

# DOCUMENTO PARA DESIGN DE GAMES

## Os Três Poderes

**Autores:** Eric Tachdjian, Gabriel Pascoli, Luiz Augusto Ferreira , Pedro Munhoz, Pedro Rezende,  
Sergio Lucas e Vinícius Souza

Data de criação: 07 de Fevereiro de 2022

Versão: 0.5

## Controle do Documento

### Histórico de revisões

Data	Autor	Versão	Resumo da atividade
11/02/2022	C.F 88	0.1	Foram preenchidos os tópicos do projeto de 1.1 a 1.5
09/03/2022	C.F 88	0.2	Foram preenchidos os tópicos do projeto: 1.6 e 1.7 - completos - <b>(1.7.4)</b>
25/02/2022	C.F 88	0.3	Foram preenchidos os tópicos do projeto: 2, 3, 4 e 6.1
07/03/2022	C.F 88	0.4	Foi reestruturado o tópico do projeto: 6.1
11/03/2022	C.F 88	0.5	Foi implementado a definição de padrões de qualidade no 5.1 / atualização do segmento 1.7.4

---

## Sumário

<b>1. Visão Geral do Projeto &lt;ADALOVE – Definir Proposta de Valor&gt;</b>	<b>6</b>
1.1 Objetivos do Jogo	6
1.2 Características gerais do Jogo	6
1.3 Público-alvo	6
1.4 Diferenciais	6
1.5 Análise do cenário: Matriz SWOT	6
1.6 Proposta de Valor: Value Proposition Canvas	6
1.7 Requisitos do Jogo <ADALOVE – Documentar requisitos>	7
1.7.1 Requisitos coletados na entrevista com o cliente	7
1.7.2 Persona	7
1.7.3 Gênero do Jogo	7
1.7.4 Histórias do jogo (Game stories) ou Histórias dos usuários (user stories)	7
1.7.5 Mecânica	7
1.7.6 Fontes de Pesquisa / Imersão	7
<b>2. Game Design &lt;ADALOVE – Elaborar fluxograma do jogo&gt;</b>	<b>8</b>
2.1 História do Jogo	8
2.2 Fluxo do Jogo e Níveis (os níveis são opcionais)	8
2.3 O Mundo do Jogo	8
2.3.1 Locações Principais e Mapa	8
2.3.2 Navegação pelo Mundo	9
2.3.3 Escala	9
2.3.4 Ambientação	9
2.3.5 Tempo	9
2.4 Base de Dados	9
2.4.1 Inventário	9
2.4.1.1 Itens Consumíveis (opcional)	9
2.4.1.2 Armamento (opcional)	10

2.4.2 Bestiário (opcional)	10
2.4.2.1 Inimigos Elementais de Água	11
2.4.3 Balanceamento de Recursos (opcional)	11
<b>3. Level Design (opcional) &lt;ADALOVE – Implementar mecânicas básicas do jogo - parte 1 a 5&gt;</b>	<b>13</b>
3.1 Fase <NOME DA FASE 1>	13
3.1.1 Visão Geral (opcional)	13
3.1.2 Layout Área (opcional)	13
3.1.2.1 Connections (opcional)	13
3.1.2.2 Layout Effects (opcional)	13
3.1.2.3 Quests e Puzzles (opcional)	13
3.1.3 Balanceamento de Recursos (opcional)	13
3.1.4 The Boss	14
3.1.5 Outros Personagens	14
3.1.6 Easter Eggs	15
<b>4. Personagens &lt;ADALOVE – Implementar mecânicas básicas do jogo - parte 1 a 5&gt;</b>	<b>16</b>
4.1 Personagens Controláveis	16
4.1.2 <NOME DO PERSONAGEM PRINCIPAL n>	16
4.1.2.1 Backstory	16
4.1.2.2 Concept Art	16
4.1.2.3 Ações Permitidas	16
4.1.2.4 Momento de Aparição	16
4.2 Common Non-Playable Characters (NPC)	17
4.2.1 <NOME DO NPC COMUM n>	17
4.3 Special Non-Playable Characters (NPC)	17
4.3.1 <NOME DO NPC ESPECIAL n>	17
<b>5. Teste de Usabilidade &lt;ADALOVE – Desenvolver relatório de resultados do playtest&gt;</b>	<b>18</b>
<b>6. Relatório - Física e Matemática</b>	<b>19</b>
6.1 Funções	19
6.2 Cinemática Unidimensional	19

6.3 Vetores	19
6.4 Cinemática Bidimensional e mais	19
<b>7. Bibliografias</b>	<b>20</b>
<b>Apêndice</b>	<b>21</b>

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

## 1.1 Objetivos do Jogo

Resumo:

Nosso jogo visa ensinar de uma forma simples e compreensível, para todas as idades, sobre a constituição brasileira, fazendo com que as pessoas entendam melhor os aspectos da nossa constituição e os tornem sujeitos mas conscientes.

Contamos com diversas diretrizes que utilizamos para realizar o projeto com sucesso, elas são:

### 1. Desafios e interesses pessoais no tipo de jogo criado

Nosso interesse pessoal é de aprender sobre a constituição e o grande desafio é traduzir o que aprendemos para o usuário de forma simples, didática e, sobretudo, divertida.

### 2. Para que serve o jogo

O jogo tem a finalidade de difundir o conhecimento sobre as leis do Brasil para a população, a fim de educá-los politicamente.

### 3. Para que o jogo está sendo criado

A criação do game visa suprir uma lacuna de conhecimento político e disseminar entendimento de fácil acesso à população, propondo a oferta tanto do conhecimento sobre estrutura e organização política, quanto noções sobre Direitos e Deveres do Cidadão e do Governo, de forma simples e didática.

### 4. Visão geral e contexto do jogo

O jogo é uma aplicação web que pode ser acessada por computadores e em dispositivos móveis, com foco principal em conteúdo educacional, para ser uma forma divertida e engajadora de aprender.

Por análise do cliente, o contexto gira em torno de um país polarizado politicamente, carente de informações imparciais sob a ótica da questão política, constitucional e social.

5. Contexto onde este jogo está sendo criado (justificar que é feito num bootcamp, que vai gerar um produto para o bootcamp)

O jogo está sendo desenvolvido por alunos do curso de graduação do Instituto de Tecnologia e Liderança (Inteli) para oferecer uma solução tecnológica ao Projeto Constituição Escola.

## **1.2 Características gerais do Jogo**

É um jogo de plataforma interativo e didático que visa imergir o usuário num mandato presidencial, valendo-se de múltiplas escolhas em que as decisões impactarão diretamente na jornada durante seu tempo de presidência.

A trama é uma simulação de um ano\* na presidência do Brasil, utilizando elementos da área do direito com linguagem simplificada, para melhor compreensão. É um jogo com vitória e derrota devidamente estabelecidos com o objetivo de se manter no poder, envolvendo tanto a população, quanto o Senado..

## **1.3 Público-alvo**

Estudantes do ensino médio e jovens adultos brasileiros, entre 14 e 24 anos de ambos os gêneros, porém também será aberto e disponibilizado para qualquer cidadão que esteja interessado em conhecer sobre as leis do país.

