAMEPLAY DO JOGO	44
FIGURA 23 – TELA ENDSCREEN DO JOGO NONDA MOSTRANDO PONTUAÇÃO E DESBLOQUEIOS DE COLETÁVEIS	44
FIGURA 24 – TELA COM INFORMAÇÃO SOBRE UM ITEM COLETÁVEL.....	45
FIGURA 25 – CRONOGRAMA COMPLETO DE ATIVIDADES SEMANAIS	45

LISTA DE SIGLAS

ABRALPE	Associação Brasileira das Empresas de Limpeza Pública e Resíduos Especiais
CNUMAD	Conferência das Nações Unidas sobre o Meio Ambiente e Desenvolvimento
DAQBI	Departamento Acadêmico de Química e Biologia
DIRGRAD	Diretoria de Graduação e Educação Profissional
MMA	Ministério do Meio Ambiente
MVC	<i>Model View Controller</i>
ODS	Objetivos do Desenvolvimento Sustentável
ONU	Organização das Nações Unidas
PGRS	Programa de Gestão de Resíduos Sólidos
PSP	<i>Personal Software Process</i>
REA	Recursos Educacionais Abertos
RSU	Resíduos Sólidos Urbanos
TIC	Tecnologia da Informação e Comunicação
UTFPR-CP	Universidade Tecnológica Federal do Paraná - Câmpus Cornélio Procópio
UTFPR-CT	Universidade Tecnológica Federal do Paraná - Câmpus Curitiba

RESUMO

De M. Fernandes, Elias. **Nonda: Serious Game na Educação de Resíduos Sólidos Urbanos através da Vermitecnologia.** 2016. 15f. Trabalho de Conclusão de Curso (Graduação) – Tecnologia em Análise e Desenvolvimento de Sistemas. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2016.

A tecnologia da informação aplicada à educação tem permitido o uso de inúmeras ferramentas com finalidade de difundir o conhecimento nas diversas áreas de ensino tais como Humanas, Exatas e Biológicas. Da fusão entre educação e o entretenimento tem nascido os jogos sérios educacionais, que demandam conteúdo sucinto e de suma importância. Pode-se elaborar jogos educacionais para temáticas como a Vermicompostagem - tecnologia de compostagem que trabalha com a biooxidação de resíduos sólidos orgânicos no envolvimento de minhocas na fauna microbiana para surgimento de húmus. Nesse contexto, o presente trabalho objetiva criar um jogo educacional como material didático de apoio sobre a Vermitecnologia. Desse modo, é apresentado uma proposta de um Serious Game nessa temática, detalhes de sua construção e desenvolvimento.

Palavras-chave: Vermitecnologia, Vermicompostagem, Compostagem, Educação, Serious Games, Mobile.

ABSTRACT

De M. Fernandes, Elias. **Nonda: Serious Game in Urban Solid Waste Education through Vermitechnology.** 2016. 15f. Trabalho de Conclusão de Curso (Graduação) – Tecnologia em Análise e Desenvolvimento de Sistemas. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2016.

Information technology applied in education has allowed the use of numerous tools with the purpose of disseminating knowledge in different educational areas such as Humanities, Physical and Biological. The fusion of education and entertainment was born the educational games that require succinct content and of critical importance. Educational games can be created using themes as Vermicomposting - composting technology that works with the Bio-oxidation of organic solid waste in the involvement of earthworms on microbial fauna to emergence of humus. In this context, the present work aims to create n educational game as a didactic material support about Vermitechnology. Thus, it is presented a proposal of a Serious Game in that subject, details of construction and development.

Keywords: Vermitechnology, Vermicomposting, Composting, Education, Serious Game, Mobile.

SUMÁRIO

1	INTRODUÇÃO	12
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	<i>Serious games</i> para interagir e envolver	16
2.2	Vermitecnologia	18
3	TECNOLOGIAS E FERRAMENTAS	20
4	MATERIAIS E MÉTODOS.....	22
4.1	Processo de Software <i>Scrum Solo</i>	22
4.2	Padrão de Projeto MVC	23
4.3	Gênero do Jogo.....	24
4.4	Enredo	25
4.5	Storyboard	25
4.6	<i>Gameplay</i> – Perspectiva Centrada no Jogador	28
4.7	Mecânica do Jogo.....	29
4.8	Game Design	29
4.8.1	Design Gráfico dos Personagens	29
4.8.2	Controle das Interações dos Personagens	30
4.8.3	Predadores.....	30
4.8.4	Interface de Interação com o Usuário	31
4.8.5	Level Design	32
4.8.6	Arte	32
5	DESENVOLVIMENTO	33
5.1	Diagrama de Sequência.....	33
5.2	Diagrama de Classe	34
5.3	Máquina de Estado do Player: Mudança de Animação.....	37
5.4	Codificação.....	38
5.4.1	Coroutines do Unity	38
5.4.2	Modelo e Controle para Predadores.....	41
6	APRESENTAÇÃO DO JOGO	42
7	CONSIDERAÇÕES FINAIS	46
	REFERÊNCIAS.....	47

1 INTRODUÇÃO

É notório o aumento dos resíduos sólidos urbanos no Brasil. Isso acarreta a preocupação relativa ao alto índice de destinação irregular desses resíduos bem como a falta de mecanismos para auxiliar na decomposição ecologicamente corretas desses resíduos. A vermicompostagem - técnica que usa minhoca para produzir húmus e adubar a terra - (NDEGWA, THOMPSON, 2001) é uma forma correta de destinação dos resíduos.

Segundo Nuernberg (2014), atualmente o país carece por serviços básicos (coleta e destinação adequada) e orientação para população de procedimento com finalidade de reduzir a contaminação do meio ambiente, diminuir o impacto na saúde pública entre outros fatores. Uma das formas que já se tem é a separação do lixo ecotóxico e coleta seletiva. Porém, a gestão para um sistema de tratamento de resíduos sólidos urbanos usando vermicompostagem ainda necessita de orientação e educação das pessoas.

Com o advento das novas Tecnologias da Informação e Comunicação (TIC), diversas tecnologias além de computadores têm sido empregadas no processo de ensino-aprendizagem, como o caso dos dispositivos móveis. Estes têm como vantagem a mobilidade e podem ser acessados em qualquer lugar, diferentemente do computador pessoal. De acordo com Tarouco (2004), a importância do uso dos computadores e das novas tecnologias na educação deve-se hoje não somente ao impacto dessas novas tecnologias (ferramentas) na nossa sociedade e às novas exigências sociais e culturais que se impõe, mas também ao surgimento da Tecnologia Educativa.

A partir do uso das Tecnologias voltadas para a educação, o objetivo dessa proposta é conscientizar as crianças entre 7 e 12 anos sobre a importância da vermicompostagem na gestão de resíduos por meio do desenvolvimento de um jogo para dispositivos móveis, material didático de apoio para o Departamento Acadêmico de Química e Biologia (DAQBI) da Universidade Tecnológica Federal do Paraná, câmpus Curitiba (UTFPR-CT).

JUSTIFICATIVA

Segundo a ABRELPE – Associação Brasileira de Empresas de Limpeza Pública e Resíduos Especiais, em pesquisa realizada em 2013, foi gerada no Brasil mais de 76 milhões de toneladas de resíduos sólidos urbanos (ABRELP, 2014) e, no Brasil a produção de lixo (21%) mais que dobrou em relação ao número de população (9,65%) nos últimos 10 anos.

O problema se encontra nos destinos finais dos Resíduos Sólidos Urbanos (RSU) que têm 58,4% destinados adequadamente e 41,6% inadequadamente. Esses números parecem promissores se comparado com a quantidade de lixo que foi produzido nos últimos 10 anos, porém tem-se uma longa jornada a fim de destinar corretamente esses resíduos (TRIGUEIRO, 2013).

Em algumas regiões do país, a iniciativa de Coleta Seletiva parece desfavorável ao número de casos que tentam apoiar o mesmo. Por exemplo, na região Nordeste, 57,2% da população não tem apoio para fazer a coleta; e, no Centro-oeste esse número chega a 62,5%. Ainda, essas duas regiões somam 30,3% das participações do total de RSU coletados (ABRELP, 2014), o que deixa uma lacuna que precisa ser preenchida.

Segundo o Ministério do Meio Ambiente, (2016):

A Lei nº 12.305/10, meta a prevenção e a redução na geração de resíduos, tendo como proposta a prática de hábitos de consumo sustentável e um conjunto de instrumentos para propiciar o aumento da reciclagem e da reutilização dos resíduos sólidos (aquilo que tem valor econômico e pode ser reciclado ou reaproveitado) e a destinação ambientalmente adequada dos rejeitos (aquilo que não pode ser reciclado ou reutilizado).

O índice de abrangência da Coleta de RSU no Brasil em 2014 foi de 90,68% (ABRELPE, 2014). Isso significa que o Brasil está crescendo na quantidade de RSU gerado, assim como a população brasileira. Algumas regiões se sobressaem, devido ao maior número de grandes centros urbanos.

Sabe-se que a incidência de Coleta Seletiva em municípios pequenos (cerca de 50 mil habitantes) é menor que nos grandes centros. É necessário levar a esses pequenos municípios o incentivo a cultivar essas boas práticas de redirecionamento adequado do lixo e da vermicompostagem, sem deixar de atender as metrópoles.

Um passo para tornar a coleta de resíduos sólidos eficiente é enfatizar o papel da população na participação da Coleta Seletiva, que pode começar domesticamente e crescer

para uma coleta de nível industrial, do tamanho de uma grande empresa ou Universidade, que é o caso da UTFPR Curitiba, por exemplo.

Uma alternativa promissora de conscientização sobre a coleta dos resíduos corretos e levar a um grande número de pessoas são os jogos digitais educativos pois esses podem ser inseridos dentro das salas de aulas que por sua vez são levados até membro familiares, fazendo a ciclo completo de divulgação do problema (Damani, B., Sardeshpande, V. & Gaitonde, U) . Embora os serious game sejam um segmento recente no Brasil, começaram a ganhar espaço assim como aconteceu com os games casuais em meados de 2008 (SAMPAIO, 2008). Comparativamente, a indústria de jogos digitais educativos tem aumentado 26% a cada ano (Innovation House Rio, 2015), desempenhando o papel de dramatizar os problemas, contribuírem para desenvolvimento de estratégias e rápidas tomadas de decisões, levando à um rápido processo de feedback.

O estudo de Nuernberg (2014) aponta resultados positivos com o uso da vermitecnologia: Primeiramente por ajudar a UTFPR em assinar o pacto da Agenda 21 – um plano de ação formulados internacionalmente perante a ONU – Organizações da Nações Unidas – e oficializado pela "Cúpula da Terra" – ECO 92 – Rio de Janeiro, que reúnem em 36 capítulos, um conjunto de metas e diretrizes básicas para o desenvolvimento sustentável (BRASIL. Ministério do Meio Ambiente) responsabilizando na redução do envio de seus resíduos para aterro sanitário em 28%.

Segundo, a aplicação da vermitecnologia permite a reciclagem dos resíduos não cozidos orgânicos produzidos no RU, que em 2016 corresponde a 50% dos resíduos orgânicos totais gerados na sede ecoville da UTFPR Câmpus Curitiba e também a mesma porcentagem para resíduos orgânicos totais gerado no Brasil. De acordo com o Capítulo 36 da Agenda 21 (BRASIL. Ministério do Meio Ambiente),

...preparar estratégias destinadas a integrar meio ambiente e desenvolvimento como tema interdisciplinar ao ensino de todos os níveis nos próximos três anos. Isso deve ser feito em cooperação com todos os setores da sociedade.

a UTFPR cumpre seu papel com a Agenda 21 e com os Objetivos do Desenvolvimento Sustentável – ODS, apoiando propostas de ações ambientais corretas já separando os resíduos para reaproveitamento e criando laços com a comunidade apresentando a vermitecnologia como uma forma de diminuir o desperdício de resíduos sólidos orgânicos por meio de palestras, aulas e jogos educativos. Com o apoio do DIRGRAD na Produção de

Recursos Educacionais Abertos (REA), foi possível a construção do jogo “Nonda” como material de apoio para ser usado em sala de aula junto com a cartilha ensinando sobre vermicompostagem.

OBJETIVOS

Com base no que foi exposto sobre a situação do Brasil no panorama da coleta de Resíduos Sólidos Urbanos e a possibilidade do uso da tecnologia aplicada em jogos digitais para incentivar formas de reutilização desses resíduos orgânicos produzido pelo próprio gerador, abaixo estão relacionados os objetivos gerais e específicos.

