

DOCUMENTO PARA DESIGN DE GAME

Eu, Presidente

Autores:

Arthur Alberto Cardoso Reis

Beatriz Hirasaki Leite

Israel Nunes Lopes Carvalho

Luiz Carlos da Silva Júnior

Marcos Aurélio Florêncio da Silva

Raissa da Silva Sabino

Stefano Tosi Butori

Data de criação: 07 de fevereiro de 2022

Versão: 1.8.0

Controle do Documento

Histórico de revisões

Data	Autor	Versão	Resumo da atividade
07/02/2022	Stefano Tosi Butori	1.0.0	Versão Inicial
10/02/2022	Diversos	1.1.0	Preenchimento dos itens 1.1 a 1.5
17/02/2022	Diversos	1.2.0	Preenchimento dos itens 1.6 e 1.7
23/02/2022	Diversos	1.3.0	Inclusão do value proposition canvas no item 1.6
25/02/2022	Diversos	1.4.0	Preenchimentos dos itens 2, 3, 4 e 6.1
08/03/2022	Diversos	1.5.0	Preenchimento do item 6.2
22/03/2022	Diversos	1.6.0	Inclusão de teoria da mudança no item 1.6
04/04/2022	Diversos	1.7.0	Preenchimento dos itens 5.1 e 6.4
08/04/2022	Diversos	1.8.0	Preenchimento do item 5.3

Sumário

1. Visão Geral do Projeto <ADALOVE – Definir Proposta de Valor>	5
1.1 Objetivos do Jogo	5
1.2 Características gerais do Jogo	5
1.3 Público-alvo	6
1.4 Diferenciais	6
1.5 Análise do cenário: Matriz SWOT	7
1.6 Proposta de Valor: Value Proposition Canvas	8
1.7 Requisitos do Jogo <ADALOVE – Documentar requisitos>	9
1.7.1 Requisitos coletados na entrevista com o cliente	9
1.7.2 Persona	9
1.7.3 Gênero do Jogo	10
1.7.4 Histórias do jogo (Game stories) ou Histórias dos usuários (user stories)	10
1.7.5 Mecânica	10
1.7.6 Fontes de Pesquisa / Imersão	11
2. Game Design <ADALOVE – Elaborar fluxograma do jogo>	13
2.1 História do Jogo	13
2.2 Fluxo do Jogo	13
2.3 O Mundo do Jogo	15
2.3.1 Locações Principais e Mapa	15
2.3.2 Navegação pelo Mundo	15
2.3.3 Escala	16
2.3.4 Ambientação	16
2.4 Base de Dados	16
2.4.1 Inventário	16
3. Level Design	18
3.1 Visão Geral do Jogo	18
3.1.2 Layout do Game	18

3.1.2.1 Connections	18
3.1.2.2 Layout Effects	19
4. Personagens <ADALOVE – Implementar mecânicas básicas do jogo - parte 1 a 5>	20
4.1 Personagens Controláveis	20
4.1.2 Personagens	20
4.1.2.1 Backstory	20
4.1.2.2 Concept Art	21
4.1.2.3 Ações Permitidas	21
4.1.2.4 Momento de Aparição	22
4.2 Common Non-Playable Characters (NPC)	22
4.3 Special Non-Playable Characters (NPC)	24
5. Teste de Usabilidade e Padrões de Qualidade	25
5.1. Teste de Usabilidade <ADALOVE – Desenvolver relatório de resultados do playtest>	25
5.2 Padrões de Qualidade <ADALOVE – Definir Padrões de Qualidade>	26
5.3 Melhoria de eficiência do código <ADALOVE – Identificar oportunidades de melhoria de eficiência do código>	28
6. Relatório - Física e Matemática	30
6.1 Funções <ADALOVE - Definir as funções matemáticas que serão utilizadas no jogo>	30
6.2 Cinemática Unidimensional <ADALOVE - Fazer orientação espacial (2D)>	33
6.3 Vetores <ADALOVE - Aplicar os conceitos matemáticos no jogo>	34
6.4 Cinemática Bidimensional e mais <ADALOVE - Fazer orientação espacial>	35
7. Bibliografias	38
Apêndice	39

1. Visão Geral do Projeto <ADALOVE – Definir Proposta de Valor>

1.1 Objetivos do Jogo

No Brasil, poucas pessoas têm conhecimento sobre política, leis e a Constituição Federal brasileira, dado que esse conteúdo não é trabalhado diretamente no ensino fundamental e médio. Na prática, discussões envolvendo temas jurídicos costumam ocorrer exclusivamente em cursos de nível superior de Direito e Ciências Jurídicas e Sociais.

Nosso jogo visa promover o interesse do público mais jovem por essa temática, essencial para o exercício pleno da cidadania e o conhecimento por cada indivíduo dos seus direitos e deveres. Seu propósito é assegurar que o público mais jovem possa entender o funcionamento das principais instituições políticas brasileiras e dos dispositivos legislativos mais utilizados tais como decretos, Medidas Provisórias (MPs), Projetos de Lei (PLs), Proposta de Emenda Constitucional (PECs) e Leis Complementares (LCs).

Para tanto, o jogador assumirá o papel de presidente do Brasil, em seu primeiro ano de mandato. No decorrer do jogo, o protagonista enfrentará eventos ou crises e deverá optar entre diversas medidas ou ações, que servirão para decidir o rumo da nação. Ele precisará equilibrar suas escolhas para manter a economia saudável, ter apoio do congresso, garantir que sua opinião pública não caia muito e assegurar a segurança nacional. Caso contrário, o mesmo sofrerá um processo de impeachment e perderá o jogo.

Esse projeto será desenvolvido no primeiro módulo do curso de graduação do Instituto de Tecnologia e Liderança (“Inteli”) e gerará um Minimum Viable Product (MVP) que será apresentado e entregue à Organização Não Governamental “Projeto Constituição na Escola”. O código fonte do jogo será integralmente livre e aberto, bem como disponibilizado na plataforma GitHub.

1.2 Características gerais do Jogo

A estrutura gráfica do jogo será baseada em “cards”, que contarão uma história engajante. Cada “card” conterá a descrição de um evento ou acontecimento crítico relacionado à presidência e, em alguns casos, apresentará uma decisão a ser tomada. Em paralelo, o jogador terá acesso a 4 (quatro) indicadores: apoio popular, apoio do congresso, situação econômica e apoio do exército.

Dependendo dos eventos e das escolhas que o jogador adotar, esses indicadores poderão subir ou descer. Caso um dos indicadores alcance um nível crítico mínimo, o jogador terá um determinado período para tentar arrumá-lo. Se não conseguir, sofrerá um processo de impeachment e perderá o jogo.

Também de acordo com as decisões tomadas, poderão existir finais diferentes. Por exemplo, se houver muito apoio do congresso mas pouco apoio popular, o presidente não sofrerá impeachment, mas também não será reeleito.

A estratégia de aprendizado se dará por meio do uso de artefatos que o jogador poderá usar durante o jogo com o objetivo de solucionar problemas e dificuldades, fazendo com que o interesse em solucionar a história guie sua estratégia para chegar até o final, sendo o aprendizado uma consequência natural.

1.3 Público-alvo

O público-alvo será composto por pessoas de 12 a 18 anos, que não conheçam a fundo as leis e a estrutura do Estado Brasileiro e tenham interesse em aprender mais sobre o tema.

Entendemos que esse público não possui um conhecimento aprofundado sobre política e, portanto, está altamente propenso a absorver conceitos fundamentais como Estado de Direito, valores democráticos, princípios civis, dentre outros.

Entendemos que, para esse público, o projeto pode ser explicado de forma lúdica e sem a apresentação de termos técnicos. Isso seria possível porque o jogo irá gerar a compreensão conceitual que será rememorada no futuro, quando os aspectos técnicos forem apresentados ao nosso público-alvo.

1.4 Diferenciais

O projeto busca trazer uma narrativa imersiva que permite ao jogador assumir o papel de presidente do Brasil e tomar decisões importantes sobre o futuro do Brasil no primeiro ano de mandato.

A partir das decisões tomadas será possível apresentar as ferramentas e dispositivos constitucionais de interesse do cliente, facilitando o processo de aprendizagem e captura da atenção do público alvo.

Além disso, também iremos incluir árvores de diálogos e possibilitem ao jogador uma sensação de recompensa a cada turno, independente do resultado específico. Ou seja, não necessariamente o impeachment traria consigo uma sensação de perda, mas sim um final lógico para uma história que estava sendo contada e que, caso o jogador tente de novo, pode levar a outro final lógico e igualmente engajante.