## 1.4 Diferenciais

**ELIMINAR:** status Quo do conhecimento jurídico;

**ELEVAR:** Diversão no aprendizado; o conhecimento base de política para jovens;

**REDUZIR:** Uso excessivo da linguagem culta;

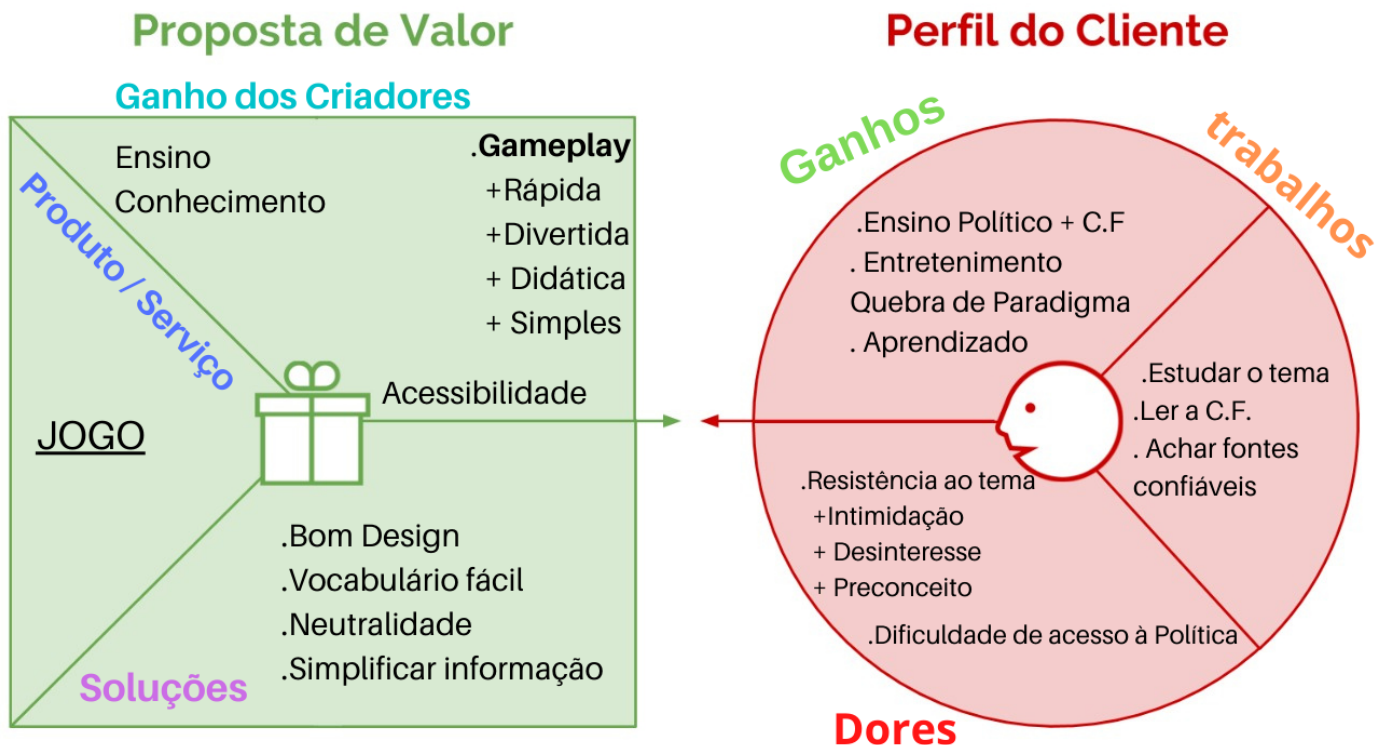
**CRIAR:** Estratégia em relação ao aprendizado dinâmico da constituição.

## 1.5 Análise do cenário: Matriz SWOT

<b>FORÇAS :</b> <ul style="list-style-type: none"><li>- Facilidade de compreensão e acessível para todos.</li><li>- Um meio didático dinâmico para escolas.</li><li>- Estimular o protagonismo no aprendizado.</li></ul>	<b>FRAQUEZAS:</b> <ul style="list-style-type: none"><li>- Área com um benchmark pouco explorado.</li><li>- É um assunto que, para muitos jovens, não desperta interesse.</li></ul>
<b>OPORTUNIDADES:</b> <ul style="list-style-type: none"><li>- Grande mercado de jogos para celular e grande possibilidade de ascensão do mercado.</li><li>- Poucos concorrentes em relação ao público alvo.</li><li>- Um tema pouco abordado na indústria dos games.</li></ul>	<b>AMEAÇAS:</b> <ul style="list-style-type: none"><li>- Falta de interesse por parte dos usuários.</li><li>- Jogos já consolidados no mercado.</li><li>- Poucas maneiras de torná-lo interessante</li></ul>



## 1.6 Proposta de Valor: Value Proposition Canvas



## 1.7 Requisitos do Jogo <ADALOVE – Documentar requisitos>

### 1.7.1 Requisitos coletados na entrevista com o cliente

<https://docs.google.com/document/d/1L7plhqpPEEYoApHSwwLbvhJ5cEG-aJa5/edit?usp=sharing&ouid=112225462967162168991&rtpof=true&sd=true> → perguntas feitas ao Felipe Neves

Que jogo é esse? Onde se passa este jogo? O que eu posso controlar? Quantos personagens eu controlo? Qual é o objetivo do jogo? O que é diferente no jogo em comparação com outros similares?

O jogo em questão trata-se do “Eu, presidente”, um game que retrata o ensino sobre política e constituição no Brasil, em que o jogador tem em mãos a possibilidade de interagir com os Três Poderes (Executivo, Legislativo e Judiciário), por meio do desenvolvimento de projetos de leis, a fim de sanar problemas populares e sobreviver a um mandato sem levar um “impeachment”.

A narrativa envolve a gestão da vida do Presidente da República, sendo que o diferencial do “Eu, presidente” é ser o pioneiro no mercado a abordar o ensino político e constitucional brasileiro de forma intuitiva e interessante, estimulando o jovem a aprender por conta própria a partir do jogo.

### 1.7.2 Persona

Apresentar para cada uma o nome, idade, ocupação, interesses, localização, etc. (relacionar com o eu foi visto nos encontros e conteúdos de autoestudo sobre definição de personas).



NOME: Enrique

IDADE: 19

OCUPAÇÃO: Estudante

**"Gosto de política e sou gamer casual"**

### Biografia:

É um jovem de 19 anos, que estudou em um Instituto Federal, sempre gostou de jogar jogos casualmente, e nunca teve um ensino sobre a constituição.

## Características (personalidade, conhecimentos, interesses, habilidades):

Gosta  
de jogos  
casuais

É  
interessado  
em política

Está no  
ensino  
superior

Tem um  
conhecimento  
básico de  
política

Classe  
econômica  
média

### Motivações com jogos:

Se diverte com jogos mais  
simples e didáticos.  
Tenta jogar alguns jogos  
para ver se despertam seu  
interesse, para assim  
continuar jogando

### Dores com jogos:

Os jogos atuais  
tem muitas  
propagandas  
que atrapalham  
a gameplay

Os jogos demandam  
muito tempo,  
atrapalhando os  
estudos e se tornando  
maçante com o tempo,  
resultando na perda de  
interesse gradativo

### Motivações com o problema:

Ter uma melhor  
perspectiva sobre  
o processo de  
eleição brasileiro.

Aprender e  
compartilhar  
com seus  
amigos

### Dores com o problema:

Ele não sabe  
sobre constituição  
e precisa  
conhecer seus  
direitos básicos  
como cidadão.

Falta de  
ensino  
qualitativo e  
que desperte  
interesse

### 1.7.3 Gênero do Jogo

“Eu, presidente” é um jogo de estratégia, uma vez que o jogador necessita tomar decisões baseadas numa lógica de jogo suficiente para que o presidente cumpra seu mandato no governo brasileiro.

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

Épico	Parte	Game Story	Complexidade (Fibonacci)	Status
<b>Apresentação</b>	<b>1 - Escolha de Lei (Executivo)</b>	Eu, jogador, preciso apresentar uma proposta de lei, para resolver problemas relacionados às temáticas apresentadas.	<b>2</b>	<b>Feito</b>
<b>Apresentação</b>	<b>2 - Minigame (Executivo)</b>	Eu, jogador, após ter terminado o mini game corrida pela saúde, quero ser pontuado.	<b>5</b>	<b>Em Progresso</b>
<b>Casa Iniciadora e Revisora</b>	<b>2- Minigame (Legislativo)</b>	Eu, jogador, após ter minha performance no minigame pontuada, quero conseguir acessar o poder legislativo.	<b>3</b>	<b>Feito</b>
<b>Análise pelas Comissões</b>	<b>3 - Debate (Legislativo)</b>	Eu, jogador, quero utilizar as funções do poder legislativo, como debate, para dar continuidade ao jogo.	<b>21</b>	<b>Pendente</b>
<b>Aprovação</b>	<b>4 - Votação do Plenário (cutscene)</b>	Eu, jogador, quero ser julgado pelo plenário, de acordo com meus indicadores, e que minhas leis sejam aprovadas (ou rejeitadas)	<b>2</b>	<b>Pendente</b>
<b>Sanção e Veto</b>	<b>5 - Aprovação do Presidente (Executivo)</b>	Eu, como poder parlamentar, quero aprovar ou desaprovar a decisão dos outros poderes para ressaltar o que é certo ou errado	<b>8</b>	<b>Pendente</b>
<b>Poder Judiciário</b>	<b>5 - Aprovação do Presidente (Judiciário)</b>	Eu, como poder judiciário, devo analisar as leis propostas para definir o progresso do jogador	<b>5</b>	<b>Pendente</b>

---

---

#### 1.7.5 Mecânica

O jogador deve interagir com os 3 poderes de maneira com que ele consiga aprovar uma lei em um processo completo.