Objetivo Geral

- Apresentar por meio do jogo móvel a importância de tratamento de resíduos sólidos urbanos (separação correta do lixo orgânico que pode ser reaproveitado daquele que não pode) por meio de um jogo educativo criado para este fim.
- Desenvolver um *serious game* como ensino diferenciado para aplicação em salas de aula, usando a tecnologia móvel como forma de absorção de conteúdo.
- Estimular e sensibilizar alunos possibilitando o contato com elementos envolvidos no processo correto de vermicompostagem apresentados de forma lúdica.
- Ambientalizar os alunos no tema vermitecnologia e cultivar boas práticas para melhor proveito de materiais sólidos orgânicos.
- Desenvolver um jogo mobile como material de apoio para educadores aplicarem em sala de aula, quando conteúdo é sobre vermicompostagem como demanda do DAQBI da UTFPR-CT.

2 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo, são apresentados conceitos sobre as duas principais abordagens para o desenvolvimento do jogo Nonda: a vermitecnologia e *serious games*, com foco maior na elaboração do *serious games*. Ainda sobre esse último tópico é apresentada a perspectiva do jogo como ferramenta de aprendizagem por meio de plataforma e fixação do conteúdos citados no capítulo anterior. Por fim, é apresentada uma noção do tema que tem base nos estudos desenvolvidos pela Ana Cláudia Nuernberg (2014) visando o desenvolvimento de um jogo mobile a partir da extração da vermitecnologia explorada por ela e transformando em linguagem de jogo para interação com o usuário.

2.1 *Serious games* para interagir e envolver

Segundo Clark (1987), um jogo é um contexto em que jogadores se enfrentam tentando alcançar objetivos a partir de regras propostas; Clark evidencia que essas características não são suficientes para definir jogos, sendo que jogadores podem cooperar para contrapor uma situação natural do jogo que não fazem deles jogadores, pois esses não possuem objetivos. Para estruturar um jogo a um desses significados acima é necessário criar subgêneros narrativos, que transpassa os modelos narrativos habituais, onde coloca-se o jogador como principal tomador de decisões e transformar o rumo do jogo, como propõem os jogos digitais (MURRAY, 2003). A aprendizagem baseada nessas tomadas de decisões necessita o constante raciocínio do jogador, podendo esse raciocínio ser para situações estratégicas envolvendo tempo ou pontuais e específicas (JOHNSON, 2005), propósito engajado nos serious games.

O termo *Serious game* surgido na década de 1970, é definido como jogo educacional proposto a qualquer faixa etária em que possa ser executado em circunstâncias diferenciadas como por exemplo na educação, formação profissional, defesa, saúde, advergames, entre outros. O objetivo é colaborar na comunicação entre conceito e fatos – devido a interpretação de um problema e a motivação – que contribuem para o desenvolvimento de estratégias e tomadas de decisões a partir de um pré-conceito, representações de papéis como proposta para rápido *feedbacks* sobre o tema (LEMES, 2014).

Interatividade descreve medições mecânicas responsáveis pela ação do usuário e resposta da plataforma. São analisadas três particularidades nessa perspectiva: velocidade, mapeamento e alcance como destacado na Figura 1 (STEUER, 1993). A primeira, também conhecida como tempo de resposta, é uma característica importante no sistema de mídia interativa que dita a velocidade em que a plataforma responde às ações do jogador (STEUER, 1993). A segunda constitui a habilidade do sistema mapear os controles e mudar em tempo real de acordo com o ambiente de forma natural e previsível (STEUER, 1993). A última, o alcance, refere-se ao número de possibilidades por ação em um tempo determinado, ou a capacidade de toque sensitivos na tela, o que significa a liberdade do jogador poder executar ações dentro do ambiente. O jogador precisa ter a sensação de liberdade em executar várias tarefas, mesmo que seja um número limitado de opções (STEUER, 1993).

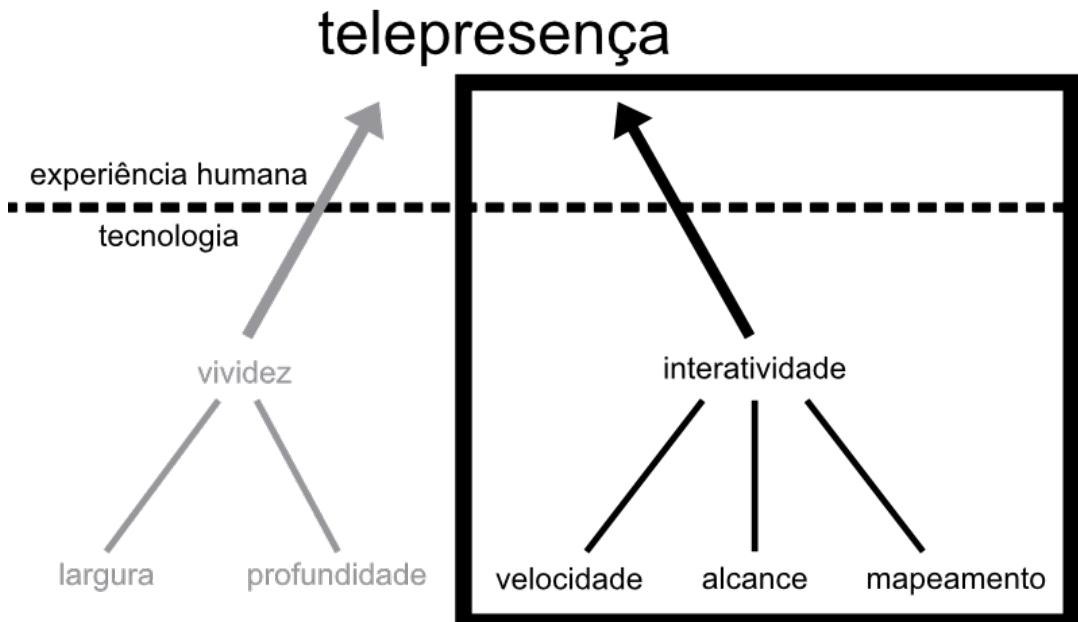


Figura 1 - Tradução de: Variáveis Tecnológicas influenciando telepresença (STEUER, 1993). Fatores que influenciam a interatividade: junção da tecnologia com a experiência humana

Kanstensmidt (2010) define a plataforma como um sistema capaz de executar jogos desenvolvidos especificamente para aquele ambiente. Cada plataforma disponibiliza recursos tecnológicos de entrada, processamento, exibição e, em alguns casos, transferência de dados. Plataforma tem função importante tanto quanto a narrativa e mecânica do jogo. Como exemplo disso cita-se a possibilidade de processar ações comandadas pelo jogador, processá-las de acordo com o comportamento de cada objeto e do sistema de regras e instantaneamente apresentar esse resultado para o jogador. Esse ambiente, juntamente com outros aspectos pode definir o grau de imersão do jogo, pois leva-se em consideração características do game design e do processo de produção, onde foca-se nas limitações técnicas dessa determinada

plataforma como por exemplo, adaptações para um gameplay mais atrativo ou simplificado (MENDES T., 2012). Como apresentado no tópico anterior, a objetividade nas tarefas estabelece uma relação entre o *game* e o jogador, em que o foco nas atividades são extremamente altos (PRENSKY, 2001). Isso ocorre porque existe um potencial imersivo através de desafio e recompensas, características da gamificação. Essas características são utilizadas na mecânica dos *games* em cenários *non games*, criando espaço de aprendizados mediado pelo desafio, prazer e entretenimento (ALVES, L. R. G. et al, 2014). Dessa maneira, o jogador sente-se motivado e engajado para continuar a realizar atividades mesmo sem dicas ou tutoriais. O envolvimento de tarefas ou uma série de tarefas mais parecidas com afazeres do que com divertimento faz com que o jogador se habitue a continuar jogando pelo motivo de ainda existir recompensa, a cada avanço dentro do contexto do jogo. Esses jogos contêm, sistematicamente, uma grande quantidade de objetos, que transmitem, de forma clara, recompensas articuladas (PRENSKY, 2001).

Com isso, o presente trabalho foca em utilizar os termos apresentados anteriormente de serious game focando na interatividade, sistema de recompensas usando uma plataforma que atende a esses requisitos e de vermicompostagem (compostagem com a ação de minhocas e microorganismos que misturam os resíduos sólidos para produzir húmus) para criar um jogo lúdico.

2.2 Vermitecnologia

A vermicompostagem é um processo biooxidativo (de resíduos sólidos), em que algumas espécies de minhocas detritívoras interagem, afetando positivamente e significativamente as taxas de degradação de uma matéria orgânica, na sua maioria devido às alterações ocorridas nas suas propriedades químicas, físicas ou microbiológicas (LOURENCO, 2015). Essas minhocas se alimentam de folhas mortas, gramíneas e outros resíduos orgânicos em diversos estágios de decomposição que são depositados no solo. Dentre esses resíduos, os principais são: esterco de animais, bagaço de cana-de-açúcar, frutas, verduras, resíduos industriais orgânicos, restos de podas, borras de café entre outros.

O resultado dessa alimentação é a produção de húmus, ou vermicomposto, um excremento das minhocas, produto natural, estável de coloração escura, rica em matéria orgânica, tendo nutrientes facilmente absorvidas pelas plantas (NUERNBERG, 2014).

Dentre muitos benefícios, de acordo com Sindifrutras (2014) melhora na porosidade e a aeração do solo, aumenta a vida biológica do solo, com o desenvolvimento de fungos

fixadores do oxigênio e bactérias, além da proliferação de microrganismos, aumenta a capacidade de *captação de nutrientes* pelas plantas e pode ser utilizado em todos tipos de culturas.

3 TECNOLOGIAS E FERRAMENTAS

No que diz respeito à tecnologia, foi utilizado a *game engine Unity*, produzida pela *Unity Technologies*. Uma *game engine* – ou simplesmente *engine*, motor de jogo em português, são *softwares* (*que contêm conjunto de bibliotecas*) capazes de auxiliar na simplificação e abstração dentro do desenvolvimento de um jogo. Dentre as funcionalidades que uma *game engine* oferece estão: motor gráfico que renderiza gráficos 2D/3D, motor para simulação de física, customização de efeitos sonoros, auxílio para animação, inteligência artificial, linguagem de programação (scripts), e outros (J. XIE, 2012).

Com *Unity* é possível desenvolver jogos para celulares e sistemas operacionais como Apple e Windows, jogos de consoles para Playstation, Xbox, Wii U, jogos baseados na Web usando Web Player e Web GL, para tecnologia de realidade virtual aumentada como Oculus Rift, Gear VR (da Samsung) além de suporte para TV usando Android TV e Samsung Smart TV. *Unity* tem diversas licenças (também chamado de produtos): *Unity Personal*, *Unity Plus*, *Unity Pro* e *Unity Enterprise*. Para o desenvolvimento do jogo Nonda, optou-se pelo uso da licença *Unity Personal*, sendo o mesmo no modo 2D.

A engine Unity pode distribuir um jogo em multiplataforma, característica usada para o jogo Nonda. Um fator importante para escolha dessa *engine* é a possibilidade de contar com assets gratuitos e atualizações compatíveis (atualmente na versão 5.4.1) com a necessidade do projeto além dos tutoriais disponíveis em caso de dúvidas ou até mesmo o uso da comunidade para dúvidas e interação.

A linguagem de programação em código aberto escolhida para desenvolver foi o C#. O *Unity* também permite o desenvolvimento na linguagem de programação *Javascript* e *Boo*. C# é uma linguagem de alto nível que permite desenvolvedores entrar facilmente no processo de desenvolvimento do jogo, aproveitando o máximo dos elementos e técnicas que a linguagem já possui. Por ser uma ramificação do C e C++, tem uma curva de aprendizado menor, e possui programação orientada a objetos como filosofia de design. Todas essas vantagens contribuem para criação de códigos fáceis de executar e depurar. Na questão de IDE, o C# é uma das linguagens mais versáteis já existente, aceitando IDEs como Visual Studio, Visual Basic e MonoDeveloper, para plataforma Mac.

