

DOCUMENTO PARA DESIGN DE GAMES

Inteli: Under Attack!

André Luís Lessa Junior

Jonas Viana Sales

Moisés Cazé de Souza Santos

Oliver Silva

13/03/2022

3.0

Controle do Documento

Histórico de revisões

Data	Autor	Versão	Resumo da atividade
<10/02/2022 >	<João Alfredo Cardoso Lamy>	<1.0>	<Realizada a primeira versão do GDD>
18/02/2022	João Alfredo Cardoso Lamy	1.1	Atualização do GDD (1.6, 1.7.1, 1.7.3, 1.7.5 e 1.7.6)
18/02/2022	Jonas Viana Sales	1.1	Atualização 1.7.4 User Stories
18/02/2022	André Luís Lessa Junior	1.2	Atualização 1.7.2 Persona e 1.7.1 requisitos
25/02/2022	André Luís Lessa Junior	2.0	Atualização sessões 1.7.5, 2.1, 2.2 e 2.3
11/03/2022	André Luís Lessa Junior	3.0	Atualização da sessão 5.0 e 5.2

Sumário

1. Visão Geral do Projeto <ADALOVE – Definir Proposta de Valor>	6
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
2. Game Design <ADALOVE – Elaborar fluxograma do jogo>	8
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
3. Level Design (opcional) <ADALOVE – Implementar mecânicas básicas do jogo - parte 1 a 5>	13
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
4. Personagens <ADALOVE – Implementar mecânicas básicas do jogo - parte 1 a 5>	16
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
5. Teste de Usabilidade para Qualidade de Software	18
5.1 Teste de Usabilidade	18
5.2 Métricas de Qualidade	18
6. Relatório - Física e Matemática	19

6.1 Funções	19
6.2 Cinemática Unidimensional	19
6.3 Vetores	19
6.4 Cinemática Bidimensional e mais	19
7. Bibliografias	20
Apêndice	21

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

1.1 Objetivos do Jogo

Atualmente temos uma falta de engenheiros e profissionais da tecnologia sendo formados no Brasil. Além disso, muitos dos cidadãos que se interessam pela área não sabem o que cada curso oferece. O propósito do nosso jogo é sanar ou ao menos amenizar as dúvidas de quem tem interesse nas carreiras relacionadas à tecnologia, proporcionando-os com as informações necessárias para que possam chegar em uma decisão bem-informada.

1.2 Características gerais do Jogo

O jogo será feito através de desafios em vários mapas utilizando um sistema de plataforma com exceção do hub, entretanto a dinâmica do jogo irá ser mudada conforme a necessidade de aplicação de novos métodos para poder ensinar os alunos sobre determinado curso, no qual é oferecido pelo Inteli, entendendo as possibilidades de carreira dessa área. O jogo foi inspirado em Celeste, no qual também é um jogo de plataforma, as fases são difíceis, porém é um jogo em que você aprende errando, sendo importante tentar várias vezes.

1.3 Público-alvo

Dado as dificuldades dos jovens em escolher uma carreira com relação a tecnologia, o nosso público alvo será jovens que estão no momento de ingresso na faculdade através dos vestibulares, e aqueles que já iniciaram a faculdade, mas ainda tem dúvidas sobre qual carreira seguir na área de tecnologia, tendo entre 18 a 23 anos.