Ele deve começar no poder executivo, passar para o legislativo e então chegar no judiciário.

Utilizando de vários botões e funções o jogador é guiado pelos poderes para que ele possa, de maneira compreensiva e fácil, criar uma lei.

### 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.

Fonte	
1.	Jogo Branches of Power
2.	
3.	

---

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

### 2.1 História do Jogo

Descrever os seguintes aspectos:

- **Tema (*storyline*)**

- O protagonista fará parte de três poderes dentro do jogo, controlando-os de forma externa para criar ou aprovar leis e projetos de leis, retratando um mandato do novo Presidente da República, de forma que seja levado em consideração, para a aprovação de leis, a aprovação do Governo e da população.

- **Conceito**

- Controlar três personagens referentes aos Três Poderes, para definir, de forma eficiente, a direção pela qual o país trilhará por meio das leis.

- **Plano de fundo da história (*backstory*) \*\*\***

- Com o passar dos anos, as leis brasileiras foram perdendo relevância, de forma que os cidadãos não mais as conheciam nem as respeitavam. Surge, nesse contexto, a figura de um Presidente responsável por estabelecer a ordem no Brasil, promovendo a criação de leis que corrijam as mazelas sociais;

- **Premissa**

- Jogador acompanhará o processo de criação de leis de sua escolha, participando ativamente do processo enquanto ele foca em manter uma boa relação tanto com o governo, quanto com a população

- **Sinopse**

- Acompanhe a saga de um Presidente da República em busca de ter o melhor mandato da história do Brasil, transitando pelos principais dilemas e escolhas que o cargo demanda e propõe no mundo real.

- **Estrutura narrativa escolhida**

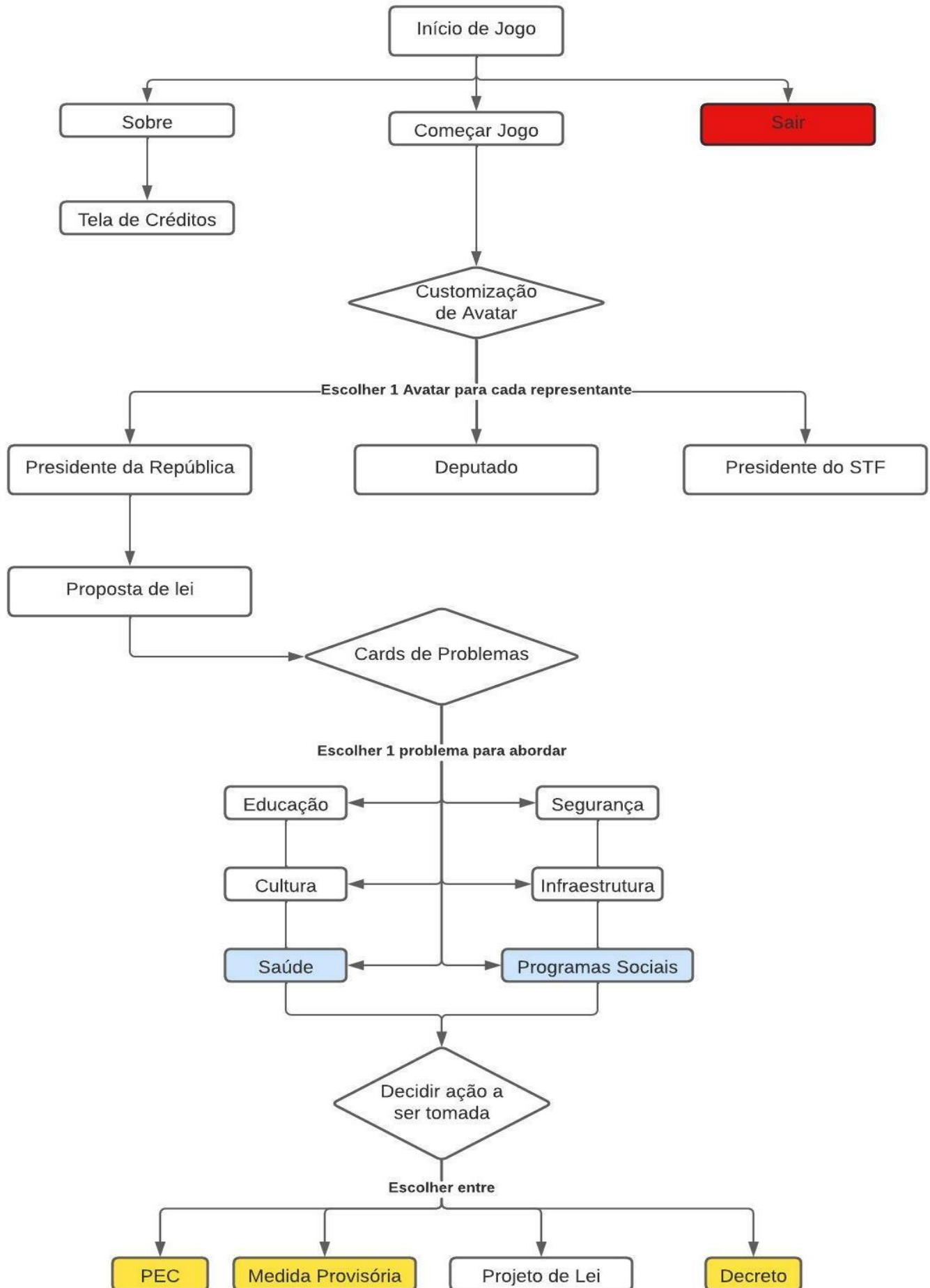
- O jogo será estruturado em uma narrativa na 3ª pessoa.

- **Elementos do roteiro para a estrutura narrativa escolhida** (animação, cut-scenes, McGuffin, diálogos, foreshadowing, inciting incident, etc) – Colocar detalhes dos elementos escolhidos por fase do jogo

- Há um jornal no qual são veiculadas notícias referentes ao mandato presidencial e as consequências das decisões tomadas pelo jogador, abordando impactos reais na vida dos cidadãos do país, sejam eles bons, sejam ruins.

- **Níveis de interatividade do jogo**

- Interatividade alta, sendo que o jogador é 100% responsável pelo caminho a ser trilhado na narrativa do jogo.



## 2.2 Fluxo do Jogo e Níveis (*os níveis são opcionais*)

Fazer o *flowchart* do jogo (gráfico representando o fluxo do jogo) e descrever que tipo de flowchart escolheu: baseado em ações, em quests, na narrativa etc? Justificar de acordo com o gênero escolhido. Por fim, quanto tempo o jogador deverá despende com o jogo?

- No início do jogo é dada uma ênfase maior para o aprendizado, isto é, “como jogar”. Posteriormente, há um aumento significativo na dificuldade e algum mecanismo de premiação pelos objetivos alcançados. (Descrever esse mecanismo de premiação). Por exemplo, objetos secretos que aparecem de acordo com a eficiência do jogador, ou seja, aparecerá algum item valendo mais pontos se o jogador alcançar uma pontuação excepcional em um determinado tempo. Uma forma para o jogo se tornar mais dinâmico é atribuir características aleatórias para o local e o tempo em que esses objetos secretos serão mostrados.

- Esse caminho será seguido, pois os jogadores de jogos de estratégia se contentam com prêmios escondidos.
- Em relação ao tempo, o jogador poderá despende cerca de 20 minutos

## 2.3 O Mundo do Jogo

### 2.3.1 Locações Principais e Mapa

Descrever as locações principais do jogo e o mapa do mundo dele (não são os mapas das fases e sim do mundo!). Apresentar um *flowchart* do mundo.

