

DOCUMENTO PARA DESIGN DE GAMES

PROBODY

Autores: Elisa de Oliveira Flemer

Giovanna Rodrigues

João Furquim de Campos Suarez

Marcos Vinicius Alves de Moura

Thomas Barton

Yasmin Vitória Rocha de Jesus

Data de criação: 11 de março de 2022

Versão: 0.4

Controle do Documento

Histórico de revisões

Data	Autor	Versão	Resumo da atividade
10/02/2022	Elisa de Oliveira Flemer	0.1	Preenchi 1.1 a 1.5
17/02/2022	Elisa de Oliveira Flemer	0.2	Preenchi 1.6 a 1.7
25/02/2022	Elisa de Oliveira Flemer	0.3	Revisei 1.7, preenchi 2, 3, 4, 6.1 e 6.3
11/03/2022	Elisa de Oliveira Flemer	0.4	Atualizamos 1.4.7 e preenchemos o 6.2/6.4

Sumário

Controle do Documento	2
Histórico de revisões	2
1. Visão Geral do Projeto <ADALOVE – Definir Proposta de Valor>	5
1.7 Requisitos do Jogo <ADALOVE – Documentar requisitos>	10
1.7.1 Requisitos coletados na entrevista com o cliente	10
1.7.2 Persona	10
1.7.3 Gênero do jogo	11
1.7.4 Histórias do jogo (Game stories) ou Histórias dos usuários (user stories)	12
1.7.5 Mecânica	15
Descrever a forma de interação do jogo (mecânica), nos aspectos: personificação, dimensão, período, conectividade, visão.	15
1.7.6 Fontes de Pesquisa / Imersão	15
2. Game Design	17
2.1 História do Jogo	17
2.2 Fluxo do Jogo e Níveis	18
2.3 O Mundo do Jogo	18
2.3.1 Locações Principais e Mapa	18
2.3.2 Navegação pelo Mundo	19
2.3.3 Escala	19
2.3.4 Ambienteção	19
2.3.5 Tempo	20
2.4 Base de Dados	20
2.4.1 Inventário	20
2.4.1.1 Itens Consumíveis	20
2.4.2 Bestiário (opcional)	22
2.4.3 Balanceamento de Recursos	23
3. Level Design <ADALOVE – Implementar mecânicas básicas do jogo - parte 1 a 5>	25

3.1 Fase <NOME DA FASE 1>	25
3.1.1 Visão Geral (opcional)	25
3.1.2 Layout Área (opcional)	25
3.1.2.1 Connections (opcional)	25
3.1.2.2 Layout Effects (opcional)	25
3.1.2.3 Quests e Puzzles (opcional)	25
3.1.3 Balanceamento de Recursos (opcional)	25
3.1.4 The Boss	26
3.1.5 Outros Personagens	27
3.1.6 Easter Eggs	27
4. Personagens <ADALOVE – Implementar mecânicas básicas do jogo - parte 1 a 5	28
4.1 Personagens Controláveis	28
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:	28
4.1.1 Backstory	28
4.1.2 Concept Art	29
Esboços do personagem.	29
4.1.3 Ações Permitidas	29
4.1.4 Momento de Aparição	29
4.2 Common Non-Playable Characters (NPC)	29
4.2.1 Agentes Cancerígenos	29
4.3 Special Non-Playable Characters (NPC)	30
4.3.1 Âncora	30
4.3.2 Corpo	30
5. Qualidade de Software	31
5.1 Teste de Usabilidade <ADALOVE – Desenvolver relatório de resultados do playtest>	31
5.2 Normas de Qualidade <ADALOVE - Definir padrões de qualidade>	31
6. Relatório - Física e Matemática	33

6.1 Funções	33
6.2 Cinemática Unidimensional	34
6.3 Vetores	38
6.4 Cinemática Bidimensional e mais	39
7. Bibliografias	43
Apêndice	44

1. Visão Geral do Projeto <[ADALOVE – Definir Proposta de Valor](#)>

1.1 Objetivos do Jogo

Definir o objetivo do desenvolvimento desse projeto. Dentre esses, devem ser considerados:

- Desafios e interesses pessoais no tipo de jogo criado
- Para que serve o jogo
- Para que o jogo está sendo criado
- Visão geral e contexto do jogo
- Contexto onde este jogo está sendo criado (justificar que é feito num bootcamp, que vai gerar um produto para o bootcamp)

Proboby é um simulador de estratégia biomédica. Nesse sentido, objetiva ensinar seu público-alvo—isto é, adolescentes—, as principais causas, fatores de risco e modos de prevenção de cânceres. Para esse fim, Proboby coloca o jogador nas fronteiras do sistema imunológico, trabalhando ativamente para destruir, conter e mitigar substâncias cancerígenas antes que elas atinjam as células do personagem principal. Ademais, cabe também ao jogador monitorar diferentes partes do corpo, associando certas atividades com uma maior presença de carcinógenos em órgãos relacionados, e selecionar os melhores hábitos saudáveis para contrabalancear um dado fator de risco. Porém, caso falhe em combater os carcinógenos, o jogador ainda terá a chance de recuperar a partida através de uma fase extra ou da troca de pontos por procedimentos de detecção precoce do câncer, como a mamografia ou o exame de próstata. Assim, procura-se demonstrar, também, que o diagnóstico de neoplasia não é uma sentença de morte, e sim apenas um obstáculo com grandes chances de ser superado com o acompanhamento correto.

A principal motivação para este projeto é o fato de que 30% a 50% dos casos de carcinoma são preveníveis, segundo a Organização Mundial da Saúde. Isso, somado à estatística de que 9,6 milhões de pessoas morreram dessa doença em 2018, justifica inequivocamente a necessidade de ferramentas lúdicas que alcancem gerações mais jovens quanto à prevenção da neoplasia.

Por fim, informa-se que este trabalho está sendo desenvolvido por uma equipe de alunos de 1º ano do Instituto de Tecnologia e Liderança para a USP-Medicina. Assim, servirá, adicionalmente, como objeto de avaliação do bimestre, portfólio para o Inteli e produto mínimo viável a ser possivelmente expandido pelo time tecnológico do cliente.

1.2 Características gerais do Jogo

Elaborar uma síntese geral do jogo. Contextualização geral do jogo.

Proboby gira em torno de um único personagem, seus maus hábitos e as consequências destes em seu corpo. O jogador, nesse sentido, é incumbido de manter o protagonista saudável através da mitigação manual de carcinógenos e da aquisição de comportamentos saudáveis. Para tanto, um contêiner na tela principal, acompanhado da figura de um âncora de TV para efeitos cômicos, relata continuamente o que o personagem tem feito, como fumar, alimentar-se mal, etc. Essas ações levam à entrada de agente cancerígenos em órgãos específicos, como o pulmão no caso do tabaco. Se nada for feito, esses agentes danificarão os tecidos do órgão até iniciarem um processo oncológico no corpo.

Assim, a cada ação prejudicial, o jogador é instigado a recordar-se de qual área ela afeta e dirigir-se a ela para tomar providências. Por exemplo, caso o contêiner indique que “o protagonista comeu salsichas no almoço”, o jogador deve se lembrar de que alimentos processados aumentam as chances de câncer no intestino. Nesse contexto, para conter os malefícios, será necessário clicar na região do intestino do personagem para trocar de cena e então combater os invasores desse órgão.

Nessa segunda tela, várias substâncias e moléculas se movimentarão em padrões pseudoaleatórios. Dessa forma, caberá ao jogador, controlando um linfócito denominado “Celinha”, identificar os carcinógenos por sua cor e formato e destruí-los através de colisões. A cada clique correto, um ponto será ganho. Esses pontos, por sua vez, poderão ser trocados por bons hábitos que trarão benefícios visíveis à jogabilidade do usuário.

Entretanto, caso o jogador não consiga deter as moléculas cancerígenas a tempo, elas começarão a se acumular nas laterais das células. Esse acúmulo representa a passagem gradual destas para o interior de células saudáveis. A esse ponto, o jogador ainda pode detê-las com mecanismos táticos variados, a aparecer na tela. Se ele falhar, as moléculas ativarão os proto-oncogenes dentro do núcleo das células, iniciando a segunda fase da oncogênese.