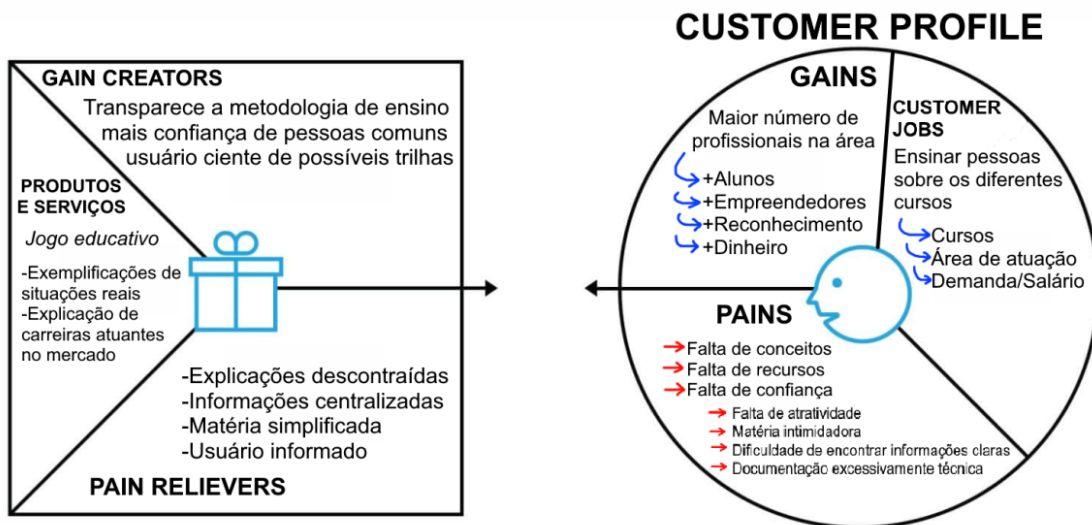
1.4 Diferenciais

O nosso jogo se diferencia por ter um foco em apresentar informações reais de maneira interessante ao usuário. Entendemos que jogos educativos podem ser (e na maioria das vezes são) atividades que não entretêm, servindo como uma maneira alternativa de aplicar o método tradicional e, para muitos, ultrapassado de ensino. Por isso, nosso grupo tem como foco central o entretenimento, assim, o usuário (entusiasta interessado por cursar carreiras tecnológicas no ensino superior) absorve muito mais as informações, tendo uma experiência divertida enquanto aprende informações importantes e relevantes no cenário de TI.

1.5 Análise do cenário: Matriz SWOT



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

- Jogabilidade clara.
- Replayability é essencial.
- Estimular a curiosidade do jogador.
- Refletir os valores do INTELI, sendo acessibilidade e liberdade.
- Abordar as capacitações que estão no site do Inteli.
- As carreiras devem ser representadas de maneiras que deixem clara que não existe caminho fixo ao concluir um curso.
- Ciência da Computação: algoritmos. Sistemas de Informação: gerenciamento. Engenharia da Computação: hardware. Engenharia de Software: projetos.
- Tudo gira em torno de POSSIBILIDADES, não de carreiras ou empregos com o mesmo nome da graduação (exemplo: cientista de dados).
- Incluir profissional Jr., Pleno e Sr. StackOverflow coloca informações relevantes sobre as áreas, importante base de consulta.
- Possibilidade de exagerar a realidade. “Fiquem livres para pensar no que vocês acham que é interessante”.
- O foco do jogo é ser legal e passar informações.
- A área de computação é extremamente dinâmica. Em constante mudança.
- Personagem que se desenvolve ao longo do jogo.
- Mudança do personagem principal.

1.7.2 Persona



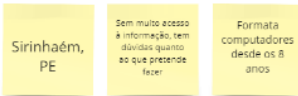
NOME: Marcelo Freitas (Marcelinho)

IDADE: 18 anos

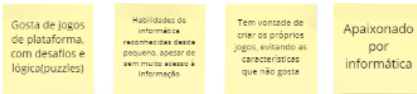
OCUPAÇÃO: Estudante (Ingressando no vestibular)

"A plenitude é a maior inimiga da evolução"

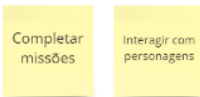
Biografia:



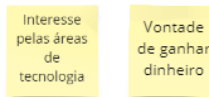
Características (personalidade, conhecimentos, interesses, habilidades):



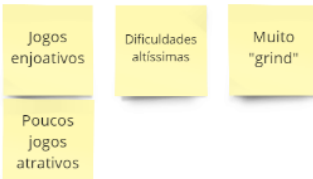
Motivações com jogos:



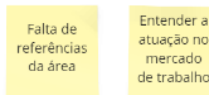
Motivações com o problema:



Dores com jogos:



Dores com o problema:



Link para melhor visualização:

https://miro.com/welcomeonboard/eGQ2YIV4YXNWT0dZWdHWMGloVlg2NUp5T0x5RUIRaEtpNTJXY29BamFJT0ZmamMzcldZV3V4MnNlajB4T1BYXWzNDU4NzY0NTE4NjlyNjI5MTY2?invite_link_id=3361785431

1.7.3 Gênero do Jogo

Aventura e Exploração (com base em jogos como Celeste, Zelda II, etc.)