- O jogo se passa em 3 cenários principais, que são: planalto executivo, o planalto legislativo, e o judiciário.

### 2.3.2 Navegação pelo Mundo

Descrever como os personagens se movem no mundo criado e as relações entre as locações – utilizar os “gráficos de escolhas”, identificando os pontos chaves do jogo como fase, descoberta de um item importante, chefe da fase etc.

- O jogador se moverá pela interface no decorrer da aprovação do projeto de lei, passando, com um clique, pelos diferentes Poderes Brasileiros, sendo que, em algum momento, será direcionado a um dos minigames propostos, como, por exemplo, um endless runner - referente ao caminho do projeto da área da saúde.

### 2.3.3 Escala

Descrever a escala usada para representar o mundo do jogo. Exemplo: os personagens são minúsculos em relação ao mundo, sendo que as portas são, por exemplo, 2 vezes maiores que a altura deles. Escala 1:2.



- A maior parte do jogo se passará em interfaces, porém a escala irá variar dependendo dos minigames propostos, por esse motivo as escalas estão abaixo:
- endless runner: O personagem se encontra em uma dimensão relativamente pequena em comparação ao mundo do mine-game, sendo uma escala de

### 2.3.4 Ambientação

Condições climáticas do mundo do jogo (se aplicáveis – verão, inverno? Dia ou noite?), condições vegetais, animais. Definir se tais condições serão estáticas (por exemplo, acontecerá uma “chuva” sempre no mesmo ponto) ou dinâmicas (por exemplo, em determinado trecho de uma fase pode aparecer uma “chuva” com uma determinada probabilidade).

- A ambientação do jogo não será muito diversa, o jogo se passará de dia sem mudanças climáticas e sem um cenário interativo.

### 2.3.5 Tempo

Como o tempo (*timer*/contador) será utilizado no jogo, se for o caso.

- O tempo será um mandato de 4 anos em que se passará em turnos de 12 rounds. Cada round é referente a um quadrimestre do ano, em que as decisões tomadas simbolizam a passagem do tempo.


## 2.4 Base de Dados

### 2.4.1 Inventário

Descrever cada um dos itens do jogo, apresentando suas características principais e usadas para programar o jogo (características que de fato afetam o funcionamento do jogo; por exemplo, suponha que o peso seja fator relevante no jogo de forma que o jogador não possa carregar muitos objetos ao mesmo tempo, portanto, listar o peso de cada item é importante porque esse é um fator relevante no jogo). É necessário que se agrupem os itens por similaridade, para facilitar consulta e organização: arma, dinheiro, itens consumíveis (cura, magia etc) entre outros. Fazer uma tabela a distribuição dos objetos do jogo todo. A seguir um exemplo de listagem de itens do jogo.


#### 2.4.1.1 Itens Consumíveis (*opcional*)

Poção Medicinal Pequena


	Descrição	Uma pequena poção que cura um pouco de energia.
	Peso	5

	Valor de cura	10
--	---------------	----

#### Poção Medicinal Média


	Descrição	Uma pequena poção que cura um pouco de energia.
	Peso	10
	Valor de cura	15

#### Poção Medicinal Grande


	Descrição	Uma pequena poção que cura um pouco de energia.
	Peso	20
	Valor de cura	25

### 2.4.1.2 Armamento *(opcional)*


#### Espada de Gelo

	Descrição	Permite congelar o inimigo. Chance de 30%.
	Peso	20
	Ataque	20

#### Espada de Fogo

	Descrição	Permite lançar fogo nos inimigos. Chance de 30%.
	Peso	20
	Ataque	20

#### Espada Imaterial


	Descrição	Permite atacar monstros do tipo <i>Fantasma</i> .
	Peso	15
	Ataque	10

### 2.4.2 Bestiário *(opcional)*


Descrever os inimigos do jogo apresentando, da mesma forma que foi feita para a listagem de itens, os fatores realmente relevantes para o jogo. A seguir alguns exemplos.

### 2.4.2.1 Inimigos Elementais de Água

#### Geleca Azul

	Descrição	Uma geleca azul. Tocar sua pele fria pode te congelar!
	HP	1600
	Defesa	10

#### Peixe Esfomeado

	Descrição	Um peixe faminto e raquítico.
	HP	2500
	Defesa	20

### 2.4.3 Balanceamento de Recursos (opcional)

Apresentar as tabelas de balanceamento aprendidas em sala (depende do estilo do jogo). Apresentar outros aspectos que as tabelas não conseguem solucionar e mostrar as soluções adotadas. A seguir estão os exemplos de tabelas vistas em sala.

#### Enemy Chart

Tipo Inimigo	W*	Fase 1	Fase 2	Fase 3	Fase 4	Fase 5	Fase 6	Total
Geleca Verde	1	3	5	2				10
Geleca Azul	2	1	5	5	5	2		15
Morcego	5			1	5	5	10	21
Flor-bomba	10					2	5	7
Total		5	15	17	35	49	100	

(No caso do *level design*, essa tabela desmembra-se em “inimigos x área por fase”).

\*Lembre-se que devemos colocar a quantidade de itens vezes o seu peso (W). A equação de dificuldade do inimigo ou peso (W) deve levar em consideração diversas

questões, como: sua IA, seu HP, sua resistência, sua velocidade de ataque, itens que pode liberar (*drop*) para o jogador etc.

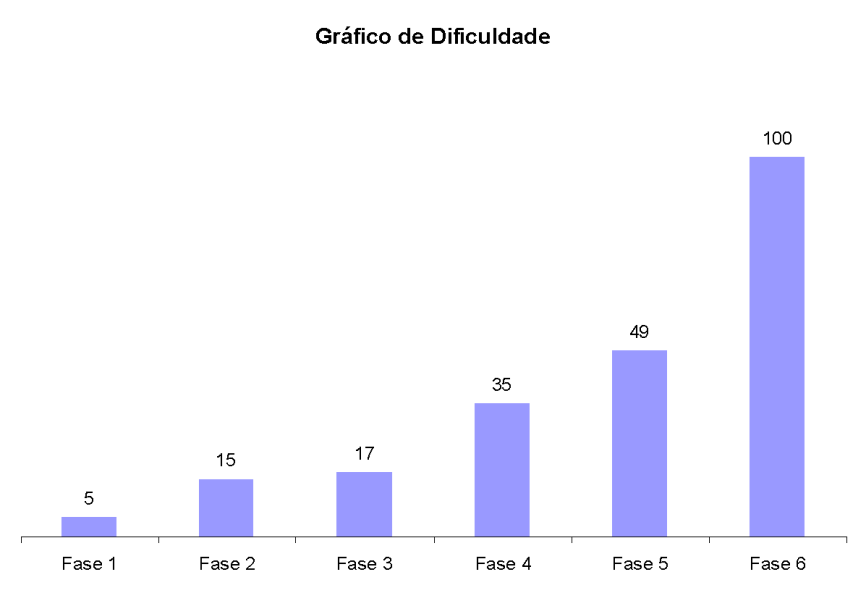


Figura 2. Gráfico de dificuldade para o jogo NOME DO JOGO.

Discutir, baseado no gráfico acima, se o balanceamento escolhido está de acordo com as teorias apresentadas por Mihaly.

Deve repetir a tabela “Enemy chart” para itens, quests, skills etc, ou seja, dependendo do tipo de jogo podem surgir outras tabelas bem como algumas desaparecer.

#### Item Chart

Tipo de Item	Fase 1	Fase 2	Fase 3	Fase 4	Fase 5	Fase 6	Total
Moeda	20	20	50	20	40	50	200
Sorvete	3	5	5	2	2	3	20
Bombinha			2			3	5
Super Estrela	1	1	1	1	1	1	6

---

### 3. Level Design (opcional) <ADALOVE – Implementar mecânicas básicas do jogo - parte 1 a 5>

Apresentação do World Diagram para, logo a seguir, especificar cada fase. Pode-se acrescentar o gameflow nessa parte, levando-se em consideração o tipo de gameflow a ser trabalhado (quest, narrativa, ação etc).

