**Pontifícia Universidade Católica de Minas Gerais**

***Campus* Poços de Caldas**

**Curso de Ciência da Computação**

**Projeto Get Out!**

**Versão 1.0**

Alexandre Peluchi de Oliveira Junior

Poços de Caldas

2/2020

**1. Introdução**

Desde a minha infância uma das formas de entretenimento e diversão que tive foi jogar video-game, incontáveis dias e fins de semana, meu pai chegava do trabalho e trazia um novo jogo e passávamos horas e horas nos divertindo e explorando aquele novo mundo.

            Apesar do conhecimento que tenho em desenvolvimento de software, em diversas linguagens e ter testado inúmeras game engines como Unity 3D, RPG Maker, Construct 3D, GameMaker Studio, Unreal Engine de todas essas a engine que trouxe de volta minha vontade de desenvolver jogos foi a Godot.

            Godot é uma engine open-source lançada em 2014, que tem como plataformas alvo PC, mobile e web. O jogo a ser desenvolvido é um zelda-like, com tema medieval e de mundo aberto.

            Graças a este engine o desenvolvimento do jogo que sempre tive vontade de desenvolver o “Get Out”, onde um meteorito cai em um reino que se torna um prisma, que gradualmente vai ganhando cada vez mais poder e fazendo com que os medos e pesadelos das pessoas se tornem realidade.

* 1. **Objetivos**

**1.1.1 Objetivos do Projeto**

O objetivo do projeto é o desenvolvimento de um primeiro game utilizando a Godot Engine, para conhecimento de tudo que a ferramenta de motor gráfico é capaz e um conhecimento mais amplo sobre a linguagem GDScript que é baseada na linguagem de programação Python.

**1.1.2 Objetivos do Produto**

O produto tem o objetivo de demonstrar, que apesar de ser um jogo 2D, top-down (qualquer jogo que ofereça um ponto de vista elevado acima da ação pode ser considerado como tendo uma perspectiva top-down/de cima para baixo) e utilizando pixel-art ele pode ser inovador, divertido e trazer temas da atualidade.

* 1. **Conceito**

“Seus medos, podem se tornar realidade.”

**1.2.1 Gênero**

Ação/Aventura e Survival-Horror.

**1.2.2 Classificação etária**

Everyone 10+ (E10+) - Contém conteúdo considerado impróprio para menores de dez anos.

**1.2.3 Trama Principal**

A ideia principal do jogo é um mundo medieval, que está cheio de monstros que tomam vida a partir dos medos e pesadelos das pessoas deste universo, o objetivo do personagem é encontrar a fonte que está causando tudo isso.

**1.2.4 Look & Feel**

A aparência do jogo é pixel arte, com uma “pegada” mais sombria inspirado no jogo **Castlevania: Symphony of the Night**.

**1.2.5 Visão geral do conceito do jogo**

O jogo que se passa em um mundo medieval pacífico sem criatura mitológicas e de fantasia, somente populado por humanos e animais, onde em uma certa noite um meteorito cai em uma vila de camponeses afastada do reino, e começam a surgir boatos do surgimento de monstros nesse local, a cada dia que passa este prisma ganha ainda mais força, fazendo com que vilas mais distantes comecem a ser atacadas por monstros também.

**2. História e Narrativa**

**2.1 Visão geral da história**

**2.1.1 Mundo do jogo**

O jogo se inicia em uma vila pacifica próxima do reino, onde camponeses criam animais e plantam seus próprios alimentos, boatos começam a surgir que a vila de seus familiares estão sendo atacadas por monstros, a uma convocação no castelo onde o rei conta para todos, as notícias que recebeu e que enviou seus melhores soldados e somente um retornou.

A cada nova vila que seu personagem chega mais sombrio fica o universo, mais destruído estão as casas, mais animais são encontrados sem vida, mais queimadas e mais escuro são os locais, constantemente encontram pessoas fugindo pelas estradas, porque cada vez mais se aproxima esse caos do reino.

**2.1.2 Personagens**

Jack (protagonista) é um camponês, casado com Louise e pai de Billy e Clara, antigamente Jack era líder do exército Rei Robert, mas devido a um ferimento por salvar a vida do rei na batalha da ponte estreita, se aposentou para levar uma vida pacata, graças a está infelicidade recebe ajudas mensais do rei como gratidão é considerado um herói e reconhecido por todos do reino.