Nesse estágio, as células afetadas começaram a ficar vermelhas conforme mais e mais carcinógenos as penetram. Quando elas ficarem vermelhas por completo, terão se tornado completamente malignas e o jogo acabará com a mensagem “a célula foi comprometida”. Para evitar isso, é preciso interromper o contato das células com carcinógenos o mais rápido possível. É nesse momento que o jogador deverá construir uma estratégia para priorizar as áreas mais afetadas e investir seus pontos em hábitos que possam aliviá-las.

Ganha-se ao alcançar ou superar uma determinada taxa de saúde, calculado a partir dos atuais hábitos do personagem e condições biológicas (por exemplo, se há proto-oncogenes ativados ou uma grande quantidade de carcinógenos presentes em certo órgão).

1.3 Público-alvo

Descreve o tipo de público a que se destina o jogo.

Jovens brasileiros entre 14 e 17 anos de todos os gêneros. São predominantemente das classes sociais DE e C1, com renda familiar até R\$2000,00 e acesso a smartphones ou computadores. Frequentam escola pública. Geralmente não investem em jogos ou consoles, preferindo baixar versões gratuitas, especialmente por conta do preço, os quais consideram muito alto. Ainda assim, jogam de 8 a 20 horas por semana e quase todos os dias, segundo dados do PGB2021.

1.4 Diferenciais

Descrever os diferenciais competitivos do jogo.

- Gerenciamento de tempo no contexto de prevenção de doenças: pouquíssimos jogos colocam o jogador na função não de médico, mas de sistema imunológico, em constante monitoramento do corpo humano. A natureza viciante de jogos de tempo com vitórias desafiadoras combinada com o inusitado cenário médico torna este jogo único.
- Valor educativo: a verossimilhança médica faz deste jogo uma introdução lúdica e medicamente correta aos efeitos biológicos maus hábitos e suas consequências. Através da exploração gradual de conceitos, o jogador se sentirá cada vez mais apto a pesquisar, aprender e interagir com materiais teóricos de neoplasia.

- Gratuito e open-source em todas as plataformas disponibilizadas: Proboddy é plenamente acessível ao oferecer experiências gratuitas para web, desktop e mobile. Além disso, o fato de ser open-source garante a integridade do código-fonte para aqueles que forem baixá-lo.

1.5 Análise do cenário: Matriz SWOT

Apresenta a Matriz SWOT.

Forças: <ul style="list-style-type: none"> • Verossimilhança médica • Ensino didático e gradual de prevenção ao câncer • Aprofundamento considerável em como o câncer se desenvolve no corpo, abrangendo desde moléculas carcinogênicas e proto-oncogenes até a metástase em si. • Partidas são relativamente rápidas 	Oportunidades: <ul style="list-style-type: none"> • Com exceção de “Bio Inc”, há pouca competição em jogos de estratégia médica e prevenção. • Potencial de expansão para outras doenças preveníveis • Potencial de monetização através da venda de boosters e pontos extras.
Fraquezas: <ul style="list-style-type: none"> • Mecânica semelhante ao jogo “Bio Inc” • Não estará disponível na Play ou App Store • Pouco conhecimento técnico da equipe desenvolvedora • Riscos de prejudicar o software ao cometer erros no GitHub • A equipe desenvolvedora é menor 	Ameaças: <ul style="list-style-type: none"> • Falta de conhecimento sobre o que outros grupos estão fazendo, de modo que não podemos garantir que nossa ideia é original. • A abordagem de assuntos sensíveis (no caso, câncer) pode ofender alguns jogadores. • Contribuições open-source futuras podem desvirtuar a visão original para o projeto.

1.6 Proposta de Valor: Value Proposition Canvas

Apresenta o Canvas de Proposta de Valor.

Proboby é um simulador de estratégia biomédico que visa informar jovens de forma lúdica acerca da prevenção contra o câncer.



1.7 Requisitos do Jogo <ADALOVE – Documentar requisitos>

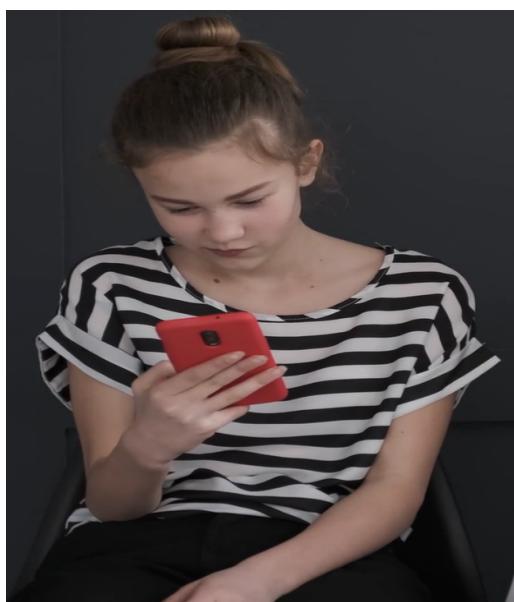
1.7.1 Requisitos coletados na entrevista com o cliente

Apresenta um resumo das respostas da entrevista com o parceiro de mercado. 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 deve:

- basear-se em uma ideia inovadora;
- direcionar-se para um público-alvo de 14 a 17 anos, estudante de escola pública;
- equilibrar o lúdico e o educativo para engajar os jogadores;
- conscientizar os jovens sobre bons hábitos para a prevenção de câncer;
- utilizar a mecânica de “ganhar” e “perder” para evidenciar os benefícios e malefícios de certos hábitos;
- impactar os jogadores positivamente, motivando-os a praticar atividades físicas, alimentar-se de forma saudável, limitar exposição solar prolongada, evitar o consumo de substâncias carcinogênicas, entre outros;
- encorajar o jovem a conscientizar a família acerca da prevenção contra o câncer;
- tratar do tema de forma sensível, excluindo situações muito dramáticas, especialmente se envolverem óbito;
- evitar passar ao jogador um sentimento de culpabilidade quanto ao surgimento de câncer.

O jogo pode envolver linfócitos, principalmente se atrelá-los ao jogador, criando a metáfora de “jogador como linfócito da família, monitorando a saúde”.



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).

Laura, 14 anos, estudante do primeiro ano do Ensino Médio do sistema escolar público. Moradora de Santana, na cidade de São Paulo. Jogadora casual em mobile.

Objetivos com o jogo: distrair-se e passar o tempo sem internet.

Dores: jogos muito longos e que exigem demais do jogador, isto é, demasiado viciantes. Jogos que demandam muita memória do celular.

Tipos de jogos preferidos: simuladores de estratégia. Interesses: conversar por Whatsapp, assistir a séries e animes, ler mangás e acompanhar canais de gameplay.

Biografia: Laura cresceu durante o auge dos youtubers gamers e acompanha muitos canais até hoje. Jogou muito Crossy Roads, Criminal Case e a linha Papa's quando criança, de modo que atualmente gosta de games de gerenciamento de tempo e simulação.

Conhecimentos sobre câncer: conhece apenas os fatores de risco mais comuns, como o hábito de fumar e a exposição prolongada ao sol. Entende que câncer é equivalente a tumor, mas não sabe como ele surge nem como se desenvolve no organismo. Tem interesse no assunto, mas não o estudaria por conta própria.

1.7.3 Gênero do jogo

Definir o(s) gênero(s) do jogo, justificando a escolha de acordo com as características do(s) gênero(s) escolhido(s).

Simulador de estratégia biomédico. O jogo não possui enredo fixo; em vez disso, o jogador assume a função de linfócito e monitora um avatar continuamente para garantir o combate a carcinógenos e a prevenção do câncer. Além disso, o aplicativo traz aspectos estratégicos por exigir um bom gerenciamento de recursos.

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

Descrever as histórias de usuários (jogadores) de acordo com o template utilizado.