#### 3.1 Fase <NOME DA FASE 1>

##### 3.1.1 Visão Geral (opcional)

Construção do *layout área* dessa fase com informações de quais áreas estão ligadas a quais Descrever o cenário desta fase: onde no mundo fica o local, como o personagem chegou ali, como é a vegetação, a temperatura etc.

Definir a meta (objetivo) do jogador na fase. Detalhar micro metas, se houver.

Descrição de onde o personagem inicia a fase, o que ele deve fazer para concluir a fase.

- Depois da inicialização do game você é direcionado ao main screen do jogo onde se encontra os três edifícios do game, sejam eles poderes- judiciário executivo e legislativo. Seu personagem é levado a um deles por escolha de usuário...

No edifício você pode propor sua lei ou passar por algum processo para sua aprovação para dar continuidade no game para te levar ao objetivo final onde é aprovada ou não.

##### 3.1.2 Layout Área (opcional)

Construção do *layout área* dessa fase com informações de quais áreas estão ligadas a quais áreas, sem se importar com itens ou o formato e detalhes de objetos da área.

—

##### 3.1.2.1 Connections (opcional)

Construção do cenário usando *connections*. Também apresentar, se for o caso, do uso das técnicas de visibilidade de cena adotadas (caixotes obstruindo visão, escadas verticais, corredores/donut rooms, portas, ambientes obscuros etc)

- Cenário de página principal onde se tem sua Hud invisível com os três prédios do poderes na sua frente botão de início/sobre/sair/voltar. no canto superior direito.

### 3.1.2.2 Layout Effects (opcional)

Legenda com informações de efeitos visuais/sons/animações CG no jogo.

- Efeitos visuais aplicados com vitória/conquista/derrota em decorrer do jogo, sons (trilha sonora feita por membro do grupo)

### 3.1.2.3 Quests e Puzzles (opcional)

Construção de quests/puzzles utilizando o quest/puzzle flow.

- Quest: aprovação de lei difícil nos três poderes

### 3.1.3 Balanceamento de Recursos (opcional)

Posicionamento de itens na fase, utilizando o layout area como base. Usar legenda especificando todos os itens/inimigos (pode-se criar uma classificação, por exemplo, de inimigos tipo 1 – fáceis e inimigos tipo 2 – difíceis).

Inserir tabela com os inimigos e a quantidade destes inimigos na fase divididos por área, para controle do balanceamento da mesma. Exemplo:

O jogo não contém combate.

Enemy Chart

Tipo Inimigo	W*	Área1	Área 2	Área 3	Área 4	Área 5	Área 6	Total
Geleca Verde	1	1	1	2		1	2	7
Geleca Azul	2		1		1	2	1	5
Morcego	5						1	1
Flor-bomba	10							0
Total		1	3	2	2	5	9	

Do mesmo modo, inserir tabela com os itens e a quantidade destes itens na fase por área, para controle do balanceamento da mesma.

#### Item Chart

Tipo de Item	Área1	Área 2	Área 3	Área 4	Área 5	Área 6	Total
Moeda	5	2	3	5			15
Sorvete	1		1		1		3
Bombinha			2			1	3
Super Estrela					1		1

#### 3.1.4 The Boss

Descrever o chefe da fase (se houver) e seu comportamento de ataque/defesa, bem como o modo previsto para o jogador derrotá-lo e a recompensa adquirida (power-up, vida, itens diversos etc).

- Temos um chefe fase em uma das etapas da criação de lei, onde a partir de um jogo de perguntas e respostas devemos responder as perguntas de uma maneira correta para que possamos ganhar pontos sobre o adversário, e caso a resposta for errada perderemos pontos. Como recompensa conseguiremos mais aprovação na câmara.

#### 3.1.5 Outros Personagens

Descrever quais e onde estão os NPCs e as ações que eles assumem perante o jogador/situação.

Definir como se dá a interação com o personagem. Criar os diálogos do NPC para a fase.

- O nosso jogo conta com um npc que irá te ajudar durante sua gameplay com dicas e meios de fazer sua jornada mais fácil. Conselheiro

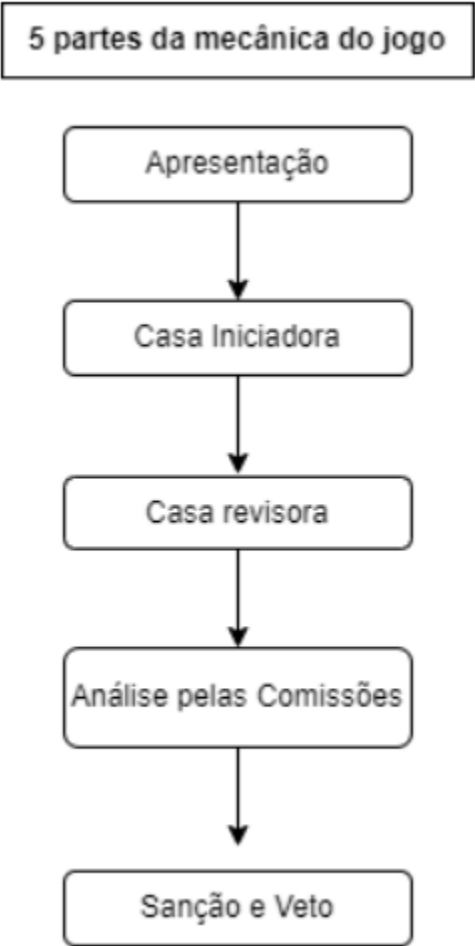
#### 3.1.6 Easter Eggs

Descrever locais/itens secretos na fase (se houver) e a forma para alcançá-los, bem como a recompensa adquirida (power-up, vida, itens diversos etc).

- Se clicar repetidamente no botão de tirar a música te leva a um menu secreto com uma animação, que faz uma referência sutil ao inteli.

---

**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** (qual turno cada um faz parte?)

Personagem	Turno 1	Turno 2	Turno 3	Turno 4	Turno 5	Turno 6	Turno 7



Presidente do Brasil	Propor a lei	Jogar o mini game					Aprovar a lei
Presidente do STF							
Deputado			Levar a lei a Comissão	Debater a lei	Votar a lei	Devolver a lei ao Presidente	
Conselheiro	Auxiliar o jogador a esclarecer os conceitos e esclarecer dúvidas						

## 4.1 Personagens Controláveis

### 4.1.2 <NOME DO PERSONAGEM PRINCIPAL *n*>

#### <Presidente da República do Brasil >

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:

**1) Presidente da República do Brasil:** Personagem principal do jogo e contém a função mais importante, ser controlado pelo jogador. Seu objetivo é conseguir a aprovação do máximo de leis possíveis, levando em conta os desejos da população e daqueles que compõem o governo. Personagem personalizável, o jogador poderá escolher sua etnia e gênero antes do início do jogo.

#### 4.1.2.1 Backstory

A princípio será um personagem com o poder de escolher o rumo do jogo e que tomará as decisões. Seu objetivo é alcançar o maior número de leis aprovadas e adquirir conhecimento político ao longo do jogo. Ele possuirá um “mentor” (conselheiro) que irá auxiliá-lo em suas escolhas e esclarecer possíveis dúvidas e explicar seus erros.

#### 4.1.2.2 Concept Art

Esboços do personagem.

#### 4.1.2.3 Ações Permitidas

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

Criação de leis com o objetivo de melhorar as condições de vida no país/ estabilizar os conflitos internos, consequentemente aumentando a sua reputação.

#### **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 aparecerá em todas as situações onde haverá uma ação do jogador, seja em algum diálogo ou algum minigame. Seu objetivo é se tornar um avatar onde o jogador pode se imaginar e aumentar a imersão do jogo.

### **4.2 Common Non-Playable Characters (NPC)**

#### **4.2.1 <NOME DO NPC COMUM *n*>**

##### **População:**

Para cada NPC comum, descrever sua história, comportamento perante o personagem (agressivo, amigável, indiferente etc), seus dados pessoais como pontos de vida e outros que forem implementados (pontos de magia, habilidades etc). O comportamento do personagem será estático ou dinâmico. Se dinâmico definir como o mesmo deve ser alterado.