Rei Robert é considerado por todos o melhor rei da história de Gran Heaven, um reino com abundância em recursos, terras férteis e pacifica após a vitória da batalha da ponte estreita, onde em desvantagem de exército de mil para um, foi vencida graças as estratégias de combate eficazes do Rei Robert.

Monstros do pesadelo, diversos monstros frutos da imaginação dos NPC’s.

Personagens estão ainda em desenvolvimento.

**2.1.3 Eventos e Ações**

O reino de Gran Heaven se tornou, pacifico, após centenas de anos de batalhas contra o reino de Sapekado e seu exército das chamas, foi uma batalha lendária onde em desvantagem de mil para um e graças a liderança, conhecimento e confiança de seu novo no rei, o exército de Gran Heaven utilizando com extrema sabedoria todo seu território a seu favor, venceu a batalha. Os reis assinaram um acordo para tornar toda a região pacifica, o rei Robert concordou em compartilhar recursos que são abundantes com o reino de Sapekado que possui escassez, com isso ambos os reinos vivem e trabalham em conjunto.

O objetivo do personagem principal é ir para a vila onde caiu o meteorito, a exata vila onde moram os pais dele (Jack) e de sua esposa (Louise), para traze-los com vida para o reino de Gran Heaven que fica distante dos assombrosos acontecimentos e como missão secreta, a pedido do Rei Robert que promete proteger sua família, enviar informações do que está acontecendo e como derrotar o surgimento desses monstros.

**2.2 Visão geral da narrativa**

**2.2.1 Integração**

Todo o jogo será desenvolvido por mim, os pixels artes, trilhas sonoras, puzzles, movimentação, combate, utilizando ferramentas gratuitas ou open-source e outros jogos como inspiração.

Será possível combater inimigos, com espadas, arco e flecha, escudos, lanças, empurrar itens, coletar itens, utilizar itens como chaves em portas.

**2.2.2 Estrutura e Progressão**

Como é um mundo aberto a história do jogo será contado, a partir de interações com NPC’s e cutscenes.

**2.2.3 Roteiro**

A ser desenvolvido.

**2.2.4 Tutoriais e manual do jogo**

Os tutorias serão apresentados ao jogador de forma sutil, onde diálogos de NPC’s e cenas do jogo serão mostradas fazendo com que o jogador entenda como utilizar as mecânicas do jogo ao seu favor.

**2.3 Cenas e sequências cinemáticas**

**3. Gameplay**

**3.1 Imersão**

Para manter a atenção dos jogadores terão diferentes armas, inimigos, cenários cada vez mais sombrios, NPC’s cada vez mais profundos e mecânicas diferentes.

**3.2 Estrutura de missões e desafios**

A missão principal do jogo é descobrir o que está causando todo este caos e impedir isso de alguma forma, as quests principais e secundárias serão recebidas conversando com NPC’s.

**3.3 Objetivos do jogo**

Existem diferentes vilas que são abertas para exploração e em cada uma delas o jogador deve descobrir o NPC, que está tendo pesadelos e fazendo seus medos se tornarem realidade eliminar o NPC ou ajudá-lo a superar isso dialogando ou com itens.

**3.4 Fluxo de jogo**

O jogador não terá nenhuma habilidade, mas descobrirá mecânicas novas para vencer os inimigos que são cada vez mais fortes.

**3.5 Mecânica do jogo**

**3.5.1 Regras implícitas e explícitas do jogo**

Os inimigos andam aleatoriamente pelo mapa causando caos por onde passam, colidem com objetos.

**3.5.2 Física**

Itens de missões podem ser pegos, assim como vidas. Inimigos quando causam danos ao jogador infligem knockback.

**3.5.3 Movimentação dos personagens**

Jogador principal pode andar para todas as direções, empurrar objetos, capturar objetos e itens e os NPC’s geralmente estão parados para darem missões, enquanto inimigos tem movimentação aleatória.

**3.5.4 Objetos**

Corações podem ser coletados para aumentar a vida, são encontrados no mapa ou dropados ao eliminar inimigos.

Chaves são coletadas para abrir portas as chaves são universais.

Portas podem ser abertas.

Salas cheias de inimigos, fecham e são abertas somente quando todos os inimigos são eliminados, etc.

**3.5.5 Gatilhos e ações**

Todos objetos são pegos ao se passar por cima deles.

Objetos são utilizados automaticamente quando o jogador se aproxima deles, portas são abertas sempre que jogador possui chave.

NPC’s fornecem informações que fazer liberar o uso de certos objetos, você não consegue utiliza-los sem conhecimento.