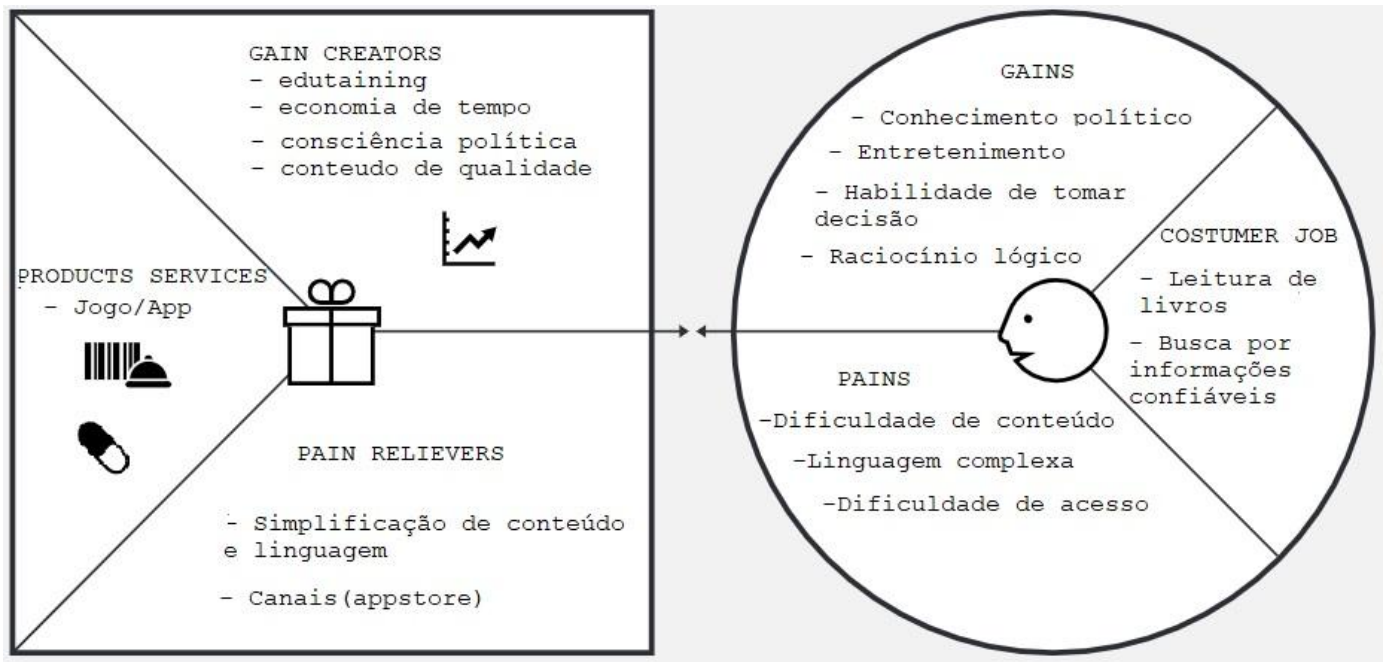
Incluiremos ainda elementos randômicos que trarão uma dimensão de aleatoriedade e novidade ao jogo, o que acreditamos que reforça o desejo de satisfazer a curiosidade acerca do arco literário a ser descoberto.

Serão apresentados obstáculos e decisões que aumentarão de dificuldade gradualmente tornando a história instigante, o que leva os participantes a se interessarem de forma ativa pelo enredo do jogo.

1.5 Análise do cenário: Matriz SWOT

<p>Strengths (Forças)</p> <ul style="list-style-type: none">- Alguns membros do time tem conhecimentos técnicos e conhecimentos de domínio- Cliente motivado em obter resultados impactantes- Apoio de professores experientes no desenvolvimento do game- Pioneirismo do projeto- Equipe com perfis diversos	<p>Weakness (Fraquezas)</p> <ul style="list-style-type: none">- Falta de experiência do time no desenvolvimento de jogos eletrônicos- Poucos assets disponíveis de imediato para o game, será preciso buscar mais- Jogo deverá rodar em um hardware simples, para ser mais acessível
<p>Opportunities (Oportunidades)</p> <ul style="list-style-type: none">- Interesse geral da população mais jovem por jogos eletrônicos pode facilitar sua conscientização sobre política- Edutainment é um setor novo e com grande potencial- Ao final o game poderá ser apresentado a estúdios de games, que poderão fornecer um feedback	<p>Threats (Ameaças)</p> <ul style="list-style-type: none">- Algumas pessoas podem não ter interesse no conteúdo temático do jogo- Competição não convencional (TikTok)- Possibilidade de surgirem outros jogos com a mesma proposta- O apelo de um game baseado em texto pode ser limitado- O conteúdo seria opcional no currículo acadêmico

1.6 Proposta de Valor: Value Proposition Canvas



Teoria da Mudança:

Insumos permitem executar...	Atividades geram...	Produtos permitem alcançar...	Resultados geram...	Impactos
Alunos Desenvolvedores	Levantar requisitos	Jogo educativo sobre a Constituição Federal e Leis do Brasil	Mais conhecimento das Leis brasileiras e da Constituição para a próxima eleição	Aumento no engajamento político dos jovens
Apoio dos Instrutores/Orientadores	Planejar, desenvolver e testar um jogo de computador			Aumento na consciência social e cidadania de crianças e adolescentes
Ferramentas tecnológicas (Godot, GitHub)				

1.7 Requisitos do Jogo <ADALOVE – Documentar requisitos>

1.7.1 Requisitos coletados na entrevista com o cliente

O jogo tem como principal objetivo ensinar, de forma simples e divertida, a constituição brasileira para adolescentes e jovens adultos. O jogo simula o sistema político brasileiro. São 9 personagens, mas apenas o presidente pode ser controlado, com o jogador tomando suas decisões ao longo do seu primeiro ano de mandato. A diferença está na vivência que o jogador terá ao decidir as questões políticas do país. Suas escolhas irão interferir no rumo da história e estarão relacionadas com o dia a dia de um presidente da república.

Requisitos funcionais:

1. O jogo deve ensinar política de forma lúdica.
 2. O jogo deve apresentar decisões.
 3. Os indicadores do jogo devem ser alterados pelas decisões do jogador.
 4. O jogo deve conter o presidente como único personagem jogável.
 5. O jogo deve apresentar múltiplos finais, acessíveis ao completar uma story branch ou ao zerar um dos indicadores.
 6. O jogo deve conter a opção de salvar, bem como de carregar um jogo salvo.
-

Requisitos não funcionais:

1. O jogo deve funcionar em WebGL e Windows.
 2. O jogo deve rodar em 30 fps de maneira estável.
 3. O jogo deve ser desenvolvido na linguagem GDScript, na plataforma Godot.
-

Regras de negócio:

1. O jogo deve conter 9 personagens.
 2. O jogo deve ter uma duração de por volta de 15 minutos.
-

1.7.2 Persona

Bruno tem 17 anos, está no terceiro ano do ensino médio na rede pública de ensino e é natural de Vitória/ES. É extrovertido, comunicativo e conhece a história da indústria de games. Tem interesse em seguir carreira como designer gráfico e tem habilidade em desenhar e criar histórias. Em seu tempo livre, Bruno gosta de jogar futebol e videogames. Gosta também de navegar pelas redes sociais, especialmente no Youtube. Bruno não gosta de jogos

entediadas, com muitos diálogos de difícil compreensão. É um jogador casual, portanto, não tem muito tempo para dedicar aos jogos.

1.7.3 Gênero do Jogo

Trata-se de um jogo de *estratégia*, baseado em escolhas feitas pelo jogador.

1.7.4 Histórias do jogo (Game stories) ou Histórias dos usuários (user stories)

Game Stories					
ID	Épico	Prioridade	Tamanho	Previsão de entrega	Game Story
1	Jogo	7	13	Parte 5	Eu, como presidente do Brasil, quero tomar as melhores decisões para me manter na presidência;
2	Indicadores	3	5	Parte 2	Eu, como jogador, posso tomar uma decisão para alterar os indicadores de suporte político, social e/ou econômico
3	Card	4	3	Parte 2	Eu, como jogador, quero ter acesso às explicações detalhadas dos "cards" para tomar melhores decisões.
4	Card	2	3	Parte 1	Eu, como jogador, devo visualizar a narrativa contida em um card para embasar minha decisão
5	Card	9	3	Parte 4	Eu, como jogador, quero passar de um mês para o outro conforme vou trocando os cards
6	Final do Turno	5	2	Parte 3	Eu, como jogador, caso tenha indicadores suficientes, posso ir para a próxima rodada, para progredir no jogo
7	Final do Turno	6	3	Parte 3	Eu, como jogador, caso não tenha indicadores suficientes, devo ser capaz de visualizar a tela de "game over" para que possa saber que o jogo acabou
8	Menu Inicial	8	2	Parte 4	Eu, como jogador, posso carregar o jogo para voltar de onde estava jogando
9	Menu Inicial	1	1	Parte 2	Eu, como jogador, devo selecionar a opção "Novo Jogo" para começar um novo jogo

1.7.5 Mecânica

O jogo será composto por *cards* que terão instruções de como o presidente poderá agir. Dependendo do card escolhido, o jogo tomará um rumo positivo ou negativo para o jogador.