Para fazer o *design* e criação de animação foram utilizados o *Affinity Designer* e *Adobe Illustrator CC*. O *Affinity Designer* é um software pago para plataforma Mac e Windows que permite criação de elementos gráficos vetorizados. A escolha deve-se ao fato do mesmo permitir o função "Export Persona" para exportação de *spritesheet* e *pixel art*.

pronta para usar no Unity. Para realizar algumas operações inexistentes no *Affinity Designer*, foi utilizado o *Adobe Illustrator*, que é da mesma categoria do *Affinity*. Em alguns momentos foi necessário o uso do *Adobe Photoshop CC* para redimensionamento de imagens, tamanho de arquivos e outros.

Para versionamento e colaboração do projeto foi utilizado o *GitHub*, que possui versão gratuita para estudante. O *GitHub* é um sistema de controle de versão para software baseado na web, podendo ser manipulado via terminal. O mesmo consiste em repositórios, onde são armazenadas as informações atualizadas de cada projeto. Através de um *link*, qualquer usuário (ou uma equipe) pode baixar, colaborar, atualizar, enviar novas atualizações sem depender de trabalho extra. Isso tudo torna o *GitHub* totalmente flexível.

Para armazenar todas as atividades a serem feitas, foi utilizado o *Trello*, que é uma ferramenta colaborativa de tarefas com versão gráts e paga. Como a versão gráts do Trello é suficiente para listar os requisitos do jogo, as atividades foram divididas em cartões (*Cards*) e listas (*Lists*). Na Figura 2, é mostrado o *Lists* contendo tarefas para um dia (*Doing*) e divisões entre diferentes áreas (Desenhos, Animação, Programação, Testes e Erros etc), marcando a *Sprint* em questão, sendo que terá um List contendo o total de *Sprints* do jogo. Na (ver Capítulo 6), está contida a lista completa de atividades.

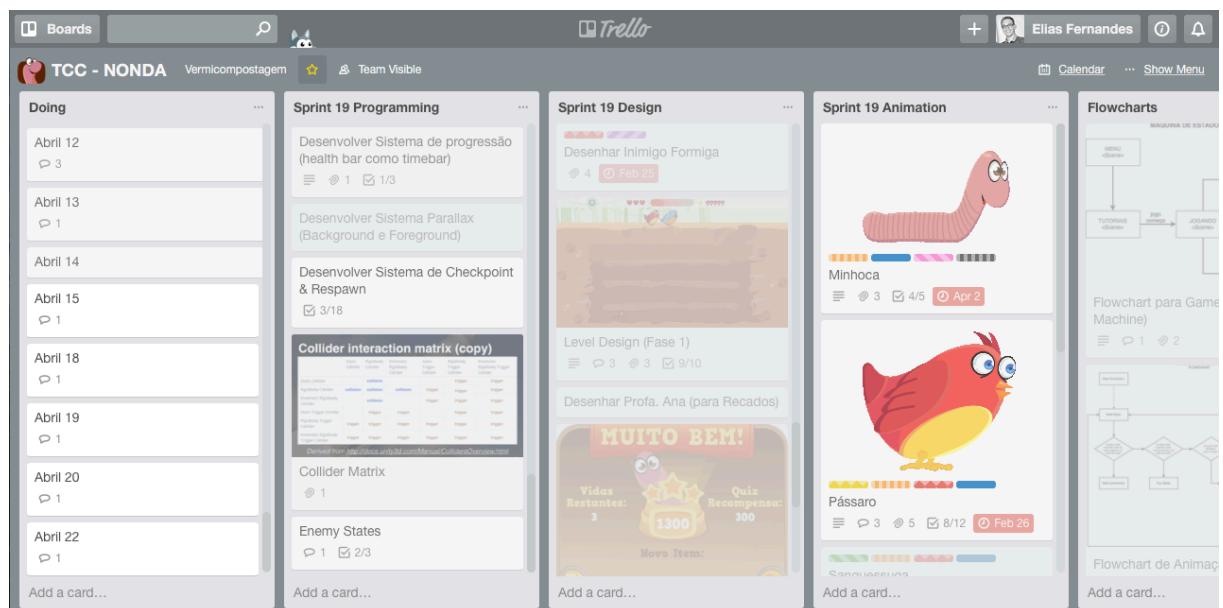


Figura 2 - Estrutura do Trello usando Scrum Solo

4 MATERIAIS E MÉTODOS

Nessa seção são descritos e discutidos os materiais e métodos utilizados nesse trabalho. A primeira subseção tem como objetivo descrever a metodologia para dar suporte no processo do jogo. A segunda subseção apresenta a arquitetura *Model View Controller* (MVC), base para o desenvolvimento do jogo Nonda. A partir da terceira subseção são descritos em detalhes, ao longo de 15 subseções, todas características do jogo que ajuda entender no ambiente de games o processo de software *Scrum Solo*.

4.1 Processo de Software *Scrum Solo*

Esse projeto foi desenvolvido de acordo com o *framework Scrum solo*, que tem base no *Scrum* e é uma metodologia Ágil. Dentre as fases do processo estão: detalhes dos requisitos, análise & design, implementação & teste, aprovação do teste, reavaliar / priorizar novas tarefas. Segundo JAMES, M. (2010), no desenvolvimento de jogos, o *Scrum* tem início a partir do levantamento dos requisitos passando pelo análise e design do UI/UX, ambiente, personagens e animação passando pela implementação da lógica do jogo, até a fase de teste onde finaliza a primeira iteração (*sprint*), sendo que nessa iteração o foco principal é estabelecer quais métodos, instâncias serão reaproveitados para reutilizar nas demais fases. O *Scrum solo*, desenvolvido por professores da UTFPR-CP, beneficia o desenvolvedor solo e tem as boas práticas do *Scrum* e Personal Software Process (PSP), processo que auxilia o desenvolvedor a entender a própria performance, desenvolvendo melhorias em suas práticas de trabalho.

O *Scrum solo* divide o projeto em várias sprints (iterações nos ciclos de desenvolvimento) que duram 1 semana e tem incrementos cada vez que um product owner identifica uma nova prioridade no product backlog (funcionalidades que o scrum team deve desenvolver no software).

Durante as iterações, o desenvolvedor codifica e testa pequenas partes do projeto. O diferencial é que entrega uma fatia do software no prazo dito acima, e se necessário existe reunião para orientação entre o grupo de validação (cliente final) e o desenvolvedor e não há reuniões diárias, como acontece no *Scrum* (T. PAGOTTO, J. A. FABRI, A. LERARIO AND J. A. GONÇALVES, 2016).



Figura 3 - Fluxo de uma sprint específica de uma entrega parcial do projeto

4.2 Padrão de Projeto MVC

Entre os engenheiros de software e arquitetos foi amplamente aceito que a concepção de aplicações em conformidade a esses *designs pattern* facilitariam a reutilização da experiência e conhecimento adquiridos por *experts* ao longo de exaustivos esforços em desenvolver um software de alto nível no mundo real (MASOVER, 2004). A arquitetura *Model View Controller* – MVC – justifica a afirmação do Masover (2004) e o fluxograma (Figura 4) possibilita a compreensão do reuso da experiência por meio da divisão das três camadas de aplicação Modelo de Negócio, Interface de Usuário e Lógica de programação. Em geral, o modelo gerencia os dados do software, a Interface é a camada de apresentação para o usuário e a Lógica trata os eventos para a interface (YONGLEI, 2002).

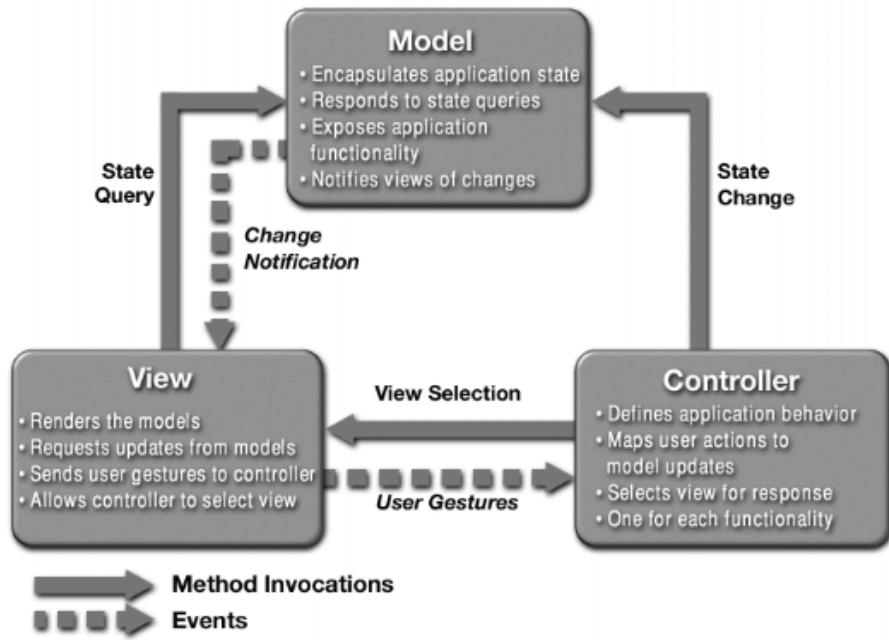


Figura 4 – Fluxograma de camadas do MVC
(MASOVER, 2014)

Trazendo o MVC para o desenvolvimento de games, é notório dizer que o fluxo de requisições está sempre na espera de uma ação do usuário ou em uma condição de disparo, seguido do envio da notificação desses eventos para a lógica do jogo que responde na conformidade dos eventos relacionados à ação disparada no início. Essa metodologia introduz outra camada de abstração que ajuda no planejamento do software. Quando divide em dados, interface e decisões há uma redução de numero de arquivos que consequentemente reduzem a complexidade de adicionar funcionalidades ou corrigir problemas (COSTA, 2015).

4.3

Gênero do Jogo

O jogo é uma combinação de dois estilos: *puzzle* e *non-stop running*. O primeiro estilo é apropriado pela arte de forçar o raciocínio do jogador antes de efetuar uma ação que resulte uma reação da parte lógica do game (TULLEKEN H., 2011). O segundo, implica que o jogador não terá domínio sobre a velocidade do personagem, ou seja, o mesmo está sempre em movimento (GIANT BOMB, 2015). Para mudar de direção do personagem, o jogador precisará deslizar na parte esquerda da tela (ver Seção 4.8.2).

4.4 Enredo

A história da minhoca Nonda acontece no minhocário da UTFPR, que fica sob a responsabilidade da Professora Ana Cláudia Nuernberg, chamada apenas de Ana, que é formada em Processos Ambientais também pela UTFPR-CT.

Ana educa a todos através de palestras como deve ser feito a vermicompostagem corretamente dentro da sala de aula ou fora, como no pátio da UTFPR ou no minhocário, onde Nonda trabalha.

O minhocário (*level design*) é composto por plataformas, que caracteriza o labirinto, e o objetivo de Nonda é cuidar para que nenhum inimigo tome posse e reproduza ou infecte a caixa de terra com agente ecotóxico. Para que Nonda continue produzindo biofertilizantes de qualidade (criando húmus ao defecar) e continue sempre forte, a Professora Ana sempre abastece a caixa de Nonda com resíduos orgânicos.

O jogo conta com 3 predadores, que são: o pássaro, o sanguessuga e a formiga. Cada um destes possui poderes diferentes, podendo ser mais lento, porém o dano é maior, como por exemplo o sanguessuga, ou rápido e invasor, como o caso do pássaro. A formiga terá uma colônia (carreira) de formigas que poderão atacar Nonda.

4.5 Storyboard

Antes de iniciar o jogo, é mostrada uma história sobre os personagens envolvidos (principal e predadores).

A concepção geral do jogo, como mostrado na Figura 5 é fazer o jogador entender o que pode e não alimentar a minhoca através da exploração do minhocário, quais os empecilhos e predadores Nonda pode ter durante o processo de alimentação para produzir húmus de qualidade e conseguir alta pontuação. A indicação de pontuação está no HUD de cada fase e, no menu direito, os itens e predadores, se houver, que estarão presentes.

A primeira fase, no tutorial, o objetivo é aprender o controle do jogo e coletar os itens que aparecem na tela. Esse ato simples de coletar itens saudáveis faz o jogador ser responsável pela minhoca Nonda.

Na segunda fase, são jogados alimentos que pode e não pode comer, fazendo o jogador decidir qual é o correto. Novamente, precisa da pontuação mínima para avançar.

A partir da segunda começa a fase dos predadores. O jogador precisa coletar os itens saudáveis e ao mesmo tempo proteger a vida da Nonda e o minhocário. A profª Ana deposita

alimentos na caixa (representado por a animação de uma mão no canto superior direito). Todas fases apresentam perigo de ataque dos inimigos. Por ataque, entende-se o fato do inimigo tocar no personagem. Cada toque (contato), o personagem perde vida. Para repulsar os inimigos, precisa tocar até fazer desaparecer da tela.

