

RPG

Universidade Federal Fluminense - PURO
 Monitor: José Victor Dias Pereira
 Professores: Alessandro Copetti e Patrick Moratori
 Programação de computadores II

O que é?

- ▶ Role Playing Game (Jogo de Interpretação de papéis).
- ▶ Jogadores são personagens da história.
- ▶ O personagem se desenvolve no decorrer da história (Ex: Sobe de nível, aumenta sua força, ganha reputação).

O que sempre tem?

- ▶ Um enredo!
- ▶ Presença de um narrador ou mestre. Neste caso, será o programa.
- ▶ O personagem deve ter poder de decisão a cada "etapa".
- ▶ Fator sorte presente (Ex: batalhas, sorteio de perguntas, sorteio de estória).

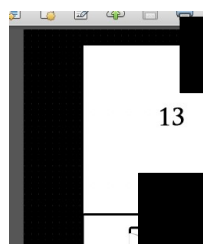


Cuidados:

- ▶ O enredo pode evitar muitos problemas na hora de programar!
- ▶ Pensem no RPG a ser desenvolvido mais parecido com um livro-jogo do que com um RPG online famoso. Motivo: o mapa e as interações!
- ▶ Em um RPG online, o usuário interage com outros usuários. No trabalho, não!
- ▶ Em um RPG online, o usuário tem um mapa muito maior. Aqui, o ideal é que o mapa seja bem "fechado".

Exemplo de mapa

O usuário deve poder escolher o caminho que ele quer seguir.



Várias salas, cada sala deve ter sua interação com o usuário prevista!

Critérios de avaliação do programa de RPG

1. Criatividade da estória.
2. Aderência aos princípios do RPG.
3. Complexidade do programa.
4. **Implementação dos requisitos.**

Requisitos: 1 - Incluir um componente aleatório.

Exemplo:

- ▶ Texto/resposta que aparece para o jogador.
- ▶ O número de personagens que o personagem principal encontrará em determinado momento.
- ▶ A força inicial do personagem principal.

Requisitos: 1 - Incluir um componente aleatório.

Bibliotecas para utilizar números aleatórios:

▶ <stdlib.h> e <time.h>

Funções nessas bibliotecas:

- ▶ srand() e time()
- ▶ rand()

Requisitos: 1 - Incluir um componente aleatório.

```
#include <stdlib.h>
#include <time.h>
```

Exemplo:

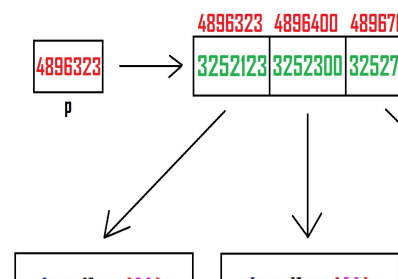
```
int main()
{
    int n, m;

    srand(time(NULL));

    n = rand() % 10; //Valor
    m = (rand() % 10) + 10; //Valor
```

Requisitos: 2 - Usar um vetor de ponteiros para estruturas

O vetor de ponteiros (para estruturas) irá guardar os dados dos personagens do jogo!



Requisitos: 2 - Usar um vetor de ponteiros para estruturas

```
typedef struct personagem
```

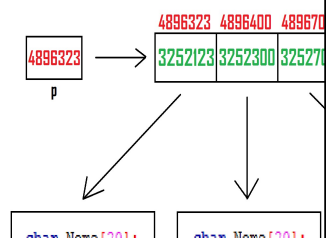
```
{
    char Nome[20];
    int Level;
    float Forca;
}
```

```
Personagem;
```

```
int main()
```

```
{
    Personagem** p;
```

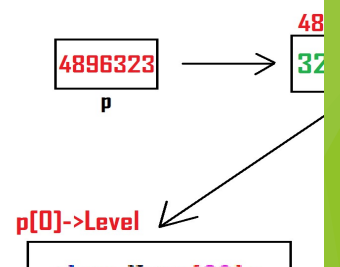
```
p = (Personagem**)malloc(3*sizeof
```



Requisitos: 2 - Usar um vetor de ponteiros para estruturas

Exemplo:

```
p[0]->Level
p[1]->Level
```



Requisitos: 3 - Criar funções parametrizadas com poucas linhas de código, com tarefas específicas e com alocação dinâmica.

Exemplo:

```
int* CriarVetor(int Quantidade)
{
    int* v;

    v = (int*)malloc(Quantidade*sizeof(int));
    if(v == NULL)
    {
        return NULL;
    }
}
```

Requisitos: 3 - Criar funções parametrizadas com poucas linhas de código, com tarefas específicas e com alocação dinâmica.

Exemplo:

```
void PreencherVetor(int* Vetor, int Quantidade)
{
    int i;

    for(i=0; i<Quantidade; i++)
    {
        printf("Digite um valor para a posicao %d: ", i);
    }
}
```

Requisitos: 4 - Usar no mínimo 5 funções das bibliotecas do C ainda não utilizadas nas listas de exercícios

Exemplo:

- ▶ sprintf()
- ▶ strcat()
- ▶ strrchr()
- ▶ toupper()
- ▶ tolower()
- ▶ etc...