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

1. Como um estudante que participa do Processo Seletivo Inteli, quero aprender sobre os cursos Inteli, a fim de escolher uma carreira profissionalizante.
2. Como um estudante que se interessa na Inteli, quero aprender as aplicações do mercado de trabalho de C.C., E.S., S.I. e E.C. A fim de escolher um curso que caiba comigo.
3. Como jogador, quero aprender de maneira divertida e educacional os cursos Inteli, para selecionar cursos que me identifique.

4. Como jogador, quero alcançar as escolhas de carreira/empresa, a fim de colocar meus treinamentos em prática.
5. Como personagem do jogo, quero conhecer trilhas acadêmicas e profissionais, para identificar qual a melhor opção para mim.
6. Como jogador, quero conversar com instrutores de cada curso, a fim de iniciar os tutoriais sobre a área a ser estudada.
7. Como jogador, irei explorar o Hub a fim de entender as carreiras que a organização Inteli proporciona, entendendo também as possibilidades no mercado de trabalho.
8. Como jogador, irei enfrentar desafios em um mapa capturando peças de um robô para entender a sua montagem e em qual carreira se aplica (Engenharia da computação).

1.7.5 Mecânica

No hub (seletor de fases e áreas), a orientação e movimentação do personagem será feita de maneira sem gravidade perpétua, como se ele estivesse no chão (sem conseguir pular). Pense no hub do Pokémon, ou Mario Bros. As fases seriam de plataforma, com mecânicas de gravidade invertida, voo, entre outras. Essas mecânicas dependem muito das fases, por exemplo: na fase de ciência da computação, você precisará controlar um robô a partir de código para desbloquear uma porta.

- Parte 1 Criação de um menu e hub (Item 7 do 1.7.4)
- Parte 2 Colocar NPC e definir funções dos personagens (Item 7 do 1.7.4)
- Parte 3 Criação da fase 1 (Item 8 do 1.7.4)
- Parte 4 Criação da fase 2
- Parte 5 Criação da fase 3

1.7.6 Fontes de Pesquisa / Imersão

Fonte
1. Control
2. Pokémon
3. Celeste
4. Legend of Zelda: A Link to the Past
5. Legend of Zelda: The Adventure of Link

6.	Super Mario World
7.	Super Mario 3
8.	Diablo (geometria)
9.	Twilio (jogo educativo)

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

2.1 História do Jogo

O jogo se passará no Inteli, onde o jogador poderá escolher um personagem sendo já determinados (Não será customizado), podendo ser homem, mulher, animais ou objetos, a organização Inteli tem o objetivo de formar profissionais altamente capacitados para as empresas que estão tendo problemas em encontrar profissionais da área de tecnologia.

O conceito do jogo é baseado em possibilidade, o jogador pode fazer fases relacionadas às diversas áreas e receberá moedas por essas missões, podendo escolher habilidades conforme a quantidade de moedas que recebeu, entendendo que uma carreira não está relacionada a apenas uma área que há diversidade, entretanto se escolher estudar determinada área estará mais próximo dessa carreira.