Glossário:

- Celinha: célula de defesa (linfócito) que combate carcinógenos;
 - Avatar: entidade responsável por supervisionar diferentes partes do corpo e comandar o sistema imunológico contra o câncer. Controlado pelo jogador e sem representação gráfica;
 - Carcinógeno: substância cancerígena que transforma células saudáveis em cancerosas;
 - Célula cancerosa: célula danificada por carcinógenos. Cresce até comprometer o jogo, resultando em punições de pontuação ao jogador;
 - Manchetes de comportamento: barra na parte superior do jogo onde as ações do corpo (ficar muito tempo no sol, fumar, alimentar-se de carnes processadas, etc.) aparecem;
 - Âncora: NPC localizado na parte superior da tela, ao lado das manchetes de comportamento, cuja animação indica que ele está informando as notícias (no caso, ações) em evidência nas manchetes de comportamento. Ilustração cômica;
 - Loja de hábitos: aba em que avatar pode adquirir bons hábitos. Esses bons hábitos diminuem as dificuldades dos ataques de agentes cancerígenos;
 - Pontos: unidades fungíveis adquiridas através da destruição de carcinógenos e de células cancerosas. Podem ser trocadas por bons hábitos na loja de hábitos.
-

Game stories

- Como avatar, quero monitorar diferentes partes do corpo para identificar carcinógenos. (Parte 2)
- Como carcinógeno, quero corromper as células saudáveis do corpo humano para estimular o surgimento de um carcinoma. (Parte 2)
- Como carcinógeno, caso o avatar esteja muito corrompido, quero atacar mais de um órgão de uma vez, para assim, com mais facilidade, contaminar células saudáveis
- Como avatar, quero transformar os maus hábitos do corpo em bons para impedir que mais carcinógenos prejudiquem a saúde de órgãos específicos de meu corpo. (Parte 3)
- Como âncora, quero informar o avatar sobre a rotina do corpo, e em quais órgãos há carcinógenos, para que ele saiba como proceder no jogo. (Parte 1)

- Como Celinha, quero neutralizar células cancerosas para que elas não se multipliquem descontroladamente e corrompam o avatar. (Parte 2)
- Como avatar, quero que os carcinógenos neutralizados sejam convertidos em pontos para poder comprar bons hábitos. (Parte 1)
- Como avatar, quero fazer upgrades nos bons hábitos do corpo para potencializá-los. (Parte 5)
- Como avatar, quero poder comprar todos os bons hábitos para minimizar o risco de câncer. (Parte 3)
- Como avatar, quero que, em meu corpo, esteja sempre tocando uma música feliz e alegre, para que combine com o design da Celinha. (Parte 4)
- Como Celinha, quero ter um design amigável para chamar a atenção do jogador. (Parte 1)
- Como célula cancerosa, quero ter um design de “vilão” para indicar qual o meu papel no jogo para o jogador. (Parte 3)
- Como avatar, quero poder visitar a loja de hábitos inúmeras vezes para ver quais são os hábitos disponíveis. (Parte 2)
- Como avatar, quero algo na tela principal que me ensine sobre diferentes cânceres e seus fatores de risco para definir minha estratégia no jogo. (Parte 1)
- Como avatar, quero clicar em carcinógenos para destruí-los. (Parte 2)
- Como avatar, quero que um menu principal me apresente ao jogo para que eu escolha o momento de iniciar. (Parte 3)
- Como avatar, quero escolher o sexo e o nome de meu corpo para me identificar mais com ele. (Parte 3)
- Como célula cancerosa, quero me desenvolver rapidamente após a colisão do carcinógeno para atingir mais partes do corpo. (Parte 2)
- Como avatar, quero que uma barra lateral me indique quais hábitos já comprei para que eu saiba o que priorizar no jogo. (Parte 3)
- Como avatar, quero que a música continue quando troco páginas, exceto quando clico em um órgão, para que haja continuidade em minha experiência. (Parte 2)
- Como corpo, quero envelhecer um estágio de vida em períodos regulares para completar meu ciclo de vida. (Parte 2)
- Como avatar, quero que os hábitos que compro tenham um efeito tangível nos fatores de risco de cada câncer para que eu entenda como certos hábitos afetam a saúde do corpo. (Parte 2)
- Como linfócito no modo NPC, quero instruir o jogador para iniciar o jogo escolhendo um corpo. (Parte 3)
- Como linfócito no modo NPC, quero ter várias expressões e olhos que acompanhem o mouse para que haja mais interação. (Parte 3)
- Como avatar, quero que efeitos sonoros acompanhem a captura de carcinógenos para que eu saiba quanto obtive sucesso. (Parte 3)
- Como avatar, quero pausar o jogo para não ser prejudicado em momentos de distração. (Parte 3)

- Como avatar, quero mutar e desmutar o jogo para ter mais conforto durante a partida. (Parte 3)
 - Como avatar, quero reiniciar o jogo para ter quantas chances desejar. (Parte 3)
 - Como avatar, quero que efeitos sonoros acompanhem o game over para que eu entenda plenamente o que houve. (Parte 3)
 - Como avatar, quero que efeitos sonoros agradáveis acompanhem a compra de hábito para que eu entenda mais intuitivamente o efeito positivo que eles têm. (Parte 3)
 - Como avatar, quero visitar e jogar no órgão da pele para aprender mais sobre como certos hábitos o afetam. (Parte 4)
 - Como avatar, quero voltar à tela principal automaticamente após vencer um ataque carcinógeno para compreender que tal evento acabou. (Parte 5)
 - Como avatar, quero que um vetor de metabolismo saudável substitua a imagem do cursor nas telas de órgão para que o jogador entenda sua função no corpo humano. (Parte 4)
 - Como jogador, quero escolher o nome do corpo para me identificar mais com ele. (Parte 4)
-

User stories

- Como Laura, quero jogar algo divertido para me distrair quando estou entediada.
 - Como Laura, quero aprender sobre câncer de forma lúdica porque não me engajo com materiais puramente acadêmicos.
 - Como Laura, quero jogar algo leve porque não tenho muita memória no celular.
 - Como Laura, quero jogar algo rápido com começo, meio e fim porque não tenho muito tempo livre para dedicar.
 - Como Laura, quero jogar algo desafiador para não ficar entediada.
 - Como Laura, quero jogar algo gradativo para aprender conceitos novos aos poucos.
 - Como Laura, quero que em momentos específicos, tenham mais de um minigame acontecendo ao mesmo tempo, para incrementar a dificuldade do jogo
-

1.7.5 Mecânica

Descrever a forma de interação do jogo (mecânica), nos aspectos: personificação, dimensão, período, conectividade, visão.

- O linfócito é movido pelos órgãos através das setas do teclado. Não há gravidade, pois ele está imerso em líquido;
 - O linfócito pode destruir as células cancerosas e/ou os agentes cancerígenos;
 - Cada célula cancerosa ou agente cancerígeno eliminados equivale ao ganho de um ponto de saúde;
 - Cada célula saudável destruída reduz a taxa de saúde, visível na barra de saúde;
 - A taxa de saúde representa a saúde atual do corpo. Aumenta com hábitos bons e diminui hábitos ruins. Atinge seu ápice quando todos os hábitos são comprados e as células cancerígenas do corpo são mitigadas, configurando vitória no jogo. Perde-se o jogo quando a taxa de saúde atinge zero, o que ocorre quando há a multiplicação descontrolada de células cancerosas, configurando o diagnóstico de câncer.
 - A loja de bons hábitos contém diversos hábitos saudáveis adequados para cada situação específica (praticar atividades físicas, passar protetor solar, não se expor excessivamente ao sol, comer alimentos não processados, se vacinar, etc). Cada bom的习惯 adquirido é refletido graficamente no órgão relacionado, cuja estrutura facilita o movimento, capacidades e alcance das metas do jogador.
 - Há um menu para iniciar, pausar, voltar ao jogo;
 - As células cancerosas atacam o linfócito no momento em que o linfócito estiver visível no campo onde estão concentradas;
 - Corpo tem regiões limitadas que poderão ser desbloqueadas apenas quando as células anômalas forem combatidas;
 - É possível desbloquear habilidades adicionais mediante a vitória em batalhas.
-

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. | Como surge o câncer:
https://www.inca.gov.br/como-surge-o-cancer#:~:text=O%20processo%20de%20forma%C3%A7%C3%A3o%20do,origem%20a%20um%20tumor%20vis%C3%ADvel . |
| 2. | Prováveis carcinógenos humanos:
https://www.cancer.org/cancer/cancer-causes/general-info/known-and-probable-human-carcinogens.html |
| 3. | Jogo Bio Inc: http://playbioinc.com |
| 4. | Carcinógeno: https://www.wikiwand.com/en/Carcinogen |
| 5. | Pac-Man |
| 6. | Mecanismos naturais que protegem contra o câncer: https://pubmed.ncbi.nlm.nih.gov/14687712/ |

2. Game Design

2.1 História do Jogo

Descrever os seguintes aspectos:

- Tema (*storyline*)
- Conceito
- Pano de fundo da história (*backstory*)
- Premissa
- Sinopse
- Estrutura narrativa escolhida
- 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
- Níveis de interatividade do jogo

Probody gira em torno de um adolescente, denominado Corpo para os fins deste documento, que possui diversos hábitos negativos. Esses hábitos, por sua vez, inserem substâncias cancerígenas em seu corpo, danificando os tecidos dos órgãos e aumentando a chance de o proto-oncogene, responsável pelo surgimento do câncer, ser ativado.

