**Roteiro de projeto: ArqueoUmbra – GAME**

**Alunos: Eliseu Jayro; Wendel Amérioco.**

**Resumo**

ArqueoUmbra – Game, é um jogo que conta a história de Marco que é um estudante que está concluído o curso de Arqueologia, e nas horas vagas ele ajuda seu pai a produzir explosivos para serem utilizados em minas de extração de ferro e outros metais. Por sua mãe ter adoecido, e ele e toda sua família estar tentando arcar com as despesas dos medicamentos muito caros para o tratamento dela; Marco se dispôs a, momentaneamente, se afastar de sua mãe e ir numa viagem de exploração de artefatos e conhecimento cultural num dos lugares mais remotos do planeta, localizado na Ilha Havelock, Arquipélago Andamão, Índia; ofertado pela faculdade, que o ofereceu uma bolsa que custeava a viagem e dava um auxílio pelo trabalho. Durante a exploração ele conhece um ancião de uma tribo que lhe conta uma história que no vulcão inativo Mauaia há uma caverna que esconde artefatos que juntos revelam o local de um tesouro escondido no fundo da caverna. O problema é que lá também existem criaturas aterrorizantes. Marco, então, se equipa de explosivos para se defender dos monstros e corajosamente vai até essa caverna a procura de novos artefatos e, esperançosamente, em busca de uma joia que a mitologia da tribo conta que trás poderes de cura a quem o possuir; essa é a oportunidade dele usar o poder do artefato para curar sua mãe.

**Mecânica/Descritivo**

O jogo será dividido em três fases, onde o personagem encontrará artefatos em cada uma delas. Para conseguir manter os itens sob a posse do personagem o jogador terá de saber agir com rapidez, pois a cada fase terá monstros para tentar matá-lo. A intenção é que o jogador consiga raciocinar os melhores caminhos a seguir e as melhores ações a tomar visto que um segundo a mais ou a menos fará a diferença entre a vida e a morte de Marco. Por isso, o jogo possuirá um visual que lembre um labirinto, porém com algumas paredes destrutíveis – para que o personagem possa utilizar as bombas – e com visuais mais naturais, afinal a ideia é fazer o jogador sentir dentro das cavernas em Havelock.

Para isso, nós utilizaremos o conceito de top down camera que fará com que o jogador veja o game de cima, assim como nos jogos Grand Theft Auto III e Sensible Soccer que podem ser vistos nas imagens abaixo.



Sensible Soccer

Exemplo de jogo que utiliza câmera Top Down

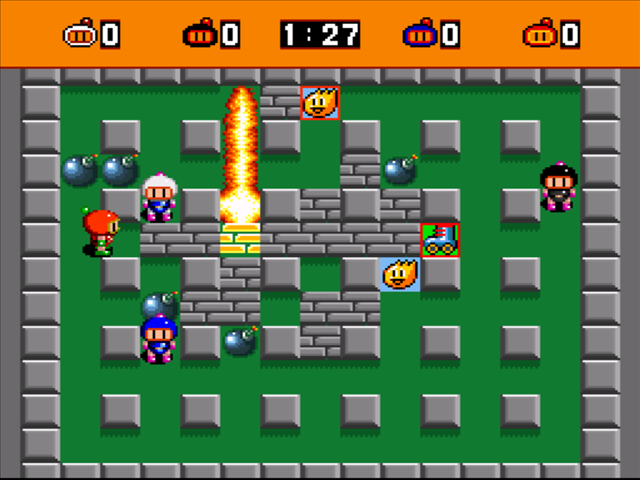


Grand Theft Auto III

Exemplo de jogo que utiliza câmera Top Down

Vale salientar que a câmera será fixa, ou seja, sem movimento dinâmico.

Como já foi dito, as ações serão realizadas dentro de cavernas e galerias rochosas, por isso tomaremos também como exemplo as técnicas aplicadas no jogo Super Bomberman 2 cuja imagem está logo abaixo.



Super Bomberman 2

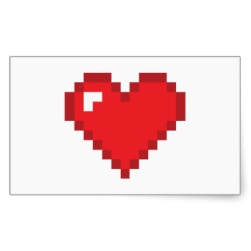
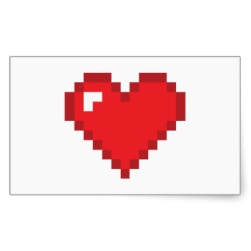
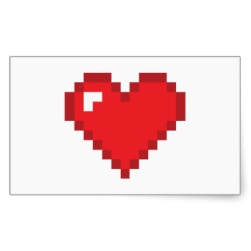
Nesse game o jogador pode destruir algumas paredes e destruir os personagens inimigos usando bombas

Assim, fica claro que utilizaremos uma matriz para implementar o cenário, uma vez que a utilização de câmera Top Down e paredes e galerias formando labirintos facilitam o uso dessa estrutura.

