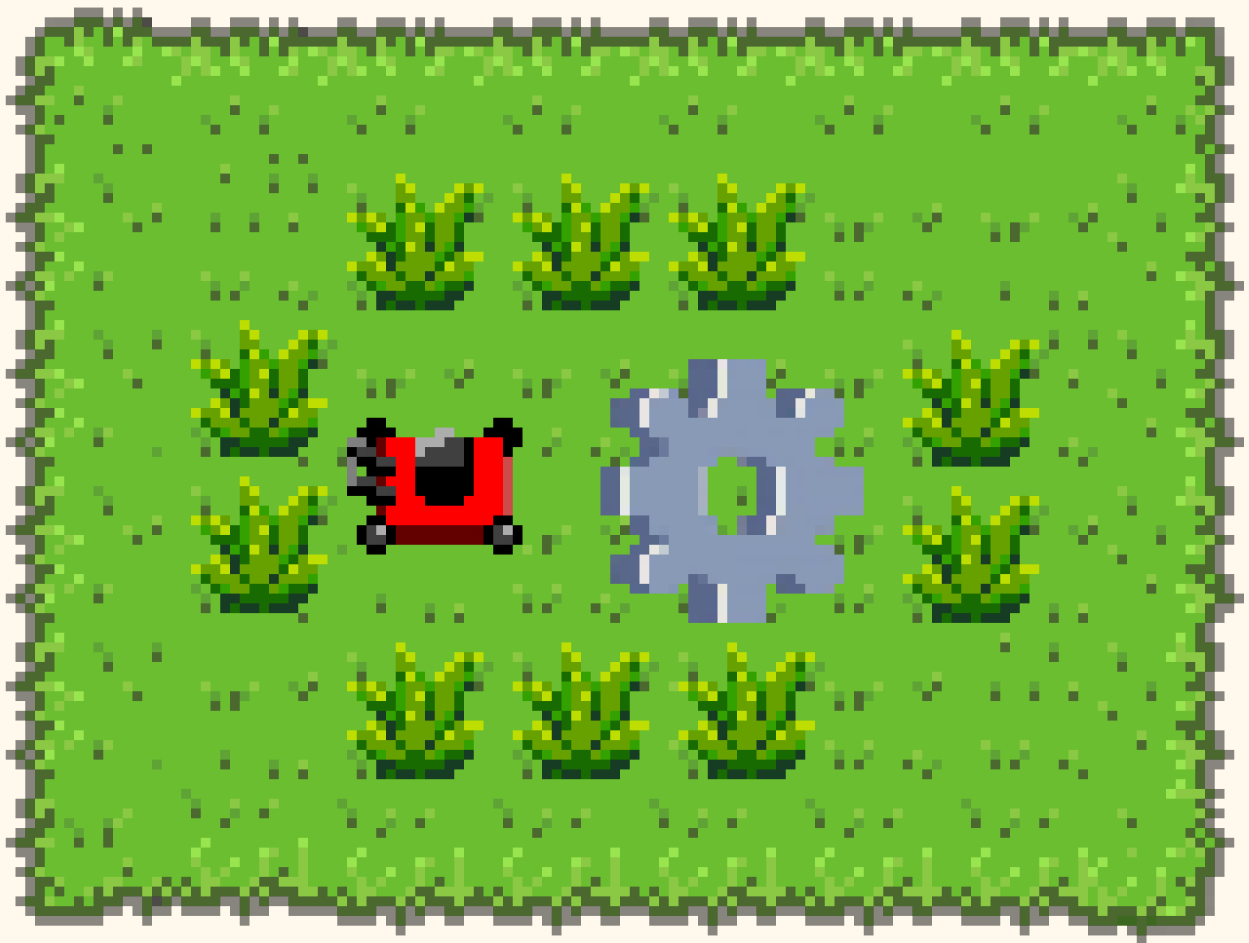


Equipe show
10 de Junho de 2022

CutCut

Game Design Document



Equipe: João neto, Julia nogueira, Ismael Rodrigues (O nível de desinteresse dos outros membros é tão grande que vou nem colocá-los como membros participantes, afinal de contas eles nem leram o documento e não viram essa frase)



Sumário

I. Visão geral essencial	3
a. Resumo	3
b. Aspectos fundamentais	3
c. Golden nuggets	3
II. Contexto do game	4
a. História do game	4
b. Eventos anteriores	4
c. Principais jogadores	5
III. Objetos essenciais do game	5
a. Personagens	5
b. Armas	9
c. Estruturas	10
d. Objetos	11
IV. Conflitos e soluções	17
V. Inteligência artificial	18
VI. Fluxo do game	18
Fases	18
Elemento desafio	19
Seleção de fases	20
VII. Controles	21
a. Principais inputs	21
b. Direcionais	22
VIII. Variações de jogo	22
IX. Definições	22
X. Referências	22
XI. Extras	23



I. Visão geral essencial

a. Resumo

Neste game, o jogador terá que controlar um cortador de gramas que possui um único objetivo: Cortar todas as gramas da fase, porém sua tarefa será dificultada com alguns quebra-cabeças, como apertar alguns botões para acessar uma área ou fugir de alguns inimigos e aproveitar de elementos do cenário para finalizar seu objetivo de cortar toda a grama. c

b. Aspectos fundamentais

Single-player: Nosso jogo será desenvolvido sobre uma concepção single-player e não fornecerá suporte multi-player.

Gráficos em duas dimensões (2D): Nosso jogo utilizará a engine GameMaker 2 para o desenvolvimento e utilizará sprites sequenciais para fazer a plotagem gráfica e animações.

Cenários diversificados: Dependendo do progresso do jogador, o cenário seguirá o estilo do mundo selecionado, para acompanhar a dinâmica de desafios do determinado mundo e suas fases.

Jogabilidade em 4 botões estilo cruz: O personagem andará apenas no eixo X e Y sem permissão para andar na diagonal, o click do botão direcional fará com que o personagem ande 1 bloco na direção desejada, caso houver uma grama ela será cortada.

c. Golden nuggets

As fases serão os puzzles separados com um único objeto: o jogador deve cortar toda a grama do cenário. Ao concluir a fase o jogador será levado para a próxima fase com uma dificuldade um pouco elevada ou um design mais divertido. Ao concluir todas as fases de um mundo será liberado o próximo mundo com novos puzzles e desafios que o jogador terá que aprender. E claro, cada fase terá seus temas lunáticos e curiosos... Cortar grama poderá ser uma tarefa um pouco mais complicada.