**3.5.6 Mecânica de combate**

Combates são vencidos quando o inimigo recebe muitos ataques e tem a vida zerada.

**3.5.7 Economia e mecânica de troca**

Itens são dropados de inimigos e podem ser adquiridos a partir de diálogos com NPC’s.

**3.6 Projeto de Fases (Level Design)**

A ser desenvolvido.

**3.7 Projeto de Interface**

**3.7.1 Sistema Visual**

a) Na HUD são exibidos os elementos de vida, chaves, arma.

b) A ser desenvolvido.

c) Câmera top-down.

**3.7.2 Sistema de controle**

Mouse e teclado interagem com os menus e interface do jogo.

**3.7.3 Fluxo de telas**

a) Menu Principal, Opções, Carregar Jogo Salvo, Som, Vídeo, Controles, Sair do Jogo.

b) A ser desenvolvido.

**3.7.4 Opções do jogo**

O jogo não terá como selecionar dificuldade.

**3.7.5 Saving e Checkpoints**

Jogo pode ser salvo pelo jogador a qualquer momento, a menos que esteja em combate ou em dialogo com NPC’s.

Após mudar de tela, conversas com NPC’s, finalização e recebimento de quests o jogo será salvo automaticamente.

Checkpoints são feitos no último jogo salvo.

**3.7.6 Sistema de ajuda (Help)**

O jogador terá um diário, que estará salvo tudo que foi feito e aprendido no jogo e quais as consequências.

**3.8 Projeto de Som (se houver)**

**3.8.1 Descrição geral**

Sons de itens são executados ao ser utilizados e do ambiente serão executados em loop.

**3.8.2 Músicas**

a) Músicas ambiente são executadas em loop.

b) Música de ações são executadas uma única vez.

c) Música de Vitória são executadas uma única vez.

d) Música de derrota são executados em loop.

**3.8.3 Efeitos Sonoros**

a) Sons ambiente são executadas em loop.

b) Sons de armas são executadas quando utilizadas.

c) Sons da interface são executados uma única vez.

d) Sons de objetos são executados quando utilizados.

**3.9 Sistema de Inteligência Artificial**

**3.9.1 Oponentes**

Alguns tem movimentação aleatória, alguns perseguem o personagem ao se aproximar e outros persegue o personagem ao serem atacados.

**3.9.2 Inimigos**

Chefes e mini chefes têm padrões de ataques.

**3.9.3 Personagens não combatentes (NPC’s neutros)**

Praticamente todos os NPC’s ficarão parados em um local do mapa e poderá dialogar com o personagem principal.

**3.9.4 Personagens amigáveis**

Se as quests de determinados NPC’s forem finalizadas eles poderão dar informações essenciais para o combate e descoberta do NPC que tem os pesadelos de determinada vila.

**3.9.5 Inteligência Artificial auxiliar**

(Ver o código do jogo)

**4. Arte**

**4.1 Guia de cores e estilos gráficos**

O visual do jogo será um pixel art 8-bits, com paleta de cores de cores mais fortes e vibrantes no inicio do game e ao decorrer do jogo as cores irão escurecendo, até se tornar 1-bit pixel art.

**4.2 Arte conceitual**

Personagens a serem desenvolvidos.

**4.3 Asset List**

Aspectos dos modelos a serem desenvolvidos.

**5. Documentação Técnica**

**5.1 Requisitos de sistema**

Por se tratar de um jogo utilizando pixel art, este são os requisitos.

**Mínimo**:

Memory:4 GB

Graphics Card: NVIDIA GeForce 6200

CPU: Intel Core 2 Duo E8400

OS:Windows

**Recomendado**:

Memory:4 GB

Graphics Card: NVIDIA GeForce 8200M G

CPU: Intel Core 2 Duo E8400

OS:Windows

**5.2 Engenharia de software**

A ser desenvolvido.