Da segunda à quarta fase, o jogador precisa enfrentar os pássaros. Da quinta à sétima fase, são as formigas que invadem o minhocário procurando atacar Nonda. Da oitava até a décima fase aparecem as sanguessugas.

Para restaurar a vida da Nonda, o jogo disponibiliza *power-ups*¹ como o Saco de chá. Quando o jogador coletar esse item, restaura 100% da sua vida, caso tenha perdido.

Além das fases, o jogo Nonda conta com Quiz, perguntas de certo ou errado, elaboradas pela Ana Claudia Nuernberg. Essas perguntas são para fixação do conteúdo e pode gerar competição, como por exemplo, a pontuação máxima sem errar as questões.

¹ Power-ups são itens que beneficiam ou adicionam habilidades extras instantânea ao jogador

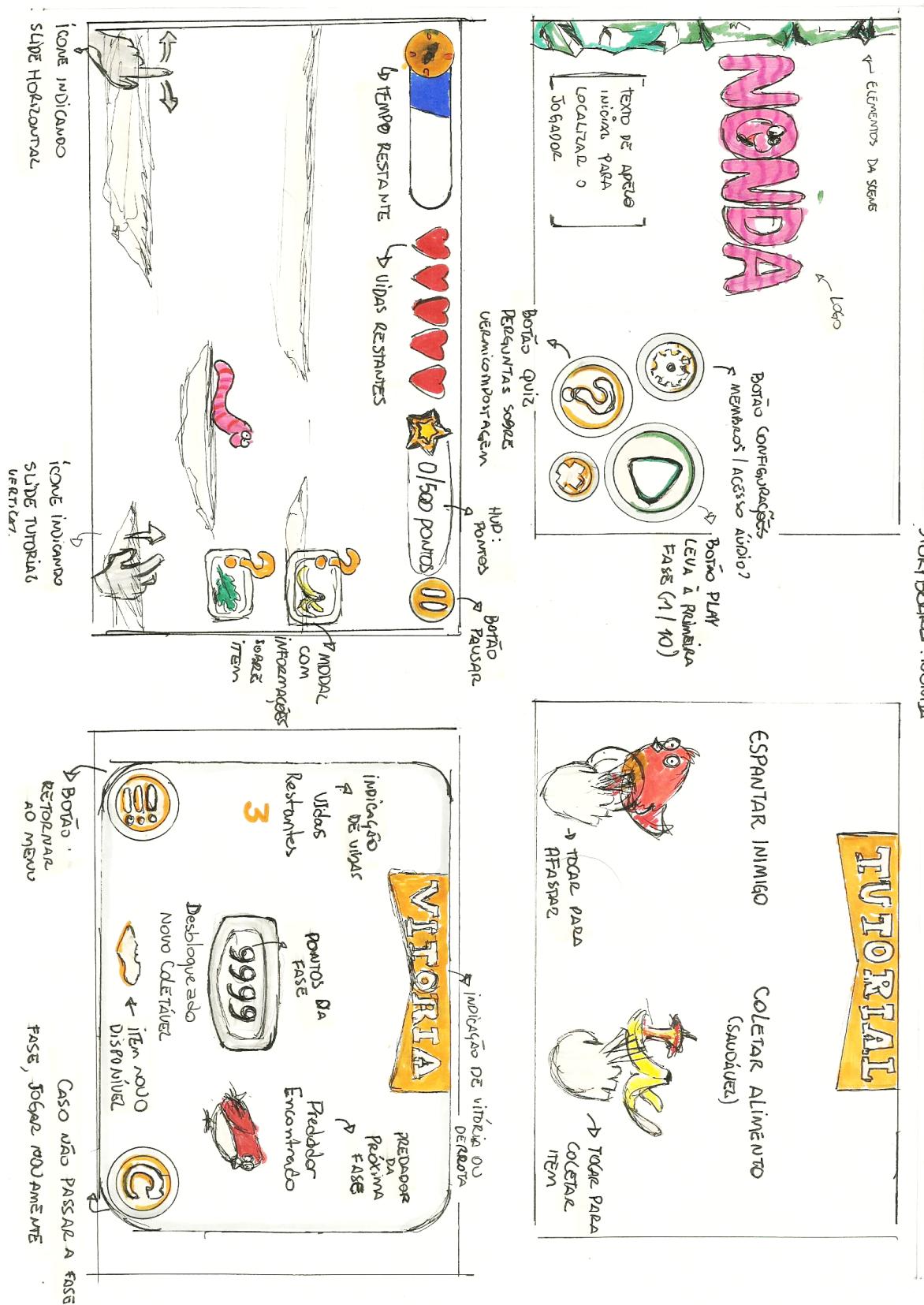


Figura 5 – Nonda: Sketch do Storyboard representado através de telas

4.6 Gameplay – Perspectiva centrada no jogador

Gameplay é o elemento central do jogo, que para muitos jogadores determina a qualidade do jogo (DJAOUDI, D. et al). Em outras palavras é a relação de conjunto de atividades que o jogador pode fazer e como é a experiência do mesmo enquanto joga.

O *gameplay*, ilustrado na Figura 6, induz o jogador a responder com ações aos elementos que interagem na tela, partindo da necessidade do personagem se alimentar, depois defendendo o ambiente dos predadores. Outro elemento importante do *gameplay* é o tempo, que tem a função de situar o jogador em que momento do jogo está e quanto ainda falta para atingir o objetivo.

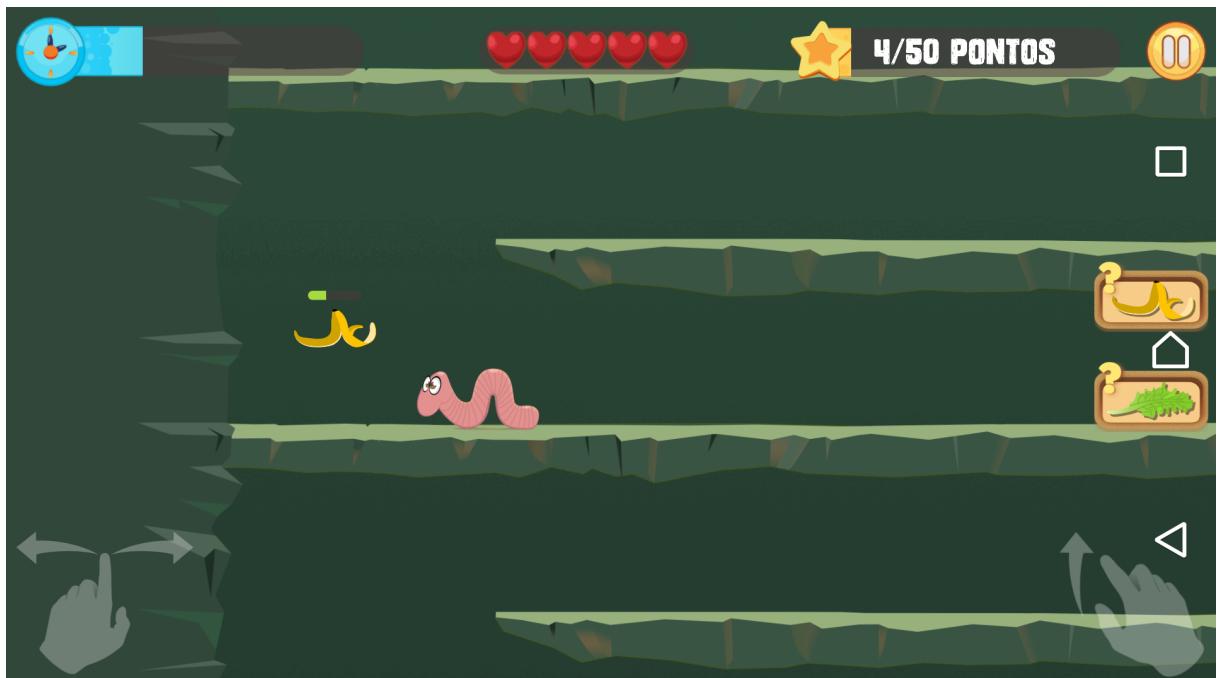


Figura 6 - Tela de gameplay

4.7 Mecânica do Jogo

Atividade e Interatividade são conceitos que fundamenta o tópico anterior. De acordo com (BRUNER, 1972), qualquer atividade lúdica envolve a interação com objetos concretos ou abstratos. É cada parte do *gameplay* individualmente. A mecânica do jogo Nonda é simples: usar slides para mover a Nonda e escapar de ataques dos inimigos, usar toque para interagir com os itens coletáveis, menu e quiz.

4.8 Game Design

É o processo do jogo onde informa toda característica do jogo, incluindo controles, jogabilidade, interfaces, personagens, inimigos, fases e outros. Todas essas serão descritas em tópicos separados abaixo.

4.8.1 Design Gráfico dos Personagens

Nonda tem 1 ano de idade, é uma minhoca (anelídeo), tem um tamanho padrão para uma minhoca. Nonda sempre sofreu com a invasão dos inimigos dentro da sua casa. Muitas vezes o solo estava quase pronto quando algum inimigo chegava e destruía tudo o que ela tinha umidificado. Tem personalidade calma, trabalhadeira e protetora e não possui poderes especiais. As ações dela dentro do jogo limitam-se em: andar, correr, pular, comer. É ilustrado na Figura 7 um *Sprite Sheet*² da personagem Nonda.

² *Sprite Sheet*: É uma série de imagens combinadas dentro de uma imagem maior para diminuir o número de arquivo de imagens. Essa série de imagem é mostrada sequencialmente como frames. (GOOGLE WEB DESIGNER HELP).

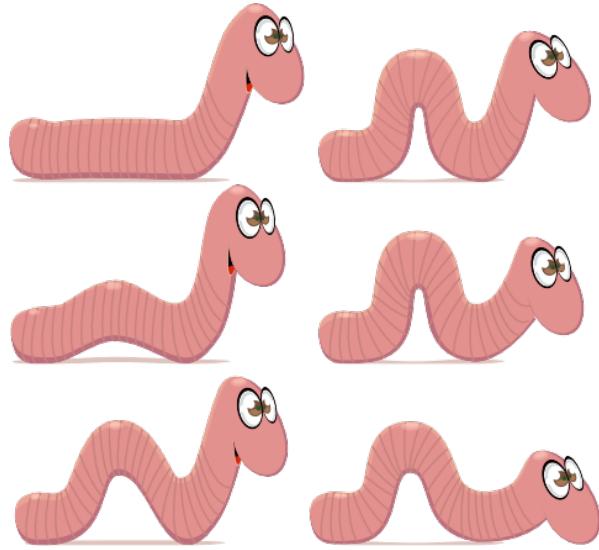


Figura 7 – Personagem Nonda: Spritesheet da animação "Pular"

4.8.2 Controle

Do personagem:

O personagem estará sempre se movendo, nunca parado. Para controlar, é necessário deslizar o dedo sobre tela (*slide*). Os controles disponíveis são:

Direcionais

- Deslizar horizontalmente no lado esquerdo da tela muda a direção do personagem
- Deslizar verticalmente (de baixo para cima) o lado direito da tela, ativa o comando pular. Se deslizar 2x seguidas, o personagem ativará a ação de pulo duplo.
- Toque (várias vezes) em cima do inimigo para dar dano.

Do predador:

- Possui movimentos randômicos e pode mudar o comportamento se o player se aproximar ou afastar.

4.8.3 Predadores

O predador no contexto do jogo são espécies distintas que aparecem para atrapalhar o trabalho de Nonda no decorrer do jogo.

Nessa versão, o jogo conta com 3 predadores: Pássaro, Sanguessuga e a Formiga ilustrados nas Figura 8, Figura 9 e Figura 10 .



Figura 8 – Jogo Nonda: Predador Pássaros

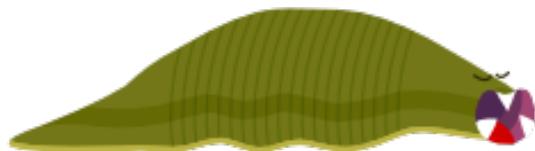


Figura 9 – Jogo Nonda: Predador Sanguessuga



Figura 10 – Jogo Nonda: Predador Formiga

4.8.4 Interface de Interação com o Usuário 31

Na interface, o foco é no HUD – *head-up display*, termo utilizado para visualizar todos os elementos de interesse do jogador, tais como barra de progresso, tempo restante, quantidade de vidas, pontuações, indicações de localidade para atacar etc – do jogo, conforme ilustrado na abaixo.