O Corpo, ao perceber tais substâncias, mobiliza suas células de defesa. Estas são representadas pela figura de Celinha, o avatar controlado pelo jogador. Nesse sentido, a Celinha tem a capacidade de atuar diretamente em regiões infestadas de carcinógenos e destruí-los através de colisões, o que rendeu ao jogador pontos. Por outro lado, caso Celinha colida com uma célula saudável, como uma hemácia, perderá-se pontos.

Entretanto, conforme o Corpo cresce, passando pelas fases de jovem adulto, adulto de meia-idade, por exemplo, seu comportamento tende a piorar. Assim, a partir de certo ponto, deixa de ser possível combater os carcinógenos apenas com a Celinha. Nesse momento, cabe ao jogador trocar seus pontos por hábitos saudáveis na loja de hábitos a fim de diminuir seu risco de câncer. Entretanto, se mesmo assim a oncogênese se completar, será possível recuperar-se através de uma fase especial.

A vitória em Probody consiste em alcançar a terceira idade em boa saúde. Não há game over, pois a fase especial no caso de câncer permite o replay infinito ou o reinício da partida como um todo. Desse modo, o jogador nunca terá de lidar com a morte de seu personagem e poderá refinar seus conhecimentos sobre a prevenção da neoplasia continuamente.

2.2 Fluxo do Jogo e Níveis

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.

Fazer o *flowchart* do jogo (grafo 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 despender com o jogo?

Proboddy não possui níveis no sentido tradicional do termo. Em vez disso, o jogo adequa-se dinamicamente tanto à performance do jogador quanto à passagem de tempo *in-game*. Nesse contexto, a dificuldade é controlada por índices de risco de câncer. Tais índices consistem em uma função matemática que leva em consideração todos os hábitos negativos do Corpo, juntamente de sua idade, e calcula suas chances de desenvolver câncer a partir deles. Assim, o jogo torna-se mais fácil quando os índices de risco são menores e mais difícil conforme eles aumentam, podendo variar de dificuldade inúmeras vezes em uma única partida.

O início do jogo, entretanto, contém um tutorial que apresenta o jogador as diferentes funcionalidades e mecânicas do aplicativo através de um combate guiado a carcinógenos de câncer de pulmão.

Flowchart: https://miro.com/app/board/uXjVOKee8Dw=/?invite_link_id=977679478303

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.

A tela principal do jogo é um painel que contém 1) o Corpo, com certos órgãos clicáveis; 2) um contêiner inferior contendo o nome e idade do Corpo , juntamente de ícones relacionados a cada tipo de câncer. Ao passar o mouse por cima de cada ícone, um pop-up aparece com mais detalhes; 3) um contêiner superior contendo a ilustração de um âncora de jornal para efeitos cômicos e um *label* que o informa o que o Corpo tem feito; 4) um ícone que leva à loja de hábitos.

Ao clicar em um órgão, a cena muda para um zoom-in da região escolhida. Nesse sentido, o “mapa” segue o formato do órgão em questão.

A loja de hábitos, acessível através da tela principal, contém diversos ícones, cada um relacionado a um hábito diferente.

Flowchart: https://miro.com/app/board/uXjVOJxfQYg=/?invite_link_id=327122251010

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.

Na tela principal, a navegação é por meio de cliques em botões (loja de hábitos, órgãos) e *hovering* (detalhes sobre cada câncer). Dentro de cada órgão, a navegação se dá por cliques no botão de “VOLTAR” e pelas teclas de seta e espaço, que movimentam a Celinha.

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.

Os carcinógenos são a menor estrutura do jogo. Se os consideramos, tem-se que a Celinha atinge 1,5. Portanto, o órgão que a circunda, ocupando a maior parte da tela, chega a cerca de 60. Evidentemente, essa escala não condiz com a realidade, em que cada órgão seria milhões de vezes maior que uma célula, e sim com o tamanho relativo das estruturas gráficas.

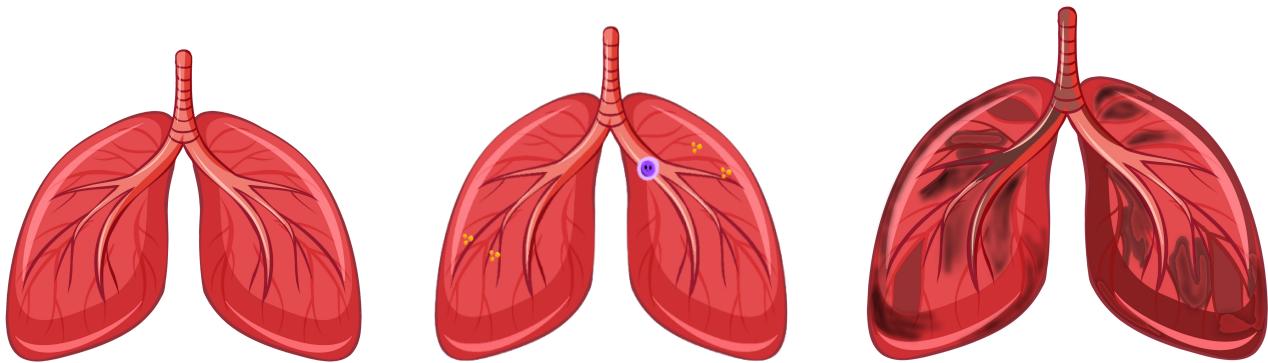
Semelhante, na tela principal, a escala real teria o corpo humano como dezenas de vezes maior que um órgão, enquanto os diversos elementos de menu seriam indefinidos por não existirem fora do computador. Já do ponto de vista gráfico, o corpo chega a 30. O menu inferior é 20 e o superior 15. O ícone para comprar hábitos é 3, assim como o *label* de pontos.

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).

O jogo ocorre dentro do corpo humano, em específico, no interior dos órgãos e cada órgão tem sua própria cena, a qual pode ser selecionada através da imagem do corpo inteiro da tela principal. Ademais, cada órgão tem seus tecidos circundantes divididos em regiões. Essas regiões contam quantas colisões sofrem com carcinógenos e mudam de cor para indicar o nível de dano acumulado. Por exemplo, se uma parte do pulmão for atacada repetidamente, sua coloração se tornará mais forte, beirando o preto. Por fim, em momentos de perseguição a

inimigos pela Celinha, haverá crescente poluição visual causada pela aglomeração não só de agentes cancerígenos, mas também de células saudáveis das quais o jogador deverá desviar.



2.3.5 Tempo

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

Probody acompanha o crescimento e consequentemente envelhecimento do Corpo. Esse processo é dividido nos estágios Adolescente, Jovem Adulto, Adulto, Meia-Idade e Terceira Idade. Cada estágio dura três minutos.

Além disso, o caráter dinâmico do jogo exige que o jogador gerencie seu tempo eficientemente. Em muitos momentos, haverá carcinógenos em diversos órgãos simultaneamente, de modo que o jogador precisará avaliar seus riscos e recursos a fim de decidir que área priorizar.

Por fim, um timer de um segundo é utilizado para criar os carcinógenos. A cada rodada, mais um é criado. Outro timer é implementado na aparição do texto na manchete, em que um caractere aparece a cada 0,1 segundos.

2.4 Base de Dados

2.4.1 Inventário

2.4.1.1 Itens Consumíveis

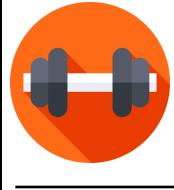
Descrever cada um dos itens do jogo, apresentando suas características principais e usadas para programar o jogo (características que de fato afetam o funcionamento do jogo; por exemplo, suponha que o peso seja fator relevante no jogo de forma que o jogador não possa carregar muitos objetos ao mesmo 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.