**5.3 Game Engine**

**Godot**, trata-se de uma engine de código-aberto, que possui diversos pontos fortes e também alguns fracos:

**Prós**

* O layout da Godot é bem intuitivo, tornando fácil de entender o seu trabalho. Além disso, tudo fica muito claro independentemente do projeto ser em 2D ou 3D.
* Ela possui uma linguagem própria, chamada de “GDScript”. Uma vez que você domina essa linguagem nativa da ferramenta, você será capaz de criar novas ferramentas no editor para qualquer coisa que precisar. Pode ser muito útil para projetos mais complexos.
* É open-source (código-aberto). Se em algum momento você sentir alguma limitação do próprio mecanismo, bastará escrever uma extensão em GDScript ou C++ para adicionar alguma alteração no editor.
* A licença totalmente gratuita torna o projeto seu. A Unity ou Unreal, por exemplo, possuem versões gratuitas, mas o mecanismo em si continuará sendo deles. Para todos os efeitos, é como se o Godot fosse uma engine sua e ninguém pode te dizer o que deve ou não fazer.
* Caso você já seja um programador experiente, esse ponto será importante: o depurador da Godot é fantástico. É poderoso e fácil de usar.

**Contras**

* A Godot pode ser muito difícil para iniciantes. O fato dela ser uma engine open-source te exigirá já possuir certos conhecimentos em programação para realizar diversas ações e não ficar limitado.
* A comunidade ainda é muito pequena. Como falei no início do artigo, esta é uma engine nova e é normal que ainda tenha uma comunidade descentralizada na internet. Alternativamente, você pode usar o grupo do Facebook da Godot Engine para tirar suas dúvidas.
* A capacidade gráfica da Godot ainda é limitada em comparação a outras engines. O renderizador atual é escrito para ser compatível com uma versão antiga do OpenGL, para suporte web e mobile, portanto, a capacidade de renderização fica limitada.
* A engine é atualizada e melhorada constantemente. Isso é bom? Claro! Mas também te forçará a sempre estar tendo cuidado com a compatibilidade ao atualizar versões – para que seu projeto não se corrompa por algum motivo.

**5.4 Programação**

Godot utiliza as linguagens de programação GDScript que é baseada em Python muito simples e intuitiva e recentemente C#, mas devido existir mais perguntas, documentação detalhada em GDScript, está foi a linguagem utilizada.

**6. Gerenciamento e análise do projeto**

**6.1 Organização inicial do projeto**

O jogo está sendo desenvolvido por uma única pessoa, Alexandre Peluchi.

**6.2 Escopo do projeto**

O jogo se inicia em uma vila bem próxima ao reino, a cada nova vila tudo será diferente, os inimigos, ambientação, NPC’s puzzles, mecânicas de jogos e bosses.

A cada nova vila o jogo será mais difícil, sombrio e com puzzles mais complexos, o personagem não tem habilidades especiais, ele é simplesmente um antigo guerreiro e protetor do Rei.

**6.3 Recursos**

**Hardware**:

Ryzen 5 3600

16gb Ram DDR4 3000Mhz

RX 570 4GB

Windows 10

**Software**:

Godot Engine

Equipe:

Alexandre Peluchi

Para executar os testes do projeto, precisa simplesmente de um computador com processador com 2 cores e placa de vídeo integrada.

**6.4 Entregas**

Primeiro está sendo desenvolvido todo o sistema de combate, recuperação de vida, interação com itens, movimentação, IA dos NPC’s e inimigos, etc. Para aí sim a história e roteiro serem escritos e iniciar de fato o desenvolvimento.

**6.5 Premissas**

Inexistente.

**6.6 Restrições**

Falta de tempo, devido ao fato de o jogo estar sendo desenvolvido após horário de trabalho.

**6.7 Cronograma**

Inexistente.

**6.8 Estimativa de custo**

Todas as ferramentas e cursos para aprendizado são gratuitos.

**6.9 Limites do projeto**

Open World

Combate em tempo real

Cutscenes

Interação com itens

Puzzles

Survival Horror

**6.10 Processos**

Kanban em especifico o gerenciador de projetos do GitHub do próprio repositório.

**6.11 Análise de riscos**

O principal risco identificado é o aumento do escopo inicial do projeto, fazendo com que não seja possível finalizar tudo, pois o projeto está sendo desenvolvido por uma só pessoa.

**6.12 Viabilidade técnica**

**6.13 Viabilidade econômica**

Todas as ferramentas são gratuitas, ou seja, custo zero.

**6.14 Plano de testes e controle de qualidade**

Lançar uma demo no Google Play Store para verificar possíveis bugs e se o projeto como um todo terá um feedback positivo.

**6.15 Ferramentas e técnicas**

**7. POST MORTEM (Resultados, conclusão e considerações finais)**

As boas lições observadas foram que as ferramentas por serem gratuitas e algumas open-source, tem mais fonte de informações, melhor documentação o que torna simples de se usarem.

A principal má lição, foi não ter definido um horário fixo para desenvolvimento do projeto, fazendo com que se atrase o desenvolvimento.