Figura 11 – Nonda: HUD – Posicionamento do UI

4.8.5 Level Design

O jogo vai se passar somente em um cenário, pois se trata de um ambiente de cativeiro da minhoca, que é característica básica da vermicompostagem. Tem a possibilidade desse cativeiro mudar de cor, conforme a qualidade do adubo.

4.8.6 Arte

A arte é baseada no minhocário da UTFPR, tendo apenas como plano de fundo a mudança de plataformas, conforme a fase. Foi escolhida o estilo *cartoon*³ para dar maior diversão ao jogador, pois não se sente na obrigação de estar em um simulador ou um jogo de primeira pessoa, por exemplo. Outro fator para escolha é pela habilidade artística adquirida com *cartoons* ao longo dos anos.

Com esse estilo de arte, foi possível expandir o design do jogo, transferindo para o papel, no formato de colorir como mostrado através da Figura 12. Essa expansão foi fundamental para ensinar as crianças do ensino fundamental durante as atividades sobre o tema.

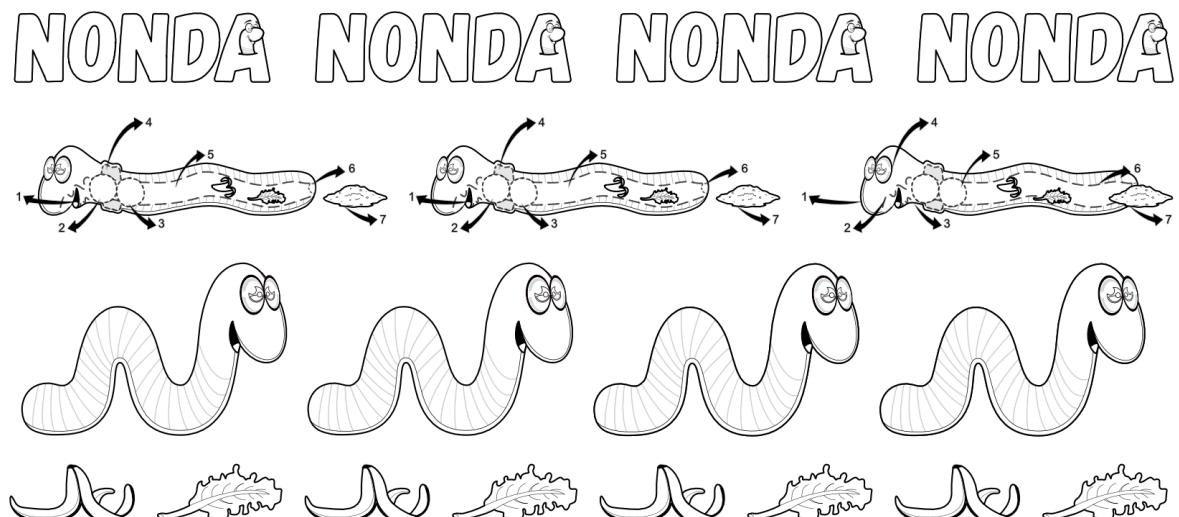


Figura 12 - Desenho para Colorir

³ Cartoon: É uma Ilustração 2D. Esse estilo artístico busca trazer um tom meio realista para o desenho ou pintura. Pode ser usado para representar sátira, humor e caricaturas (Y. ZHANG et al).

5 DESENVOLVIMENTO

Até o capítulo 4, este trabalho concentrou-se em mostrar a fundamentação teórica seguida do ANEXO A – GAME DESIGN DOCUMENT DO JOGO NONDA para entender aspectos de criação do jogo Nonda, as ferramentas que faz parte do processo de desenvolvimento e os métodos a serem utilizados para atingir o objetivo. Nesse capítulo será abordado o processo de construção, incluindo diagramas, trechos de códigos e considerações.

Para entender o diagrama de sequência é necessário relembrar o *storyboard* do jogo Nonda (Figura 5).

5.1 Diagrama de Sequência

No contexto do jogo Nonda, o diagrama de sequência narra o fluxo do jogo, deixando claro quais cenas necessitarão de quais elementos e em qual ordem. Na Figura 13, primeiro é mostrado ao usuário a tela principal do jogo, seguido da tela de tutorial (que se repete a cada fase). Logo após surge a tela do jogo para o usuário interagir, pontuar e avançar de fase.

É importante mencionar que existem telas que podem ser acessadas durante o jogo como o Tutorial ou o Menu de Opções. Seguindo o fluxo do jogo, a próxima tela informa a performance do jogador na fase específica, conhecido como *End Screen*, que dá acesso a uma próxima fase, jogar novamente a fase, caso não pontuar suficiente ou ao menu principal.

Essa sequência do storyboard faz o *game design* fluir organicamente.

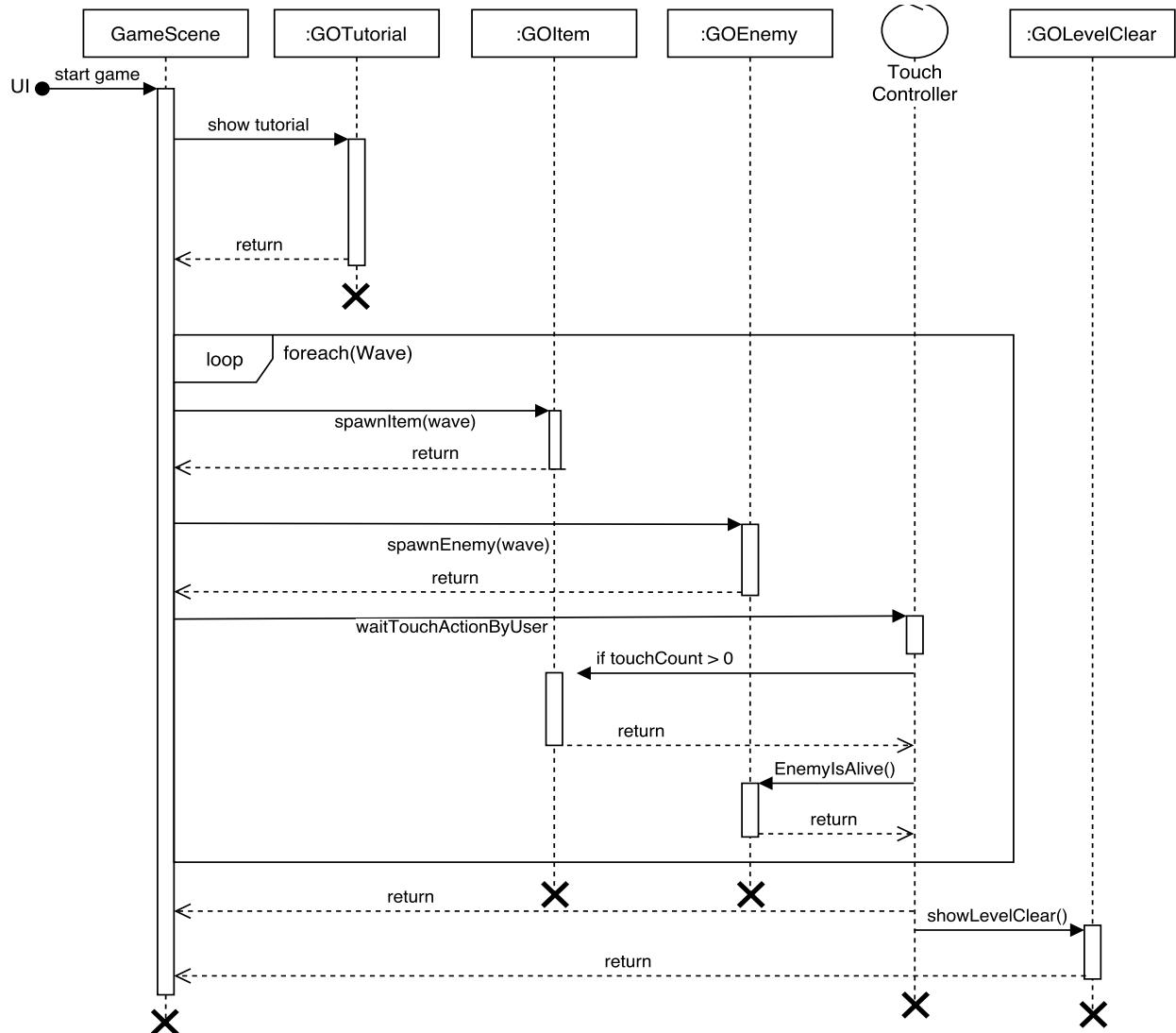


Figura 13 - Diagrama de Sequência

5.2 Diagrama de Classe

O diagrama de classe foi modelado seguindo o conceito da arquitetura Model View Controller descrito na seção 4.2 separando a modelagem de negócio da lógica de programação e da interface de usuário. O diagrama é composto de 21 classes e as três principais são Character, Level Manager e Item Controller; as demais classes tem associações e herança com

essas classes como mostrado na Figura 14 . Pode ser observado que a Classe *Character* é a base tanto para implementar o jogador quanto para o predador. Outra classe importante a citar é o *Spawner Controller*, que foi implementada para dar base às outras como *spawnar*⁴ itens coletados e *spawnar* predadores.

⁴ *Spawn* – Termo que descreve o nascimento de um jogador, inimigo item, etc. *Respawn* é o termo usado para um jogador voltar à vida dentro do jogo (PULSIPHER, L., 2010).

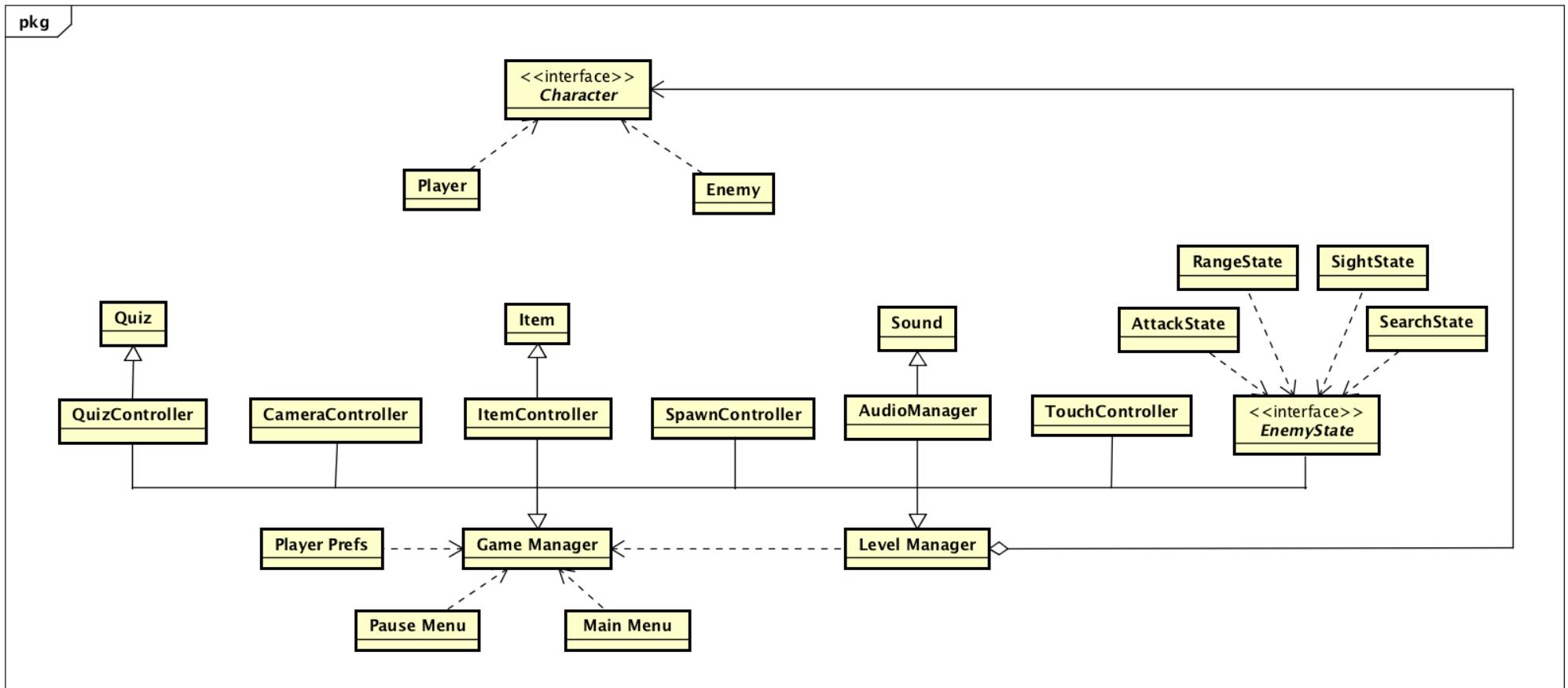


Figura 14 – Diagrama de Classe seguindo o conceito MVC

5.3 Máquina de Estado do Player: Mudança de Animação

A personagem Nonda possui diversos estados de animação, de acordo com um comportamento gerado pelo *gameplay* ou pela intervenção do usuário através do toque na tela. São atribuídos cinco estados diferentes que Nonda: Andar, Correr, Pular, Cair, Pousar. Esses estados de animação são definidos abaixo, na Figura 15. Para mudar de estado, algum outro estado precisa intervir em tempo de execução do jogo, no método *Update()*, que é disparado em um *frame por segundo*⁵. O estado de animação padrão da Nonda é Andar. Isso significa que sempre a velocidade é maior do que 0 e, se o jogador executar ações do tipo pular ou mudar de direção, implicará mudança de estado de animação resultando em conjunto de sprites totalmente diferente para que represente tal estado de animação.

