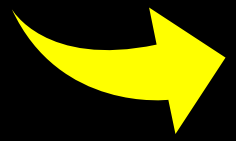
Além do matagal

Nossa ideia

- A nossa ideia foi desenvolver um jogo arcade de sobrevivência, onde o protagonista foi um lenhador/minerador ilegal. Nesse jogo, você deve juntar recursos e se preparar durante o dia, para sobreviver durante a noite.

Requisito 1: Structs

- Como sugerido na proposta do trabalho, utilizamos os structs para definir nossas entidades como Item, Player, Inimigo e etc.



```c
typedef struct Item
{

  int id;
  int type[2];          /*respectivamente, o tipo  de item e o quanto ele recupera */
  char name[25];
  char description[200];
  float chance;
  int damage;
  int throwable[2];       /*respectivamente, jogavel sim ou não. o quanto de dano causa */
  int craftable[2];       /*respectivamente, craftavel sim ou não, id do craft */
  int durability[2];      /*Se tem durabilidade sim ou não, quanto de durabilidade tem (max 10) */

} Item;

typedef struct Enemy
{

  char name[25];
  int status[2]; /*respectivamente, vida e força*/
  int max_life;
  int chance;         /*chance de aparecer */
  Item in_hand;
  int body;
  char parts[4][25];

  int first_encounter[2];   /*respectivamente, se já encontrou ou não e quanto de sanidade irá perder no primeiro encontro */

} Enemy;
```

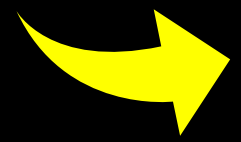
Requisito 2: Passagem por referência

- Aqui escolhemos as funções mais fáceis de visualizar. Como pode ser notado, eu preciso acessar a estrutura e não uma cópia.

```c
/*função que dropa um item da mochila do jogador (talvez retornar o item)*/
int drop_item_from_bag (int index, Player * player, Item nada)
{

  if (index < 0 || index > 4) {
    print("É, não vai funcionar.");
    return -1;
  }

  if (player->bag[index].id == nada.id)
  {
    return -1;
  }

  player->bag[index] = nada;
  return 1;
}

/*função que dropa um item da mão do jogador*/
int drop_item_from_hand (Player * player, Item nada)
{

  if (player->in_hand.id == nada.id)
  {

    return -1;
  }

  player->in_hand = nada;
  return 1;

}
```

Requisito 3: Alocação dinâmica

- Durante o jogo, pensamos que seria uma boa ideia dar a capacidade do jogador aumentar o tamanho do baú, para o mesmo comportar mais itens, por isso decidimos utilizar a alocação dinâmica.



```
int a, n;
    n = 25;
    shack.chest = malloc(n * sizeof(int));
    for (a = 0; a < n; a++)
    {
        shack.chest[a] = nada;
    }
```

Requisito 4: Manipulação de vetores

- Neste primeiro trecho, podemos ver que no vetor status, cada espaço representa um atributo a ser modificado durante o jogo

```c
typedef struct Player
{

  char name[25];
  int status[6];        /*São respectivamente vida, força, energia, sanidade, notoriedade e fome */
  int max_life, max_hunger, max_energy;
  Item in_hand;
  Item bag[6];

} Player;
```

- Já neste segundo trecho, podemos ver isso na pratica, pois o status interage diretamente com o tipo do item, assim determinando o que ele recupera e o quanto ele recupera.

```c
/*Função para usar um item e recuperar algum status*/
int use_item(int index, Player* player, Item nada) {
    if (index < 0 || index > 5) {
        print("Eu não sei o que eu estou tentando fazer, mas provavelmente não vai funcionar...");
        return -1;
    }

    if (player->bag[index].type[0] == -1) {
        print("Não vai dar pra usar isso.");
        return -1;
    }

    Item item = player->bag[index];
    player->status[item.type[0]] += item.type[1];
    drop_item_from_bag(index, player, nada);
    balance(player);
    clear();
    printf("\nAcho que %s vai prestar por agora.\n", item.name);
    getchar();
    getchar();
    clear();
    return 1;
}
```

Requisito 5: Salvar e carregar jogo

- Aqui, utilizamos um arquivo .txt para salvar a quantidade de dias sobrevividos pelo jogador na sua última jogatina.

```c
/*tela de game over que salva no arquivo a quantidade de dias sobrevividos*/
void game_over(Player* player) {
    if (player->status[0] <= 0) {
        FILE* f = fopen("C:\\Users\\Win\\CLionProjects\\demo_matagal\\score.txt", "w");
        if (f == NULL) {
            printf("Erro ao abrir o arquivo\n");
            exit(1);
        }

        int resultado = fprintf(f, "%d \n", player->days);
        if (resultado < 0) {
            printf("Erro ao escrever no arquivo!\n");
            exit(1);  // Encerra o programa em caso de erro
        }

        fclose(f);
        print("Acabou, você escuta seu oração parar aos poucos enquanto sua visão fica turva, mas...\n");
        exit(1);
    }
}

/*Printa o ultimo score do jogador*/
void score() {
    FILE *f;
    char linha[1024];

    f = fopen("C:\\Users\\Win\\CLionProjects\\demo_matagal\\score.txt", "r");
    if (f == NULL) {
        printf("Erro ao abrir o arquivo.\n");
        exit(1);
    }

    while (fgets(linha, sizeof(linha), fp) != NULL) {
        printf("O fortuna do seu ultimo ciclo durou %s dias", linha);
    }

    fclose(fp);
}
```

Créditos

- Link para as imagens usadas:
  https://www.reddit.com/r/Darkwood/comments/nwbwqw/two_wallpaper_by_me/

  https://wallpapercave.com/darkwood-wallpapers

Trabalho feito por: Artur M. Santana e João Costa.