

Sohlepse

Um Game Design Document

Larissa Goto Sala, Pedro Paulo de Queiroz Bambace, Nathalia Orlandi Borin

November 26, 2018

1 Resumo

Sohlepse é um jogo de puzzle-plataforma que faz uso de reflexões para proporcionar o controle de mais de um personagem com os mesmos comandos.

Isso pode ser feito de forma paralela (os dois personagens se movem ao mesmo tempo) ou sequencial (a mesma sequência de movimentos de um personagem é repetida enquanto você move o outro personagem).

Cada fase apresentará diferentes obstáculos que te obrigará a fazer uso criativo de suas reflexões para superá-los.

2 Jogabilidade

2.1 Ações do jogador

- **Andar:** \rightarrow e \leftarrow ou A e D
- **Pular:** \uparrow ou W
- **Empurrar caixas:** E enquanto anda na direção da caixa
- **Pressionar botões:** permanecer em cima dele
- **Acionar alavancas:** E

2.2 Objetos do jogo

- **Botões:** ativam ou desativam uma quantidade arbitrária de paredes e espinhos divididos em 2 tipos:
 - Os que se mantêm ativos enquanto estão pressionados
 - Os que pressionados uma vez, se mantêm ativos até o fim da fase
- **Alavancas** - Se mantêm ativos até serem acionadas novamente - ativam ou desativam uma quantidade arbitrária de paredes e espinhos
- **Caixas** - Podem ser empurradas para formar escadas ou acionar botões
- **Chão quebrável** - Se quebram quando o jogador passa por ele duas vezes
- **Água** - Jogador pode passar por esse tile normalmente, mas sua movimentação fica mais lenta
- **Lava** - Jogador morre se encostar nesse tile
- **Espetos** - Jogador morre se encostar nesse tile - pode ser desativado por um botão/alavanca
- **MovingEspetos** - Como um tile de espeto, mas que se move

2.3 EndGoal

Espelho que representa o ponto onde o Jogador deve chegar para terminar a fase, ao interagir com ele, a próxima fase começa(dependendo da fase podem haver cutscenes antes da inicialização)

3 Organização das cenas

3.1 Menu Principal

- **Jogar** - Carrega o última fase que o jogador ainda não completou,armazenado na variável global `unlocked_stage`.
- **Seleção de fases** - Carrega o menu de seleção de fases. Suporta batches de 12 fases cada.
- **Opções** - Carrega o menu de opções, que possui controladores de volume (para músicas, efeitos sonoros ou ambos) assim como indicação dos controles (em teclado ou em joypads)
- **Créditos** - Carrega os créditos
- **Sair** - Fecha o jogo.

3.2 Stage Manager

Faz o trabalho de carregar a fase atual nos dois viewports, assim como organiza as telas horizontal ou verticalmente, como definido na variável `MODE`.

1. **Setup** - Configura as câmeras de cada viewport para seguir cada player, de acordo com a variável `PLAYER`, que está em `Stage`. Também limita as câmeras aos retângulos definidos como `Real` e `Mirrored`.
 - (a) **Viewports** - Contidos em containers verticais ou horizontais, o `Setup` define qual será usado.
 - (b) **Stage** - Apresenta as variáveis `ACT`, `PLAYERS`, `MAX_CLONES`, e `MODE` (1 é vertical, 2 é horizontal).

2. Pause Menu

- (a) Voltar ao jogo - Retoma o jogo.
- (b) Menu - Volta ao menu principal
- (c) Recomeçar - Reinicia a fase.
- (d) Opções - Mostra a tela de opções.
- (e) Sair - Fecha o jogo, mas pede confirmação primeiro.

3.3 Stage

Nó raiz de cada fase.

- **TileMap** - Layout da fase.
- **Real e Mirrored** - ReferenceRect, definem os limites da câmera.
- **Endgoal e EndgoalMirror** - Objetivo final de cada fase.
- **Players** - Um Node normal, cujos filhos são instâncias do Player.
- **Reality e Mirror** - dois Nodes normais, cujos filhos são os nós Objects e Hazardz nas fases Real e Espelhada respectivamente
- **Objects** - Um Node normal, cujos filhos são botões, paredes, alavancas, etc.
- **Hazards** - Um Node normal, cujos filhos são limites, firearea, espetos, etc.

3.4 Player

O Player é o personagem humano controlado pelo jogador, que através da movimentação e interação com os objetos da fase, deve chegar no EndGoal, evitando cair e tocar em Hazards

- **InputHandler** - Lida com os inputs do jogador. Tem dois modos de funcionamento - um espera inputs do teclado, enquanto outro lê de um vetor de inputs (usado para as gravações do Ato 3)
- **Recorder** - Responsável pela gravação do movimento do jogador. Para cada frame, grava o "estado" do player. O estado nesse caso consiste de um vetor booleano com quatro entradas, que indicam se os botões \rightarrow , \leftarrow , \uparrow e E , respectivamente, estavam pressionados nessa frame.