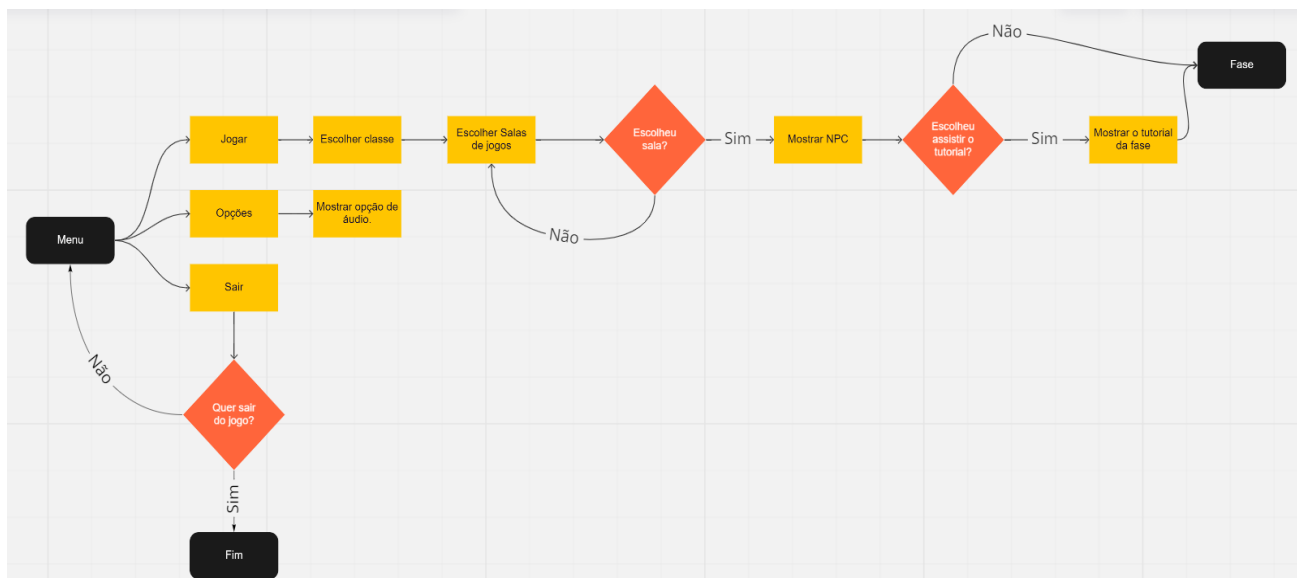
Essas habilidades serão definidas classes, a partir das quantidades de moeda o jogador poderá comprar uma classe que seria um título de um profissional no mercado, além de receber instruções antes de comprar, entendendo aquela classe, ou seja, como aquele profissional atua no mercado de trabalho.

Dentro dessa escolha de possibilidades, o jogador poderá escolher qual carreira quer seguir de início entrando em algum ateliê, na quais serão divididos por área, tendo um personagem não jogável para explicar um breve resumo desta carreira, ao escolher o ateliê que quer jogar os desafios iniciaram, sendo desafios diferentes em cada fase.

As fases serão padronizadas em mapa de estilo plataforma contendo diversos desafios dentro desses fazendo que tenha funcionalidades padrões que são para frente, para trás e pula, porém como cada fase terá um desafio diferente será necessário que em cada fase haja um recurso específico para alcançar o objetivo, por exemplo, na fase 1, foi utilizado um sistema de captura de itens.

O jogo poderá ter vários níveis lançados conforme o passar do tempo como em Candy Crush, no qual, a cada dia eles lançavam 5 fases, por nosso jogo ser mais complexo, levaria mais tempo, o jogo pode ser jogado várias vezes, além de possibilitar o jogador comprar vários títulos.

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



Link para visualização:

https://miro.com/welcomeonboard/a1I0c29NVTBveGVxaHU4cGpwY2RmWDJzTDFkTTZNRUISNWNiQ3VscTNJRVB3Rkdkd3BqYU95dGFTbGdWU0o0b3wzNDU4NzY0NTE4NjlyNjI5MTY2?invite_link_id=129224484930

2.3 O Mundo do Jogo

2.3.1 Locações Principais e Mapa

A principal locação do mapa é o Hub após entrar no game que será baseado no Inteli, a partir disso, as fases vão sendo adaptadas podendo continuar no Inteli ou ir para ambientes corporativos, isso dependerá de cada fase e os objetivos.

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.

2.3.3 Escala

Os personagens são minúsculos comparado ao mundo em que ele deve explorar. Com relação ao player comparado ao recursos do jogo o tamanho está equilibrado, sendo quase do mesmo tamanho, com exceção dos portais que são duas vezes maiores que os personagens para dar uma sensação de “portas”, tendo a escala 1:2.

2.3.4 Ambientação

A ambientação ainda não foi aplicada, entretanto o modo em que foi estruturado lembra um espaço fechado, na qual dá uma sensação de estar dentro de um prédio que em nosso jogo seria no Inteli.

2.3.5 Tempo

Não é aplicável em nosso jogo.


2.4 Base de Dados

2.4.1 Inventário


O itens serão agrupados de acordo com fases, já que o intuito será o mesmo em uma mesma fase, por exemplo, na fase 1 de engenharia da computação os itens terão que ser capturados para montar um robô, logo eles estarão agrupado na mesma categoria

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.