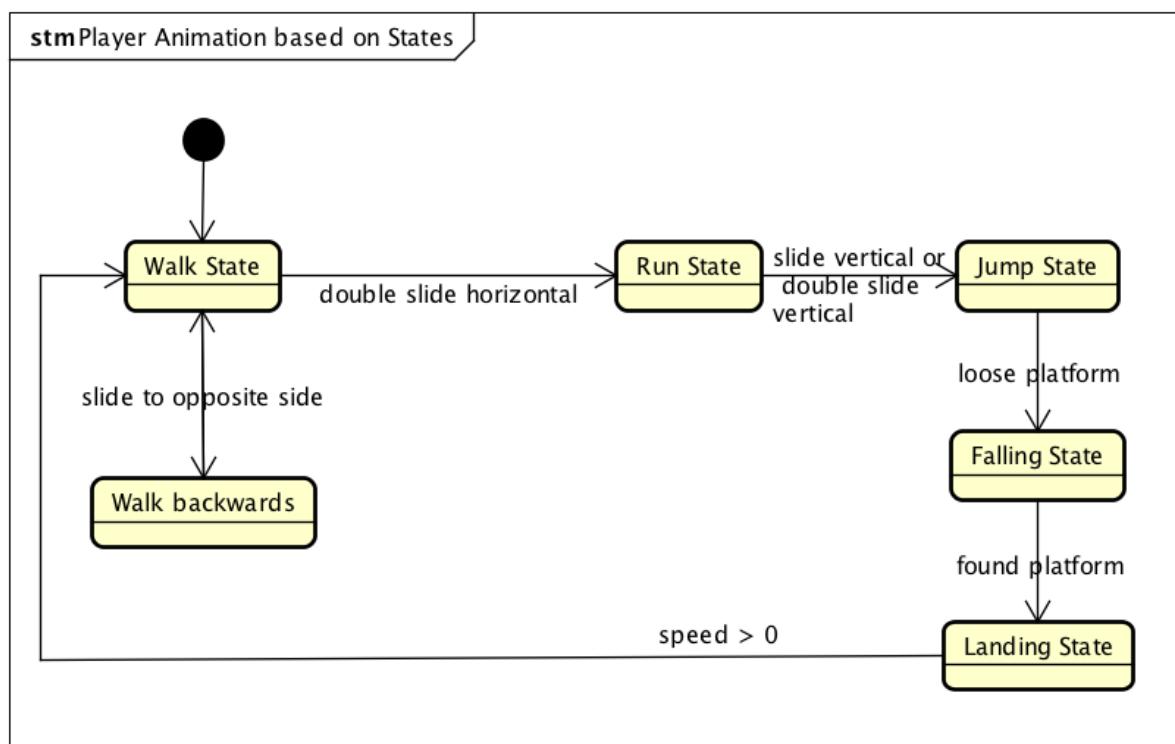


Figura 15 – Máquina de Estado do personagem Nonda (Player)

⁵ *Frame por segundo*: É a capacidade máxima que o processador consegue reproduzir imagens dentro de 1 segundo (FLICK J., 2015).

5.4 Codificação

O presente capítulo procura demonstrar trechos de códigos para o desenvolvimento do jogo, usando a linguagem C#, tendo subtópicos para códigos que implementaram função descritas no Diagrama de Classe usando o padrão de projeto MVC.

5.4.1 Coroutines do Unity

Coroutine (Co-rotina, em português) é uma função da própria *game engine Unity* que fornece execução de uma tarefa sequencial, sem divisão de processador (Z. LI E E. KRAEMER). Coroutine funciona da seguinte forma: a função salva o último ponto que parou, e no próximo frame executa o restante do código. O ganho em relação às *threads* (*threads* ou *multithreads* são funções que usam sincronia de várias funções distintas para executar animações, colisões, entradas fornecidas pelo usuário, entre outros) é que não exige alto processamento computacional, especialmente em caso de dispositivo móvel com baixa memória (DESENVOLVEDOR DE JOGOS, 2016). O Unity 3D usa a interface *IEnumerator* para implementar Coroutine, isso significa que une duas interfaces (*IEnumerator* e *IEnumerable*) para que possa numerar, através de um cursor interno para o índice atual e fazer o gerenciamento para o ponto real eliminando verificações que espendem memória. É usado a declaração *yield* como marcador para que continue após essa declaração na próxima vez que for chamado. Semelhantemente ocorre com a declaração *return* (DESENVOLVEDOR DE JOGOS, 2016). Abaixo, na Figura 16 pode-se observar o código extraído da classe *ItemController.cs* que tem adiciona itens na tela em forma de *waves*⁶. No método *SpawnWave(Wave wave)* dentro do um laço de repetição *for* encontra-se uma chamada para o método *SpawnItem(_wave.item)* que adiciona na jogo a quantidade de items definidos pelo tamanho do laço de repetição. Logo após, é chamada a declaração *yield* que retorna uma função *WaitForSeconds(1f/rate)*, esperando um tempo passado entre parâmetros para dar

⁶ Waves - Termo usado para caracterizar um grupo de inimigos, itens, power-ups que vem em um certo tempo em uma quantidade pré-determinada durante o gameplay de um jogo (ADAMS, ERNEST, 2010).

continuidade na execução do método. Logo após sair do laço, o Enum muda de estado da função para *Waiting* e passa a executar o que vem após a declaração *yield*.

```

void Update() {
    if(state == SpawnState.WAITING) {
        if(!ItemHasEnergy()){
            WaveCompleted();
        }else{
            return;
        }
    }
    if(waveCountdown <= 0){
        if (state != SpawnState.SPAWNING){
            HandController.Instance.PlayHandAnimation();
            StartCoroutine( SpawnWave( waves[nextWave] ) );
        }
        else{
            waveCountdown -= Time.deltaTime;
        }
    }
}
bool ItemHasEnergy(){
    searchCountdown -= Time.deltaTime;
    if(searchCountdown <= 0f){
        searchCountdown = 1f;
        if(GameObject.FindGameObjectWithTag ("Item") == null){
            return false;
        }
    }
    return true;
}

IEnumerator SpawnWave(Wave _wave) {
    state = SpawnState.SPAWNING;
    for (int i = 0; i < _wave.count; i++){
        SpawnItem(_wave.item);
        yield return new WaitForSeconds(1f/_wave.rate);
    }
    state = SpawnState.WAITING;
    yield break;
}

public void SpawnItem(Transform _item, Transform _touchColorItem){
    Transform _spawnPoint = itemSpawnPoint[ Random.Range (0,
    itemSpawnPoint.Length) ];
    Instantiate(_item, _spawnPoint.position, _spawnPoint.rotation);
}

public void SpawnItem(Transform _item){
    Transform _spawnPoint = itemSpawnPoint[ Random.Range
(0,itemSpawnPoint.Length) ];
    Instantiate(_item, _spawnPoint.position,
    _spawnPoint.rotation);
}

```

Figura 16 - Código usando Coroutines

Com uso de coroutines, foi possível seguir o diagrama de classe e conectar classes, solidificou dependências apenas criando métodos específicos. Outra contribuição importante

foi a capacidade de eliminar processamento de memória (dito anteriormente) com a habilidade de pausar a execução e retornar o controle para o Unity, e quando volta a execução continua a partir do ponto onde foi deixado.

Coroutines foi a base para implementar todas classes que envolve criação de objetos em tempo de jogo como as classes *ItemController.cs*, *SpawnerController.cs* que representa os Itens e Predadores, respectivamente. Porém, existe uma classe que é exceção: *ItemSpawner.cs*. Essa classe deposita os itens não saudáveis em tempos aleatórios e faz desaparecer independente se todos foi tocado pelo jogador ou não. A classe traz um conceito simples de inteligência artificial para jogo, embora o desenvolvimento não tem foco nessa área. Na Figura 17, o trecho do código da classe em questão verifica se é hora de spawnar novos objetos na cena, caso seja positivo é chamado o método *isTimeToSpawn()*. Esse método verifica se existe algum item na cena e se ainda pode ser depositado item. Caso seja positivo, é executado o método *SpawnWave()* com parâmetros necessários.

```

void Update () {
    if (waveCountdown <= 0) {
        isTimeToSpawn ();
    } else {
        waveCountdown -= Time.deltaTime;
    }
}

void isTimeToSpawn (){
    waveCountdown = timeBetweenWaves;
    if (GameObject.FindGameObjectWithTag ("ItemN") == null) {
        if (nextWave > waves.Length - 1) {
            nextWave = 0;
            Debug.Log ("All bad waves completed! Looping..."); 
            return;
        } else {
            SpawnWave (waves [nextWave]);
            nextWave++;
        }
    }
}

```

Figura 17 - Trecho de código da classe *ItemSpawner*

5.4.2 Modelo e Controle para Predadores

Como o jogo possui várias fases e é alimentado por diferentes predadores, é necessário implementar uma classe modelo para predadores que seja dinâmico e fácil de controlar. O modelo desenvolvido apresenta a classe *Character* que é base tanto para o implementar a personagem Nonda quanto os predadores. Através da herança, foi criado classes de controller para implementar os estados de animação que incluem: *Idle*, *Range*, *Search* e *Attack*. O estado *Idle* deixa o predador sem movimento (inativo) por alguns segundos. O estado *Search* busca pelo jogador baseado no estado *Range* que verifica se existe o jogador próximo dele para que possa habilitar o estado *Attack*, que efetiva o estado de atacar o jogador.

O Controller dos predadores tem estrutura para construir inúmeras *waves* de predadores diferentes, isso através do trecho de código representado na Figura 18.

```
[System.Serializable]
public class Wave{
    public string name;
    public Transform enemy;
    public int count;
    public float rate = 2f;
}
public Wave[] waves;
```

Figura 18 - Código que estrutura criação de Waves

A classe *Wave* possui quatro parâmetros que são setados dentro do Unity Editor e são armazenados dentro de cada posição do array *Wave[] waves*. Com isso a criação e adição de novos predadores fica flexível além de facilitar na manutenção e refatoração do código em versões futuras.

6 APRESENTAÇÃO DO JOGO

Para mostrar os resultados obtidos, essa seção apresenta o jogo com imagens sequenciais do jogo descrevendo o que cada tela representa.

É mostrado na Figura 19 o *Splashscreen* do jogo com uma pequena animação e um apelo textual para o jogador sobre o que será tratado o jogo.

