MANUAL DE UTILIZADOR

Trabalho realizado por:

Luís Monteiro nº 70742

José Sampaio nº 70647

Introdução:

Este projeto foi desenvolvido no âmbito da componente pratica da Unidade Curricular de Arquitetura de Computadores, onde nos foi dado como objetivo desenvolver uma aplicação na linguagem *Assembly* no emulador x8086.

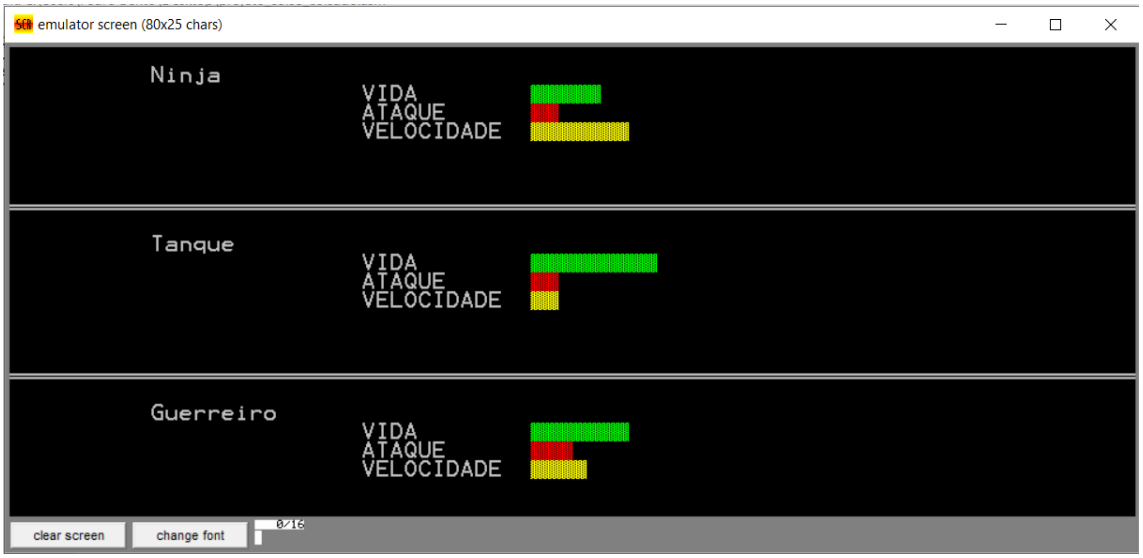
O grupo decidiu desenvolver um jogo no estilo Combat-RPG.

Como Jogar:

O jogo apenas pode ser jogado com duas pessoas, cada uma escolhe uma classe com estatísticas diferentes sendo elas, vida, ataque e velocidade. Cada partida tem uma duração média de 1 a 2 minutos.

TELAS INICIAIS:

ESCOLHA DE CLASSES:

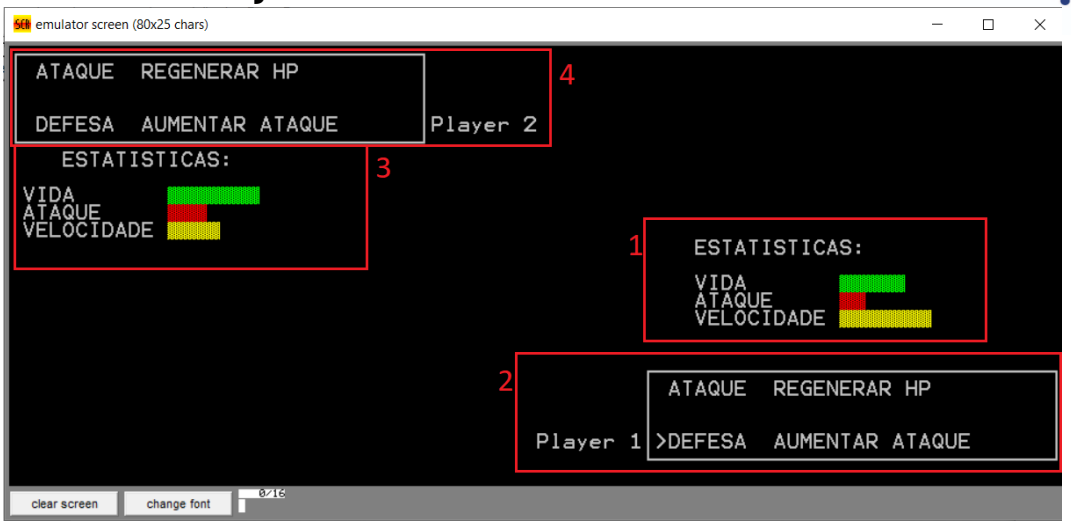


Esta tela vai aparecer para ambos os jogadores, onde vão ter de escolher que classe querem levar para o jogo. De forma a o jogo estar equilibrado, todas as classes contem pontos fortes e fracos:

- **NINJA:** Vida baixa, ataque baixo, velocidade bastante alta, o que permite ser sempre o primeiro no turno.
- **TANQUE:** Vida no máximo, ataque e velocidade baixo, sempre o último a atacar, mas consegue sempre aguentar mais dano.
- **GUERREIRO:** Vida, Velocidade e Ataque equilibrados.

Para interagir com o menu, deve de se utilizar as setas do teclado e depois clicar na tecla ENTER.

INTERFACE DO JOGO:



Esta tela consiste no jogo em si:

1. Estatísticas do jogador 1 (Player 1);
2. Lista de movimentos disponíveis ao jogador 1 (Player 1);
3. Estatísticas do jogador 2 (Player 2);
4. Lista de movimentos disponíveis ao jogador 2(Player 2);

Movimentos:

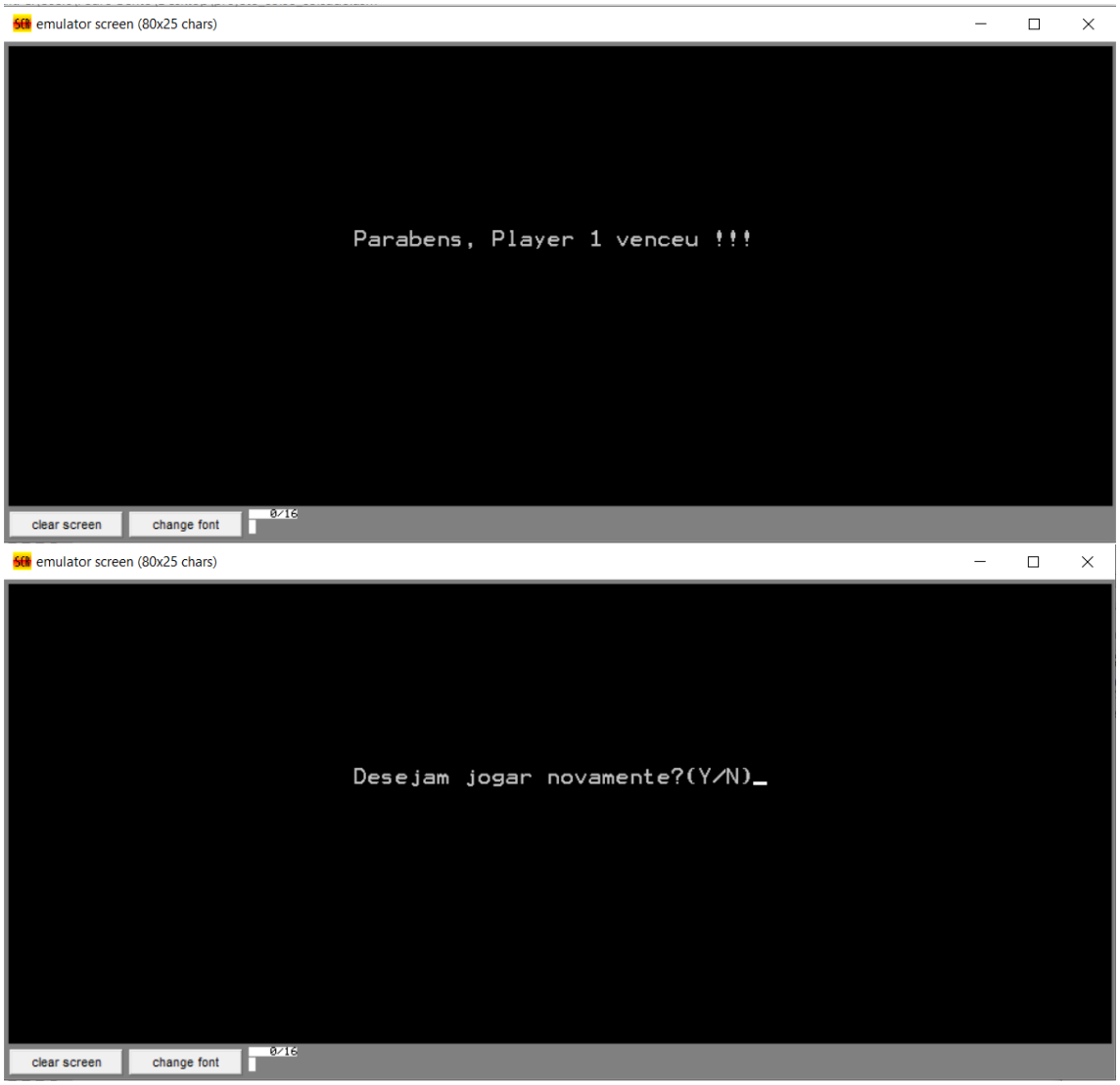
Os movimentos são os mesmos independentemente da classe ou do jogador.