Serão uma quantidade significativa de personagens que representam os desejos da população e reagirão conforme as leis propostas pelo presidente sejam aprovadas ou negadas, essas reações podem ser tanto positivas quanto negativas

### **4.3 Special Non-Playable Characters (NPC)**

#### **4.3.1 <Conselheiro>**

Para cada NPC especial (mini-boss, boss, mentor/guia etc), descrever sua história, comportamento perante o personagem (agressivo, amigável, indiferente etc), seus dados pessoais como pontos de vida e outros que forem implementados (pontos de magia, habilidades etc). O comportamento do personagem será estático ou dinâmico. Se +dinâmico definir como o mesmo deve ser alterado.

O conselheiro consiste em um NPC que não é envolvido em batalha, e tem como missão guiar o jogador a tomar as decisões corretas e explicar o porquê das decisões erradas estarem erradas. Ele servirá para direcionar e instruir o personagem principal e, conseqüentemente, o jogador ao conceito de leis e a implementação da mesma.

---

## 5. Teste de Usabilidade e Padrões de qualidade <[ADALOVE](#) – [Desenvolver relatório](#)

[de resultados do playtest](#)>

### 5.1 - Usabilidade

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:**

**Pontos positivos (observados nos testes em geral):**

**Pontos de melhoria (observados nos testes em geral):**

Número do teste:

Nome e perfil dos tester:

**O que observar e perguntar durante o teste:**

**-Observar e registrar:**

Conseguiu começar o jogo? Entendeu a mecânica do jogo? Aprendeu como jogar?

Conseguiu controlar o jogo?

Progrediu no jogo? Passou de fase? Fez pontos? Chegou ao final? Perdeu rápido?

Entendeu as regras do jogo? Teve dificuldade de compreensão? Teve dificuldade ao jogar?

Foi muito fácil? O jogo foi desafiador?

**-Perguntar a quem testou:**

Numa escala de 0 a 10, quanto você se divertiu nesse jogo?

O que você gostou no jogo?

O que poderia melhorar no jogo?

Definir padrões de qualidade

CARACTERÍSTICAS	SUBCARACTERÍSTICAS	SIGNIFICADO
<p><b>Funcionalidade</b></p> <p>O conjunto de funções satisfazem as necessidades explícitas e implícitas para a finalidade a que se destina o produto?</p> <p>→ Sim, a codificação do jogo possibilita tanto que o usuário aprenda sobre o processo de criação de leis dentro da Constituição Brasileira (explícito), quanto que se divirta pelos mini games (implícito).</p>	<p><b>Adequação</b></p>	<p><b>Propõe-se a fazer o que é apropriado?</b></p> <p>→ Mesmo que a ideia do jogo seja diferente daquela esperada, foi aprovada pelo cliente e apresentada em prática.</p>
	<p><b>Acurácia</b></p>	<p><b>Gera resultados corretos ou conforme acordados?</b></p> <p>→ Sim, gera resultados na medida em que ensina o usuário sobre o processo de criação de leis, difundindo conhecimento</p>

		sobre a função de cada setor da política brasileira.
	Interoperabilidade	<p>É capaz de interagir com os sistemas especificados?</p> <p>→ Sim, o programa utilizado, sendo ele o Godot, no momento, apresenta um uso mais específico para Web, sendo possível (futuramente) a adaptação para apps.</p>
	Segurança de acesso	<p>Evita o acesso não autorizado, acidental ou deliberado a programas e dados?</p> <p>→ O código é open source, logo é de fácil acesso para qualquer um, contudo quem pode modificar o</p>

		código é controlado pelos desenvolvedores.
	Conformidade	<p>Está de acordo com normas e convenções previstas em leis e descrições similares?</p> <p>→ Sim, está de acordo com todas as normas jurídicas previstas.</p>
<p>Confiabilidade</p> <p>O desempenho se mantém ao longo do tempo e em condições estabelecidas?</p> <p>→ Tanto o jogo quanto o grupo se mantém fiel a sua ideia inicial, respeitando-a sempre que possível.</p> <p>As únicas mudanças que ocorreram</p>	Maturidade	<p>Com que frequência apresenta falhas?</p> <p>→ O grupo apresenta eventuais discussões e diferentes visões, contudo nada sério e nem levado para o lado pessoal</p>

<p>foram para a melhora/ melhor aproveitamento, mantendo sempre um bom e constante desempenho, com pouca oscilação.</p>	<p><b>Tolerância a falhas</b></p>	<p><b>Ocorrendo falhas como ele reage?</b></p> <p>Com resiliência, o grupo se dispõe a procurar o problema em grupo e, quando nosso conhecimento é insuficiente para resolução, buscamos orientação dos professores.</p>
	<p><b>Recuperabilidade</b></p>	<p><b>É capaz de recuperar dados após uma falha?</b></p> <p>→ De acordo com os comentários e o salvamento do jogo em diferentes formas (nuvem do GitHub e computadores dos desenvolvedores), é possível a recuperação dos dados do jogo.</p>



		<p>Em questão de salvamento de dados dentro do jogo, será necessário a implementação de um banco de dados que suporte esse quesito.</p>
<p><b>Usabilidade</b></p> <p>É fácil utilizar o software?</p> <p>→ A nossa interface é intuitiva, sendo assim, fácil para o usuário se locomover pelo jogo e suas mecânicas, tornando o software de fácil utilização.</p>	<p><b>Inteligibilidade</b></p>	<p>É fácil entender os conceitos utilizados?</p> <p>→ Sim, nosso jogo é intuitivo e de fácil entendimento da interface.</p> <p>→ Sem falar que a maior parte do código é a movimentação entre interfaces. Sendo assim, grande parte dos conceitos utilizados no código é de fácil utilização</p>

	<p><b>Apreensibilidade</b></p>	<p><b>É fácil aprender a usar?</b></p> <p>→ Nosso jogo tem uma mecânica básica e já conhecida, sendo assim, fácil a adaptação para aprender as mecânicas do nosso jogo.</p>
	<p><b>Operacionalidade</b></p>	<p><b>É fácil de operar e controlar a operação?</b></p> <p>→ Sim, o nosso jogo é maior parte movido pelo mouse para se movimentar entre as interfaces, com exceção dos minigames, como por exemplo o “endless runner”, que está implementado onde o jogador somente usa a</p>

		<p>barra de espaço para pular e</p> <p>completar o objetivo.</p>
<p><b>Eficiência</b></p> <p>Os recursos e os tempos utilizados são compatíveis com o nível de desempenho requerido para o produto?</p> <p>→ O processamento dos recursos do jogo em relação ao tempo é extremamente bom, o desempenho requerido é simples e rápido como nosso produto entrega.</p>	<p><b>Comportamento em relação ao tempo</b></p>	<p><b>Qual é o tempo de resposta e de processamento?</b></p> <p>→ O jogo possui um tempo de resposta rápido, uma vez que se trata de um jogo simples (a respeito das mecânicas).</p>
	<p><b>Comportamento em relação aos recursos</b></p>	<p><b>Quanto recurso utiliza?</b></p> <p>→ O jogo no momento é extremamente leve, sem muitos requerimentos para o usuário.</p>

<p><b>Manutenibilidade</b></p> <p>Há facilidade para correções, atualizações e alterações?</p> <p>→ Sim, o código concorda com as éticas de programação e é muito bem comentado.</p>	<p><b>Analisabilidade</b></p>	<p><b>É fácil encontrar uma falha quando ocorre?</b></p> <p>→ Sim, por facilitação do Godot, não há dificuldades em identificar os problemas.</p>
	<p><b>Modificabilidade</b></p>	<p><b>É fácil modificar e remover defeitos?</b></p> <p>→ Requer análise da estrutura do código, tanto da mensagem de erro do Godot, quanto da possível mudança no código, porém, as mudanças não afetam o resto do código de uma maneira com que o mesmo pare de funcionar.</p>

	Estabilidade	<p>Há grandes riscos de bugs quando se faz alterações?</p> <p>→ Na estrutura atual, não se apresentam bugs.</p>
	Testabilidade	<p>É fácil testar quando se faz alterações?</p> <p>→ A transição de desenvolvimento para teste é quase imediata.</p>
<p>Portabilidade</p> <p>É possível utilizar o produto em diversas plataformas com pequeno esforço de adaptação?</p>	Adaptabilidade	<p>É fácil adaptar a outros ambientes sem aplicar outras ações ou meios além dos fornecidos para esta finalidade no software considerado?</p>