HÁBITOS SAUDÁVEIS (PARA TROCA DE PONTOS)

Exercício físico

	Descrição	Exercício físico que diminui as taxas de risco de todos os tipos de câncer.
	Valor de cura	Diminui 100% no fator de exercício físico de todos os cânceres.

Vacina contra o HPV

	Descrição	Vacina que diminui as chances de câncer de colo de útero.
	Valor de cura	Diminui 100% no fator específico de câncer de colo de útero.

Alimentação saudável

	Descrição	Alimentação balanceada que diminui a taxa de risco de todos os tipos de câncer.
	Valor de cura	Diminui 100% no fator de alimentação de todos os cânceres.

Exame de rotina no urologista

	Descrição	Exame de rotina que diminui os riscos contra câncer de próstata.
	Valor de cura	Permite que o usuário pule a fase especial caso um câncer de mama se desenvolva.

Protetor solar

	Descrição	Uso de protetor solar protege contra os raios UV e diminui o risco de câncer de pele.
	Valor de cura	Diminui 75% do fator específico de câncer de pele.

Mamografia

	Descrição	Fazer mamografia regularmente diminui os riscos de câncer de mama.
	Valor de cura	Permite que o usuário pule a fase especial caso um câncer de mama se desenvolva.

Parar de beber bebidas alcoólicas

	Descrição	Não consumir bebidas alcoólicas diminui os riscos de câncer de fígado.
	Valor de cura	Diminui 75% do fator específico do câncer de fígado.

Parar de comer alimentos processados

	Descrição	Evitar o consumo de alimentos processados diminui os riscos de câncer de intestino.
	Valor de cura	Diminui em 75% do fator específico de câncer de intestino.

Parar de fumar

	Descrição	Evitar o consumo de cigarros ou fumar, no geral, diminui os riscos de câncer de pulmão.
	Valor de cura	Diminui em 75% do fator específico de câncer de pulmão.

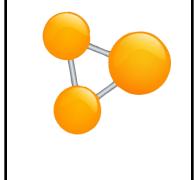
2.4.2 Bestiário (opcional)

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.

Ações do Corpo (textos a aparecer no contêiner superior, indicando os maus hábitos do jogador)

Câncer de pulmão	"O Avatar fumou"
------------------	------------------

Agente cancerígeno

	Descrição	Substância cancerígena responsável por atacar os órgãos.
---	-----------	--

2.4.3 Balanceamento de Recursos

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

Não se aplica totalmente a Proboddy no momento devido à adequação de dificuldade dinâmica, conforme discutido anteriormente. Entretanto, é possível dividir o jogo em quatro modos, cuja dificuldade varia de acordo com a pontuação do jogador. Nesse sentido, é o acúmulo de cânceres em um curto espaço de tempo e as muitas informações que devem ser memorizadas (como a relação ação-órgão-prevenção) que alteram o nível de dificuldade. Na tabela abaixo, P refere-se ao número de pontos que o jogador acumulou.

Tipo Inimigo	Tutorial	P <= 100	100 < P <= 200	200 < P
Câncer de pulmão	sim	sim	sim	sim
Câncer de pele	não	sim	sim	sim
Câncer de fígado	não	sim	sim	sim
Câncer de cólon	não	não	sim	sim
Câncer de estômago	não	não	sim	sim
Câncer de próstata	não	não	não	sim
Câncer de colo de útero	não	não	não	sim
Câncer de mama	não	não	não	sim

Total	1	3	5	8
-------	---	---	---	---

Dificuldade do jogo Proboddy por módulo

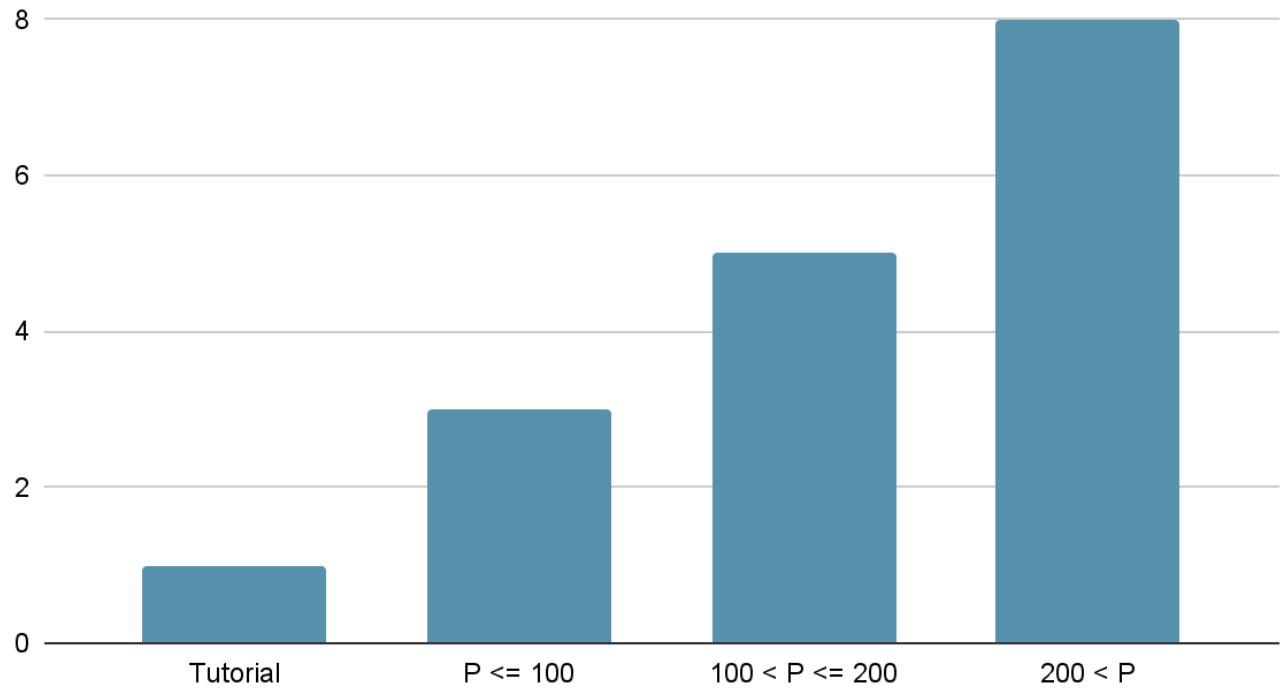


Figura 2. Gráfico de dificuldade para o jogo PROBODY.

A tabela e gráficos acima baseiam-se em estimativas. Almeja-se obedecer à teoria de *flow* de Mihaly, porém só será possível averiguar tal métrica e ajustar nossos parâmetros quando mais cânceres forem implementados, nas próximas semanas.

Habits (upgrades) chart

Em discussão.

Tipo de Item	Tutorial	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 <ADALOVE – Implementar mecânicas básicas do jogo - parte 1 a 5>

(não se aplica – as diferenças entre os módulos citados na última seção referem-se apenas à quantidade de cânceres possíveis em um dado momento)

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.

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).

3.1.2.2 Layout Effects (opcional)

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

3.1.2.3 Quests e Puzzles (opcional)

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

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:

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
<hr/>								
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).

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.

NPC informativo: fornece informações ao jogador

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).

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	Estréia
Celinha	Tutorial
Agente cancerígeno	Tutorial
Âncora	Tutorial
Corpo	Tutorial

Os personagens do jogo são a célula Celinha, os agentes cancerígenos, o âncora e o Corpo. No jogo, a célula Celinha e os agentes cancerígenos apresentam papéis antagônicos, enquanto o âncora atua como narrador. Já o Corpo consiste no corpo que Celinha deve proteger.

4.1 Personagens Controláveis

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:

Celinha

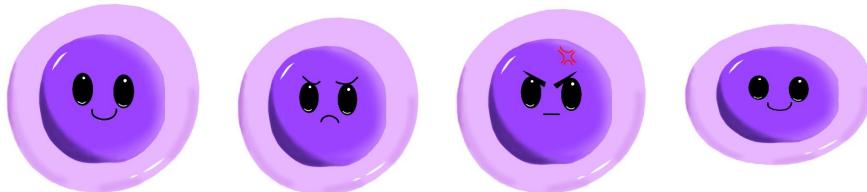
4.1.1 Backstory

Backstory (pano de fundo) do personagem.