Alguns atributos:

- **Terrain e In_terrain** - Escalar que multiplica a velocidade do player (deixa ele mais lento na água, por exemplo) e indica se o player está em algum terreno diferente, respectivamente.
- **Pushing** - Indica se o player está empurrando/puxando uma caixa - diminui a velocidade do player nesse caso.
- **Crushing** - Indica se o player está sendo esmagado - mata o player se ele permanecer nesse estado por um certo período de tempo.

- **Clone e Red** - Indica se o player é uma gravação ou o player real, e se ele tem olhos vermelhos ou não, respectivamente.
- **Dead** - Indica se o player está morto, e nesse caso, ignora quaisquer inputs recebidos (mas ainda aplica força gravitacional no cadáver do player).
- **Carry** - Indica se o player está sendo carregado por uma plataforma - nesse casos a plataforma deve setar sua posição manualmente (a física da engine não funcionava como esperado).

3.5 Botões e paredes

Nesta seção, paredes se referem à Walls, MovingPlatforms e Thorns. E botões se refere à Button, ButtonInst ou Levers.

- 1 parede -> 1 botão:
Coloque a parede num grupo com o nome do botão.
- 1 parede -> 2-3 botões (alternado):
Coloque a parede em 2-3 grupos, cada um com o nome do botão desejado. A parede deve ter activation = 0.
- 1 acionável -> 2-3 ações (cumulativas):
Coloque a parede em 2-3 grupos, cada um com o nome do botão desejado. A parede deve ter activation = 1.
- 2-3 paredes -> 1 botão
Coloque as 2-3 paredes no grupo do botão.
- Parede 1 -> Botão A e B (alternado), Parede 2 -> Botão B
Coloque a Parede 1 nos grupos A e B, e a Parede 2 no grupo B. A parede 1 deve ter activation = 0.
- Parede 1 -> Botão A e B (cumulativo), Parede 2 -> Botão B
Coloque a Parede 1 nos grupos A e B, assim como os próprios botões, e a Parede 2 no grupo B. A parede 1 deve ter activation = 1.
- Parede 1 -> Botão A, B e C (cumulativo), Parede 2 -> Botão B e C (alternado), Parede 3 -> Botão A
Coloque a Parede 1 nos grupos A.B e C. Coloque a Parede 2 nos grupos B e C. Coloque a Parede 3 no grupo A. Coloque os botões A,B e C em seus respectivos grupos. A parede 1 deve ter activation = 1, e a parede 2 deve ter activation = 0.
- Parede 1 -> Botão A, B e C (cumulativo), Parede 2 -> Botão B e C (alternado), Parede 3 -> Botão A
Coloque a Parede 1 nos grupos A.B e C. Coloque a Parede 2 nos grupos

B e C. Coloque a Parede 3 no grupo A. Coloque os botões A,B e C em seus respectivos grupos. As paredes 1 e 2 devem ter `activation = 1`

- **CASO ESPECIAL:**

Para facilitar a organização, em botões do tipo `ButtonInst` que **ativam mais de dois grupos de paredes de forma acumulada**, você pode colocar o nome dos grupos na array `names`, inserir o `ButtonInst` nesses dois grupos, e colocar as paredes acionadas por eles em seu respectivo grupo. Ex: Parede 1 -> Botão A e C (alternada) Parede 2 -> Botão A e B (acumulada) Parede 3 -> Botão B e C (alternada) `Names(A) = ["A", "A2"]`, `Names(B) = ["B", "B2"]` Botão A está nos grupos A e A2, Botão B está nos grupos B e B2, Botão C está no grupo C. Parede 1 está nos grupos A e C, Parede 2 está no grupos A2 e B2, Parede 3 está nos grupos B e C.

3.6 Events e Cutscenes

Nó que permite a automatização da criação de "eventos" - isto é, áreas que trigger algumas linhas de diálogo dentro de uma fase. Seus atributos são:

- **Events** - lista de posições na fase onde ficam as áreas que triggerão os eventos
- **Labels** - lista de textos que serão mostrados
- **Pos** - lista de posições na fase onde as labels devem surgir
- **Areascale** - lista de escalares para as áreas dos events
- **Playanyway** - indica se os diálogos podem ser mostrados fora de ordem. Por padrão, a segunda área na lista de events ativaria o segundo label apenas se o player já tiver passado pela primeira, por exemplo.

No caso de cutscenes, introduzimos os atores da cena como clones e setamos seu vetor de inputs para os movimentos que queremos que eles façam. Isso é feito usando alguns vetores previamente definidos tais como `MOVE_LEFT`, `MOVE_RIGHT`, `JUMP` ou `INTERACT` que consistem apenas na realização desses movimentos atômicos. Construímos a cena com a ajuda de `Áreas2D` que, ao serem cruzadas pelo player, atualiza seu vetor de inputs para a próxima sequência de comandos.