O jogo terá 9 personagens: presidente (personagem principal, controlado pelo próprio jogador, capaz de tomar as decisões que influenciam o país); ministros da economia, saúde, educação e meio ambiente (são a parte técnica e aconselham o presidente em suas escolhas nas suas respectivas áreas); presidente do senado (representará o senado, avalia as ações do presidente e coloca os projetos em votação, além de opinar se o projeto é viável ou não); presidente do STF (personagem que representa o STF, julga as ações do presidente, independentemente se o projeto passou ou não na votação. É o único capaz de abrir um processo de impeachment contra o presidente da república); general (representa o exército. Capaz de zelar pela estrutura de defesa nacional contra manifestações públicas violentas. O personagem se preocupa com a rebelião popular e avisa sobre as possíveis ameaças no Brasil); secretário (é o braço direito do presidente, por meio dele que o jogador receberá todas as suas informações e notícias sobre sua gestão. A personagem “secretário” é fundamental para que o jogador entenda o que está acontecendo em seu governo e como melhorar sua imagem e suas ações no decorrer do jogo); população (opinião pública que avalia as ações do presidente. Personagem que é afetada diretamente pelos atos do presidente e pode influenciar na permanência do jogador no cargo ou não. É também capaz de influenciar a tomada de decisão de todos os outros personagens).

A dimensão do jogo é 2D. O jogo será relativamente rápido, de 10 a 15 minutos. O jogo será disponibilizado inicialmente via web e, posteriormente, poderá ser acessado através de aplicativo disponível na Play Store e na Apple Store.

O jogo promoverá o entendimento de política de forma lúdica e explicativa.

1.7.6 Fontes de Pesquisa / Imersão

Indicar as principais fontes de pesquisa do jogo para a criação de conteúdo (feitos em aula): imagens, filmes, animações, livros e outros que realmente foram usados para a etapa de imersão ao tema. Fazer um sumário do conteúdo pesquisado a partir dos referidos materiais.

Fontes	
1.	Materiais de Conteúdo Jurídico Básico fornecidos pelo Projeto Constituição na Escola (Materiais - Conteúdo Jurídico.docx) - através desses materiais foi possível realizar uma pesquisa e imersão em temas jurídicos importantes para o jogo como: separação vertical e horizontal dos Poderes, Medida Provisória, PEC, Impeachment, Sopesamento de Princípios e Sistema de Freios e Contrapesos.
2.	AssetLib do Godot (Biblioteca de assets gratuita da ferramenta de criação de jogos Godot) - essa fonte serve para termos uma ideia do que é capaz de ser feito com as ferramentas técnicas com que estamos trabalhando e obtermos uma inspiração para nosso jogo

- | | |
|----|--|
| 3. | Lista de Game Assets e Áudio fornecida pelo Prof. Daniel Paz (Game Assets e Audio) - uma série de recursos que podem auxiliar a criar nosso mundo do jogo, com imagens, sons e animações disponíveis gratuitamente |
| 4. | Jogo "Reigns" - Jogo de estratégia baseado em cards, que serviu de inspiração para a mecânica que adotamos |

2. Game Design <ADALOVE – Elaborar fluxograma do jogo>

2.1 História do Jogo

O personagem é um estudante comum que reclama de todas as decisões do seu representante, mesmo tendo pouco conhecimento sobre os assuntos tratados. Em um dia, na sala de aula, ele dorme e sonha que é o presidente da república. Sua missão é tomar as melhores decisões para sua nação e para manter-se no governo.

A história é desenvolvida através das decisões do jogador, a dificuldade das decisões vão aumentando com o progresso do jogo, suas decisões implicam os rumos para o qual a história vai. Entretanto, o jogador sempre vai passar pelos conceitos básicos que a gente quer que ele aprenda.

O objetivo do jogador é se manter como presidente durante 1 ano de mandato, tendo que tomar decisões sobre assuntos como: quociente eleitoral, liberdade de expressão, direito de livre circulação, orçamento público, teto de gastos, meio ambiente, estatuto da criança, direito à educação e a saúde.

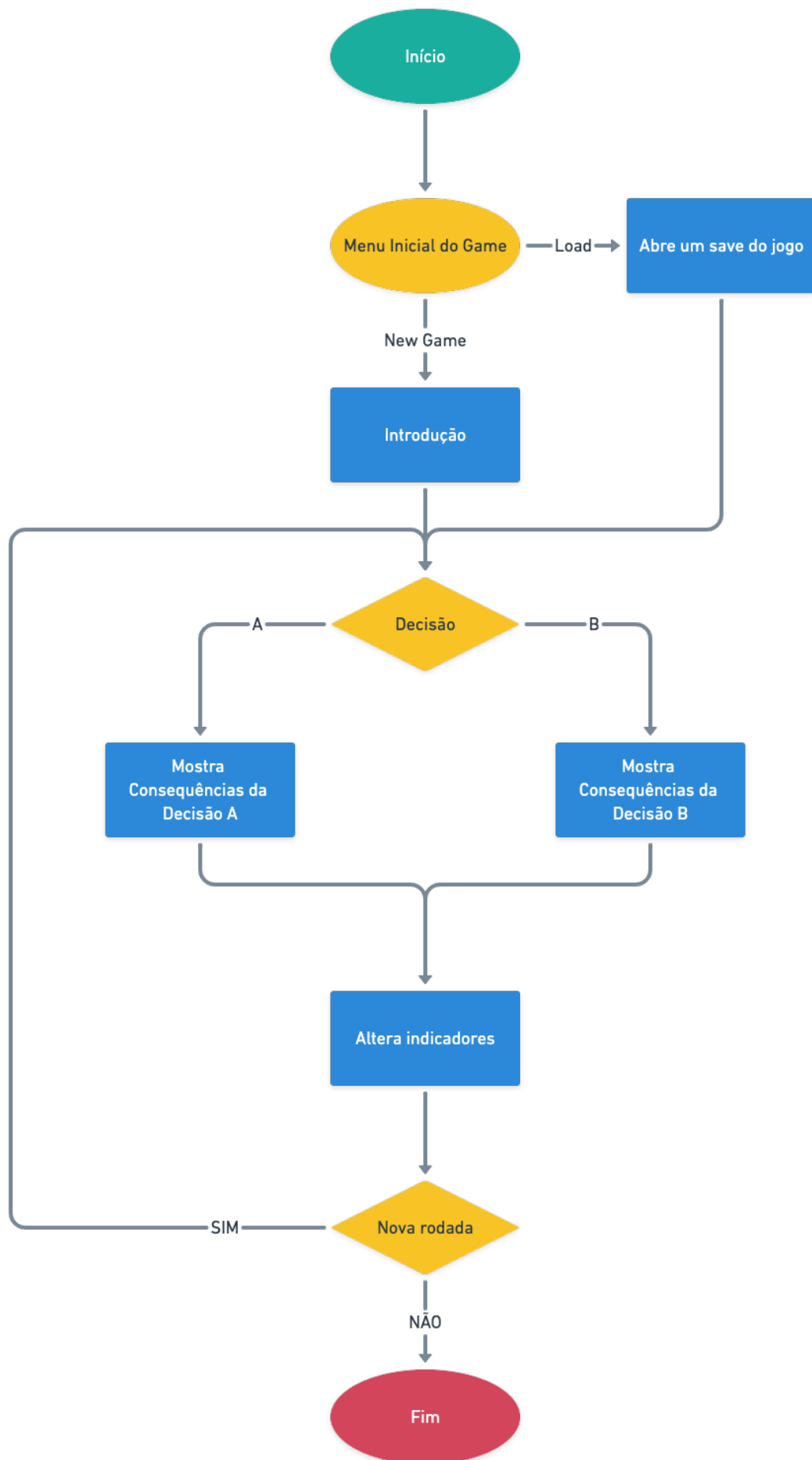
O jogador toma as suas decisões movimentando os cards, para direita ou esquerda, que decidirão a história e levarão a caminhos diferentes.

O tema do jogo, portanto, consiste na tomada de decisões políticas dentro de um mandato presidencial, como uma forma de fazer o jogador

Plano de fundo (backstory): cenário político brasileiro.

2.2 Fluxo do Jogo

De acordo com o *flowchart* abaixo, as ações do jogador serão tomadas de modo binário (sim ou não, concorda ou não concorda, etc), de modo a combinar um estilo de *gameplay* narrativo com elementos de jogos de estratégia. Isso torna o fluxo do jogo relativamente simples, na medida em que existe um "loop" contínuo de decisões que vão impactar os indicadores e as rodadas seguintes. Estimamos que o tempo total do início ao fim do *flowchart* será de aproximadamente 15 minutos. Também consideramos a possibilidade de *replays*, que pode estender esse tempo, uma vez que vamos implantar múltiplos finais com linhas narrativas próprias.



No início do jogo é dada uma ênfase maior para o aprendizado, de modo a orientar o jogador sobre “como jogar”. Passada essa fase de aprendizado, o fluxo do game será focado no descobrimento da história e na tomada de decisões, além de algum mecanismo de recompensa pelos objetivos alcançados. Tal recompensa se dará por meio de pontuações nos três indicadores do jogo (povo, congresso e economia). Por exemplo, uma determinada pontuação poderá levar a um gameplay no qual o Presidente obtém reconhecimento do povo, apoio dos legisladores e sucesso no aspecto econômico. Além disso, caso o jogador obtenha uma pontuação excepcional em um determinado momento, itens adicionais compartilháveis em mídias sociais podem ser disponibilizados. Por fim, pretendemos incluir aspectos de aleatoriedade no game, de modo a aumentar a dinamicidade e fatores surpresa.