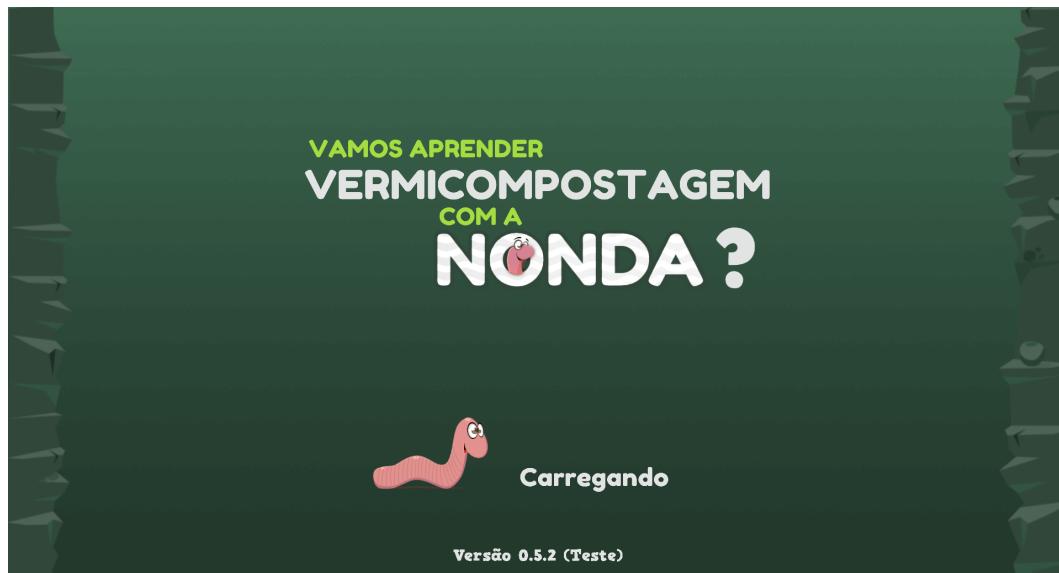


Figura 19 – Tela Splashscreen

Na tela principal do jogo, (Figura 20) é possível escolher entre responder Quiz (ícone com símbolo de interrogação) ou Jogar (símbolo representado por um triângulo). O botão Configurações, representado pelo símbolo da engrenagem, mostra informações dos criadores do projeto. O botão menor (representado pela letra xis), sai do jogo.

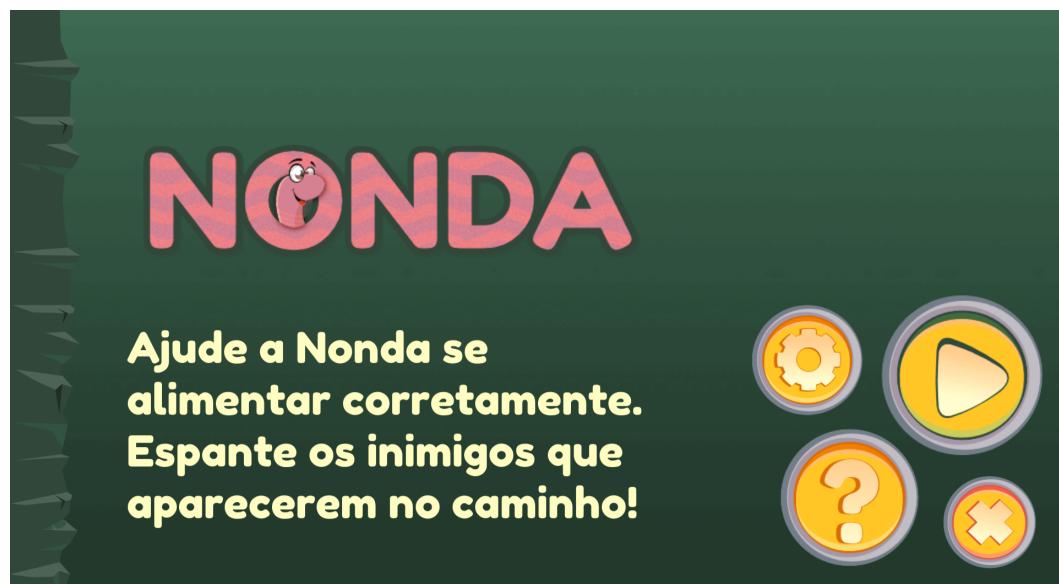


Figura 20 - Tela Principal: Menu do Jogo

Primeiro, na tela do Quiz (Figura 21), aparecem perguntas sobre aulas de Vermicompostagem como forma de fixar conteúdo. Essa tela possui 2 botões de interação: "Correto" e "Errado". Através desses botões é gerado uma pontuação baseado no conhecimento do jogador. Há também um terceiro botão que reseta o a maior pontuação caso tenha competição entre alunos para quem acertar mais perguntas. São 40 perguntas sorteadas aleatoriamente.

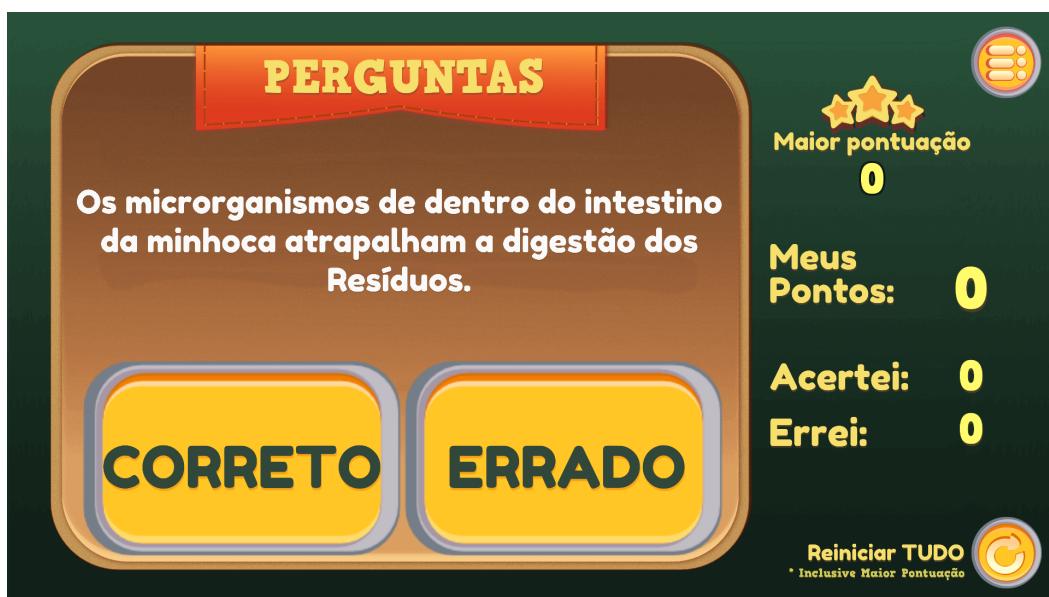


Figura 21 – Tela do Quiz

O jogo Nonda desafia o jogador a conseguir mais pontos em menos tempo, coletando o máximo de itens saudáveis ao longo de 10 fases com o nível de dificuldade aumentando a cada fase. Como descreve o GDD, o jogador pode passar de fase de duas formas: coletar o mínimo de pontos e se aproximar da estrela que aparece no fim da fase ou esperar o tempo acabar coletando o máximo de pontos possíveis. Na Figura 22 é mostrado o *gameplay* do jogo que se passa em um ambiente só, em uma vermicomposteira. O jogo possui 5 itens para coletar; os demais itens que aparecem não são saudáveis, portanto o jogador deve evitá-los.

Para coletar um item, é preciso tocar em cima do item até retirar toda energia contida na barra que fica na parte superior do item. Essa barra indica também se o item pode ser coletado ou não: se for verde ou amarelo, o jogador pode coletar sem perder pontos, caso contrário, perderá pontos. A implementação dessa característica atende a afirmação de Steuer, citada no tópico 2.1 – Serious Games para interagir e envolver.

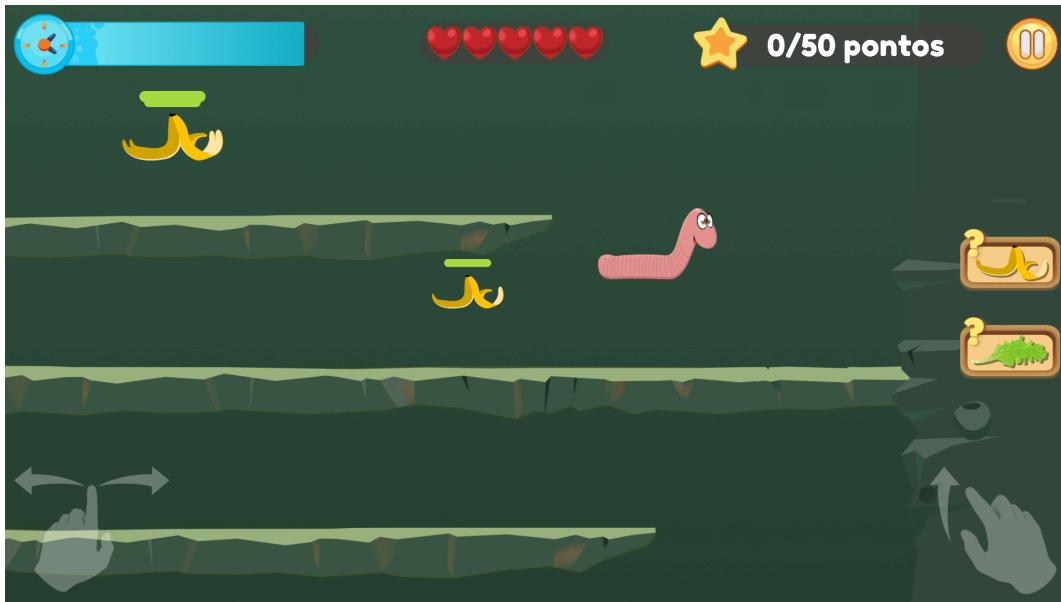


Figura 22 - Gameplay do Jogo

De acordo com a mecânica do jogo, cada fase possui um tempo diferente para acabar. Após o término, é mostrado o *Level Clear* (também chamado de *EndScreen*), ou a tela de aproveitamento da fase, mostrado na Figura 23. Essa tela consiste em mostrar ao jogador o rendimento em pontos e qual os próximos itens a serem coletados na fase seguinte. Caso o jogador não acumule os pontos suficientes, há duas opções: Reiniciar a fase ou sair para o menu principal.



Figura 23 - Tela EndScreen do jogo Nonda mostrando pontuação e desbloqueios de coletáveis

Durante o jogo, é possível acessar ao menu lateral (representado por um retângulo com um símbolo de interrogação) para obter informações sobre os da fase atual. Nas fases que

apresentar predadores, também é informado qual o perigo que esse predador pode causar ao jogador. Abaixo, na Figura 24 é mostrado a informação de um item clicado através do menu direito (retângulo com bordas arredondadas e símbolo de interrogação).

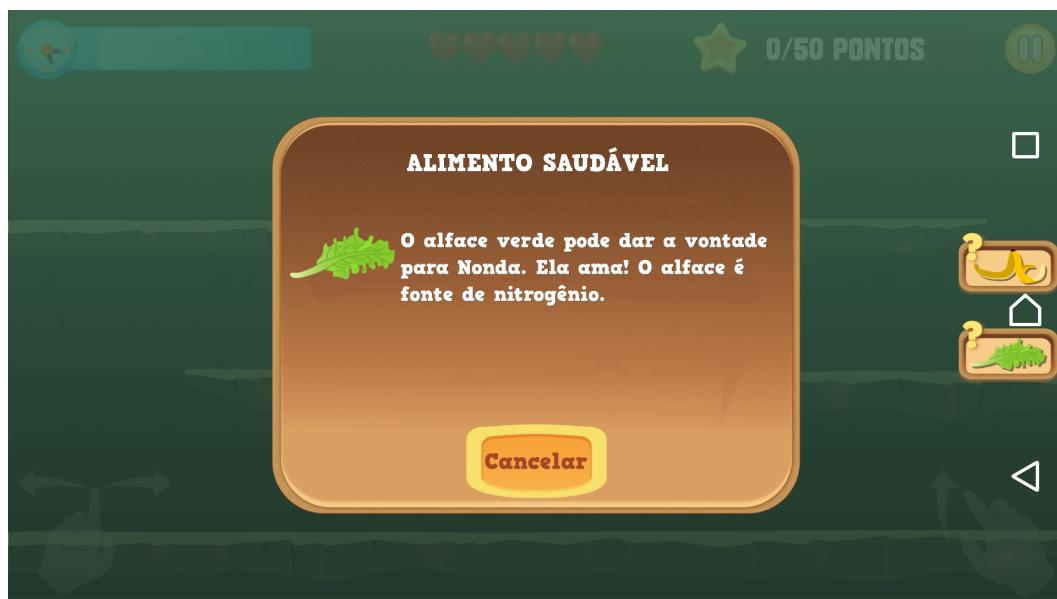


Figura 24 - Tela com informação sobre um item coletável

O jogo possui três predadores, como mostrado nas Figura 8, Figura 9 e Figura 10 e cada aplica dano diferente no personagem Nonda, indicado na tabela, localizada em "Os predadores" do Anexo A. Para afastar o predador é necessário tocar sobre ele e, caso o predador atacar (tocar na Nonda), use o movimento slide para um dos lados ou para cima para pular e evitar danos. Caso o jogador tome 5 danos, o jogo termina, levando à tela *Level Clear*.