4 Ato I

Duas telas espelhadas, podendo a divisão ser na horizontal ou vertical, dependendo da fase. O movimento do personagem na primeira tela

é refletido de maneira idêntica na segunda tela. Uma das telas pode ser obscurecida e/ou conter partes quebradas/cobertas dependendo da fase. Nesse ato, o principal objetivo é introduzir o jogador às mecânicas do jogo. As fases são idênticas em ambos lados do espelho (salvo possíveis exceções onde o espelho será obscurecido/quebrado).

4.1 Mecânicas

Lista das mecânicas do ato

- Plataformas que empurram caixas que pressionam botões.
- Plataformas que pressionam botões.
- Paredes acionáveis que derrubam caixas e/ou o player.
- Dessincronização usando paredes.

4.2 Fases

- **Fase1** : Tutorial de movimento, interação e pausa
- **Fase2** : Introduz caixas para subir e alavancas
- **Fase3** : Introduz água, lava e espinhos
- **Fase4** : Introduz caixas para apertar botões
- **Fase5** : Introduz espinhos acionáveis e plataformas one-way
- **Fase6** : Introduz falling floor e plataformas que se movem
- **Fase7** : Quebrar falling floors de propósito é uma estratégia válida, introduz alavancas cumulativas
- **Fase8** : Plataformas e caixas: não fazer o óbvio primeiro (tacar no buraco)
- **Fase9** : Tutorial de restart, não fazer o óbvio primeiro (novamente)
- **Fase10** : Introduz mecânica de plataforma que empurra caixa, morte por esmagamento
- **Fase11** : Uso de falling floors, alavancas que ativam mais de uma coisa - patience platforming (pequenas diferenças entre água e lava)
- **Fase12** : Fase final do Ato 1 - plataformas ativando botões, traps com espinhos, uso de dessincronização com paredes acionáveis, falling floor com timing preciso.

4.3 História

Jogador acorda em uma sala fechada, com apenas um espelho, sem saber onde está. Ao interagir com o espelho, é puxado pela própria reflexão para o "mundo espelhado". Percorre o laboratório e vai percebendo sutis mudanças na reflexão das fases. Eventualmente, encontra uma saída, e acaba perdido numa floresta desconhecida. Termina o ato jurando que voltará para sua família.

5 Ato II

Duas telas que representam fases diferentes, podendo a divisão ser na horizontal ou vertical, dependendo da fase. O movimento do personagem na primeira tela é refletido na segunda tela, mas como os cenários são diferentes, pode haver uma dessincronização entre os movimentos. Nesse ato, o mais longo do jogo, o objetivo é brincar com a noção de realidades paralelas, onde você deve planejar bem seus movimentos para controlar dois personagens em cenários distintos com os mesmos inputs. Ao longo das fases, a barreira entre as duas realidades começa a se quebrar, e você pode acabar interagindo com sua "reflexão" de maneiras diferentes.

Lista das mecânicas do ato

- Dessincronização usando paredes, caixas, plataformas e água.
- Fases em que um dos players tem seu movimento restrito ou pode morrer.
- Fases em que um dos players pode restringir o movimento do outro para evitar sua morte.
- Fases em que acontecimentos no lado espelhado podem dar dicas sobre o que fazer no lado real e vice-versa.

5.1 Fases

- **Fase13** : Introduz mecânica de alavancas que ativam mais de uma coisa. O puzzle é descobrir a ordem certa de ativar as alavancas.
- **Fase14** : Clone fica ativando/desativando plataformas para o player - uso da caixa para sincronizar os movimentos sem que o player caia no espinho.
- **Fase15** : Clone fica ativando/desativando plataformas para o player, tomando cuidado para não ser esmagado e/ou desativar o próprio chão.