II. Contexto do game

a. História do game

Você é apenas um jovem garoto querendo curtir suas férias de verão e talvez conseguir uma grana extra, morando em um bairro comum cheio de gramas para cortar e vizinhos bem... incomuns... Um desses vizinhos é seu próprio tio Aldercort Yubabo (conhecido como tio yu ou tio bababu). Aldercort é conhecido por ser um mago arrogante, seu único objetivo de vida é ser um cara chato, querendo sempre ser superior aos seus descendentes, como por exemplo o seu sobrinho, que o viu desocupado e em busca de uma grana extra nas férias, decide então ser o primeiro cliente do próprio sobrinho, para ajudá-lo a conseguir uma renda extra.

Porém, Yubabo se sentiu cheio de fúria ao ver o garoto cochilando embaixo de uma árvore no seu jardim e não completando a sua missão. Indignado, o transforma no próprio cortador de grama como punição, e só será libertado quando aprender sua lição sobre RESPONSABILIDADE. Assim, o garoto estará fadado ao destino de cortar grama até que entenda a mensagem que seu tio quer mostrar... ou será que não...

Logo após concluir todo o jardim, seu tio ainda fica indignado, mas seus vizinhos querem o misterioso cortador de grama que corta tão bem. Assim, seu tio o empresta para as criaturas mágicas de que fez amizade ao longo de sua vida. Mal sabendo que o protagonista irá dar fim a essas criaturas mágicas sem ao menos perceber... fazendo com que o Yubabo fique mais irritado ao longo da jornada. Colocando obstáculos e desafios ainda maiores para testar a paciência do jovem garoto que só queria ficar em paz.

b. Eventos anteriores

Aldercort Yubabo era um jovem mago louco por conhecimento, criou amigos e criaturas mágicas para treinar suas habilidades, como o dragão malvado e o Mr. Snowman. Além de criar criaturas fantásticas que apelidou de Angus Seed Oak, que gritavam barulhinhos engraçados de "*Oa Oas*", então os apelidou desse nome. Depois de muito tempo e muitas eras, nasceu seu sobrinho, afim de que ele se tornasse um grande Bruxo, quis ensinar uma lição desde cedo no garoto, transformando-o em um objeto bem chato e dando uma tarefa bem trabalhosa pela frente.



c. Principais jogadores

Player Principal

Nome: No-name (o personagem será o jogador)

Biografia: Você é uma entidade cortadora de gramas com um único objetivo em mente: cortar a grama de todos os quintais para no fim ser libertado de uma situação terrível e constrangedora: Voltar a ser uma inocente criança que aprendeu sua lição.

Motivações: Tornar-se de volta uma criança de verdade, que nunca queria cortar nenhuma grama desde o princípio.

Relacionamento:

- Antagonista: tio bababu; herói das flores felizes;
- inimigos-principais: chefes de cada mundo(Fogo: quintal do dragão malvado, Gelo: quintal do Mr. Snowman, Floresta intensa: a decidir, deserto/eletricidade : Mr. Barman.). sobrinho do antagonista.
- Inimigos-secundários: Minions dos respectivos chefes

III. Objetos essenciais do game

a. Personagens

Player Principal

Nome: No-name (o personagem será o jogador)

Biografia: Você é uma entidade cortadora de gramas com um único objetivo em mente: cortar a grama de todos os quintais para no fim ser libertado de uma situação terrível e constrangedora: Voltar a ser uma inocente criança que aprendeu sua lição.

Motivações: Tornar-se de volta uma criança de verdade, que nunca queria cortar nenhuma grama desde o princípio.

Relacionamento:

- Antagonista: tio bababu; herói das flores felizes;
- inimigos-principais: chefes de cada mundo(Fogo: quintal do dragão malvado, Gelo: quintal do Mr. Snowman, Floresta intensa: a decidir, deserto/eletricidade : Mr. Barman.). sobrinho do antagonista.
- Inimigos-secundários: Minions dos respectivos chefes



Antagonista

Nome: Tio Aldecort Yubabo - (tio yu ou tio bababu)

Biografia: Um mago arrogante, seu único objetivo de vida é ser um cara chato, quer sempre ensinar seus familiares a forma correta de cortar uma simples grama. Afinal de contas a única coisa que ele entende é um entediante jardim bem cortado, então criou criaturas místicas e quintais envolventes para seus amigos curtirem o CHATO trabalho de cortar uma simples grama... Grama de fogo? Grama que parece um sorvete? Grama eletrizante? Deixe tudo com ele, criou todas as criaturas e estilos de grama para passar seu tempo como um mega mago chato.

Motivação: Quer fazer seu sobrinho aprender uma lição. "Responsabilidade". Para isso transforma o próprio parente em um cortador de grama. E Ele quer que o sobrinho passe por todo o tédio de cortar uma grama, afinal de contas, é um mago chato que não tem nada melhor pra fazer...

Relacionamento: Tio do Protagonista;

- Amigo do dragão malvado e criador de todos seus minions;
- Amigo do Mr. Snowman : Ele mesmo criou em uma brincadeira de infância;



- Amigo do mexicano : Mexicano adora seus quintais desérticos/ eletrizantes (A DECIDIR)
Inimigo das flores felizes: Odeia coisas "fofinhas" como as lindas flores que o sobrinho insiste em deixar-las intactas.
- Amigo do Dono do quintal florestal: A DECIDIR.

Flores Felizes :

Entidade - Baby's Breath

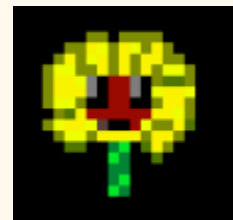
Scientific Name: Gypsophila paniculata Biografia: Um conjunto de flores simpáticas: elas representam Inocência, alegria.



Flores Felizes :

Entidade - SunFlower

Scientific Name: Helianthus annuus Biografia: Um conjunto de flores simpáticas: elas representam desejos, Lealdade.



Flores Felizes :

Entidade - Azalea

Scientific Name: Rhododendron atlanticum



Biografia: Um conjunto de flores simpáticas: elas representam fragilidade, paciência.

Motivações: São flores que cresceram em quintais estranhos. São uma prova viva que o mago não tem controle sobre seus poderes e que suas lições motivacionais não passam de uma desculpa para ser um mago tão teimoso e chato.



Relacionamento: O Tio bababu odeia essas coisas fora de controle, o protagonista adora ver quintais cheios dessas florzinhas fofinhas. São neutros em relação aos outros inimigos do protagonista.