7 CONSIDERAÇÕES FINAIS

Espera-se que esse trabalho contribua de forma positiva para o Programa de Recursos Educacionais Digitais e fortifique o laço com Acordo de Cooperação Técnicas entre a UTFPR Câmpus Curitiba e as secretarias de Educação dos municípios de Piraquara, São José dos Pinhais, Pinhais e Curitiba para promover o jogo desenvolvido juntamente com a criação de cartilhas e conteúdos digitais disponibilizados gratuitamente, em específico, às instituições de ensino fundamental. O resultado desse projeto também será incorporado pelo Programa Jogada Certa – coleta Seletiva da UTFPR, como recurso de capacitação de funcionários dos serviços gerais no processo de gestão de resíduos sólidos.

O propósito do presente trabalho foi distribuir um jogo educacional que integrasse a abordagem do conteúdo aprendido com a tecnologia de ensino para que além de familiarizar os alunos no meio de recursos educacionais digitais instigasse uma progressão lógica de aprendizado e um autocontrole.

Embora este trabalho tenha mostrado que é importante saber quais resíduos sólidos orgânicos separar e reaproveitá-los dentro do minhocário, existem limitações em termos técnicos como não mostrar o processo de construção de um vermicomposteira, a escolha de qual espécie de minhoca é adequada para a criação de adubos, os cuidados para manter o minhocário livre de mal odor e de predadores. Em termos tecnológicos, a limitação reside no fato de o aplicativo final ser distribuído apenas para dispositivo Android.

REFERÊNCIAS

- ABT, CLARK C. *Serious Games*. UNIVERSITY PRESS OF AMERICA, 1987.
- ADAMS, ERNEST. Fundamentals of Game Design 3th Edition. 2010. Disponível em: <<https://books.google.com.br/books?isbn=0133435717>> Acesso em 15 nov. 2016 23:47
- AFFINITY DESIGNER, 2016. Affinity Designer: the perfect tool for UI and UX design. Disponível em: <<https://affinity.serif.com/en-us/ui-design-software/>> Acesso em: 24 nov. 2016 10:55
- ALVES, LYNN R. G, MINHO, MARCELLE R. S, DINIZ, MARCELO V. C. Pimenta Cultural 2014. **Gamificação: diálogo com a educação**. Disponível em: <http://www2.dbd.puc-rio.br/pergamum/docdigital/PimentaCultural/gamificacao_na_educacao.pdf> Acesso em: 23 fev. 2016 16:30
- BIGG SHARK. Why Using C# with Unity is Better Than Using Boo or JS for Your Mobile Game, 2015. Disponível em: <<http://biggshark.com/why-using-c-with-unity-is-better-than-boo-and-js-for-your-next-mobile-game/>> Acesso em: 03 fev. 2016 16:35
- BRASIL. MINISTÉRIO DO MEIO AMBIENTE (MMA). Conferência das Nações Unidas sobre Meio Ambiente e Desenvolvimento (1992: Rio de Janeiro, RJ), 471 p. Capítulo 28, p. 381-382. Disponível em: <<http://www.mma.gov.br/responsabilidade-socioambiental/agenda-21/agenda-21-global>>. Acesso em 15 out. 2016 10:35
- BRASIL. MINISTÉRIO DO MEIO AMBIENTE (MMA). Gestão de Resíduos Orgânicos. Disponível em: <<http://www.mma.gov.br/cidades-sustentaveis/residuos-solidos/gestao-de-residuos-orgnicos>> Acesso em 15 out. 2016 19:40
- BRASIL. Senado Federal. Agenda 21: Conferência das Nações Unidas sobre Meio Ambiente e Desenvolvimento (CNUMAD). Brasília: SF, 2001.
- Bruner, J. S. (1972), “Nature and uses of immaturity”, American Psychologist, Vol. 27, No. 8, In Bruner, J. S., Jolly, A. and Sylva, K. (eds.) (1976), Play. Its role in development and evolution. Penguin Books, New York.
- CORTI, K. (2006). Games-based Learning; a serious business application. PIXELearning Limited. <http://www.pixelearning.com/docs/games_basedlearning_pixelearning.pdf> Acesso em: 04 mar. 2016 09:35
- COSTA, E. DIAS. Toptal Developers. Unity with MVC: How to Level Up Your Game Development Disponível em: <<http://www.toptal.com/unity-unity3d/unity-with-mvc-how-to-level-up-your-game-development>> Acesso em: 24 fev. 2016 14:40
- DAMANI, B., SARDESHPANDE, V. & GAITONDE, U. J. Comput. Educ. (2015) 2: 493. doi:10.1007/s40692-015-0045-y Acesso em 08 jul. 2016 15:45

DESENVOLVEDOR DE JOGOS, 2016. Coroutines no Unity 3D Disponível em: <<http://www.desenvolvedordejogos.com/2016/09/16/coroutines-no-unity-3d-parte-1/>> Acesso em: 14 abr. 2016 13:22

DJAOUTI, D. et al. "A Gameplay Definition through Videogame Classification" International Journal of Computer Games Technology, vol. 2008, Article ID 470350, 7 pages, 2008. doi:10.1155/2008/470350 Acesso em 12 ago. 2016 16:34

FABRICATORE, C. **Gameplay and Game Mechanics Design.** Gameplay and Game Mechanics Design: A Key to Quality in Videogames. Disponível em: <<http://www.oecd.org/edu/ceri/39414829.pdf>> Acesso em: 04 mar. 2016 17:41

GIANT BOMB. Endless Runner. Games where the Player character is constantly running. 2015. Disponível em: <<http://www.giantbomb.com/endless-runner/3015-7179/>> Acesso em: 07 jan. 2016 00:35

GOOGLE WEB DESIGNER HELP. The Sprite Sheet component. Diposnível em: <<https://support.google.com/webdesigner/answer/6317541?hl=en>> Acesso em: 12 nov. 2016 20:06

INNOVATION HOUSE RIO (IHR). Disponível em: <<https://innovationhouserio.wordpress.com/2015/09/03/brazilian-gaming-studios/>> Acesso em: 22 jan. 2016 16:05.

J. XIE, "Research on key technologies base Unity3D game engine," *Computer Science & Education (ICCSE), 2012 7th International Conference on*, Melbourne, VIC, 2012, pp. 695-699.

doi: 10.1109/ICCSE.2012.6295169 Acesso em: 20 mar. 2016 20:45

JAMES, M. Scrum Reference Card. Disponível em: <<http://scrumreferencecard.com/scrum-reference-card/>> Acesso em: 24 fev. 2016 15:20

JASPER, F. Frames Per Second – Measuring Performance. 2015. Disponível em: <<http://catlikecoding.com/unity/tutorials/frames-per-second/>> Acesso em: 15 dez. 2016 22:50

KANSTENSMIDT, C. Revistas eletrônicas. **Os impactos das Tecnologias dos Jogos Digitais Multijogadores na Jogabilidade Social.** Disponível em: <<http://revistaseletronicas.pucrs.br/ojs/index.php/famecos/article/view/7789/5531>> Acesso em: 23 fev. 2016 5:45

LEMES, David de Oliveira. ABRELIVROS. **Artigo: Serious Games – Jogos e Educação.** Disponível em: <<http://www.abrelivros.org.br/home/index.php/bienal-2014/resumos-e-fotos/5647-primeiro-resumo>> Acesso em: 23 fev. 2016 15:11

LÉVY, P. (1999) "Cibercultura", São Paulo SP ed.34.

Z. LI E E. KRAEMER, "PROGRAMMING WITH CONCURRENCY: THREADS, ACTORS, AND Coroutines," *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013 IEEE 27th International*, Cambridge, MA, 2013, pp. 1304-1311. doi: 10.1109/IPDPSW.2013.193 Acesso em: 05 nov. 2016

LOURENCO, NELSON. **Manual de Vermicompostagem e Vermicultura para Agricultura Orgânica**, 2014. Disponível em: <<https://goo.gl/9sNeiW>> Acesso em: 04 fev. 2016 13:15

MASOVER, S. IST-SIS. 2014. **Model-View-Controller: A Design Pattern for Software** Disponível em: <<https://ist.berkeley.edu/as-ag/pub/pdf/mvc-seminar.pdf>> Acesso em: 24 fev. 2016 12:00

MURRAY, J. Hamlet no Holodeck: o futuro da narrativa no ciberespaço. São Paulo: Itaú Cultural, 2003.

NDEGWA, P. M., and S. A. THOMPSON. "Integrating composting and vermicomposting in the treatment and bioconversion of biosolids." *Bioresource technology* 76.2 (2001): 107-112.

NUERNBERG, ANA CLAUDIA. Vermicompostagem: estudo de caso utilizando resíduo orgânico do restaurante universitário da UTFPR Câmpus Curitiba/Sede Ecoville. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/handle/1/3911> Acesso em: 23 jan. 2016 11:30.

PANORAMA DOS RESÍDUOS SÓLIDOS NO BRASIL, 2015. Disponível em: <<http://www.abrelpe.org.br/Panorama/panorama2014.pdf>> Acesso em: 21 jan. 2016 17:15.

PRENSKY, M. Digital Game-Based Learning. St. Paul: Paragon House, 2001.

PRENSKY, M. Digital Game-Based Learning, McGraw-Hill Pub. Co., 2004

PULSIPHER, L. Glossary for Game Designers. 2010. Disponível em: <<http://pulsiphergames.com/glossaryforgamedesigners.pdf>> Acesso em: 13 nov. 2016 21:45

SAMPAIO, H. **Jogos casuais tomam conta da Indústria**. Disponível em: <<http://jogos.uol.com.br/reportagens/ultnot/2008/06/24/ult2240u131.jhtm>> Acesso em: 02 mai. 2016 17:31

SPECOUT BY GRAPHIQ, 2014. QUALCOMM SNAPDRAGON MSM 8255. Disponível em: <<http://system-on-a-chip.specout.com/1/264/Qualcomm-Snapdragon-MSM8255>> Acesso: 10 fev. 2016 22: 02

STEUER, J. Department of Communication, Stanford University. 1993. Defining Virtual Reality: Dimension Determining Telepresence. Disponível em: <<http://www.cybertherapy.info/pages/telepresence.pdf>> Acesso em: 23 fev. 2016 23:10

T. PAGOTTO, J. A. FABRI, A. LERARIO AND J. A. GONÇALVES, "Scrum solo: Software process for individual development," *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*, Gran Canaria, 2016, pp. 1-6.
doi: 10.1109/CISTI.2016.7521555 Acesso em: 03 nov. 2016 21:45

TAROUCO, L. M. R.; FABRE, Marie-Christine Julie Mascarenhas ; ROLAND, Letícia Coelho; KONRATH, Mary Lúcia Pedroso . **Jogos educacionais**. RENOTE. Revista Novas Tecnologias na Educação, Porto Alegre, v. 2, n. 1, p. 1-7, 2004.

The Best Development Platform for Creating Games. Disponível em: <<https://unity3d.com/unity>> Acesso em: 03 fev. 2016 16:00.

TULLEKEN H. How are puzzle games designed? (Introduction). Disponível em: <<http://devmag.org.za/2011/04/16/how-are-puzzle-games-designed-introduction/>> Acesso em: 10 jan. 2016, 13:45.

UNITY (GAME ENGINE). Disponível em: <[http://research.omicsgroup.org/index.php/Unity_\(game_engine\)](http://research.omicsgroup.org/index.php/Unity_(game_engine))> Acesso em: 12 set. 2016. 21:40.

YONGLEI Tao, "Component- vs. application-level MVC architecture," *Frontiers in Education*, 2002. FIE 2002. 32nd Annual, 2002, pp. T2G-7-T2G-10 vol.1.

Y. ZHANG ET AL. "Data-Driven Synthesis of Cartoon Faces Using Different Styles," in *IEEE Transactions on Image Processing* , vol.PP, no.99. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7742963&isnumber=4358840>> Acesso em: 12 nov. 2016. 20:35.

ANEXO A – GAME DESIGN DOCUMENT DO JOGO NONTA