UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

Relatório de Laboratório de Computação I (AA783)

POR

Gabriel Augusto Lourenço da Conceição,
Célio dos Santos
e
Francisco Sancas

Ao curso de Ciência da Computação

Nova Iguaçu, 17 de fevereiro de 2017

O presente jogo foi requisito ao Laboratório de Computação I(AA783), sendo por sua vez um projeto complementar do conteúdo da disciplina de Computação I (IM406), lecionada como regra nos períodos primeiros do curso de Ciência da Computação, na Universidade Federal Rural do Rio de Janeiro, que no presente semestre segue pela orientação do professor Leandro Alvim.

Como é aplicado nos primódios do curso, esse é para cada aluno verdadeiro desafio à aplicação de seus conhecimentos básicos sobre programação, pseudo código, algoritmos e linguagens de programação.

Endereço de postagem do vídeo do Jogo: https://youtu.be/n8UEOSaem6E
Endereço GitHub: https://github.com/dart-sit/AA_Comp_I

Projeto

O projeto foi selecionado dentre as opções de jogos de plataforma, com o envolvimento de lutas marciais com mais de um personagem e temática com fim lúdico, para entretenimento.

Além de possuir um menu para escolha de situações anteriores ao jogo, o mesmo exibe escolha para o que se pode fazer depois.

Há uma inserção maciça de imagens dentro de retângulos abstraídos no código fonte do jogo , para que por meio dos mesmos seja possível a alocação, movimentação e troca dessas imagens, que dão sensação de inteligência ao projeto como um todo. As imagens são alocadas conforme os comandos de chaves específicas do teclado e a janela (espaço de jogo) pode ser fechado através do mouse (rato) e/ou chave “esc” do teclado.

Temática

Dentro do projeto são ativadas funções que acionam três tipos de som(chute, soco e música de fundo), o que causa os efeitos de sonoplastia. As personagens são parte de uma sério de TV e internet, intitulada: **Dragon Ball Super**. Claro que o projeto deixa margem para que outros tipos de temáticas possam também ser utilizados dentro do mesmo código fonte, com pequenas adaptações. São duas as personagens (Goku e Beerus Sama), e o código fonte do jogo foi totalmente elaborado em 2D, o que não desmerece sua funcionalidade.

As linguagens de programação utilisadas não foram de uso livre, mas de escolha do Orientador da disciplina. Fomos liberados para utilizar a linguagem SDL (Simple DirectMedia Layer), cuja utilização é de excelência em anexo à linguagem de programação C e suas derivadas.

SDL possui bibliotecas para funções específicas e com poucas linhas de código, o que dinamizou a programação tanto no tamanho do código fonte quanto no tempo de elaboração do projeto.

Em suma foram inclusos o uso da linguagem SDL, da linguagem C/C++ e compilação pelo compilador G++, que suposta compilação de códigos em C e C++.

O cabeçalho recebeu a chamada de quantro bibliotecas de C (stdio.h, stdlib.h, time.h e string), quatro de SDL (SDL.h, SDL_image.h, SDL_mixer.h e SDL_ttf.h) e cinco constantes distribuídas pelas funções existentes.

O código recebeu a declaração de cinco tipos (um typedef e quatro structs), para que a linguagem C pudesse aceitar operações booleanas e simplificação nas chamadas de alguns elementos, que após a resolução poderam ser chamados de maneira mais simples e objetiva.

Para declaração de variáveis nos valemos de SDL_Surface*, para declaração de variáveis inteiras, floats e de caracteres. As variáveis de SDL_Surface* recebem IMG_Load como passagem de parâmetro, para exibição de imagens, enquanto que a variável responsável por exibir a janela principal recebe SDL_SetVideoMode como passagem de parâmetro. SDL_WM_SetCaption escreve mensagem na barra da tela.; o comando SDL_SetVideoMode cria e modela a janela principal.

SDL_PollEvent(&event) fica dentro do **while** principal e comanda a ocorrência de todos os eventos dentro do loop de cada função que possui um **while**.

O comando “event.type” aguarda eventos externos (seu parâmetro para pressionamento de teclas: SDL_KEYDOWN), provocados pelo usuário; event.key.keysym.sym verifica qual tecla foi apertada. Comando que reconhece a tecla de direção para baixo pressionada: SDLK_DOWN; Comando que reconhece a tecla de direção para cima pressionada: SDLK_UP; Comando que reconhece a tecla de direção para direita pressionada: SDLK_LEFT; Comando que reconhece a tecla de direção para baixo pressionada: SDLK_RIGHT ou SDLK_RETURN.

O comando SDL_BlitSurface envia, blita, a imagem em seu destino; SDL_FreeSurface(superficie) libera a superfície que foi utilizada, permitindo que outra assuma seu lugar; IMG_Quit() fecha a imagem que foi exibida, fazendo uma limpeza da tela e/ou permitindo o fechamento comum da SDL, os comandos desse parágrafos geralmente são usados ao final de sua respectivas funções; SDL_Quit() fecha o SDL, pode ser usado também o comando SDLK_ESCAPE.

Retângulos, sons e efeitos

Para declaração de variáveis nos valemos de Mix_Chunk*, para declaração de variáveis de efeitos sonoros, que diferem do som provocado por música de fundo, que “pertence” ao comando Mix_Music*; Mix_OpenAudio é similar ao SDL_SetVideoMode, mas é para criação e modelagem dos sons. Mix_LoadMUS carrega a música de fundo; Mix_LoadWAV carrega os efeitos sonoros com a extensão .WAV; já o comando Mix_PlayMusic(music, -1) determina quantas vezes a música será tocada. Mix_FreeChunk(effect) fecha a ação do efeito, também usado no final de suas respectivas funções; Mix_PlayChannel(-1, effect, 0) executa o efeito sonoro e determina a quantidades de vezes de sua execução.

SDL_GetTicks() comanda a execução de contagem por meio do cronômetro. TTF_Font *fontAplic; fontAplic = TTF_OpenFont("fontes/FineCollege.ttf", 26) determina tamanho, cor e tipo de fonte; SDL_Color colorChrono = {255, 255, 255} abre a paleta de cores do SDL; SDL_Event event abre a ocorrência de eventos; SDL_Rect posicao cria retângulos como ponteiros; TTF_Init() inicia o controle de fontes; TTF_Font* declara variável para uso de fontes; Uint8* teclas regulamenta o uso internacional de símbolos nas chaves do teclado; Uint32 red, green, blue determina o to dos pixels da tabela RGB;

Referências

http://lazyfoo.net/SDL_tutorials/

http://equipe.nce.ufrj.br/adriano/c/apostila/sdl/

https://www.youtube.com/watch?v=7xFaWRzWqr0&list=PL894088275284BDEA

https://www.youtube.com/watch?v=94pPyuS1E_M