- **ATAQUE:** Retira vida ao adversário, a quantidade de vida retirada depende do valor da sua própria estatística “ATAQUE”;
- **DEFESA:** Bloqueia o próximo ataque do adversário;
- **AUMENTAR ATAQUE:** Aumenta em +1, o valor do ataque;
- **REGENERAR HP:** Aumenta em +1, a vida;

O JOGO:

Cada partida vai decorrer por turnos, onde a classe mais rápida vai atacar primeiro, ganha quem conseguir colocar a vida do adversário a 0.

FIM DO JOGO:



Quando surgem estas duas telas significa que um dos jogadores venceu a partida, após isso o programa pergunta se pretendem jogar novamente. Após isso devem premir Y – sim, ou N – não.

Código:

O código final contém 2282 linhas, mas muitas são espaços em branco e repetição de código. Nesta secção do relatório é explicado como tudo funciona.

```
include emu8086.inc

JMP frente
      hp_1 db 10
      hp_2 db 10
      atk_1 db 10
      atk_2 db 10
      velocidade_1 db 10
      velocidade_2 db 10
      class_id db 1
      DIRECAO db 0
      ESCOLHA_PLAYER db 0
      TURN db 0
      DEFESA_P1 db 0
      DEFESA_P2 db 0

      DIREITA   EQU 4DH
      BAIXO     EQU 50H
      ESQUERDA  EQU 4BH
      CIMA      EQU 48H
      ENTER     EQU 1CH

frente:
```

Começamos por utilizar o include emu8086.inc para ter acesso aos comandos GOTOXY, PUTC e PRINT.

Após isso surgem as variaveis criadas e os DEFINES, as primeiras 6 variaveis são das estatisticas dos jogadores, as outras são variaveis que utilizamos para a escolhas das classes, dos movimentos e na DEFESA que teve que ter um cuidado em especial. As ultimas 5 linhas deste printscreen são os DEFINES para facilitar o uso do teclado durante o jogo.

```
GOTOXY 35,12
PRINT "BEM VINDO          "                    ;
GOTOXY 35,12
PRINT "          "

GOTOXY 25,12
PRINT "Player 1 escolha uma classe:          "
GOTOXY 25,12
PRINT "                                      "
```

Aqui utilizamos o comando PRINT em vez do MOV AH, 9 para dar um efeito melhor enquando se joga, mais a frente utilizamos o MOV AH, 9 pois escreve a string no ecra ao mesmo tempo enquando o comando PRINT escreve careter a careter, o que faz com que noutro caso demoraria muito a executar e a jogar.

```
gotoxy 10,1
MOV AH, 9
LEA DX, class_ninja
INT 21H

gotoxy 25, 2
MOV AH, 9
LEA DX, vida
INT 21h

gotoxy 25, 3
MOV AH, 9
LEA DX, ataque
INT 21h

gotoxy 25, 4
MOV AH, 9
LEA DX, velocidade
INT 21h
```

Isto serve apenas para escrever texto.

```
GOTOXY 37,2
MOV AL, 20h
MOV BH, 0
MOV BL, 0000_1010b
MOV CX, 10
MOV AH, 9
INT 10h
MOV AH, 9
LEA DX, barra_5
INT 21h
GOTOXY 37,2
MOV AL, 20h
MOV BH, 0
MOV BL, 0000_1111b
MOV CX, 10
MOV AH, 9
```