<p>→ Deverá haver uma análise de posicionamento, porém com essas alterações feitas é fácil de realizar.</p>		<p>→ O jogo não foi produzido pensando em plataformas além da Web e Windows. Para se adaptar a outras plataformas (como a plataforma Mobile, Android e iOS), seria necessário uma análise da ferramenta Godot e uma reapropriação dos recursos.</p>
	<p>Capacidade para ser instalado</p>	<p>É fácil instalar em outros ambientes?</p> <p>→ Utilizando a ferramenta Godot para adequar o jogo desenvolvido a outras plataformas, não se apresenta problemas neste quesito.</p>

	Capacidade para substituir	<p>É fácil substituir por outro software?</p> <p>→ Pode ser alterado e adaptado de forma fácil.</p>
	Conformidade	<p>Está de acordo com padrões ou convenções de portabilidade?</p> <p>→ O desenvolvimento foi feito de acordo com as normas e éticas de código (Indentação, Comentários, etc.).</p>

## 6. Relatório - Física e Matemática

### 6.1 Funções

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.

<[ADALOVE - Definir as funções matemáticas que serão utilizadas no jogo](#)>

→ As funções matemáticas utilizadas no **mini game** estão presentes no PULO do personagem e na GRAVIDADE exercida nele. (O game principal ainda não possui estruturas que exijam funções matemáticas)

O exemplo matemático usado foi do Lançamento Vertical (sendo esse o do PERSONAGEM) e o Lançamento Horizontal (que tecnicamente seria do CENÁRIO, o qual se move naturalmente ao decorrer do jogo), sendo um exemplo bem parecido com o jogo do “Dinossauro do Google”, de quando cessa a internet.

EQUAÇÕES MATEMÁTICAS: **1. Verticais** - Pulo do personagem / Gravidade exercida **2. Horizontais** - Cenário

- Obstáculos e objetos: personagem estará na posição X para uma velocidade de 6.3 do cenário. Isso formará uma escala de 1: 6.3, pois a cada posição da personagem o cenário estará rolando a esta velocidade.

O jogador deve fazer o caminho de entrar no jogo, onde ele será apresentado com o menu principal, e clicando em “Começar Jogo”, ele é levado a cena dos 3 poderes, onde ele terá que clicar no prédio do poder executivo, clicar no botão de nova proposta, clicar no botão saúde, onde ele será transportado de volta para o menu do executivo, e então ele poderá clicar no botão “Imprensa” e entrar no jogo “Corrida Pela Saúde”.

No minigame, quando o usuário pressionar a barra de espaço, o código entende que a função “Pulo” foi apertada, o código faz uma passagem de estados (do estado “CORRER” para o estado “PULAR”) e executa o seguinte código

```
velocidade = Vector2.ZERO
velocidade.y -= forca_pulo
animacao.play("Pulo") e
```

```
1  extends KinematicBody2D
2
3  var velocidade = Vector2.ZERO
4  >| ## Criamos a variável velocidade, e como o jogo é um "auto-runner" (corre
5  >| ## automaticamente), nós queremos que a velocidade inicial seja 0, logo,
6  >| ## utilizamos do Vector2, que é a referência cartesiana do Godot.
7
8  export var forca_pulo = 600.0
9  export var gravidade = 20.0
10 >| ## Utilizamos variáveis exportadas pois queremos alterar elas sem ter que
11 >| ## voltar no script, fazendo assim com que elas possam ser alteradas
12 >| ## diretamente pelo "Inspector", embaixo de "Script Variables". Esse método
13 >| ## é de comum utilidade em testes.
14
15 enum {
16 >| ## enum é a função de enumerador. Estamos utilizando dela para fazer 3 estados
17 >| ## possíveis em que o jogador pode se encontrar; Pulando, Correndo
18 >| ## ou no meio do ar.
19 >| PULAR,
20 >| CORRER,
21 >| AR
22 }
23
24 var estado = CORRER
25
```



```

26 onready var animacao = $AnimatedSprite
27 >I ## Identificamos o nó de animação no código utilizando o símbolo do sifrão
28 >I ## e o guardamos em uma variável
29
30 ▾ func _ready():
31 >I pass
32
33 ▾ func _physics_process(delta):
34 >I ## Função do Godot que cuida das propriedades físicas de acordo com o
35 >I ## FPS (Frames por segundo)
36 ▾ >I match estado:
37 >I >I ## Aqui definimos o que cada estado fará em questão de código
38 ▾ >I >I CORRER:
39 >I >I >I animacao.play("Corrida")
40 >I >I >I >I ## Animação para o personagem correr.
41 ▾ >I >I PULAR:
42 >I >I >I velocidade = Vector2.ZERO
43 >I >I >I >I ## Aqui, é bom ter um reset da velocidade, para que o pulo não saia
44 >I >I >I >I ## como algo na diagonal.
45 >I >I >I velocidade.y -= forca_pulo
46 >I >I >I >I ## Como já foi explicado, o plano cartesiano se encontra "invertido"
47 >I >I >I >I ## então, para irmos para cima, precisamos subtrair a força do pulo.
48 >I >I >I animacao.play("Pulo")
49 >I >I >I >I ## Utilizamos o bloco de animação que foi criado no

```

```

49 >I >I >I >I ## Utilizamos o bloco de animação que foi criado no
50 >I >I >I >I ## Animated Sprite > Frames > Sprite Frames, e chamado de
51 >I >I >I >I ## "Pulo".
52 >I >I >I estado = AR
53 >I >I >I >I ## Se o jogador pulou, ele está no meio do ar.
54 ▾ >I >I AR:
55 >I >I >I pass
56 >I velocidade.y += gravidade
57 >I >I ## Em 2D, graças a inversão do Godot, caso tenhamos um y positivo,
58 >I >I ## nosso personagem se encontrará na direção abaixo, e caso negativo,
59 >I >I ## ele se encontrará acima. Logo, o efeito da gravidade será positivo
60 >I >I ## ao invés de negativo, então utilizamos a soma.
61 >I move_and_collide(velocidade * delta)
62 >I >I ## O "move_and_collide" serve para realizar o movimento do personagem,
63 >I >I ## e aqui, multiplicamos a velocidade pelo delta, que é uma maneira de
64 >I >I ## fazer com que a velocidade escale de acordo com o FPS
65
66 ▾ func _input(event):
67 >I ## Detecta qualquer input que entra, dentro de seu próprio loop no Godot.
68 >I ## Aqui, estamos utilizando ela para eventos, com a função básica de
69 >I ## assistir aos botões do teclado.
70 ▾ >I if estado == CORRER:
71 >I >I ## Se o jogador estiver no chão
72 ▾ >I >I if event.is_action_pressed("Pulo"):
73 >I >I >I ## Adicionamos uma condição que faz com que toda vez que o botão de

```

```

73 >| >| >| ## Adicionamos uma condição que faz com que toda vez que o botão de
74 >| >| >| ## pulo (configurado no menu Project > Project Settings > Input Map)
75 >| >| >| ## for pressionado, algo aconteça.
76 >| >| >| estado = PULAR
77 >| >| >| >| ## Se o jogador pressionar o botão de pular, ele está pulando.
78
79
80 →| func _on_Pes_body_entered(body):
81 >| ## Aqui, o nó "Pes" checa se algum outro corpo de colisão entrou em
82 >| ## contato com ele. Neste caso, o nó representa o pé do jogador. E nossa
83 >| ## intenção é saber se os pés dele estão no chão.
84 →| if body is StaticBody2D:
85 >| >| ## Aqui nós checamos se o corpo que está em contato com os pés do
86 >| >| ## jogador é o StaticBody2D, que, na cena "Frente" representa a caixa
87 >| >| ## de colisão do chão.
88 >| >| estado = CORRER
89 >| >| >| ## Se os pés estão no chão, o jogador está correndo.
90
91
92 →| func _on_Pes_body_exited(body):
93 >| ## Aqui, o nó "Pes" checa se algum corpo que estava em colisão com ele
94 >| ## deixou de estar. Neste caso, o nó representa o pé do jogador. E nossa

```

```

92 →| func _on_Pes_body_exited(body):
93 >| ## Aqui, o nó "Pes" checa se algum corpo que estava em colisão com ele
94 >| ## deixou de estar. Neste caso, o nó representa o pé do jogador. E nossa
95 >| ## intenção é saber se os pés dele saíram do chão.
96 →| if body is StaticBody2D:
97 >| >| ## Aqui nós checamos se o corpo que parou de entrar em contato com os pés ## do jogador é o StaticBody2D, que, na cena "Frente" representa a
98 >| >| ## caixa de colisão do chão.
99 >| >| estado = PULAR
100 >| >| >| # Se os pés estiverem no ar, o jogador está pulando.
101

```

---

Utilizamos da documentação de Shaders do Godot

([https://docs.godotengine.org/en/3.0/tutorials/shading/shading\\_language.html](https://docs.godotengine.org/en/3.0/tutorials/shading/shading_language.html)) para fazer o código.