Dragão malvado

Nome: Sr mr. dragão



Biografia: Sr mr. dragão conheceu o mestre Bababu em alguma de suas aventuras, conhecido por ter um bafo não muito agradável, ele é dono de alguns quintais que o tio Bababu deu uma arrumada, afinal de contas um dragão com um quintal cheio de grama? Pegaria fogo muito rápido. Então seus quintais são com gramas de brasa e chamas! Sim, fogo, cresce e tem sementes de fogo que o tio Bababu criou, afinal de contas é um mago chato que não quer ter

nenhum furo de roteiro. Motivações: Seu quintal está um caos! Olha para toda essa lava, brasa, fogo e espetinhos de salsicha! Alguém tem que cortar esta grama que dói quando encostamos. Todo esse desastre foi causado por seus filhos, sim, é difícil ser um pai solteiro.

Relacionamento: Amigão do Titio; É pai de muitos filhos mas com esposas diferentes(esses filhos estarão nos cenários e serão seus mini-inimigos)



Mr. Snowman



Nome: Frosty Berry Junior

Biografia: Frosty é uma criatura que foi concedido a dádiva da vida pelo Grandíssimo mago Aldecort Yubabo. É um ser gigante mas que não se movimenta muito bem. Seu sonho sempre foi curtir as praias do Havaí, mas infelizmente não sabe passar um protetor solar... Sorte que possui um quintal incrivelmente frio para cuidar enquanto resolve esse pequeno probleminha.



Motivações: Frosty Berry precisa deixar seus quintais preparados para poder viajar para o Havaí, mas com esse frio todo ele precisa de alguém que consiga andar por esse chão tão escorregadio, afinal... Frosty Berry não possui pernas.

Relacionamento: Amigão do Titio; Inimigo indireto do protagonista.

Minion: Plantador de sementes

Nome: Angus Seed Oak (Oa Oas)

Biografia: São pequenas criaturas que vivem em todos os quintais, aprenderam desde muito cedo que precisam jogar grãos no chão. Esses grãos rapidamente se tornam gramas fresquinhas que adoram o cheiro. Buscam sempre plantar a maior quantidade de grama. Ficam tristes ver sua linda graminha cortada. Foram criados por um mago em sua adolescência, em uma aula de botânica no ensino médio.

Motivação: Aprenderam a jogar sementes no chão por diversão, agora são viciados em quintais cheios de grama alta com muito local para pularem e se divertirem. Relacionamento: Odeiam o protagonista: Como alguém enxerido vem cortar nossa maravilhosa grama? Reconhecem o mago: A cara do mago é bem familiar, porém parece que se passaram alguns anos. Sendo assim, não se importam de seguir as ordens do Yubabo e foram espalhados por vários quintais por ordens dele



b. Armas

A principal habilidade do nosso jogador será cortar as gramas altas. Sem presença de power-ups secundários. Essa habilidade estará atrelada a movimentação do jogador. O jogador não se defenderá porque não existirá ameaças para ele, já que o foco do gameplay é a resolução do puzzle.



c. Estruturas

Paredes: Serão os arredores da fase, não necessariamente será retangular, depende em qual fase o jogador está e qual o estilo de design escolhido para aquela fase. Porém, terá um limite de 12x20 blocos, onde cada bloco corresponde a uma sprite do jogo do tamanho de 16x16 pixel.

- 1 estados:

1->Parede estática

- Variações? Sim

1-> Parede lateral esquerda,

2-> Parede lateral direita,

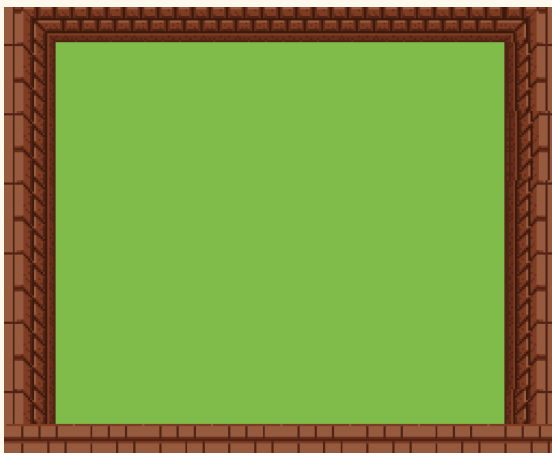
3-> Parede superior,

4-> Parede inferior,

5-> cantos: Inferior-direito, inferior-esquerdo, superior-direito, superior-esquerdo.

- Observação: Fazer apenas um esquema(1 sprite de cada) para poder manipular apenas na programação - economizar memória e espaço.

Rascunho referência:





OBS: Observem que foi necessário 2 sprites para manter o efeito de CONTÍNUO sem doer a visão do User.

d. Objetos

Objetos Definidos para level design e modelagem da programação.

Grama Alta: São o objetivo principal do player, deve saber que precisa passar por cima.

- 2 estados:

1->Grama Alta

2-> Grama Alta cortada

- Variações? Sim, poucas variações que não saiam muito do padrão (2 ou 3)

1-> SIM (2 ou 3 variações de Grama Alta)

2-> SIM (2 ou 3 variações de Grama Alta cortada)

- Animado? -> Não. Decisão de level design por retirar o foco do jogador -> muitas gramas em uma fase trás sensação de desconforto visual.

- Observação: Cada variação de Grama Alta estará atrelado a sua versão Grama Cortada -
Rascunho referência:



1.

2.

1. grama alta

2. grama alta cortada



Campo Caminho Livre: Player pode passar por esses campos, sem interações, são apenas caminhos seguros para o player andar

- 1 estados:

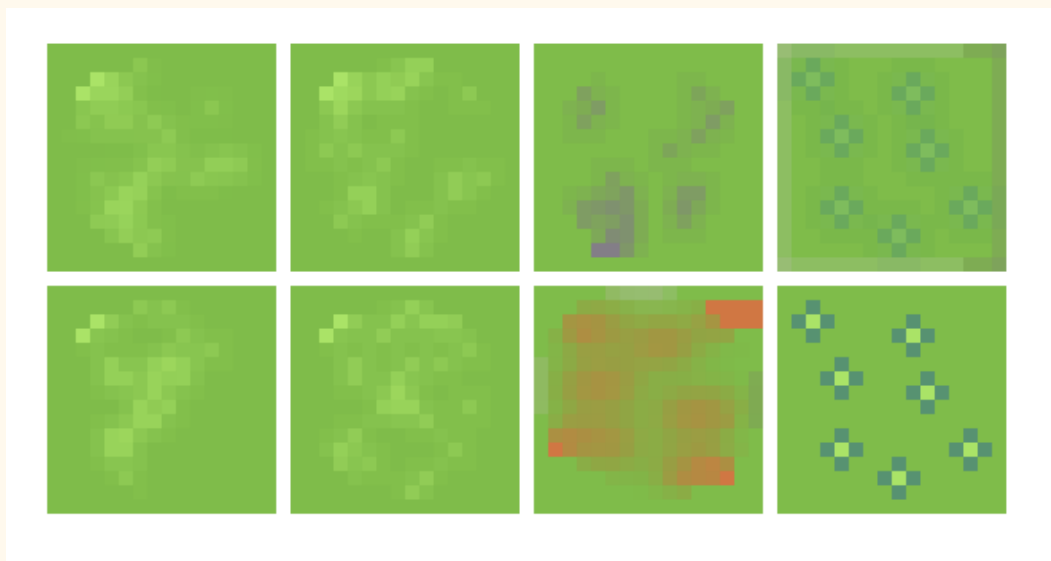
1->Campo estático

- Variações? Sim, muitas (5 ou mais)

Podemos ter campos de grama baixa, pequenas pedrinhas, pequenos objetos estáticos que não interagem com nenhum objeto(pedaços de caminhos, matinhos baixos, etc)

- Animado -> Não
- OBS: Palheta de cor que diferencie levemente do objetivo principal: Grama alta (escolha de level design)

Rascunho referência:



Plataforma-Campo-falso: Lógica: Campo que permite a passagem do player 1 vez, quando o player sair: um buraco é aberto e o player não consegue passar mais por esse caminho.



- 4 estados:

1-> Campo-falso - Sem Grama Alta

2-> Campo-falso - Com Grama Alta

3-> Campo-falso - Variação de campo aberto (A DECIDIR)

4-> Campo-falso - Buraco aberto

- Variações? Talvez

> dependerá do resultado final da arte, CASO não combine com muita variação de Campo Caminho Livre: necessário criar variações parecidas com Campo Caminho Livre.

- Animado-> Sim

>Animação da transição 1 -> 4 (buraco se abrindo): Instante após o player sair, a plataforma-falsa cai e é exibido um buraco aberto. Decisão de level design: a Animação começa imediatamente, pois o player não poderá voltar para esse caminho no instante logo após a saída do local.

- OBS: Requisição de level design: O jogador terá que visualizar um Campo-falso e saber diferenciar com o resto do cenário, porque a resolução de algumas fases precisa de movimentações específicas do jogador e para isso precisa reconhecer a estrutura e passar apenas 1 vez em algumas posições.
- Rascunho referência:



1.

2.

3.

4.

1. plataforma em campo aberto 2. Plataforma com grama alta 3. Plataforma Com campos 4. plataforma estado buraco aberto



Rochas: Rocha onde o player não poderá andar.

- 1 estado:
 - > Rocha estático
- Variações? Sim, poucas (1 ou 2, máximo 3)
- Animado? Não

Obs: Fazer UMA ou DUAS rochas que combinem com o cenário de campo aberto em várias situações: Rocha do lado de outras rochas, rocha do lado de variações de campos abertos, rochas do lado de grama alta, rocha do lado de grama alta cortada, rocha do lado de grama cortada, rocha do lado de flores-vibrantes. Necessário: efeito de repetição de Sprite.

- Rascunho referência:



Player: Jogador principal, movimentado pelo User.

- 16 estados (8 estados OBRIGATÓRIOS + 8 estados A DECIDIR)

-> Player PARADO: esquerda, direita, cima, baixo;
 -> Player MOVIMENTANDO: esquerda, direita, cima, baixo;
 -> Player CORTANDO: esquerda, direita, cima, baixo; (A DECIDIR AINDA)
 -> Player CAINDO EM PLATAFORMA FALSA BURACO-ABERTO (A DECIDIR)

- 4 variações:

-> Esquerda - personagem virado para esquerda;
 -> Direita - personagem virado para direita;

 -> Cima - personagem virado para cima;
 -> Baixo - personagem virado para baixo;

- Animado: SIM

-> Animação para os 16 estados: PARADO(4 direções), MOVIMENTANDO(4 direções), CORTANDO(A DECIDIR), CAINDO(A DECIDIR).

- PARADO: O User não apertou nenhum comando no teclado e o personagem não sai do local de onde está. User olha pro cenário buscando forma de resolvê-lo.

-> Ideia: Um cortador de grama não se move muito, mas é uma máquina ligada, imaginem um motorzinho que está parado, exemplo: um motor de um carro quando está parado.

- MOVIMENTANDO: O User está apertando botões de movimentação, o player sairá do ponto de origem e irá parar apenas quando o User soltar o botão de movimentação.

Lógica de animação ->

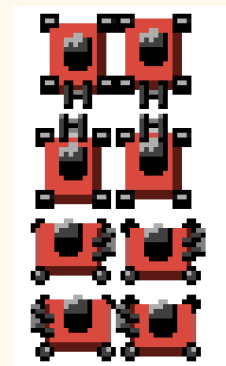
- PARADO vai_para MOVIMENTANDO, MOVIMENTANDO vai_para PARADO.

- CORTANDO(A DECIDIR): Quando player passar por grama alta, ainda não sabemos se terá apenas partículas de grama saindo por cima do Player: CASO tiver, pode ser implementado por PARTÍCULAS (A DECIDIR - por enquanto não essencial).

-CAINDO: O User, mesmo com a plataforma-Campo-Falso querer passar pelo ESTADO-BURACO ocorrerá um aviso visual(Talvez animação), se na 2a tentativa ele consegue cair: A DECIDIR - DECISÃO de Level Design não analisada ainda - Prototipagem necessária.

- OBS: Não façam nenhuma Sprite de CORTANDO ou CAINDO por enquanto.

- OBS: Notem que o player vai ter movimentação em GRID(andar em bloquinho pré-definidos sem a possibilidade de estar entre dois bloquinhos ao mesmo tempo). Assim, a Sprite deve respeitar essa limitação EM TODAS AS DIREÇÕES, então é necessário um design que seja proporção 1:1 ou que pelo menos se encaixe nas GRIDS. Observem a referência, é notório que o Sprite Não é 16x16, mas mesmo assim as rodas são o limitante do nosso GRID, a parte que sai





do 1:1 será a alça do Player. A parte da Alça ficará em cima de Sprite de todos os outros objetos da fase.