Celinha é uma célula de defesa do Corpo. Ela deve ajudá-lo a combater substâncias cancerígenas através da mitigação direta dentro de cada órgão afetado. Tende a enfurecer-se quando um carcinógeno danifica as paredes do órgão.

4.1.2 Concept Art

Esboços do personagem.



(Imagens autorais)

4.1.3 Ações Permitidas

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

Celinha é capaz de movimentar-se livremente nos órgãos do corpo. Seu único impedimento são os próprios tecidos e delineados de cada região. Também pode capturar agentes cancerígenos através da colisão.

4.1.4 Momento de Aparição

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

Sua primeira aparição é no tutorial do jogo, sendo apresentada pelo âncora. Após isso, Celinha aparece nos órgãos do corpo humano, cada vez que o jogador os seleciona na tela principal.

4.2 Common Non-Playable Characters (NPC)

4.2.1 Agentes Cancerígenos

Para cada NPC comum, descrever sua história, comportamento perante o personagem (agressivo, amistoso, 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.

No jogo, os agentes cancerígenos atuam de maneira a afetar os órgãos do corpo humano, sendo, portanto, os antagonistas do game. Estes agentes cancerígenos aparecem quando o corpo humano realiza alguma atividade não

saudável, como por exemplo, fumar. Ao aparecerem, elas irão começar seus ataques no corpo humano, e cabe a Celinha defendê-las desses agentes. Seu comportamento será de se mover em direções aleatórias, a fim de encostar nas paredes dos órgãos, caso consigam, irão corromper certa quantidade de células, caso contrário, serão eliminadas.

4.3 Special Non-Playable Characters (NPC)

4.3.1 Âncora

Para cada NPC especial (mini-boss, boss, mentor/guia etc), descrever sua história, comportamento perante o personagem (agressivo, amistoso, 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.

Este NPC será o guia de Celinha. Tem a aparência de um âncora de telejornal, por motivos cômicos, e serve à função de informar o que o Corpo tem feito. Suas ações serão limitadas a barras de texto na tela, contendo resumos de como o Corpo se comportou, como suas escolhas afetam seus órgãos, como combater agentes cancerígenos e como escolher os melhores hábitos para deixar o Corpo mais saudável.

4.3.2 Corpo

O Corpo é, de certa forma, o co-protagonista de Proboddy. É nele que Celinha mora e é ele que ela deve proteger. O Corpo inicia o jogo como um adolescente e envelhece com o decorrer do tempo. Seus hábitos tendem a tornar-se cada vez mais negativos se providências não forem tomadas. Não é considerado um personagem controlável porque pode ser apenas indiretamente influenciado pelos hábitos adquiridos pelo jogador.

5. Qualidade de Software

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:

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:

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

Consegui 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?

5.2 Normas de Qualidade <[ADALOVE - Definir padrões de qualidade](#)>

Adequação	Propõe-se a fazer o que é apropriado?	Sim, a concepção de Proboddy atende às expectativas e requisitos do cliente, conforme estabelecido no GDD.
Acurácia	Gera resultados corretos ou conforme acordados?	Sim, nossos resultados atendem corretamente aos requisitos estabelecidos pelo cliente e a nossa documentação.
Interoperabilidade	É capaz de interagir com os sistemas especificados?	Sim, o jogo funcionará em todas as plataformas especificadas.
Segurança de acesso	Evita o acesso não autorizado, acidental ou deliberado a programas e dados?	Não é aplicável. O jogo, em sua versão final de executável, não terá agentes aos quais o acesso será restrito. Também não coletará dados privados de seus usuários.
Conformidade	Está de acordo com normas e convenções	Sim, não infringimos nenhum tipo de lei, como a proteção a direitos autorais, e ainda tomamos cuidado para tratar do assunto

	previstas em leis e descrições similares?	de câncer da forma mais sensível possível.
Maturidade	Com que frequência apresenta falhas?	Raramente. O jogo apresentou falhas durante o seu desenvolvimento e testagem, mas sempre o atualizamos e consertamos para apresentações ao cliente e deploy.
Tolerância a falhas	Ocorrendo falhas como ele reage?	Caso haja uma falha, o programa se fecha automaticamente. Pretende-se estabelecer <i>handlers</i> para os erros mais prováveis nas próximas sprints.
Recuperabilidade	É capaz de recuperar dados após uma falha?	Não, afinal, não temos nenhum tipo de mecanismo para salvar os dados após o encerramento de uma sessão.
Inteligibilidade	É fácil entender os conceitos utilizados?	Sim, tanto a documentação quanto o código são escritos de forma clara, objetiva e, em sua maior parte, em voz ativa e períodos simples.
Apreensibilidade	É fácil aprender a usar?	Sim, o jogo é bem intuitivo, além de apresentar tutoriais para o jogador.
Operacionalidad e	É fácil de operar e controlar a operação?	Sim. A interface com todos os seus botões, áreas clicáveis e elementos dinâmicos apresentam design simples e intuitivo.
Comportamento em relação ao tempo	Qual é o tempo de resposta e de processamento?	O jogo é leve, por este motivo, o tempo de resposta e processamento dele são bem rápidos (< 0,1s)
Comportamento em relação aos recursos	Quanto recurso utiliza?	Também por ser leve, o jogo utiliza pouco recurso das máquinas nas quais ele roda.
Analisabilidade	É fácil encontrar uma falha quando ocorre?	Ainda não. Pretende-se estabelecer <i>handlers</i> com descrições úteis de falhas nas próximas sprints.
Modificabilidade	É fácil modificar e remover defeitos?	Sim. Uma vez que o código possui descrições bem claras das funcionalidades do jogo, torna-se mais fácil a correção de possíveis defeitos.
Estabilidade	Há grandes riscos de bugs quando se faz alterações?	Sim. A estrutura de nós do Godot faz com que alterações pequenas em um script afetem potencialmente toda uma cena do jogo.
Testabilidade	É fácil testar quando se faz alterações?	Sim. A exportação é rápida e simples, permitindo a testagem em poucos minutos após o fim do desenvolvimento.
Adaptabilidade	É fácil adaptar a outros ambientes sem aplicar outras ações ou meios além dos fornecidos para esta finalidade no software considerado?	Sim. O ambiente de desenvolvimento Godot permite a exportação para diversas plataformas.
Capacidade para ser instalado	É fácil instalar em outros ambientes?	Sim. Na versão final, o jogo poderá ser jogado online ou através de executáveis de fácil instalação em desktop ou Android.
Capacidade para substituir	É fácil substituir por outro software?	Sim. Há outros simuladores biomédicos disponíveis no mercado.
Conformidade	Está de acordo com padrões ou convenções de portabilidade?	Sim, o jogo roda em Web e Windows. Além disso, os planos são para que futuramente rode também para dispositivos Android.

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>

Função para movimento do personagem (implícita no move_and_collide())

Distância relativa ao ponto inicial = vetor de velocidade * delta de tempo

No código, o cálculo da distância a ser percorrida pelo personagem é dividido em duas partes. Na primeira, a velocidade é identificada através de um vetor de movimento relativo. Nesse sentido, por estar associada a certas teclas, pode-se interpretá-la como Vector2/toques na tecla. Já na segunda parte, esse vetor é enviado ao move_and_collide(), que é chamado continuamente pelo _physics_process(), mas só executa o deslocamento quando as teclas indicadas são acionadas e velocidade (Vector2), consequentemente, é diferente de zero. Logo, toda a estrutura resume-se a distância = (distância / teclas) * teclas.

```
if Input.is_action_pressed("space"):
    if Input.is_action_pressed("ui_up"):
        moving = true
        velocity = Vector2(0, -2)
    if Input.is_action_pressed("ui_down"):
        moving = true
        velocity = Vector2(0, 2)
    if Input.is_action_pressed("ui_left"):
        moving = true
        velocity = Vector2(-2, 0)
    if Input.is_action_pressed("ui_right"):
        moving = true
        velocity = Vector2(2, 0)
```

```
var collision = $CeliaCell.move_and_collide(velocity)
```

Função para calcular risco de câncer

$$\text{Risco (em \%)} = 0.25 * \text{má alimentação} + 0.25 * \text{sedentarismo} + 0.50 * \text{maus hábitos específicos}$$

Proboddy apresenta barras de risco de desenvolvimento de câncer para cada neoplasia tratada. A função geral leva em consideração os fatores gerais de má alimentação e sedentarismo em conjunto com fatores específicos que variam para cada tipo de câncer. Os valores associados a esses fatores dependem dos hábitos adquiridos e melhorados na loja de hábitos. Assim, quanto mais saudáveis os hábitos, menores serão os índices de má alimentação e sedentarismo, por exemplo, o que levará à diminuição do risco de câncer.