**8. Estudo sobre monetização do GDD**

Como a Godot Engine consegue exportar o projeto para as plataformas **PC, mobile e web,** a escolha e principal foco para distribuição do jogo será para dispositivos mobile com foco em Android e se possível IOS.

**Custos para lançar um jogo no Google Play Store**

A publicação de um jogo no Google Play Store tem um custo relativamente. Para se tornar um desenvolvedor, é preciso pagar uma taxa única de US$25. Após isso, o Google cobra uma comissão em cada venda realizada.

Apesar disso, vale a pena colocar um jogo na loja do Google. Afinal, ela possui uma base de usuários em todo o mundo, o que aumenta as chances do conteúdo ser bem vendido. Além disso, existe a possibilidade de monetizar o game com os patrocinadores do Google.

Devido ao modelo de negócio do projeto que será um jogo, primeiramente será desenvolvido um demo do mesmo, para viabilizar certas características do jogo a partir de feedback dos usuários, este sem publicidades.

Ao lançamento final, o jogo inicialmente utilizará **publicidade** que é a maneira mais usual de se ganhar dinheiro, não de maneira forçada para todos aproveitarem o jogo, e também uma **versão paga do jogo** com um valor a ser decidido, este sim sem publicidade. A decisão de se ter o mínimo de publicidade é por causa do gênero do jogo, que terá aspectos de **survival horror** e uma história profunda, a imersão do jogador neste universo é essencial para uma completa experiência.

Os primeiros meses será para correção de bugs, balanceamento sempre ouvindo o feedback dos usuários e após, chegar a uma versão válida e que traga uma experiência de valor, será desenvolvido outras formas de monetização como tornar o jogo **Freemiun,** disponibilizando uma loja venda de itens cosméticos e de relevância para os jogadores.

Se o jogador na versão gratuita comprar qualquer coisa da loja, ele receberá uma chave para desbloquear a versão sem publicidade está é mais uma maneira de motivar os jogadores a adquirirem algo do jogo e contribuírem para o desenvolvimento.

Após o lançamento da loja do jogo, o foco será a distribuição para outras plataformas como para PC e Apple Store.

Para a versão PC o objetivo será colocar o jogo na Steam, para isso é preciso parar uma tarifa no valor de $100.00 USD para cada produto que desejar distribuir na loja. A tarifa não é reembolsável, mas você receberá o valor de volta no pagamento feito assim que o produto alcançar uma receita bruta ajustada de, no mínimo, $1,000.00 USD por meio da Loja Steam e compras no aplicativo.

**Regulamentos e diretrizes para distribuição de jogos na Steam**

Temos algumas diretrizes para o conteúdo que pode ser distribuído pelo Steam. Analise-as atenciosamente antes de prosseguir com a distribuição.

**O que não se deve publicar no Steam:**

1. Conteúdo adulto que não está adequadamente classificado nem possui verificação de idade
2. Declarações caluniosas ou difamatórias
3. Conteúdo do qual não é proprietário ou não possui os direitos adequados
4. Conteúdo que viole as leis de qualquer jurisdição na qual seja disponibilizado
5. Conteúdo que explore crianças de alguma forma
6. Aplicativos que modifiquem os computadores dos clientes de formas inesperadas ou prejudiciais (ex.: vírus, softwares mal-intencionados)
7. Aplicativos que tentem adquirir dados confidenciais, como as suas credenciais do Steam ou dados financeiros (como dados do cartão de crédito), de forma fraudulenta

**Tipos de conteúdo aceitos**

Atualmente aceitamos os seguintes tipos de conteúdo: jogos e experiências de realidade virtual. Softwares que não sejam jogos não serão aceitos pelo Steam Direct. Caso tenha um software que não seja um jogo para distribuição no Steam, consulte a página [Distribuindo softwares que não sejam jogos](https://help.steampowered.com/wizard/HelpWithPublishing?issueid=925).

Além dos itens acima, o conteúdo publicado no Steam deve, obviamente, obedecer ao Acordo de Distribuição do Steam, ao [Acordo de Assinatura do Steam](http://store.steampowered.com/subscriber_agreement/), às diretrizes da [Conduta Online do Steam](http://store.steampowered.com/online_conduct/) e à [Política de Privacidade do Steam](http://store.steampowered.com/privacy_agreement/).

Estas são as formas de monetização escolhidas e que melhor se encaixam ao projeto a curto, médio e longo prazo.