- Rascunho referência:

Flores-vibrantes: Lógica: Flores maiores que o Player não poderá passar por cima, caso passe, reinicie a fase(A DECIDIR - Decisão de Level Design Reiniciar ou limitar número de flores que poderá pisar? Protótipo necessário)

- 1 estado:

1-> Flor-paradinha;

- Variações? Sim - 3

- > Entidade - SunFlower
- > Entidade - Baby's Breath (Talvez)
- > Entidade - Azalea

- Animado-> Sim

>Animação para cada entidade de: DANÇAR, PARADO, OLHAR-LADOS.

DANÇAR: Flor alegre dança no mesmo local que está.

PARADO: Flor Antes ou depois de DANÇAR e Antes ou depois de

OLHAR-LADOS. *(florzinha vibrante não quer Fazer nenhuma das outras ações, ela quer apenas... ser uma florzinha bonitinha)*

OLHAR-LADOS: Flor Antes ou depois de DANÇAR e Antes ou depois de PARADO. *(Florzinha quer ver o que está em sua direita e esquerda, mas sem virar para olhar para trás nem para frente)*

>Escolha do estado da animação: Aleatório para cada entidade presente na fase.

>Ainda não analisado: CASO Player passar por cima: Animação da florzinha sendo esmagada OU animação FullScreen de Fim de fase (A DECIDIR - decisão de art design - prototipagem necessária).



- Rascunho referência:



1. SunFlower

2. Baby's Breath

3. Azalea

Minion "Oa Oas": São pequenas criaturas encontradas em algumas fases, estão sempre plantando sementes de grama, porém choram ao ver seu caminho ser destruído e param de plantá-las.

- 4 estados:

1-> Plantando enquanto caminha

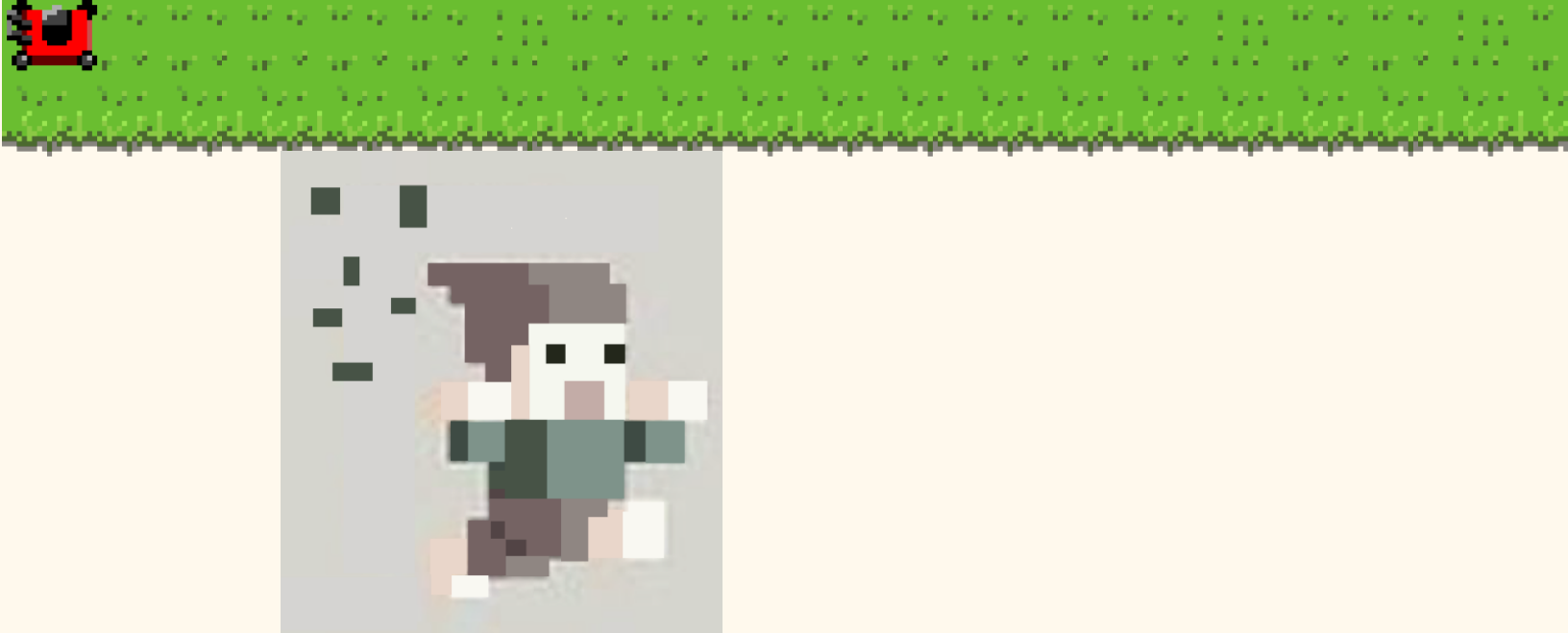
2-> Chorando no chão

- Variações? Não
- Animado-> Sim

>Animação de movimentação: Nas 4 direções(Up, Down, Left, Right).

>Animação de tristeza: Após o combo de cortes do player, o Minion senta no chão atual e derrama lágrimas em loop.

- OBS: Necessariamente o minion precisa aparentar estar colocando sementes por onde ele passa, pode ser keyframe dele jogando para trás enquanto caminha.
- Rascunho referência:



IV. Conflitos e soluções

Conflitos que achamos durante o planejamento e o que deve ser feito.

- a. Player caminha e colide com um Minion
 - i. Solução: Nada acontecerá, o player pode caminhar tranquilamente e o minion ficará em uma layer inferior ao player, os dois objetos podem ocupar o mesmo espaço, assim o player poderá cortar a grama de onde o minion está plantando.
- b. Player fica preso entre locais sem saída e não consegue finalizar o level.
 - i. Pode acontecer isso, já que temos elementos como a plataforma falsa, por isso, nesse tipo de situação-evento, decidimos criar um botão de restart do nível atual (Esse botão aparecerá na interface quando o player não mostrar progresso durante o jogo, e também estará disponível ao apertar Pause). Facilitando o desenvolvimento do nosso jogador a resolver o puzzle, além de prevenir outros bugs inesperados como, por exemplo, não terminar uma fase mesmo cortando todas as gramas e etc.