O cálculo ainda não foi implementado, mas sua representação gráfica será coordenada pela função abaixo:

```
func set_cancer_bar(value):  
    $CancerBar.value = value
```



Função para calcular pontos no pulmão:

$$\sum_{n=0}^{\infty} f(t) = \begin{cases} 1 & n \leq 30 \\ 0 & n > 30 \end{cases}$$

O tutorial do jogo ensina o jogador a ganhar pontos no pulmão. Para isso, o jogador deve movimentar seu avatar (a célula Celinha) em direção aos trinta carcinógenos que surgirão no órgão. A cada colisão, um ponto é adicionado. Desse modo, esse processo pode ser descrito pela função matemática acima e simplificado pela função de GDScript abaixo. Nesse caso, o limite de trinta pontos não é levado em consideração porque exatamente trinta inimigos são gerados previamente, de modo que é impossível atingir mais do que essa quantia.

```
func change_score(score):  
    var current_score = int($ScoreNumber.text)  
    new_score += int(score)  
    $ScoreNumber.text = str(new_score)
```

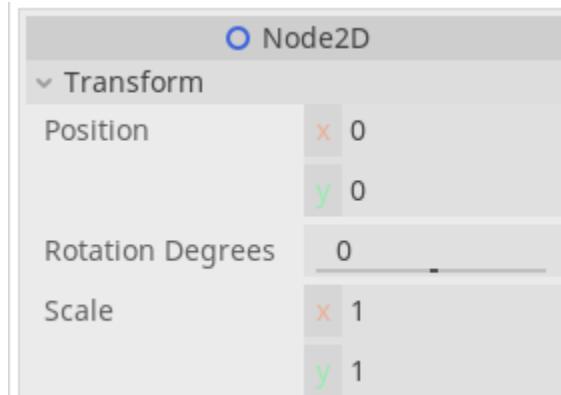
6.2 Cinemática Unidimensional

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>

Proboddy utiliza as grandezas de posição e velocidade. Em primeiro plano, conforme o padrão do Godot, todos os nós derivados de Node2D possuem um atributo de posição que salva a localização do elemento na cena.



Exemplo de atributos de posição em um Node2D

Nesse sentido, a geração de carcinógenos nos órgãos utiliza o “position” para definir o ponto onde esse agente cancerígeno surgirá. Assim, a cena de carcinógenos, que serve como template para todos os inimigos, é instanciada inúmeras vezes, tendo seu ponto inicial sorteado a cada “spawn”.

```

var badMoleculeScene = badMolecule.instance() # Cria o carcinógeno
var locationPoint = rng.randi_range(1, 8) # Sorteia um número entre 1 e 8
# O bloco abaixo escolhe uma posição de onde o carcinógeno vai surgir segundo
# o locationPoint. Os nós PositionX para encontrar os pixels de posição.
# Já a velocidade indica também a direção na qual o carcinógeno será criado.

if locationPoint == 1:
    badMoleculeScene.position = Vector2(600, 200)
    badMoleculeScene.velocity = Vector2(rand_range(-100, -20), rand_range(-160, -70))
elif locationPoint == 2:
    badMoleculeScene.position = Vector2(337, 385)
    badMoleculeScene.velocity = Vector2(rand_range(-100, -20), rand_range(-160, -70))
elif locationPoint == 3:
    badMoleculeScene.position = Vector2(436, 382)
    badMoleculeScene.velocity = Vector2(rand_range(160, 70), rand_range(-160, -70))
elif locationPoint == 4:
    badMoleculeScene.position = Vector2(226, 144)
    badMoleculeScene.velocity = Vector2(rand_range(160, 70), rand_range(160, 70))
elif locationPoint == 5:
    badMoleculeScene.position = Vector2(140, 306)
    badMoleculeScene.velocity = Vector2(rand_range(-100, -20), rand_range(-100, -20))
elif locationPoint == 6:
    badMoleculeScene.position = Vector2(226, 144)
    badMoleculeScene.velocity = Vector2(rand_range(160, 70), rand_range(160, 70))
elif locationPoint == 7:
    badMoleculeScene.position = Vector2(358, 229)
    badMoleculeScene.velocity = Vector2(rand_range(-160, -70), rand_range(160, 70))

```

Ademais, a grandeza velocidade é empregada implicitamente no movimento da Celinha e dos carcinógenos. Isso se dá através do método move_and_collide(). Nesse caso, o argumento passado é um Vector2 que indica tanto a direção do movimento, por meio da soma vetorial de suas coordenadas, quanto a velocidade, de acordo com a magnitude do vetor resultante.

```

var badMoleculeScene = badMolecule.instance() # Cria o carcinógeno
var locationPoint = rng.randi_range(1, 8) # Sorteia um número entre 1 e 8
# O bloco abaixo escolhe uma posição de onde o carcinógeno vai surgir segundo
# o locationPoint. Os nós PositionX para encontrar os pixels de posição.
# Já a velocidade indica também a direção na qual o carcinógeno será criado.

if locationPoint == 1:
>I badMoleculeScene.position = Vector2(600, 200)
>I badMoleculeScene.velocity = Vector2(rand_range(-100, -20), rand_range(-160, -70))
elif locationPoint == 2:
>I badMoleculeScene.position = Vector2(337, 385)
>I badMoleculeScene.velocity = Vector2(rand_range(-100, -20), rand_range(-160, -70))
elif locationPoint == 3:
>I badMoleculeScene.position = Vector2(436, 382)
>I badMoleculeScene.velocity = Vector2(rand_range(160, 70), rand_range(-160, -70))
elif locationPoint == 4:
>I badMoleculeScene.position = Vector2(226, 144)
>I badMoleculeScene.velocity = Vector2(rand_range(160, 70), rand_range(160, 70))
elif locationPoint == 5:
>I badMoleculeScene.position = Vector2(140, 306)
>I badMoleculeScene.velocity = Vector2(rand_range(-100, -20), rand_range(-100, -20))
elif locationPoint == 6:
>I badMoleculeScene.position = Vector2(226, 144)
>I badMoleculeScene.velocity = Vector2(rand_range(160, 70), rand_range(160, 70))
elif locationPoint == 7:
>I badMoleculeScene.position = Vector2(358, 229)
>I badMoleculeScene.velocity = Vector2(rand_range(-160, -70), rand_range(160, 70))

```

Sorteio de posição de ponto inicial para cada carcinógeno

```

func get_direction():
>I var direction = Vector2.ZERO
>I if isMoving:
>I >I if Input.is_action_pressed("ui_up"):
>I >I >I direction += Vector2(0, -1)
>I >I if Input.is_action_pressed("ui_down"):
>I >I >I direction += Vector2(0, 1)
>I >I if Input.is_action_pressed("ui_left"):
>I >I >I direction += Vector2(-1, 0)
>I >I if Input.is_action_pressed("ui_right"):
>I >I >I direction += Vector2(1, 0)
>I return direction.normalized()

```

Método que define direção do vetor da Celinha a partir do input do jogador

```

func get_movement():
>|   var direction = get_direction()
>|   if Input.is_action_pressed("shift"):
>|       >|   return direction * 5.5
>|   else:
>|       >|   return direction * 2.5

```

Método que define vetor de velocidade da Celinha ao multiplicar o vetor de direção pela magnitude da velocidade desejada