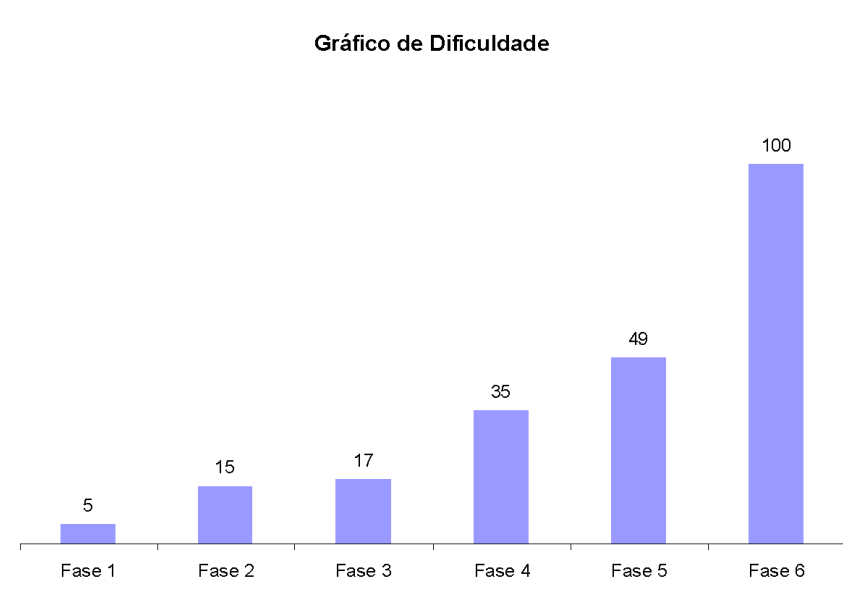


Figura 2. Gráfico de dificuldade para o jogo NOME DO JOGO.

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.

3.1.2 Layout Área (opcional)

Construção do *layout area* 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
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.

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>

Todos os personagens estarão disponíveis antes de entrar nas fases, ou seja, antes de adentrar ao hub os jogadores poderão escolher os personagens no qual gostar ou se identificar, entretanto no nosso protótipo existirá apenas um personagem que será o Chimbinha deixando a escolha do cliente adicionar mais personagens jogáveis ou não.

Character Appearance Chart

Personagem	Hub
Chimbinha	

4.1 Personagens Controláveis

4.1.2 < Personagens principais n>

O personagem será uma luva chamada Chimbinha.

4.1.2.1 Backstory

Chimbinha é uma luva que já teve várias experiências na área de tecnologia durante o ensino médio, entretanto ainda não decidiu a carreira que quer seguir, sendo selecionada para ser aluna do Inteli e encontrar o seu caminho.

4.1.2.2 Concept Art

A seguir iremos apresentar a pixel arte do chimbinha, que foi criado com base nas luvas que são entregues no almoço do Inteli. Na parte frontal está a luva com um rosto como se fosse desenhado com caneta, enquanto atrás o camisa 10 refere-se o quão talentoso os alunos do Inteli são, também aparece um dos parceiros do Inteli, tanto que fornece projetos, bolsas e notebook.



4.1.2.3 Ações Permitidas

Na fase 1 de engenharia da computação, chimbinha por ser uma luva, consegue capturar os itens para concluir a

4.1.2.4 Momento de Aparição

Irá aparecer no Hub.

4.2 Common Non-Playable Characters (NPC)

4.2.1 <NOME DO NPC COMUM *n*>

Ana Cristina é uma professora da organização Inteli, ela é muito legal e atenciosa com os alunos, tendo uma didática incrível fazendo qualquer aluno aprender, ficando localizada no Hub do ateliê 1 e também na fase 1 desse hub na qual refere-se ao curso de engenharia da computação. Ana é um NPC na qual irá dar a introdução do ateliê 1, falando um pouco sobre engenharia da computação e na fase 1 ela explica como funciona a fase e faz um fechamento quando o player concluí-la.

4.3 Special Non-Playable Characters (NPC)

4.3.1 <NOME DO NPC ESPECIAL *n*>

Recurso não aplicado em nosso jogo.

5. Teste de Usabilidade para Qualidade de Software <ADALOVE – Desenvolver

relatório de resultados do playtest>

5.1 Teste de 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?

5.2 Métricas de Qualidade

As métricas de qualidades foram estabelecidas com as funções atuais do jogo e seus funcionamentos relacionando-as como o objetivo final do jogo, estando sujeito a alterações conforme forem aplicadas novas funções para que o jogo atenda as necessidades dos clientes e dos usuários.