2.3 O Mundo do Jogo

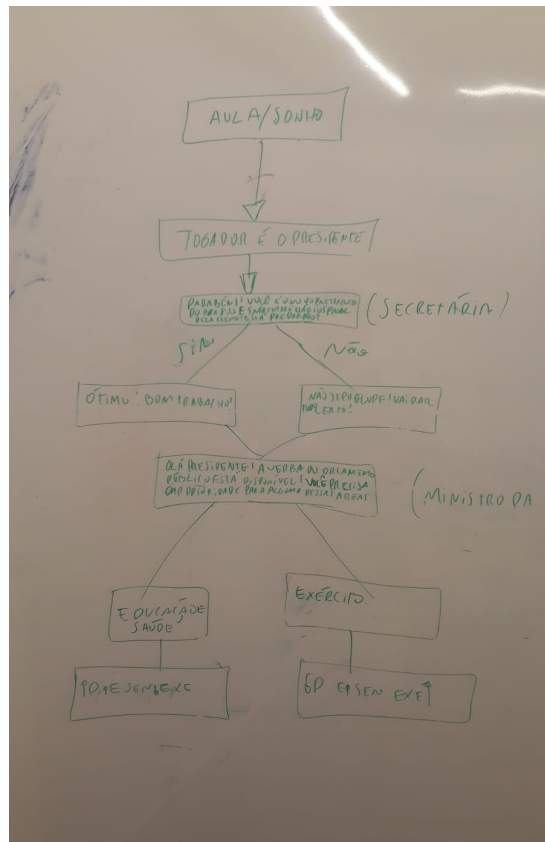
2.3.1 Locações Principais e Mapa

O nosso jogo terá um mapa que simula um ambiente político de Brasília. O jogador pode se movimentar livremente com o bonequinho do presidente, explorando a região encontrando Easter Eggs, realizando tarefas e é claro indo para o congresso nacional, palácio do planalto e STF para resolver as tarefas contidas no jogo.

2.3.2 Navegação pelo Mundo

A navegação pelo mundo se dará através de uma *árvore de decisões*. Sendo a história desenvolvida através das decisões do jogador, a dificuldade destas irá aumentando com o progresso do jogo - suas escolhas implicam os rumos para o qual a história vai. Entretanto, o jogador sempre vai passar por alguns conceitos básicos, conforme os requisitos apresentados pelo cliente. Além disso, para que o jogador chegue a essa árvore de decisões é preciso que o mesmo passe pelo minimap, e, através dele, ocorrerá a principal movimentação do jogador antes dos processos de decisão.

A imagem abaixo pode fornecer um exemplo de como funciona a mecânica de escolhas binárias:



2.3.3 Escala

Não possuímos escala.

2.3.4 Ambientação

A ambientação ocorre esse pelos personagens e pelo texto. Mas também o jogo possui uma luminosidade agradável e viva com cores que simulam o centro de Brasília. As cores são fortes e quentes e contam como uma paisagem viva e natural. O bonequinho do presidente se movimenta de maneira 2D e consegue visitar todos os pontos do mapa em questão.

2.4 Base de Dados

2.4.1 Inventário

Por ser um jogo de estratégia baseado em "cards", não há um inventário na versão inicial. Acreditamos que essa *feature* não é um recurso fundamental para esse tipo de jogo. Eventualmente podemos acrescentar itens que sejam capazes de desbloquear caminhos diferentes, protejam contra alguns eventos indesejados ou aumentem um dos

indicadores, mas esses recursos apenas serão considerados caso as funcionalidades principais do jogo já estejam completamente implementadas.

3. Level Design

3.1 Visão Geral do Jogo

O jogo será um *mix* de *card game* e *graphic novel*, com aspectos de jogos de estratégia, com base em uma árvore de decisões com múltiplos finais e elementos randômicos que, na nossa opinião, são capazes de aumentar a retenção dos jogadores ao mesmo tempo em que viabilizam a construção do conhecimento desejado dos fundamentos constitucionais abordados pela história do jogo. Com relação ao aspecto instrucional, tentaremos gerar o desejo de aprofundar conhecimentos por meio de *brain teasers*, situações em que as decisões do jogador levam a desfechos contra-intuitivos que serão explicados por meio de *cards* opcionais, a serem acessados por um botão especial de Informações.

3.1.2 Layout do Game

Após a introdução e o menu inicial, todo o jogo será apresentado por meio de telas individuais, nas quais haverá uma imagem destacando o contexto situacional, um *snippet*, ou seja, um parágrafo de texto suficiente para esclarecer o momento do jogo, o nome do personagem que estará interagindo com o presidente naquele momento do jogo e uma data que informará acerca do tempo de mandato do Presidente.

Além disso, no topo da tela, haverá três indicadores que serão inicializados com metade da sua capacidade (a princípio, a capacidade irá variar entre 0 e 10, com o jogo iniciando em 5). Esses indicadores irão variar de acordo com as decisões do jogador e servirá como *feedback* em relação às consequências mediatas e imediatas das decisões tomadas.

3.1.2.1 Connections

O jogo contará com cinco cenas, sendo: introdução, menu, main, card e final.

As cenas de introdução e final darão, respectivamente, o contexto da história e sua conclusão, que dependerá das decisões do jogador durante a *gameplay*.

O menu possibilitará a visualização das principais *features* do jogo, a serem implantadas no decorrer do desenvolvimento do projeto.

As cenas principais serão a main e a card, que funcionarão como repositórios centrais das funcionalidades de interação entre o *software* e o jogador, tais como funções de *drag & swipe*, mudança de pontuação e *display* do progresso por meio de indicadores visuais.

3.1.2.2 Layout Effects

Os efeitos a serem utilizados incluem efeitos visuais dos botões (por exemplo, uma iluminação de fundo diferente quando o jogador clicar no botão e/ou algum movimento que chame a atenção para uma determinada ação a ser tomada), um sistema de *drag & swipe* dinâmico, que torna a navegabilidade por meio de dispositivos móveis mais atraente e dinâmica para o jogador, dentre outras focadas na experiência do usuário durante sua interação com a interface do jogo.

4. Personagens <ADALOVE – Implementar mecânicas básicas do jogo - parte 1 a 5>

Aqui, descrever brevemente a relação dos personagens na história e uma tabela com os personagens do jogo, apontando a fase em que aparecem (se houver personagens).

Character Appearance Chart

Personagem	ato 1	ato 2	ato 3	ato 4	ato 5	ato 6	ato 7
Presidente							
min. economia							
min.educação							
min.meio ambiente							
secretário							
min.saúde							
presidente do senado							
presidente do stf							
população							

4.1 Personagens Controláveis

4.1.2 Personagens

Para cada personagem (se houver mais de um), descrever como foi criado, qual é a sua *backstory*. É interessante que apareça os esboços (desenhos) do mesmo. Deve existir algum mecanismo inicial para a seleção de personagem, quando for o caso. Deve permitir seleção de itens básicos iniciais para o personagem, quando for o caso. Para cada personagem, detalhar:

4.1.2.1 Backstory

Backstory (pano de fundo) do personagem.

O presidente da república Pedro Augusto é uma pessoa correta, justa e que toma decisões importantes para a construção do país. Ele ouve o ministro da economia e se preocupa com os assuntos orçamentários da nação. O personagem não precisa ser selecionado, pois ele é apresentado na tela inicial do jogo e o usuário precisa controlar apenas suas decisões. Quando o jogador pressiona o botão iniciar já aparece o card da secretária parabenizando pela vitória na eleição e que ele precisa estar preparado para os novos desafios. Por exemplo, se a educação e a saúde precisarem de reformas, aparece uma decisão de investir ou não nesses setores. Se realmente o líder

máximo do executivo precisa construir novas escolas e hospitais ou se os investimentos estão bem distribuídos. O presidente da república necessita do conselho do ministro da economia a fim de obter uma resposta sobre o destino do orçamento para essas áreas básicas. Caso ele decida investir, a parte da economia diminui e o apoio popular aumenta, já que essas áreas quando atendidas são muito bem avaliadas pela opinião pública. Caso ele não invista a economia não se altera, porque a verba destinada não foi gasta. a população fica revoltada e protesta, pois os serviços populares não foram cumpridos e a saúde e a educação ficam sucateadas. O presidente precisa lidar com a situação e se manter no poder com suas decisões.

4.1.2.2 Concept Art



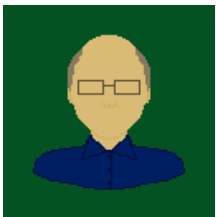
Presidente



Ministra da economia



Ministro da saúde



Secretário



População

4.1.2.3 Ações Permitidas

Habilidades físicas/ações no jogo (tem que estar relacionadas à psicologia e à sociologia do personagem).

O presidente da república é um homem que tem capacidades técnicas para tomar decisões durante seu mandato. Ele consegue ser ao mesmo tempo frio e calculista a fim de tomar decisões mais corretas e certas possíveis (mesmo que seja impossível), ele não gosta de errar e é muito perfeccionista. O personagem principal sente muita raiva e muito magoado com suas decisões tomadas de maneira incorreta. Ao mesmo tempo, consegue ser muito atencioso e ouvir com maestria seu corpo técnico formado pelos ministros da educação, saúde, economia e o meio ambiente. O mesmo é muito comunicativo e consegue dialogar com os outros representantes do poder legislativo e judiciário, com o presidente do senado e o presidente do STF.