Nesta porção utilizamos a interrupçao 10h para mudar a cor das barras de vida, ataque e velocidade que vão ser utilizadas, após isso são colocadas novamente a branco. Estas sequencias são utilizadas para apresentar no ecra as 3 classes disponiveis aos jogadores.

```
gotoxy 0,8
MOV AH, 9
LEA DX, linha_meio_grande
INT 21H

gotoxy 0,17
MOV AH, 9
LEA DX, linha_meio_grande
INT 21H
```

Aqui são desenhadas duas linhas para dividir a consola em 3 e preparar para apresentar as classes.

```
ESCOLHA_CLASSE:
    MOV DIRECAO, 0
MOV_NINJA:
    CMP DIRECAO, 1
    JE ELIM1_TANK
    CMP DIRECAO, 2
    JE ELIM1_WARRIOR
    JMP ELIM1_FIM
ELIM1_TANK:
    GOTOXY 9, 10
    PUTC ' '
    JMP ELIM1_FIM
ELIM1_WARRIOR:
    GOTOXY 9, 19
    PUTC ' '
    JMP ELIM1_FIM
ELIM1_FIM:
    MOV DIRECAO, 0
    GOTOXY 9, 1
    PUTC '>'

    MOV AH, 00H
    INT 16H
    CMP AH, ENTER
    JE FIM_NINJA
    CMP AH, BAIXO
    JE MOV_TANK
    CMP AH, CIMA
    JE MOV_WARRIOR
```

Esta porção serve para colocar “>” no lugar da classe ninja e um espaço nas outras classes. Após isso pede ao utilizador para ou utilizar o ENTER ou as SETAS, o ENTER faz com que esta parte acabe e seja o jogador dois a escolher a sua classe, e as setas mudam de escolha. Este codigo repete-se nas 3 classes.

Da linha 280 à linha 435 é a porçao do código onde se da o uso das setas para a escolhes as classes para o jogador 1.

```
FIM_ESCOLHA_PLAYER1:

CALL CLEAR_SCREEN
CMP class_id, 1
JE class_change_ninja

CMP class_id, 2
JE class_change_tank

CMP class_id, 3
JE class_change_warrior

class_change_ninja:
mov hp_1, 5
mov atk_1, 2
mov velocidade_1, 7
jmp player2_class

class_change_tank:
mov hp_1, 9
mov atk_1, 2
mov velocidade_1, 2
jmp player2_class

class_change_warrior:
mov hp_1, 7
mov atk_1, 3
mov velocidade_1, 4
jmp player2_class
```

Aqui os valores da classe escolhida são atribuidas as variaveis do jogador 1.

Todo o codigo ate agora é repetido para o jogador 2, ou seja so se começa a jogar apartir da linha 735.

```
GOTOXY 63,13
MOV AL, 20h
MOV BH, 0
MOV BL, 0000_1010b
MOV CX, 10
MOV AH, 9
INT 10h
```

Este bloco de código já foi utilizado e aqui é reutilizado para mudar a cor das barras que vão ser colocadas no ecra de jogo. Cada bloco muda 10 lugares da consola de cor.

Da linha 808 até 931, o codigo baseia-se na interface e em utilização de comandos já falados.

```
MOV BH, velocidade_2
CMP velocidade_1, BH
JGE TURN_GREATER
JMP TURN_LOWER
```

Compara as velocidades para ver quem vai poder utilizar os movimentos primeiro.

```
CMP hp_1, 10
JE printHP_1_10

CMP hp_1, 9
JE printHP_1_9

CMP hp_1, 8
JE printHP_1_8
```

Esta parte do codigo repete-se em todas as estatisticas e tem como objetivo verificar qual é o valor da vida do jogador e apresentar no ecra a barra em questão. Este codigo vem depois com outra parte em anexo que é a barra em questão, e repete-se para todas as estatisticas e para todos os jogadores.

Após isso surge outra porção de codigo igual à já utilizada para escolher as classes com setas e ENTER. Isso que gera pequenos jumps para realizar os movimentos.

No final há uma verificação da vida dos jogadores e se algum estiver a zero o jogo termina e surge a tela de termino de jogo, onde existe a opção de sair ou repetir.