Funcionalidades	Adequação	O jogo é adequado para pessoas que não tem conhecimento das carreiras de tecnologias e que queiram aprender sobre elas.
	Acurácia	O jogo gera conhecimento das áreas de tecnologias: Ciências da computação, Engenharia da computação, Engenharia de software e Sistema da informação.
	Interoperabilidade	O jogo está sendo entregue em uma versão executável para Windows e uma versão HTML para ser disponibilizada em um servidor web.
	Segurança de acesso	O jogo não solicita dados pessoais, não impedindo o jogador de utilizá-lo, logo não armazenamos dados.
	Conformidade	Para a aplicação do conteúdo dos cursos foram seguidas as diretrizes curriculares nacionais do Ministério da Educação.
Confiabilidade	Maturidade	O jogo apresentou apenas uma falha na sprint 3, mas que não atrapalha a jogabilidade, que em comparação as sprints anteriores tinha diversos erros de código, logo nota-se que o código está mais limpo.
	Tolerância a falhas	Caso haja uma falha inesperada o jogo irá continuar rodando normalmente, podendo não executar algum tipo de função, entretanto não irá atrapalhar a jogabilidade.
	Recuperabilidade	Não se aplica.
Usabilidade	Inteligibilidade	As mecânicas são fáceis de utilizar, sendo no geral para frente, para trás e para pular, já as habilidades específicas de cada fases serão explicadas através de NPC's.
	Apreensibilidade	O jogo usa mecânicas básicas do computador, exigindo habilidades motoras básicas como (WAD) ou (Cima, Lado esquerdo e direito), também estratégicas.

		Na fase 1 é implementado o espaço para captura de itens.
Eficiência	Comportamento em relação ao tempo	Existe um intervalo imperceptível entre as mudanças de cenas, não atrapalhando a jogabilidade.
	Comportamento em relação a recursos	Não é aplicado no jogo.
Manutenibilidade	Analísabilidade	O jogo é fácil de encontrar falhas, devido a organização dos códigos, estando padronizado.
	Modificabilidade	O jogo é fácil de ser modificado, por causa da estrutura de códigos e as cenas estarem individuais, então alterações pontuais são facilitadas.
	Estabilidade	Há alto risco de bugs ao alterar o código, por causa que está tudo integralizado.
	Testabilidade	A testabilidade é facilmente realizada pela ferramenta do godot.
Portabilidade	Adaptabilidade	O jogo é facilmente adaptável, tendo o modo executável e html.
	Capacidade para ser instalado	Não é preciso ser instalado, é necessário ter um navegador.
	Capacidade para substituir	Não é possível substituir.
	Conformidade	Padrões do Godot.

6. Relatório - Física e Matemática

6.1 Funções

Para usar as funções foi necessário primeiro declarar as variáveis e constantes, estas são grandezas que podem adquirir diversas finalidades estabelecidas, em nosso jogo utilizamos uma constante chamada gravidade e uma variável chamada velocidade.

```
const GRAVITY:float = 1700.0
```

A constante gravitacional (const GRAVITY) está relacionada a gravidade do jogo, que forçará o personagem ficar no chão. O valor float é utilizado para determinar numeração com decimais, outro ponto importante é o valor positivo, pois no plano cartesiano da programação o eixo das abscissas é invertido, ou seja, os números positivos estão direcionados para baixo nesse plano, logo a gravidade para empurrar o boneco para baixo é positiva. Esta função mostra como funciona nossa gravidade, e logo abaixo o código:

$$f(x) = x + g \cdot d$$

```
_out.y += GRAVITY * get_physics_process_delta_time()
```

```
# delta_time é o tempo que se passou entre dois frames, utilizado principalmente para manter consistência.
```

6.2 Cinemática Unidimensional

Na cena do Hub não utiliza a aplicação da física, apenas na colisão do personagem para que ele se mova apenas nos lugares demarcados e não na área inteira da tela, foi criada uma var de direção e dizendo que nela não se aplica no vetor, também a função move_and_slide() faz com que o personagem se mova na tela através dos pixel e não pelo trabalho da força, mantendo uma velocidade constante. Utilizando apenas cinemática unidimensional, pois ele consegue se mover pelo pixel sem que uma força interfira como a gravidade.

```
15 > > > "Hub":  
16 > > > var _direction: = get_non_physics_direction()  
17 > > > _velocity = calculate_non_physics_move_velocity(_direction, SPEED)  
18 > > > _velocity = move_and_slide(_velocity)  
19 > > >
```