4.1.2.4 Momento de Aparição

Momento em que o personagem vai aparecer (identificar de acordo com as fases planejadas, utilizar o apoio do flowchart, se necessário).

O personagem, por ser principal, aparece em todo momento do jogo desde o início (introdução do jogo, onde o personagem é apresentado como o presidente da república até o final, quando o personagem sofre o processo de impeachment ou continua para a próxima fase.) Esse será o processo em toda parte do jogo.

4.2 Common Non-Playable Characters (NPC)

Secretária- Braço direito do presidente. É por meio dela que o jogador presidente receberá todas as informações e notícias sobre sua gestão. A personagem secretária é fundamental para que o jogador entenda o que está acontecendo e como melhorar sua imagem e suas ações no decorrer do jogo.

Personagem tranquilo e calmo e sempre comenta de forma sincera as principais notícias sobre sua gestão.

Atributo Físico- Homem branco de cabelos loiros

Ministra Da Economia- Personagem técnico, capaz de auxiliar o presidente na tomada de decisão de todas as situações que estão relacionadas à economia. Ele irá auxiliar qual é o melhor caminho a seguir no ramo da economia. Cabe ao presidente ouvi-lo ou não.

Atributo Físico- Mulher Branca de cabelo preto.

O personagem é estático, não possui movimentação e é amistoso. Sempre respeitará o presidente e suas decisões.

Ministro da Saúde- Personagem técnico, capaz de auxiliar o presidente na tomada de decisão de todas as situações que estão relacionadas à saúde. Ele irá auxiliar qual é o melhor caminho a seguir no ramo da economia. Cabe ao presidente ouvi-lo ou não.

O personagem é estático, não possui movimentação e é amistoso. Sempre respeitará o presidente e suas decisões.

Atributo físico- Homem Negro

Ministra Da Educação- Personagem técnico, capaz de auxiliar o presidente na tomada de decisão de todas situações que estão relacionadas à educação. Ele irá auxiliar qual é o melhor caminho a seguir no ramo da educação. Cabe ao presidente ouvi-lo ou não.

O personagem é estático, não possui movimentação e é amistoso. Sempre respeitará o presidente e suas decisões.

Atributo físico- Mulher Negra

Ministro do Meio Ambiente- Personagem técnico, capaz de auxiliar o presidente na tomada de decisão de todas situações que estão relacionadas ao meio ambiente. Ele irá auxiliar qual é o melhor caminho a seguir no ramo do meio ambiente. Cabe ao presidente ouvi-lo ou não.

O personagem é estático, não possui movimentação e é amistoso. Sempre respeitará o presidente e suas decisões

Atributo Físico- Homem Índio

General- Personagem capaz de lidar com a segurança pública do país. Através dele que o presidente pode conter as manifestações de maneira mais violenta ou branda.

O personagem é muito sincero e possui uma opinião muito forte. Entretanto sempre respeita as decisões do presidente.

Atributo Físico- Homem Branco de cabelo preto.

Presidente do Senado- Representa o Senado. Através dele que o senado pauta as decisões do presidente. O presidente do Senado pode aconselhar se o projeto do presidente da república tem chance de passar ou não e cabe ao mesmo decidir a continuidade do projeto. Além disso, o presidente do Senado coloca em votação o processo de impeachment contra o presidente da república.

O personagem é neutro e sempre será influenciado e opinar em cima da opinião pública ou do presidente.

Atributo Físico- Mulher branca de cabelo loiro.

Presidente do STF- Representa o STF. Através dele que o STF coloca em prática o processo de investigação contra o presidente. Essa investigação independe se o projeto passou ou não pelo Senado e sim avalia de fato a atitude e a intenção do presidente da república. Depois da investigação concluída, o presidente do STF manda um comunicado para o presidente do Senado e coloca o processo em votação.

O personagem é neutro e sempre será influenciado e opinar em cima da opinião pública ou do presidente.

Atributo Físico- Homem Negro

População- Opinião pública que avalia as ações do presidente. Personagem que é afetado diretamente pelas ações do presidente e podem influenciar sua permanência no cargo ou não. É capaz de influenciar a tomada de decisão de todos os outros personagens. Ela pode fazer manifestações quando estão insatisfeitas com a política e a economia do país.

Conjunto de pessoas com uma opinião forte, podendo ser até agressiva de acordo com as decisões do presidente.

4.3 Special Non-Playable Characters (NPC)

O jogo não tem um “boss” específico ou um NPC especial. A maior preocupação do jogador é fazer uma boa gestão - não sofrer impeachment, agradar o Senado e a população, manter a economia sadia - para que possa continuar exercendo o cargo. Diversos desafios serão enfrentados através dos cards, mas sem a existência de um NPC especial que barre o caminho.

5. Teste de Usabilidade e Padrões de Qualidade

5.1. Teste de Usabilidade <ADALOVE – [Desenvolver relatório de resultados do playtest](#)>

Ao longo dos encontros vocês tiveram oportunidade de colocar pessoas para testar seu jogo.

Descreva aqui quantas pessoas testaram o jogo, quem são elas e os principais pontos de aprendizado.

Número de testes: 33 testes

Pontos positivos (observados nos testes em geral): Os pontos positivos do nosso jogo, segundo os testes apresentados foram marcados pelo nível de clareza e facilidade de abordar o conteúdo de maneira prática e divertida, prendendo a atenção do usuário e assim transformar um conteúdo maçante em simples e interessante. Tivemos elogios na parte mecânica, visto que o jogo não possui bugs durante a sessão de playtest e as escolhas de cards possuem um fluência muito boa e não tem partes de travamento em nenhum momento do jogo. A integração está boa também, segundo as pesquisas, a história possui conexão e segue uma sequência lógica durante o jogo.

Pontos de melhoria (observados nos testes em geral): Os resultados de balanceamento foram os pontos abordados para nossa melhoria. Muitas vezes a jogatina ficou fácil e as decisões que a personagem toma no meio do jogo são simples e não provocam o usuário no processo de escolha. A duração do jogo, de acordo com a avaliação das pessoas, poderia ser maior, ou seja, em suma maioria os testadores queriam que acrescentássemos mais pontos de escolha em nossa história.

Número do teste: Teste número 5

Nome e perfil dos tester: O nome e o perfil dos tester permanecerão no anonimato. A única informação disponibilizada é que eles pertencem ao ateliê 3 do inteli e outros cinco são de outros ateliês. O que podemos informar é que os testers possuem um perfil muito jovem(entre 17 e 25 anos). Eles não gostam de jogos entediantes, adoram ser desafiados e gostam de testar jogos estratégicos que necessitam de um pensamento crítico mais desenvolvido. Além disso, não gostam de conteúdo político, mas se tiverem um jogo que aborda esse assunto de maneira lúdica podem se tornar possíveis cidadãos atuantes na política nacional.

O que observar e perguntar durante o teste:

-Observar e registrar:

Conseguiu começar o jogo? Entendeu a mecânica do jogo? Aprendeu como jogar? Os testadores conseguiram começar o jogo, entenderam a mecânica e aprenderam como jogar. Apenas em 4 casos esse nível de dificuldade foi salientado.

Conseguiu controlar o jogo? Os testadores também conseguiram controlar o jogo e mover os cards na hora da tomada de decisão.

Progrediu no jogo? Passou de fase? Fez pontos? Chegou ao final? Perdeu rápido? Os testadores progrediram no jogo, passaram na fase do jogo, conseguiram pontuar no jogo, chegaram ao final, não perderam rápido (em média 80 % conseguiram finalizar e 20% pararam em alguma das etapas) e conseguiram controlar o jogo

Entendeu as regras do jogo? Teve dificuldade de compreensão? Teve dificuldade ao jogar? Esse quesito ficou um pouco dividido. Alguns testadores(estamos falando de aproximadamente 40 %) tiveram dificuldades de entender o jogo, muito por conta de os indicadores não ficaram claros o suficiente(muitos não entenderam o que faz o usuário perder pontos na política, economia e população). Já outros testadores conseguiram entender a proposta do jogo(estamos falando de aproximadamente 60%) e assim progredir e tomar as melhores decisões e assim terem bons indicadores, quase terminando o jogo ou até mesmo terminando o jogo.

Foi muito fácil? O jogo foi desafiador? A maioria dos testadores não sentiram tanta dificuldade em finalizar o jogo, muito por conta da falta de balanceamento do jogo, assim as decisões tomadas ficaram muito simples e

precisam ser melhoradas para tornar o jogo mais interessante, visto que foi quase pouco desafiador. **-Perguntar a quem testou:**

Numa escala de 0 a 10, quanto você se divertiu nesse jogo? A média sobre a nota de diversão no jogo foi 7,5

O que você gostou no jogo? A história e a mecânica dos cards de decisão.

O que poderia melhorar no jogo?Balanceamento, ou seja, o nível de dificuldade poderia ser um pouco maior.

