

Aluno: Gustavo Magalhães Moura

Projeto do TRB2B - Jogo base: Worms Armageddon

O jogo Worms Armageddon:

O jogo ao qual pretendo implementar utilizando HTML5 se baseia no jogo [Worms Armageddon](#) (Figura 1).



Figura 1: Cenário e times do jogo Worms Armageddon

O jogo é do gênero estratégia/ação, podendo ser *singleplayer* ou *multiplayer*, no qual cada o jogador controla um time de minhocas, cujo objetivo é derrotar os outros times. Para tal cada time possui um conjunto de recursos que dá habilidades às minhocas (Figura 2). Estes recursos podem ser armas, golpes e ferramentas. Entre as armas destacam-se lança-mísseis, granadas e armas exóticas como a ovelha voadora.



Figura 2: Recursos possíveis de cada time

A jogabilidade do jogo é baseada em turnos onde um time é escolhido a partir de uma sequência pela *engine* do jogo. A cada turno, o jogador só pode utilizar uma minhoca, escolhida pelo time durante um tempo estabelecido para movê-la pelo cenário escolhida pela *engine* do jogo. A minhoca pode andar, pular e utilizar algum recurso disponível durante seu turno.

Cada minhoca começa a rodada com uma quantidade específica de saúde, que é perdida quando é acertada por uma arma ou golpe, está no raio de explosão de alguma arma explosiva ou cai de uma certa altura. A minhoca morre quando sua saúde atinge zero ou quando cai na água.

O jogo inclui uma grande variedade de armas, armas corpo a corpo, projéteis, explosivos, ataques aéreos. Algumas são baseadas na vida real, como taco de *baseball*, pistola, espingarda, bazuca, grana de mão. Outras são únicas, como ovelhas, gambás, e bananas. Cada arma possui um poder de destruição, influência ou não do vento. A direção e sentido é escolhido pelo jogador, bem como a força aplicada no lançamento do projétil. Outros recursos que podem ser utilizados para auxiliar a minhoca a mover pelo mapa, como cordas, pontes e teletransporte.

Todas estas opções de movimentos para os personagens, fazem com que os jogadores trassem uma boa estratégia para vencer, além de ter precisão nos disparos de todas as armas.

Outra característica importante do jogo é que o cenário é pode ser modificado por explosões e ferramentas que o atinjam, fazendo com que os jogadores tenham que se adaptar às novas condições do campo de batalha.

O jogo é atraente, também, por utilizar formas cartunescas nos personagens e cenários, as frases ditas pelas minhocas, pelo menos em português, são divertidas e adaptadas às várias situações do jogo, como por exemplo, ao acertar a primeira minhoca adversária, a frase é “O primeiro de muitos!” ou ao ser acertada por uma minhoca do mesmo time a frase é “Seu mané!”. A voz das minhocas é fina, o que deixa o jogo mais divertido.

Proposta inicial de implementação:

Para tornar a minha implementação do jogo atraente, utilizarei os elementos do jogo original que considero fundamentais, que são: as imagens e sons originais e a variedade de armas, o vento no cenário, que influencia na trajetória de alguns projéteis.

Os pontos que considero desafiadores para implementar é a interação das explosões com o cenário, que pode ser feito utilizando a técnica de *pixel perfect* para renderizar o cenário e personagens e interações entre eles. E por uma busca pela internet, não encontrei uma fonte boa para este tipo de técnica, as informações estão espelhadas.

Por isso, para o protótipo, não utilizarei esta técnica, não haverá a interação das armas com o cenário, utilizarei as técnicas de plataforma abordada em sala de aula para fazer a movimentação dos personagens pelo cenário. Isto será uma melhoria futura.

Outra simplificação é a quantidade de armas, inicialmente implementarei a bazuca e granada de mão, implementando outras conforme os resultados atingidos.

Sendo assim as implementações que proponho a fazer, inicialmente, são:

1. Movimentação dos personagens, incluindo pulo;
2. Inclusão dos sons e frases dos personagens;
3. Implementação das armas bazuca e lança granada, interação com os personagens e vento (caso da bazuca).

Após estas implementações tentarei implementar a técnica de *pixel perfect* para as colisões e interações dos elementos do jogo.

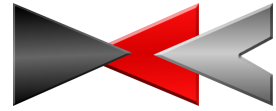
Dificuldades encontradas durante a implementação:

Devido ao conjunto de estados possíveis dos personagens do jogo, fiz um esforço de implementação para manter organizado as mudanças de estado, criando uma máquina de estados para substituir variáveis de controle. Assim conseguiu-se um ganho na depuração dos erros de execução que ocorriam.

Devido à complexidade de ações e elementos, a implementação demorou mais que o previsto. A pesquisa por *sprites* foi demorada, uma vez que os disponíveis na internet não contém todos os movimentos possíveis. Encontrou-se uma ferramenta que extraí os *sprites* originais do jogo, assim poderia ser feito animações próximas ao jogo original. Para armazenar os *sprites* foi utilizada a implementação de biblioteca de imagens aprendida em sala de aula.

Devido ao número alto de *frames* destes *sprites*, as primeiras tentativas de realizar as animações foram frustrantes. Entretanto, nos trabalhos do semestre passado, encontrei uma classe implementada pelo aluno Douglas que facilita a execução de animações. Com isso, foi possível prosseguir na implementação. A dificuldade ficou em encontrar as coordenadas e passo de cada *sprite*, o que demanda um tempo considerável.

Conseguiu-se também, através do jogo base para PC, os sons originais. Assim, seria possível fazer com a experiência do jogador fosse próxima ao jogo original. Para armazenar e executar os sons, utilizou-se a implementação de biblioteca de áudio aprendida em sala de aula, facilitando a execução de áudios durante o jogo.



Implementação atual:

Cenário:

O cenário foi implementado utilizando uma matriz de posições, tal qual aprendido em sala de aula. A altura máxima da primeira coluna é escolhida aleatoriamente e as próximas são escolhidas dentro do intervalo $[-1, 0, 1]$ unidades não ultrapassando os limites mínimos e máximos estabelecidos. A seguir, uma função lê a matriz e desenha os elementos do cenário.

Esta é uma simplificação do jogo, uma vez que utilizar a técnica de *pixel perfect* demandaria muito tempo de aprendizado;

Elementos do jogo:

O personagem do jogo é capaz de se movimentar no mapa e com o áudio original, mas a implementação das colisões com o mapa, salto e queda, ainda estão em implementação.

Até a data limite de entrega, implementarei os elementos faltantes.