V. Inteligência artificial





Não haverá inimigos, assim, não focaremos em nenhum algoritmo de inteligência artificial.

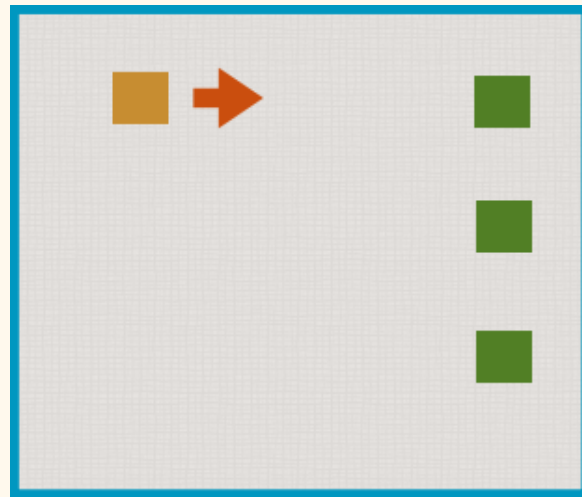
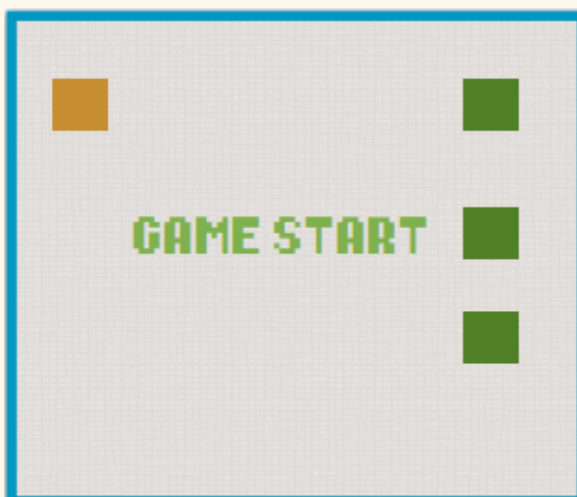
Porém, em casos onde o Angus Seed Oak (Oa Oas) aparecem, eles terão passos programados de acordo com suas fases. Por exemplo, o Level design planejou uma fase onde teremos um Oa Oas andando em círculos curtos 3x3, logo, teremos que colocar o objeto pré-programado para virar sempre em 3 passos para a direção subsequente. Ou queremos colocar para andar 10 passos e dar meia volta, faremos a pré-programação para ele de 10 passos e em seguida vire para a direção subsequente 2 vezes para dar meia volta.

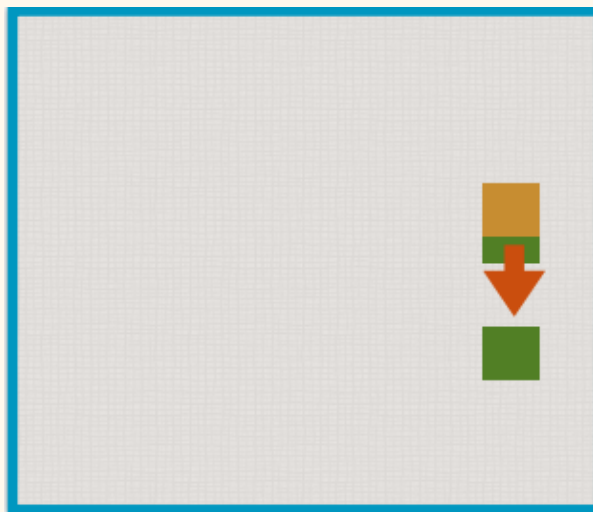
VI. Fluxo do game

Fases

A Fase será composta por uma grid (tamanho variado) com paredes que limitam os caminhos onde o jogador pode caminhar. Os espaços para andar: nesses espaços teremos alguns elementos como inimigos, as gramas ou as mecânicas de puzzle.

	Paredes
	Espaço para andar
	Player
	Blocos com grama





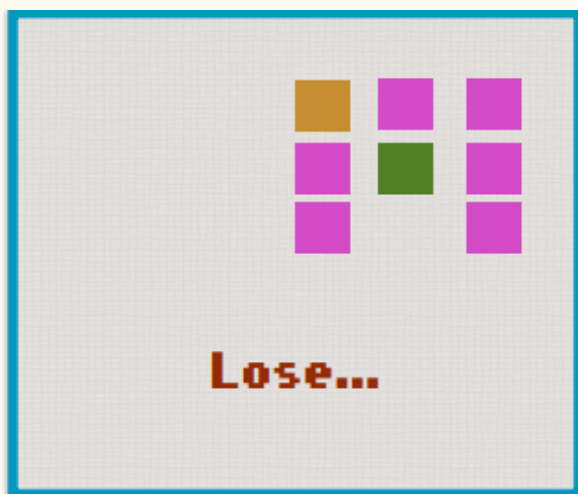
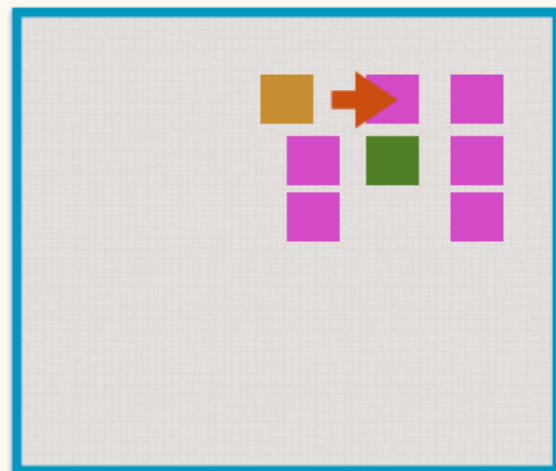
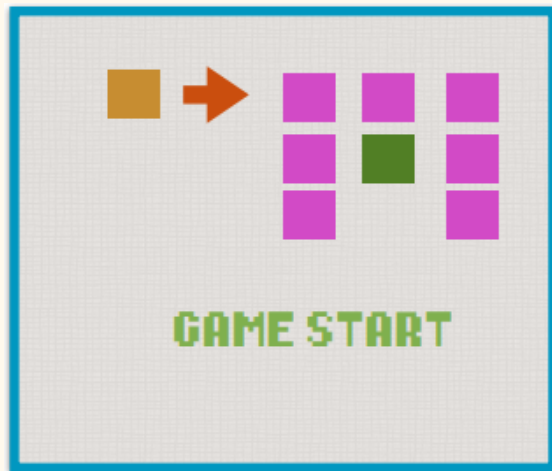
Elemento desafio

Com o passar de fases, novos elementos serão adicionados para dificultar o objetivo do jogador, por exemplo:

Elemento Flores: Evitar passar pelas flores e cortar elas:



Blocos de flores



O jogador perde a fase, aparece uma tela de Perda e o motivo da perda.