Na cena abaixo mostra quais os movimentos essa cena sem física pode realizar, movendo-se em um plano cartesiano na qual não é aplicado uma força física sobre o personagem sendo gerado movimentos na função de get_non_physics_direction. No código percebe-se o Vector2 significa que é um corpo em um plano 2D, realizando as ações para direita, esquerda, para baixo e para cima (Apesar de no código está escrito jump, o personagem não pode realizar pulos, ao invés disso ele vai para cima).

```
26 > func get_non_physics_direction() -> Vector2:  
27 > > return Vector2(Input.get_action_strength("move_right") - Input.get_action_strength("move_left"),  
28 > > Input.get_action_strength("move_down") - Input.get_action_strength("jump"))  
29 >
```

6.3 Vetores

As funções aqui serão para calcular os vetores tanto num espaço sem física aplicada (Hub) quanto um espaço com física aplicada (Nível 1).

Na função abaixo usa-se uma função para calcular o vetor desse corpo na qual não está sendo aplicada processos físicos, primeiro foi adicionado speed e direction na função, fazendo o corpo receber essas grandezas, depois foi criado uma variável, para que normalize a velocidade sendo aplicado na variável “_out.” os valores do ponto x. e y, fazendo a movimentação nos pontos e multiplicando pela direção desses pontos.

```
46 ▾ func calculate_non_physics_move_velocity(  
47   ▸   ▸   _direction,  
48   ▸   ▸   SPEED  
49   ▸   ) -> Vector2:  
50   ▸   var _out = _velocity.normalized()  
51   ▸   _out.x = SPEED.x * _direction.x  
52   ▸   _out.y = SPEED.y * _direction.y  
53   ▸   return _out
```

Já nesta função na qual há física sendo aplicada segue os mesmos primeiros procedimentos para aplicar o cálculo do vetor adicionando uma interromper os pulos, ou seja, se o personagem pular, mas em determinado ponto apertar o botão para baixo, este personagem irá parar o pulo e começará a cair.

Na parte de calcular é necessário estabelecer que o y irá receber uma força oposta que seria a gravidade, utilizando delta_time calculando o tempo entre dois frames, mantendo a consistência para suavizar os movimentos com relação ao frame.

```
31 ▾ func calculate_physics_move_velocity(  
32   ▸   ▸   _direction,  
33   ▸   ▸   SPEED,  
34   ▸   ▸   _is_jump_interrupted  
35   ▸   ) -> Vector2:  
36   ▸   var _out = _velocity  
37   ▸   _out.x = SPEED.x * _direction.x  
38   ▸   _out.y += GRAVITY * get_physics_process_delta_time() # delta_time é o tempo que se passou entre dois frames, utilizado principa  
39 ▾ ▸   if _direction.y == -1.0:  
40   ▸   ▸   _out.y = SPEED.y * _direction.y  
41 ▾ ▸   if _is_jump_interrupted:  
42   ▸   ▸   _out.y = 0.0  
43   ▸   return _out
```

6.4 Cinemática Bidimensional e mais

Na cinemática bidimensional pode-se utilizar a gravidade, ou seja, serão aplicadas diversas forças no personagem. Primeiro irá iniciar declarando uma variável na qual impede o pulo que seria jump interrupted usando uma lógica booleana, ou seja, declarando falso e zerando a velocidade no y.

Depois foi necessário definir a direção, fazendo a variável direction receber processos físicos, calculando a direção, movimentação e o jump interrupted. Por fim, aplicar na movimentação no move and slide a velocidade e no corpo acrescentando pulo.

```
8 ~ func _physics_process(_delta: float) -> void:
9 ~ >| match get_tree().current_scene.name: # Analisa quais cenas devem receber física e quais não devem.
10 ~ >| >| "Level01":
11 ~ >| >| >| var _is_jump_interrupted:bool = Input.is_action_just_released("jump") and _velocity.y < 0.0
12 ~ >| >| >| var _direction: = get_physics_direction()
13 ~ >| >| >| _velocity = calculate_physics_move_velocity(_direction, SPEED, _is_jump_interrupted)
14 ~ >| >| >| _velocity = move_and_slide(_velocity, Vector2.UP)
```

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.