Agora, tratando da parte de movimentação faremos cenários estáticos, ou seja, assim como no Bomberman o personagem poderá se movimentar pelo cenário sem que seja necessário trocá-lo (o cenário) constantemente. Essa troca só ocorrerá quando o personagem se mover para além dos limites da tela de forma a estabelecer uma noção de continuidade para os locais. Os movimentos do personagem e dos elementos de cena serão guiados pelos eixos da matriz, assim sendo o movimento para a direita é feito pelo acréscimo de um passo (considerando o caminhar) da posição atual elemento no eixo x, para esquerda é necessário decrescer um passo no mesmo eixo, já os movimentos verticais são controlados através do eixo y em que o incremento da posição atual resulta num movimento para cima e o decremento movimenta para baixo.

A tela do jogo além mostrar a cena propriamente dita exibirá o “life”, os itens encontrados e a pontuação do jogador. O life será implementado utilizando o tipo abstrato fila representado por uma série de corações de forma que a perda de uma vida resultará na saída do primeiro coração e o ganho na entrada mais um coração ao fim da fila.

Início Fim



\*prox \*prox

Os itens encontrados também serão armazenados numa lista, mais especificamente numa pilha para que seja possível visualizar os elementos começando do mais recentemente encontrado.

A fase será um tipo abstrato de dados (TAD) e de forma genérica possuirá três características básicas, o cenário, os elementos de cena e os sons. A medida que o jogador avança se faz necessário salvar o jogo exigindo então uma lista para guardar o local e a situação em que o jogo parou.

**Objetivo**

O objetivo do projeto é criar um game 2D, de gênero Aventura. O jogo é voltado em um personagem, arqueólogo, que busca uma joia numa caverna. A plataforma do game possui semelhanças com a do jogo Super Bomberman 2 (Nintendo).

**Telas e Interfaces**

O software deve possuir as seguintes telas:

ı **Menu principal:**

Possuirá a opção “1 - Novo Jogo”; “2 - Continuar”, “3 – Intruções” e “4 – Sair”. Onde as opção poderão ser captadas pelo click do mouse e comandos no teclado.

**ı Instruções:**

Manual para o usuário sobre o software com a descrição de comandos; objetivo e dinâmica geral do jogo.

**ı Levels:**

Possuirá a dinâmica do jogo, recebendo comandos e manipulação do personagem e demais elementos do jogo.

* Level 1:

Iniciará com o personagem no canto superior esquerdo da tela, e a intenção é encontrar a passagem para o próximo nível.

*- Recurso extra:*

* Level 2:

Terá as mesmas características do level anterior, mas os objetos em cena serão modificados, e possuirá um nível de dificuldade maior que o anterior.

* Level 3:

Para finalizar o jogo, o level 3 é o mais difícil de todos os outros, e terá as mesmas características dos anteriores, mas com um número maior de adversários e, objetos, bônus ou não, presentes no jogo.

**Possíveis recursos extras**

São exemplos de recursos extras que podem ser implementados:

ı Poderão existir prêmios bônus, dependendo do lugar onde o personagem passar;

ı Sons de fundo do jogo Implementados em uma Lista Circular.

ı Implementar mais de um Level (a intenção é criar três levels para o jogo).

**Desenvolvimento**

No desenvolvimento do projeto devem ser utilizados os diversos conceitos de Estruturas de Dados e Linguagem de Programação vistos em sala de aula, tais como:

* Existirá um TAD principal, chamada GameObject, onde os demais TADs (Personagem, GameobjectList, e outros) herdarão as características dele.
* O Life do personagem será uma sequência de corações (vistas em tela), que será implementado por fila;
* Os prêmios do jogo serão implementados em um TAD;
* O usuário poderá salvar o jogo, e para isso será implementado um TAD com os dados atuais do personagem, pontos e level atual;
* A TAD Personagem possuirá, basicamente: Vida; Pontuação; Itens encontrados; Fase do jogo;
* Cada fase também será uma TAD e terá as seguintes características: Cenário; Objetos (elementos da cena); Som;

O TAD GameObject possui as seguintes características:

#ifndef \_GAMEOBJECT\_H\_

#define \_GAMEOBJECT\_H\_

class GameObject {

public:

std::string nome;

float posicao\_x, posicao\_y;

float velocidade\_x, velocidade\_y;

float atrito;

enum Direcao {CIMA, BAIXO, DIREITA, ESQUERDA};

Direcao direcao;

GameObject();

void Start(); // RODA SOMENTE UMA VEZ QUANDO O OBJETO É CRIADO

virtual void Update();

virtual void Render();

virtual int mouseDown\_Move(float x, float y, int comportamento

};

#endif

// Start -> RODA SOMENTE UMA VEZ QUANDO O OBJETO É CRIADO

// Update -> RODA A CADA FRAME E DEVE CONTER AS MUDANCAS FISICAS DE MOVIMENTO

// Render -> TAMBÉM RODA A CADA FRAME E DEVE CONTER AS ROTINAS DE DESENHO DO OBJETO NA TELA No GameObject Render() não faz nada, deixando esse comportamento para suas subclasses

// MouseDown\_Move -> VERIFICA SE O CLIQUE DO MOUSE FOI SOBRE UM GAMEOBJECT E TAMBEM PODE VERIFICAR SE O CURSOR ESTA SOBRE UM GAMEOBJECT DEPENDENDO DO COMPORTAMENTO ENVIADO, 1 - MouseDown(clique) / 2- MouseMove