5.2 Padrões de Qualidade <ADALOVE – Definir Padrões de Qualidade>

Para mostrar os padrões de qualidade seguidos pelo jogo, usamos a *tabela da ISO/IEC 9126 (NBR 13596)*:

CARACTERÍSTICAS	SUBCARACTERÍSTICAS	SIGNIFICADO
Funcionalidade O conjunto de funções satisfazem as necessidades explícitas e implícitas para a finalidade a que se destina o produto?	Adequação	Estamos fazendo o jogo que foi solicitado pelo cliente.
	Acurácia	Atualmente está de acordo com o esperado.
	Interoperabilidade	Consegue interagir com os sistemas solicitados.
	Segurança de acesso	Não aplicável, não temos banco de dados.
	Conformidade	Está de acordo com as normas e a lei.
Confiabilidade O desempenho se mantém ao longo do tempo e em condições estabelecidas?	Maturidade	Apresenta falhas semanalmente, pois ainda está sendo desenvolvido

	Tolerância a falhas	continua rodando, entretanto com alguns bugs.
	Recuperabilidade	Ainda não é capaz de voltar ao estado normal quando ocorrem falhas.
Usabilidade É fácil utilizar o software?	Inteligibilidade	É de simples interação.
	Apreensibilidade	Fácil de aprender, tendo em vista que não é muito complexo.
	Operacionalidade	É fácil de jogar, porque o design é bem intuitivo.
Eficiência Os recursos e os tempos utilizados são compatíveis com o nível de desempenho requerido para o produto?	Comportamento em relação ao tempo	60 fps
	Comportamento em relação aos recursos	Não temos esses dados.
Manutenibilidade Há facilidade para correções, atualizações e alterações?	Analísabilidade	A falha é facilmente encontrada.
	Modificabilidade	É fácil de modificar porque o código está bem organizado e comentado
	Estabilidade	Estável, não ocorre muitos bugs quando alterado.
	Testabilidade	É fácil de testar.
Portabilidade	Adaptabilidade	Adaptável a outros

É possível utilizar o produto em diversas plataformas com pequeno esforço de adaptação?		programas.
	Capacidade para ser instalado	É fácil instalação em diversos ambientes.
	Capacidade para substituir	Não é de fácil substituição.
	Conformidade	Está de acordo com os padrões de portabilidade.

5.3 Melhoria de eficiência do código <ADALOVE – Identificar oportunidades de melhoria de eficiência do código>

- **Oportunidade 1**

No script "Card.gd", há duas funções muito parecidas, "_on_LeftSwipeHitbox_input_event" e "_on_RightSwipeHitbox_input_event". Essas funções executam a mesma lógica quando ocorre a mudança do card, com um clique para a esquerda ou para a direita. Desse modo, toda vez que temos que alterar alguma coisa na transição dos cards (como salvar o jogo, por exemplo), é necessário mudar as duas funções, descumprindo o princípio DRY ("Don't Repeat Yourself"). Seria melhor passar essa lógica para uma outra função, que seria executada nos dois casos.

Implementamos essa oportunidade de melhoria, como pode ser visto abaixo:

Antes:

```

117 ▾ func _on_LeftSwipeHitbox_input_event(_viewport, event, _shape_idx):
118 ▾ | if event is InputEventMouseButton \
119 ▾ | | and event.button_index == BUTTON_LEFT \
120 ▾ | | and event.pressed:
121 ▾ | | | change_card_sfx(random_sfx, ChangeCardSfx1, ChangeCardSfx2, ChangeCardSfx3)
122 ▾ | | | swiped_left = true
123 ▾ | | | yield(get_tree().create_timer(CARD_INTERVAL), "timeout")
124 ▾ | | | swiped_left = false
125 ▾ | | | current_card = functionA.function # Necessário registrar o card atual para a feature de save/load
126 ▾ | | | infoBtn.visible = false # Se houver informações a serem mostradas, rodar a função update_info no próprio card
127 ▾ | | | save_game()
128 ▾ | | | functionA.call_func()
129 ▾ | | | check_scores()
130
131
132 ▾ func _on_RightSwipeHitbox_input_event(_viewport, event, _shape_idx):
133 ▾ | if event is InputEventMouseButton \
134 ▾ | | and event.button_index == BUTTON_LEFT \
135 ▾ | | and event.pressed:
136 ▾ | | | change_card_sfx(random_sfx, ChangeCardSfx1, ChangeCardSfx2, ChangeCardSfx3)
137 ▾ | | | current_card = functionB.function # Necessário registrar o card atual para a feature de save/load
138 ▾ | | | infoBtn.visible = false # Se houver informações a serem mostradas, rodar a função update_info no próprio card
139 ▾ | | | save_game()
140 ▾ | | | swiped_right = true
141 ▾ | | | yield(get_tree().create_timer(CARD_INTERVAL), "timeout")
142 ▾ | | | swiped_right = false
143 ▾ | | | functionB.call_func()
144 ▾ | | | check_scores()

```

Depois:

```

118 # CLICK LISTENERS
119 # Esse conjunto de funcoes lida com o clique do mouse para selecionar uma das opcoes binarias
120 ▾ func card_transition(fn, fn_name, direction):
121 ▾ | change_card_sfx(random_sfx, ChangeCardSfx1, ChangeCardSfx2, ChangeCardSfx3)
122 ▾ | if direction == 'left':
123 ▾ | | swiped_left = true
124 ▾ | else:
125 ▾ | | swiped_right = true
126 ▾ | yield(get_tree().create_timer(CARD_INTERVAL), "timeout")
127 ▾ | swiped_left = false
128 ▾ | swiped_right = false
129 ▾ | current_card = fn_name # Necessário registrar o card atual para a feature de save/load
130 ▾ | infoBtn.visible = false # Se houver informações a serem mostradas, rodar a função update_info no próprio card
131 ▾ | save_game()
132 ▾ | fn.call_func()
133 ▾ | check_scores()
134
135 ▾ func _on_LeftSwipeHitbox_input_event(_viewport, event, _shape_idx):
136 ▾ | if event is InputEventMouseButton \
137 ▾ | | and event.button_index == BUTTON_LEFT \
138 ▾ | | and event.pressed:
139 ▾ | | | card_transition(functionA, functionA.function, 'left')
140
141
142 ▾ func _on_RightSwipeHitbox_input_event(_viewport, event, _shape_idx):
143 ▾ | if event is InputEventMouseButton \
144 ▾ | | and event.button_index == BUTTON_LEFT \
145 ▾ | | and event.pressed:
146 ▾ | | | card_transition(functionB, functionB.function, 'right')

```

- Oportunidade 2

Nosso código tinha originalmente um movimento de rotação nos cards, mas nos testes com usuários o movimento sem rotação foi mais aceito. Assim, optamos por remover esse código que não era mais utilizado.

```
74     if hover_left:
75         # portrait.rotate(-PI/200) <- velocidade que testei
76         # portrait.rotate(PORTRAIT_ANGULAR_VELOCITY * delta * -1)
77         portrait.position.x -= PORTRAIT_SPEED_X * delta
78     elif hover_right:
79         # portrait.rotate(PI/200) <- velocidade que testei
80         # portrait.rotate(PORTRAIT_ANGULAR_VELOCITY * delta)
81         portrait.position.x += PORTRAIT_SPEED_X * delta
82     else:
83         portrait.position.x = PORTRAIT_ORIGINAL_X
84         # portrait.rotation_degrees = 0
85         # Para evitar que a imagem rotacione alem de 45 graus ou saia da tela,
86         # limitamos aqui os valores possiveis para essas propriedades
87         portrait.rotation_degrees = clamp(portrait.rotation_degrees, -25, 25)
```

- Oportunidade 3

Nosso código pode ter mais comentários mais elaborados e padronizados em alguns trechos, apesar de já estar razoavelmente claro.

- Oportunidade 4

Os arquivos do projeto possuem sprites adicionais, tiles disponíveis e outros elementos que viabilizam uma expansão do escopo do projeto atual. Nesse sentido, o mapa pode ser populado com NPCs adicionais e as construções faltantes podem ser incorporadas para viabilizar "idas" e "voltas" do jogador entre o cenário do mapa e o cenário do jogo de cartas. Isso aumentaria a sensação de dinâmica do jogador e, acredita-se, a sua atração pelo jogo.

- Oportunidade 5

A implementação de sinais do tipo *drag* e *swipe* para dispositivos móveis é relativamente simples e o modelo de *layout* utilizado para o jogo é adaptado para esse modo de reprodução, de modo que não requer adaptações adicionais no que se refere ao seu uso em *smartphones*.

6. Relatório - Física e Matemática

6.1 Funções <ADALOVE - Definir as funções matemáticas que serão utilizadas no jogo>

Quais funções são usadas no jogo desenvolvido neste projeto? Coloque os trechos do programa no Godot onde elas aparecem e explique sua utilidade no jogo.

O jogo envolve diversas funções matemáticas. Elencamos a seguir algumas delas, a título exemplificativo, com seus respectivos blocos de código.