Seleção de fases

Existem dois casos que vamos tratar: Caso seja a primeira vez que o jogador inicia o game, nosso jogo terá uma introdução curta para explicar o objetivo principal do game e os controles básicos. Logo após será mostrado uma tela com alguns mundos bloqueados e apenas o primeiro aberto. Caso o jogador tenha resolvido algumas fases e desbloqueado alguns mundos, será exibido como disponível os mundos já liberados.



VII. Controles

a. Principais inputs

Os principais controles serão:

CASO TECLADO:

AWSD ou setas direcionais - movimentação principal pelo menu, movimentação na gameplay;

Enter - Confirmação em menus. Interações para cutscenes (pressionar e segurar pula a cutscene, pressionar e soltar vai para o próximo diálogo)

Escape - Saída entre menus, pause/start na gameplay.

CASO CONTROLE GENÉRICO:

Direcionais ou analógico - movimentação principal pelo menu, movimentação na gameplay;



Fire button 1 (Botão principal: XBOX("A button"), PLAYSTATION("X button"), NINTENDO(" A - left button")) - Confirmação em menus. Interações para cutscenes(pressionar e soltar vai para o próximo diálogo).

Fire button 2 (Botão secundário: XBOX("B button"), PLAYSTATION("O button"), NINTENDO(" B - down button")) - Saída entre menus,

Start Button - pause/start na gameplay, confirmação em menus. Interação para cutscene(pressionar e segurar para pular a cutscene)

b. Direcionais

O personagem andar apenas no eixo X e Y sem permissão para andar na diagonal, o click do botão direcional fará com que o personagem ande 1 bloco na direção desejada, caso houver uma grama ela será cortada.

VIII. Variações de jogo

Não haverá variações drásticas de gameplay, entre um mundo e outro será temáticas diferentes, porém seguindo o mesmo fluxo do game.

Talvez, seja disponibilizada uma versão demonstração futuramente, onde o jogador percorrerá uma fase de maneira infinita e sem progredir, apenas para exibição em Stands.

IX. Definições

a. Estilo artístico:

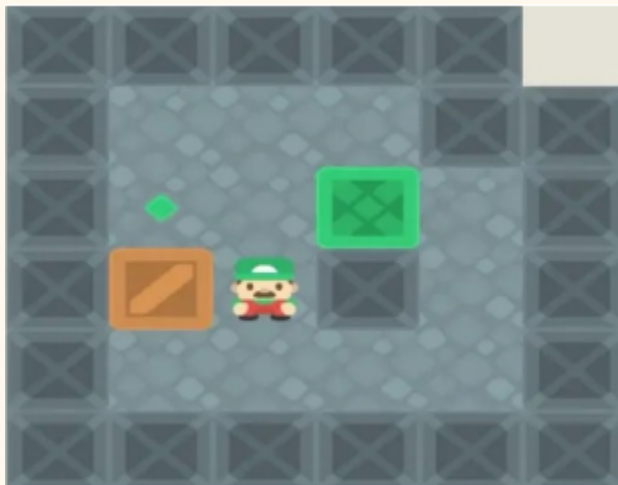
Pixelart: Era 16 bits - Mega drive, Neogeo, Super Nintendo.

b. Músicas e trilha sonora:

Música instrumental sem vocal, com efeito de loop e não necessariamente Chiptune. Músicas mais voltadas para tons mais finos e instrumentos mais infantis: xilofone, violão, violino.

X. Referências

Bomberman(1983): é uma série de jogos de estratégia, inicialmente desenvolvido pela fabricante Hudson Soft. O objetivo principal, em todos os jogos da série, é completar as fases depositando bombas em ordens e lugares estratégicos para destruir obstáculos e inimigos.



Sokoban(1982) (倉庫番 sōkōban, zelador do armazém): é um tipo de jogo de transporte e movimentação de cubos ou engradados em um armazém. O objetivo é pegar e estocar o engradado em determinada posições. O jogo é geralmente apresentado como vídeo game. Sokoban foi criado em 1981 por Hiroyuki Imabayashi, e publicado em 1982 por Thinking Rabbit, uma empresa de software localizada em Takarazuka.

XI. Extras

- a. Definições artísticas
 - > Proporção de imagem

Nosso jogo terá ESTILO 16x16, isso não quer dizer que TODOS os objetos terão que seguir a risco essa definição, mas claro, respeitando a resolução planejada para dar a



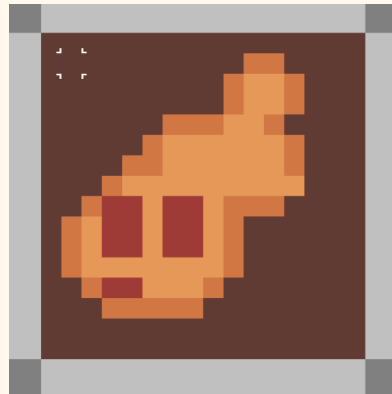
sensação da estilização. Por exemplo, nosso rascunho do player não está 16x16 e sim 16x18. Isso vale para as cutscenes e outros elementos gráficos.

> Escalabilidade para detalhamento da imagem

Pode ser usado uma técnica de escalabilidade da Sprite. CASO precise fazer uma arte de proporção maior, escale sua arte SEMPRE duplicando ela (2 em 2). Assim você pode trabalhar com mais detalhes, mas claro, respeitando os conceitos de arte 16x16.

Por exemplo

Temos essa pixel arte simples 16x16



Problema: Queremos colocar "um pixel" em um espaço impossível, que seria o intermediário entre a imagem 2 e a imagem 3, não é possível já que o pixel é a menor representação de uma imagem:

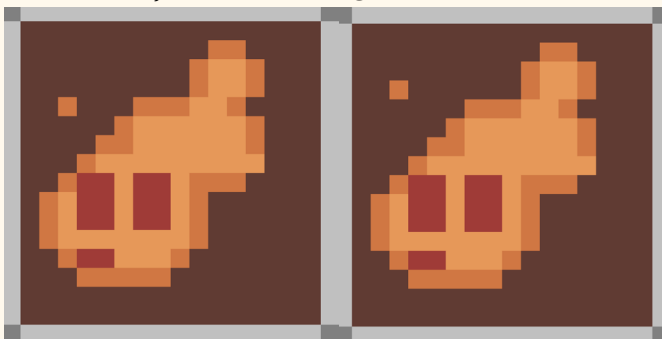


imagem 2

imagem 3

Solução: Escalamos a imagem para a próxima resolução: 32x32, fazemos isso até conseguirmos resolver o problema, neste caso já resolvemos o problema pois nosso pixel ficou representado por 2x2. Então conseguimos colocar entre o intermédio de dois pixels.