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>

Conforme detalhado na seção 6.1, Vector2 é utilizado para determinar a velocidade e direção da movimentação dos personagens Celinha e carcinógenos. Nesse sentido, aparecem na instância abaixo, no script Lung.gd, para definir a velocidade da Celinha.

```

if Input.is_action_pressed("space"):
>|   if Input.is_action_pressed("ui_up"):
>|       moving = true
>|       velocity = Vector2(0, -2)
>|   if Input.is_action_pressed("ui_down"):
>|       velocity = Vector2(0, 2)
>|       moving = true
>|   if Input.is_action_pressed("ui_left"):
>|       velocity = Vector2(-2, 0)
>|       moving = true
>|   if Input.is_action_pressed("ui_right"):
>|       velocity = Vector2(2, 0)
>|       moving = true
>
>|   . . .
var collision = $CeliaCell.move_and_collide(velocity)
-- -- -- -- --

```

Similarmente, na criação de carcinógenos, suas velocidades e direções são definidas a partir de Vector2s. Em ambos os casos, a direção é derivada do sinal das coordenadas x e y, enquanto a velocidade vem de suas magnitudes.

```

if locationPoint == 1:
>|    badMoleculeScene.position = Vector2(600, 200)
>|    badMoleculeScene.velocity = Vector2(-50, 100)
elif locationPoint == 2:
>|    badMoleculeScene.position = Vector2(337, 385)
>|    badMoleculeScene.velocity = Vector2(-50, -100)
elif locationPoint == 3:
>|    badMoleculeScene.position = Vector2(436, 382)
>|    badMoleculeScene.velocity = Vector2(100, -100)
elif locationPoint == 4:
>|    badMoleculeScene.position = Vector2(226, 144)
>|    badMoleculeScene.velocity = Vector2(100, 100)
elif locationPoint == 5:
>|    badMoleculeScene.position = Vector2(140, 306)
>|    badMoleculeScene.velocity = Vector2(50, -50)
elif locationPoint == 6:
>|    badMoleculeScene.position = Vector2(226, 144)
>|    badMoleculeScene.velocity = Vector2(100, 100)
elif locationPoint == 7:
>|    badMoleculeScene.position = Vector2(358, 229)
>|    badMoleculeScene.velocity = Vector2(-100, 100)
elif locationPoint == 8:
>|    badMoleculeScene.position = Vector2(652, 392)
>|    badMoleculeScene.velocity = Vector2(-100, -100)

```

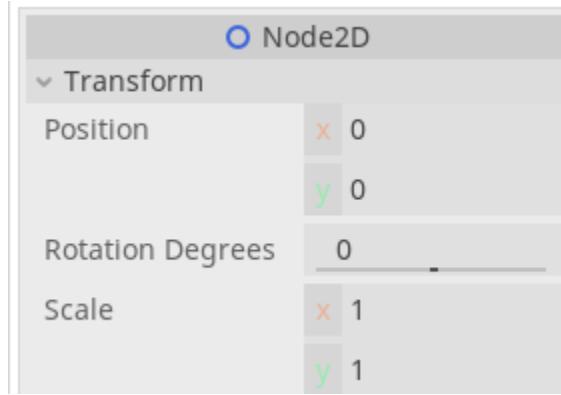
6.4 Cinemática Bidimensional e mais

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.

<ADALOVE - Fazer orientação espacial (2D)>

Proboddy utiliza as grandezas de posição e velocidade. Em primeiro plano, conforme o padrão do Godot, todos os nós derivados de Node2D possuem um atributo de posição que salva a localização do elemento na cena.



Exemplo de atributos de posição em um Node2D

Nesse sentido, a geração de carcinógenos nos órgãos utiliza o “position” para definir o ponto onde esse agente cancerígeno surgirá. Assim, a cena de carcinógenos, que serve como template para todos os inimigos, é instanciada inúmeras vezes, tendo seu ponto inicial sorteado a cada “spawn”.

```
var badMoleculeScene = badMolecule.instance() # Cria o carcinógeno
var locationPoint = rng.randi_range(1, 8) # Sorteia um número entre 1 e 8
# O bloco abaixo escolhe uma posição de onde o carcinógeno vai surgir segundo
# o locationPoint. Os nós PositionX para encontrar os pixels de posição.
# Já a velocidade indica também a direção na qual o carcinógeno será criado.

if locationPoint == 1:
    badMoleculeScene.position = Vector2(600, 200)
    badMoleculeScene.velocity = Vector2(rand_range(-100, -20), rand_range(-160, -70))
elif locationPoint == 2:
    badMoleculeScene.position = Vector2(337, 385)
    badMoleculeScene.velocity = Vector2(rand_range(-100, -20), rand_range(-160, -70))
elif locationPoint == 3:
    badMoleculeScene.position = Vector2(436, 382)
    badMoleculeScene.velocity = Vector2(rand_range(160, 70), rand_range(-160, -70))
elif locationPoint == 4:
    badMoleculeScene.position = Vector2(226, 144)
    badMoleculeScene.velocity = Vector2(rand_range(160, 70), rand_range(160, 70))
elif locationPoint == 5:
    badMoleculeScene.position = Vector2(140, 306)
    badMoleculeScene.velocity = Vector2(rand_range(-100, -20), rand_range(-100, -20))
elif locationPoint == 6:
    badMoleculeScene.position = Vector2(226, 144)
    badMoleculeScene.velocity = Vector2(rand_range(160, 70), rand_range(160, 70))
elif locationPoint == 7:
    badMoleculeScene.position = Vector2(358, 229)
    badMoleculeScene.velocity = Vector2(rand_range(-160, -70), rand_range(160, 70))
```

Ademais, a grandeza velocidade é empregada implicitamente no movimento da Celinha e dos carcinógenos. Isso se dá através do método move_and_collide(). Nesse caso, o argumento passado é um Vector2 que indica tanto

a direção do movimento, por meio da soma vetorial de suas coordenadas, quanto a velocidade, de acordo com a magnitude do vetor resultante.

```
var badMoleculeScene = badMolecule.instance() # Cria o carcinógeno
var locationPoint = rng.randi_range(1, 8) # Sorteia um número entre 1 e 8
# O bloco abaixo escolhe uma posição de onde o carcinógeno vai surgir segundo
# o locationPoint. Os nós PositionX para encontrar os pixels de posição.
# Já a velocidade indica também a direção na qual o carcinógeno será criado.

if locationPoint == 1:
    badMoleculeScene.position = Vector2(600, 200)
    badMoleculeScene.velocity = Vector2(rand_range(-100, -20), rand_range(-160, -70))
elif locationPoint == 2:
    badMoleculeScene.position = Vector2(337, 385)
    badMoleculeScene.velocity = Vector2(rand_range(-100, -20), rand_range(-160, -70))
elif locationPoint == 3:
    badMoleculeScene.position = Vector2(436, 382)
    badMoleculeScene.velocity = Vector2(rand_range(160, 70), rand_range(-160, -70))
elif locationPoint == 4:
    badMoleculeScene.position = Vector2(226, 144)
    badMoleculeScene.velocity = Vector2(rand_range(160, 70), rand_range(160, 70))
elif locationPoint == 5:
    badMoleculeScene.position = Vector2(140, 306)
    badMoleculeScene.velocity = Vector2(rand_range(-100, -20), rand_range(-100, -20))
elif locationPoint == 6:
    badMoleculeScene.position = Vector2(226, 144)
    badMoleculeScene.velocity = Vector2(rand_range(160, 70), rand_range(160, 70))
elif locationPoint == 7:
    badMoleculeScene.position = Vector2(358, 229)
    badMoleculeScene.velocity = Vector2(rand_range(-160, -70), rand_range(160, 70))
```

Sorteio de posição de ponto inicial para cada carcinógeno

```
func get_direction():
    var direction = Vector2.ZERO
    if isMoving:
        if Input.is_action_pressed("ui_up"):
            direction += Vector2(0, -1)
        if Input.is_action_pressed("ui_down"):
            direction += Vector2(0, 1)
        if Input.is_action_pressed("ui_left"):
            direction += Vector2(-1, 0)
        if Input.is_action_pressed("ui_right"):
            direction += Vector2(1, 0)
    return direction.normalized()
```

Método que define direção do vetor da Celinha a partir do input do jogador

```
func get_movement():
>|    var direction = get_direction()
>|    if Input.is_action_pressed("shift"):
>|        return direction * 5.5
>|    else:
>|        return direction * 2.5
```

Método que define vetor de velocidade da Celinha ao multiplicar o vetor de direção pela magnitude da velocidade desejada

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.