O primeiro conjunto de funções envolve um avatar do presidente, controlado pelo jogador, executando movimentos em um mapa do jogo, conforme o código abaixo. Essa parte do jogo será na verdade um "minigame" introdutório, antes do cenário geral descrito previamente, de escolhas baseadas em "cards". Destacamos que o script a seguir foi gerado ao seguirmos a excelente série de vídeo tutoriais *"Make an Action RPG in Godot 3.2"* do desenvolvedor *"HeartBeast"*, disponível em <https://www.youtube.com/watch?v=mAg8Oi-SvQ>:

```
7  const ACCELERATION = 1000 # Aceleracao do avatar
8  const MAX_SPEED = 200 # Velocidade maxima do avatar
9  const FRICTION = 1000 # Friccao do avatar
10
11  var velocity = Vector2.ZERO # Inicializar vetor
12
13  onready var animationPlayer = $AnimationPlayer # Animacoes do avatar
14  onready var animationTree = $AnimationTree # Maquina de estados
15  onready var animationState = animationTree.get("parameters/playback") # Estado
16
17  func _physics_process(delta):
18      var input_vector = Vector2.ZERO
19
20      input_vector.x = Input.get_action_strength("ui_right") - Input.get_action_strength("ui_left")
21      input_vector.y = Input.get_action_strength("ui_down") - Input.get_action_strength("ui_up")
22      input_vector = input_vector.normalized()
23
24      if input_vector != Vector2.ZERO:
25          animationTree.set("parameters/Idle/blend_position", input_vector)
26          animationTree.set("parameters/Run/blend_position", input_vector)
27          animationState.travel("Run")
28          velocity = velocity.move_toward(input_vector * MAX_SPEED, ACCELERATION * delta)
29      else:
30          animationState.travel("Idle")
31          velocity = velocity.move_toward(Vector2.ZERO, FRICTION * delta)
32
33      velocity = move_and_slide(velocity)
```

Como é possível observar nesse trecho de código, o avatar do jogador tem o movimento controlado por um vetor, que por sua vez tem as suas coordenadas no eixo y e no eixo x modificadas pelo "input" do jogador.

Se existe um input do jogador (variável "input_vector"), o método "velocity.move_toward()" é executado para acelerar a velocidade do personagem em direção ao sentido apontado pelo jogador, com determinada rapidez (constante "MAX_SPEED"). Esse processo é realizado incrementalmente, sendo que a cada frame a velocidade

aumenta o valor de `"ACCELERATION * delta"`, sendo `"ACCELERATION"` uma constante e `"delta"` a diferença em segundos entre um frame e outro. Esse aumento incremental persiste até que a velocidade máxima seja alcançada, momento em que a velocidade do personagem permanecerá constante.

Se não houver input do jogador (variável `"input_vector"` for igual a zero), o mesmo processo descrito para a aceleração acontecerá, porém o método `"velocity.move_toward()"` será executado para reduzir a velocidade do personagem até zero, quando, novamente, permanecerá constante. Essa redução ocorrerá em incrementos de `"FRICTION * delta"`, sendo `"FRICTION"` uma constante.

O método `"move_toward"` é uma abstração do Godot para o movimento uniformemente variado, cuja fórmula é:

$$S(t) = S_0 + V_0 t + \frac{at^2}{2}$$

Sendo `"S"` a distância, `"t"` o tempo, `"S0"` a posição inicial, `"V0"` a velocidade inicial e `"a"` a aceleração (se positivo) ou fricção (se negativo).

Portanto, enquanto o personagem está acelerando ou desacelerando ($a \neq 0$), temos uma função quadrática. Já quando o personagem atingiu a velocidade máxima ou está em repouso, temos uma função afim, pois o terceiro termo será eliminado:

$$S(t) = S_0 + V_0 t$$

Essas fórmulas serão aplicadas tanto para o eixo `"x"` quanto para o eixo `"y"`. Em termos práticos:

$$x(t) = x_0 + V_0 t + \frac{at^2}{2} \text{ se } a \neq 0$$

$$y(t) = y_0 + V_0 t + \frac{at^2}{2} \text{ se } a \neq 0$$

$$x(t) = x_0 + V_0 t \text{ se } a = 0$$

$$y(t) = y_0 + V_0 t \text{ se } a = 0$$

Sendo: `"x"` a posição horizontal final do personagem na tela; `"y"` a posição vertical final; `"x0"` a posição horizontal inicial; e `"y0"` a posição vertical inicial. Note que no Godot, quando mais positivo o `"y"` mais para "baixo" o personagem se encontrará, sendo esse eixo invertido em relação ao que estamos acostumados ao trabalhar com o plano cartesiano tradicional.

O segundo conjunto de funções é relativamente mais simples, sendo usado para girar um `"card"` do personagem. O trecho de código relevante pode ser observado abaixo:


```

41 # Chamada todo frame. 'delta' é o tempo (em segundos) desde o último frame.
42 ~ func _process(delta):
43 >I # Essas condicoes servem para determinar se a imagem deve se mexer para
44 >I # a esquerda ou direita, quando o jogador mover o mouse naquela direcao
45 ~ >I if swipe_left:
46 >I >I # portrait.rotate(-PI/200) <- velocidade que testei
47 >I >I portrait.rotate(portrait.angular_velocity * delta * -1)
48 >I >I portrait.position.x -= 2
49 ~ >I elif swipe_right:
50 >I >I # portrait.rotate(PI/200) <- velocidade que testei
51 >I >I portrait.rotate(portrait.angular_velocity * delta)
52 >I >I portrait.position.x += 2
53 ~ >I else:
54 >I >I portrait.position.x = portrait_original_x
55 >I >I portrait.rotation_degrees = 0
56 >I # Para evitar que a imagem rotacione alem de 45 graus ou saia da tela,
57 >I # limitamos aqui os valores possiveis para essas propriedades>I
58 >I portrait.rotation_degrees = clamp(portrait.rotation_degrees, -45, 45)
59 >I portrait.position.x = clamp(portrait.position.x, 250, 450)
60

```

A função "portrait.rotate()" imprime uma velocidade angular para o card, descrita pela seguinte fórmula:

$$\omega = \frac{\Delta\theta}{\Delta t}$$

Sendo ω a velocidade angular, $\Delta\theta$ a mudança na rotação angular (em radianos) e Δt a mudança no tempo.

Se o jogador movimentar o mouse para a esquerda, a rotação será negativa, no sentido anti-horário e se o jogador movimentar o mouse para a direita a rotação será positiva, no sentido horário. Novamente, o Godot inverte a notação matemática tradicional, pois estamos acostumados a identificar os ângulos no plano cartesiano como sendo positivos no sentido anti-horário e negativos no sentido horário.

6.2 Cinemática Unidimensional <ADALOVE - Fazer orientação espacial (2D)>

Quais grandezas da cinemática são usadas no jogo desenvolvido neste projeto? Coloque os trechos do programa no Godot onde elas aparecem e explique sua utilidade no jogo.

As grandezas da cinemática unidimensional que identificamos no jogo são a posição "x" original do retrato contido no "card" do jogador ("PORTRAIT_ORIGINAL_X"), a velocidade de movimento quando o jogador coloca o mouse à esquerda ou à direita do retrato ("PORTRAIT_SPEED_X") e o modificador dessa velocidade quando o card está sendo retirado de cena ("SPEED_SWIPE_MODIFIER"):

```
onready var PORTRAIT_ORIGINAL_X: float = portrait.position.x # Valor da posicao x quando o card é renderizado
const PORTRAIT_SPEED_X: int = 400 # Rapidez com que o card se move horizontalmente
const SPEED_SWIPE_MODIFIER: int = 5 # Coeficiente da rapidez quando o card sai de cena
```

Se o jogador colocar o mouse à direita ou à esquerda do retrato do personagem no card, a posição "x" deste será atualizada para subtrair ou acrescentar o valor da velocidade ("PORTRAIT_SPEED_X"), que será multiplicado a cada frame por "delta" (a quantidade de tempo entre um frame e outro). Caso contrário, o card voltará para sua posição "X" original ("PORTRAIT_ORIGINAL_X"):

```
# Essas condicoes servem para determinar se a imagem deve se mexer para
# a esquerda ou direita, quando o jogador mover o mouse naquela direcao
if hover_left:
>| portrait.position.x -= PORTRAIT_SPEED_X * delta
elif hover_right:
>| portrait.position.x += PORTRAIT_SPEED_X * delta
else:
>| portrait.position.x = PORTRAIT_ORIGINAL_X
```

Já se o jogador clicar à direita ou à esquerda do retrato do personagem no card, este será retirado de cena rapidamente, acelerando sua velocidade padrão por uma escala previamente definida ("SPEED_SWIPE_MODIFIER"):

```
if swiped_left:
>| portrait.position.x -= PORTRAIT_SPEED_X * delta * SPEED_SWIPE_MODIFIER
>| return
if swiped_right:
>| portrait.position.x += PORTRAIT_SPEED_X * delta * SPEED_SWIPE_MODIFIER
>| return
```

6.3 Vetores <ADALOVE - Aplicar os conceitos matemáticos no jogo>

Quais vetores são usados no jogo desenvolvido neste projeto? Coloque os trechos do programa no Godot onde eles aparecem e explique sua utilidade no jogo. Obs.: Anexar ao relatório as atividades realizadas em aula (prints, fotos, etc.).