- A função utiliza de uma variável (velocidade\_rolamento) para utilizar de incremento no código, utilizando de UVs (mapeamento de pixels), guardados na variável u, incrementados no eixo X, e ocorrendo de acordo com o tempo (TIME)

```

1  ▾ shader_type canvas_item;
2  ▸  /*
3  ▸  Aqui definimos o tipo de Shader, o tipo de Shader para jogos 2D é
4  ▸  sempre canvas_item no Godot.
5  ▸  */
6
7  ▾ uniform float velocidade_rolamento;
8  ▸  /*
9  ▸  Aqui temos uma variável uniforme, do tipo de dado float, onde o que
10 ▸  declaramos é a velocidade de rolamento da tela. A função dessa linha de
11 ▸  código é, basicamente, para mudar o valor no "inspector", e a partir daí,
12 ▸  poderemos manipular o script tanto pelo inspector, quanto pelo código.
13 ▸  */
14 ▾ void fragment()
15 ▸  /*
16 ▸  Aqui utilizamos a shader "fragment", que tem como função executar em cada pixel
17 ▸  da cena que contem o nó "FundoInfinito", e então fará algumas modificações
18 ▸  para que ela consiga mover os UVs (Um mapeamento de onde certos pixels da
19 ▸  textura estão localizados no componente).
20 ▸  */
21 ▾ {

```

```

22 ▾ ▸  vec2 u = UV;
23 ▸  ▸  /*
24 ▸  ▸  Aqui estamos pegando o UV e o colocando em uma variável
25 ▸  ▸  */
26 ▾ ▸  u.x += velocidade_rolamento * TIME;
27 ▸  ▸  /*
28 ▸  ▸  Permite a velocidade do rolamento ser incrementada conforme o tempo, no eixo X
29 ▸  ▸  */
30 ▾ ▸  vec4 cor = texture(TEXTURE, u);
31 ▸  ▸  /*
32 ▸  ▸  Aqui utilizamos vec4, que é a cor, e armazenamos em uma variável, e então
33 ▸  ▸  a colocamos na textura, utilizando a coordenada UV. Basicamente, pegamos um
34 ▸  ▸  fragmento da textura.
35 ▸  ▸  */
36 ▾ ▸  COLOR = cor;
37 ▸  ▸  /*
38 ▸  ▸  Aqui, atribuímos a variável de volta a cor, utilizando o fragmento que
39 ▸  ▸  pegamos. Funcionalmente, resetando o plano de fundo.
40 ▸  ▸  */
41 ▾ }

```

## 6.2 Cinemática Unidimensional

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.

<[ADALOVE - Aplicar os conceitos matemáticos no jogo](#)>

## 6.3 Vetores

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.).*

<[ADALOVE - Definir as funções matemáticas que serão utilizadas no jogo](#)>

## 6.4 Cinemática Bidimensional e mais

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.

Utilizamos a Gravidade, Força pulo e Velocidade

```
7 var velocidade = Vector2.ZERO
8 # Criamos a variável velocidade, e como o jogo é um "auto-runner" (corre automaticamente), nós queremos que a velocidade inicial seja 0, logo, utilizamos do Vector2, que é a referência cartesiana do Godot.
9
10 export var força_pulo = 650.0
11 export var gravidade = 25.0
12 # Utilizamos variáveis exportadas pois queremos alterar elas sem ter que voltar no script, fazendo assim com que elas possam ser alteradas diretamente pelo "Inspector", embaixo de "Script Variables". Esse método é de comum utilidade em testes.
13
14
15
16 enum {
17     # enum é a função de enumerador. Estamos utilizando dela para fazer 3 estados possíveis em que o jogador pode se encontrar; Pulando, Correndo ou no meio do ar.
18     PULAR,
19     CORRER,
20     AR
21 }
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38 func _physics_process(delta):
39     # Função do Godot que cuida das propriedades físicas de acordo com o FPS (Frames por segundo)
40     match estado:
41         # Aqui definimos o que cada estado fará em questão de código
42         CORRER:
43             animacao.play("Corrida")
44             # Animação para o personagem correr.
45         PULAR:
46             velocidade = Vector2.ZERO
47             # Aqui, é bom ter um reset da velocidade, para que o pulo não saia como algo na diagonal.
48             velocidade.y = força_pulo
49             # Como já foi explicado, o plano cartesiano se encontra "invertido" então, para irmos para cima, precisamos subtrair a força do pulo.
50             animacao.play("Pulo")
51             # Utilizamos o bloco de animação que foi criado no Animated Sprite > Frames > Sprite Frames, e chamado de "Pulo".
52             estado = AR
53             # Se o jogador pulou, ele está no meio do ar.
54         AR:
55             pass
56             velocidade.y += gravidade
57             # Em 2D, graças a inversão do Godot, caso tenhamos um y positivo, nosso personagem se encontrará na direção abaixo, e caso negativo, ele se encontrará acima. Logo, o efeito da gravidade será positivo ao invés de negativo, então utilizamos a soma.
58             move_and_collide(velocidade * delta)
59             # O "move_and_collide" serve para realizar o movimento do personagem, e aqui, multiplicamos a velocidade pelo delta, que é uma maneira de fazer com que a velocidade escale de acordo com o FPS
60
61 func _input(event):
62     # Detecta qualquer input que entra, dentro de seu próprio loop no Godot. Aqui, estamos utilizando ela para eventos, com a função básica de assistir aos botões do teclado.
63     if estado == CORRER:
64         # Se o jogador estiver no chão
65         if event.is_action_pressed("Pulo"):
66             # Adicionamos uma condição que faz com que toda vez que o botão de pulo (configurado no menu Project > Project Settings > Input Map) for pressionado, algo aconteça.
67             estado = PULAR
68             # Se o jogador pressionar o botão de pular, ele está pulando.
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```

71 ~ func _on_Pes_body_entered(body):
72 ~ | ## Aqui, o nó "Pes" checa se algum outro corpo de colisão entrou em contato com ele. Neste caso, o nó representa o pé do jogador. E nossa intenção é saber se os pés dele estão no chão.
73 ~ | if body is StaticBody2D:
74 ~ | | ## Aqui nós checamos se o corpo que está em contato com os pés do jogador é o StaticBody2D, que, na cena "Frente" representa a caixa de colisão do chão.
75 ~ | | estado = CORRER
76 ~ | | | ## Se os pés estão no chão, o jogador está correndo.
77
78
79 ~ func _on_Pes_body_exited(body):
80 ~ | ## Aqui, o nó "Pes" checa se algum corpo que estava em colisão com ele deixou de estar. Neste caso, o nó representa o pé do jogador. E nossa intenção é saber se os pés dele saíram do chão.
81 ~ | if body is StaticBody2D:
82 ~ | | ## Aqui nós checamos se o corpo que parou de entrar em contato com os pés do jogador é o StaticBody2D, que, na cena "Frente" representa a caixa de colisão do chão.
83 ~ | | estado = PULAR
84 ~ | | | ## Se os pés estiverem no ar, o jogador está pulando.
85

```

<ADALOVE - Fazer orientação espacial (2D)>

---

## 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.