- **Fase16** : Traps envolvendo espinhos e caixas, depende de um "leap of faith" do player e seu clone, que desativa os espinhos da fase antes do player cair neles.
- **Fase17** : Uso de falling floors no lado espelhado, de forma que os movimentos do clone não pode ser 1:1 os do player. O segredo da fase está na sincronização do movimento dos dois no pulo da terceira plataforma para a quarta (o clone deve aproveitar esse momento para chegar na alavanca e ativá-la). Introduz mecânica de espelho que se move.
- **Fase18** : Fase "Mario Maker" - jogador não pode pular e nem ir muito para os lados, ou o clone morre. A ideia é sincronizar as plataformas que se movem de antemão, e depois só seguir o fluxo.
- **Fase19** : Fase em que o player deve ativar diversos botões ao mesmo tempo com seu clone, precisando usar da dessincronização para isso.
- **Fase20** : Fase em que o player precisa usar um tile bem colocado para dessincronizar seu movimento em relação a uma plataforma que se move.
- **Fase21** : Fase "Mario Maker Invertido" - jogador só pode pular em uma pequena área na fase e seu clone deve ativar e desativar plataformas de forma a levar o espelho até ele. É preciso tomar cuidado para não mandar o espelho pro abismo.
- **Fase22** : Plataformas "troll" e traps com espinhos. Uso de uma parede acionável para dessincronizar os jogadores (callback para **fase12**). Uso de informação no lado espelhado para guiar ações do player no lado real (o movimento da plataforma que carrega o espelho está sincronizado com a plataforma espelhada).
- **Fase23** : Fase de corrida - o player deve desativar as paredes e derrubar as caixas antes que elas sejam levadas por outra plataforma - essa é a única maneira de atravessar a lava. Introduz mecânica de espinho que se move.
- **Fase24** : Fase final do Ato 2 - espinhos que se movem, dessincronização usando paredes. espiral da morte com espinhos, timear bem o pulo dos dois players.

5.2 História

O jogador percebe que sua reflexão agora tem olhos vermelhos. Ao longo das fases, sente-se mal e começa a esquecer acontecimentos importantes em sua própria vida. Cooperar com sua reflexão contra sua vontade, tendo sempre um pouco de desconfiança com relação à esse.

Na fase final, a realidade espelhada e a real se juntam numa só, e o player finalmente chega em um lugar que parece sua casa. Sua reflexão o aconselha a não entrar. O jogador a ignora e entra em sua própria casa.

6 Ato III

Nesse ato, a realidade do espelho se fundiu quase completamente com a realidade, então as fases se passam numa única tela. O diferencial desse ato é o uso de "clones", cujo número pode ser limitado ou não, dependendo da fase, que repetem o que você faz, como numa gravação. A ideia é sincronizar o seu movimento com o dos seus "clones" para conseguir atingir seu objetivo.

Algumas mecânicas do ato

- Fazer uma gravação e ativar/desativar coisas para garantir sua sobrevivência
- Fazer uma gravação e ativar/desativar coisas para matá-la e usar seu cadáver para pisar sobre tiles perigosos.
- Começar uma gravação em spots estratégicos - antes de um botão importante, sobre plataformas que ainda serão ativadas e coisas do tipo.
- Platforming preciso - momentos em que plataformas são ativadas e desativadas sucessivamente, de modo que o player deve ter pressa.

6.1 Fases

- **Fase25** : Callback da primeira fase: mas dessa vez o espelho está no "teto". Introduz mecânica de pular nos clones para atingir seu objetivo.
- **Fase26** : Callback da primeira fase do ato 2: dessa vez quem ativa as plataformas é um clone gravado, e não um clone do espelho.
- **Fase27** : Player tem acesso a apenas um clone e deve ativar todos os botões para abrir a parede que protege o espelho.
- **Fase28** : Introduz mecânica de matar clones para prosseguir - player deve usar uma plataforma com espinhos para atingir seu objetivo, e para isso precisa ficar sobre o cadáver de um clone.
- **Fase29** : Como na fase anterior, temos apenas um clone, que deve percorrer a fase e ativar um botão que abre a parede que protege o espelho. Porém, o sucesso da gravação depende do

player ativando/desativando plataformas e paredes no momento certo.

- **Fase30** : Mostra a importância de começar sua gravação no momento certo - é possível fechar o espelho apertando um botão muito bem posicionado. A gravação deve ativar e desativar espinhos/paredes em um ritmo no qual o jogador possa passar por eles com facilidade.
- **Fase31** : Nesta fase é preciso timear bem suas gravações: um movimento que ativava uma alavanca inicialmente pode ser reutilizado para ativar outra, se você usar bem seus recursos. Introduz mecânica de carregar um clone em sua cabeça.
- **Fase32** : Fase final - corrida contra o tempo. Essa fase apresenta um timer e está repleta de armadilhas. O player deve ter pressa e timear muito bem seus pulos para atingir o final.

6.2 Progresso

- **Ato 1** : 12/12 fases - Completo!
- **Ato 2** : 12/12 fases - Completo!
- **Ato 3** : 8/8 fases - Completo!

6.3 História

O jogador novamente acaba na sala inicial do laboratório. Exasperado, recomeça sua busca por uma saída. Quando finalmente a encontra, seu clone aparece e lhe diz que apenas um deles pode sair daquele lugar vivo, e ativa um mecanismo de auto-destruição do laboratório. O jogador declara que não deixará que o clone roube sua identidade, e começa então uma corrida para a saída. Se o jogador conseguir escapar a tempo, ele chega em casa e o clone acaba morrendo na explosão. Em casa, entretanto, depara-se com uma surpresa - sua esposa tem olhos vermelhos, como o seu clone.