O projeto usa vetores essencialmente para controlar o movimento do avatar do jogador. Inicialmente, o avatar começa com um vetor nulo ("zero vector"), com magnitude igual a zero e sem direção:

```
var input_vector = Vector2.ZERO
```

A quantidade de input do jogador para a direita é subtraída pela quantidade de input do jogador para a esquerda. O resultado passa a ser a orientação "x" (eixo horizontal ou das abscissas) do vetor:

```
input_vector.x = Input.get_action_strength("ui_right") - Input.get_action_strength("ui_left")
```

Já a quantidade de input do jogador para baixo é subtraída pela quantidade de input do jogador para a cima. O resultado é definido como a ser a orientação "y" (eixo vertical ou das ordenadas) do vetor:

```
input_vector.y = Input.get_action_strength("ui_down") - Input.get_action_strength("ui_up")
```

Esse vetor resultante, com "x" e "y" determinados, passa por um processo de "normalização", para assegurar que seu módulo será sempre igual a 1, no máximo. Dessa forma, o personagem não se movimenta mais rápido na diagonal, quando seu módulo seria normalmente superior a 1:

```
input_vector = input_vector.normalized()
```

Se o vetor não for neutro (ou seja, há um input do jogador naquele frame), uma série de instruções é executada para acelerar o movimento do jogador:

```
if input_vector != Vector2.ZERO:
>I animationTree.set("parameters/Idle/blend_position", input_vector)
>I animationTree.set("parameters/Run/blend_position", input_vector)
>I animationState.travel("Run")
>I velocity = velocity.move_toward(input_vector * MAX_SPEED, ACCELERATION * delta)
```

Já se o vetor for igual a zero ("x" e "y" iguais a zero por não existir input do jogador ou eles se anularem), uma série de instruções é executada para desacelerar o movimento do avatar, através de atrito:

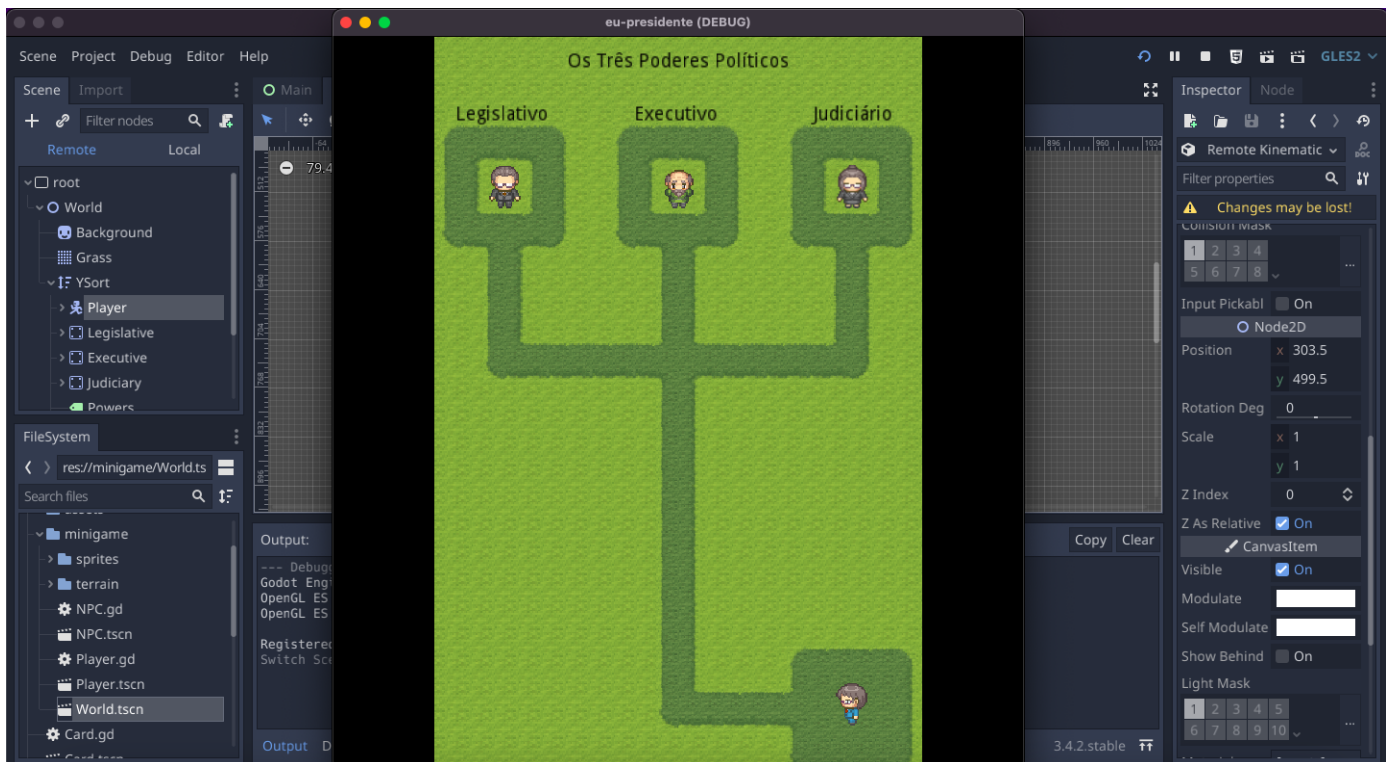
```
else:
>I animationState.travel("Idle")
>I velocity = velocity.move_toward(Vector2.ZERO, FRICTION * delta)
```

6.4 Cinemática Bidimensional e mais <ADALOVE - Fazer orientação espacial>

Quais as grandezas físicas (não trabalhadas nos encontros anteriores) são usadas no jogo desenvolvido neste projeto? Coloque os trechos do programa no Godot onde elas aparecem e explique sua utilidade no jogo.

As grandezas da cinemática bidimensional que identificamos no jogo são a posição (x,y), a aceleração, a velocidade máxima e o atrito do avatar do jogador enquanto se movimenta no mapa do jogo. Trabalharemos cada um desses itens a seguir.

1. Posição



Exemplo de mapa do jogo em modo "debug" - o personagem controlado pelo jogador está em azul, com três NPCs acima. No canto direito da tela, podemos ver que a posição "x" do jogador é de 303.5 pixels e "y" de 499.5 pixels.

O personagem controlado se movimenta com base em um vetor calculado a partir do input recebido do jogador. Basicamente, ele começa em uma posição "x" e "y", em pixels, calculada a partir do canto superior esquerdo. Sendo assim, quanto mais para a direita o personagem estiver, maior será sua posição "x", em pixels e quanto mais para baixo, maior será sua posição "y", também em pixels. Em uma analogia ao mundo real, seria equivalente a um campo de futebol em que a posição de um jogador, em metros, seria descrita a partir da distância "x" e "y" do canto superior esquerdo do campo.

2. Aceleração

A aceleração do personagem é uma grandeza física, medida no Sistema Internacional de Unidades (SI) em metros por segundo ao quadrado. No jogo, trata-se da constante "ACCELERATION" que pode ser modificada no código fonte:

```
const ACCELERATION = 1000 # Aceleracao do avatar
```

Essa constante é utilizada para calcular o intervalo de incremento da posição do personagem no jogo, através da função "move_toward" do Godot:

```
velocity = velocity.move_toward(input_vector * MAX_SPEED, ACCELERATION * delta)
```

3. Velocidade Máxima

A velocidade do personagem também representa uma grandeza física, medida no Sistema Internacional de Unidades (SI) em metros por segundo. No jogo, trata-se da constante "MAX_SPEED":

```
const MAX_SPEED = 200 # Velocidade maxima do avatar
```

Essa grandeza serve para limitar a velocidade máxima que o personagem pode alcançar, a partir da aceleração mencionada acima:

```
velocity = velocity.move_toward(input_vector * MAX_SPEED, ACCELERATION * delta)
```

4. Atrito

O atrito também modifica o movimento do personagem, porém atua no sentido oposto ao da velocidade. Em nosso jogo, o atrito é a constante "FRICTION":

```
const FRICTION = 1000 # Atrito do avatar
```

A função dessa grandeza consiste em determinar o intervalo de redução da velocidade até que a mesma chegue a zero, através da função "move_toward":

```
velocity = velocity.move_toward(Vector2.ZERO, FRICTION * delta)
```

7. Bibliografias

Toda referência citada no texto deverá constar nessa seção, utilizando o padrão de normalização da ABNT). As citações devem ser confiáveis e relevantes para o trabalho. São imprescindíveis as citações dos *sites* de *download* das ferramentas utilizadas, bem como a citação de algum objeto, música, textura ou outros que não tenham sido produzidos pelo grupo, mas utilizados (mesmo no caso de licenças gratuitas, *royalty free* ou similares).

Apêndice

Os apêndices representam informações adicionais que não caberiam no documento exposto acima, mas que são importantes por alguma razão específica do projeto. Em geral, os apêndices do GDD podem incluir os rascunhos das fases, outros *concept arts* do jogo, diagramas diversos etc.