Precauções: Perceba que mesmo escalando o item, o visual e resolução continuam os mesmos, isso porque nosso pixel apenas está representado como 2x2. A Partir de agora não podemos trabalhar com o pixel 1x1 como unitário, senão nossa arte ficaria com aspecto de 32x32.

Exemplo de mau uso da escalabilidade.

Perceba que a aparência da nossa imagem ficou com estilo mais próximo de 32x32. Pois suavizamos muito, esse efeito nós não queremos

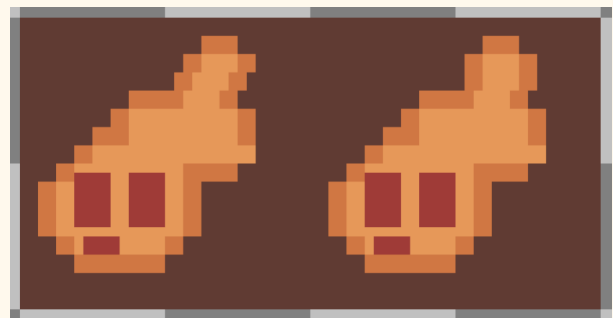


Um bom uso de escalabilidade:

Nesse exemplo temos duas modificações importantes:

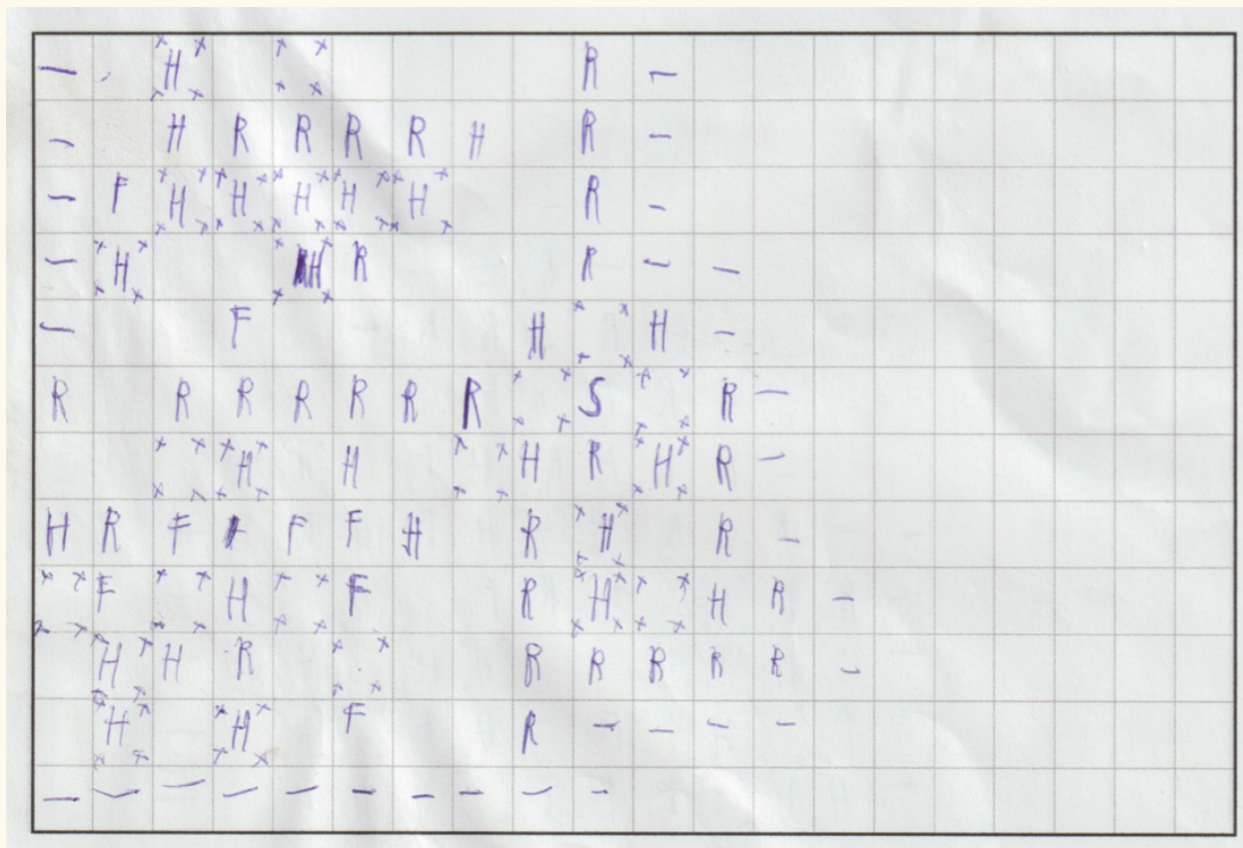
Efeito de "Layer" no rosto: Mesmo que não respeite a unidade, causa impressão de camadas diferentes de pixel art. Criando uma ilusão satisfatória sem redefinir o aspecto 16x16.

Quadros de animação (1-2): Caso precise fazer animações mais fluídas, as transições dessas animações podem quebrar o conceito do pixel unitário ser 2x2.



b. Prototipagem

Algumas fases de testes já foram criadas e validadas à mão. Na próxima imagem,



está representada uma fase de dificuldade média-difícil e foi entregue para alguns estudantes de Game Design resolverem. Foi observado engajamento entre os avaliadores que persistiram em continuar tentando resolver o level. Nessa prototipagem, foi fornecida a informação de cada elemento da cena: S - Onde o jogador começa. R - Rochas. F - Flores que não podem pisar. H - a grama que deve ser cortada. Os blocos com 4 pontas São plataformas falsas (caem depois que o jogador passar pela primeira vez). Os hífen são os limites do mapa.

Observação: As primeiras fases não serão assim, será mais no estilo introdutória para cada mecânica separadamente, esse protótipo de exemplo estará perto de fases mais avançadas.