Requisitos: 5 - Fazer uso intensivo de arquivos

Exemplo:

- ▶ Salvar o progresso do personagem;
- ▶ Exibir algum diálogo no jogo;

Requisitos: 5 - Fazer uso intensivo de arquivos

```
printf("Você entra na cadeia, você se lembra vagamente dela e  
printf("Quatro celas quadradas, duas de cada lado. É um espaço  
printf("Atrás de uma mesa um homem com barba por fazer e roupa camplada.  
printf("Você nunca o viu, porém subentende que ele deve ser o tal capitão.  
printf("Ao seu lado esquerdo uma linda moça com jeito de Femme Fatale e  
printf("A sua direita um homem atarracado e medroso. Que você reconhece como John Grimm.  
printf("Ele era um vereador muito conservador.
```

cap.txt - Bloco de notas

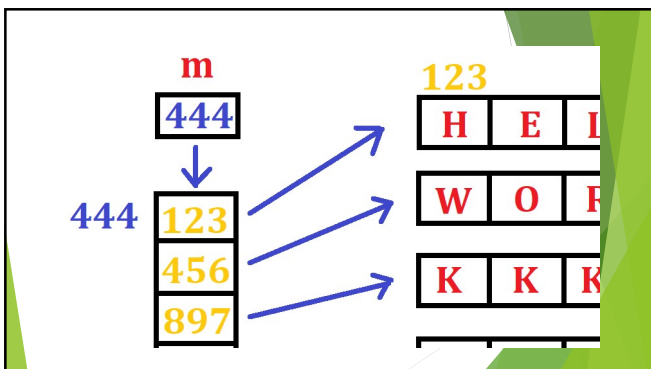
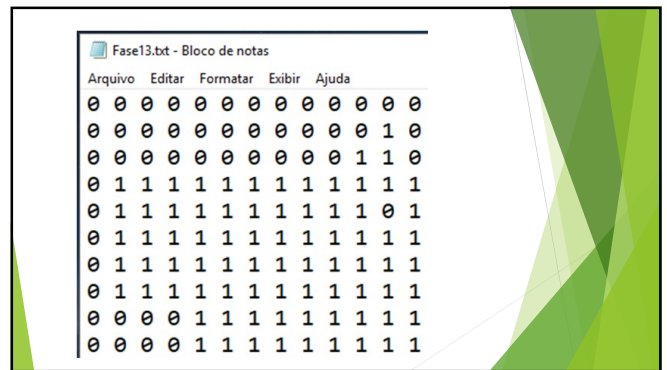
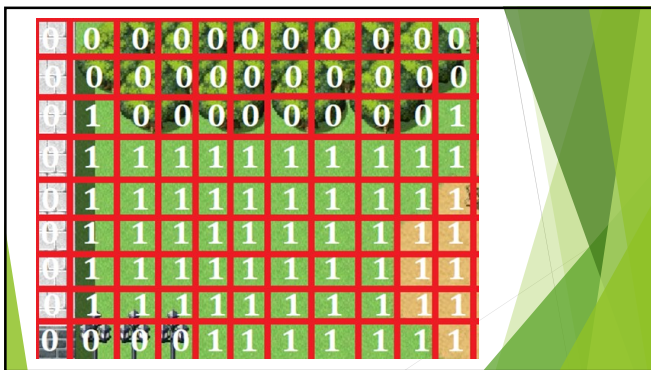
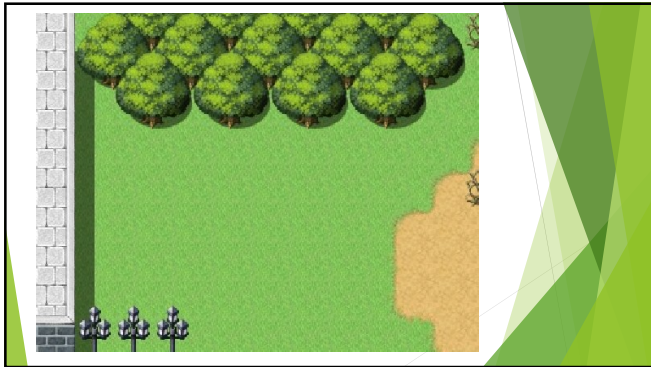
Arquivo Editar Formatar Exibir Ajuda

Você entra na cadeia, você se lembra vagamente dela e dá um risinho.
Quatro celas quadradas, duas de cada lado. É um espaço amplo entre elas.
Atrás de uma mesa um homem com barba por fazer e roupa camplada.
Você nunca o viu, porém subentende que ele deve ser o tal capitão.
Ao seu lado esquerdo uma linda moça com jeito de Femme Fatale e
a sua direita um homem atarracado e medroso. Que você reconhece como John Grimm.
Ele era um vereador muito conservador.

Requisitos: 6 - Manipulação de matrizes

Exemplo:

- ▶ Controlar o cenário do jogo
- ▶ Vetor de strings



Requisitos: 7 - Usar a biblioteca gráfica Allegro

- Uma biblioteca gráfica é um conjunto de comandos e funções que permitem que o programador consiga facilmente exibir e manipular imagens na tela do seu programa.

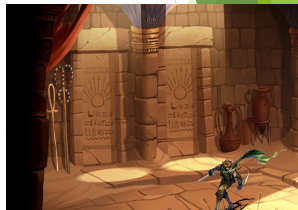
Requisitos: 7 - Usar a biblioteca gráfica Allegro

Victor é policial desmuniado e que busca chegar ao alto escalão do comando de um serviço secreto de inteligência.

Depois de todas essas tentativas frustradas, Victor vê a oportunidade certa, mas... Victor vai procurar o autor de um assassinato do prefeito da cidade de Berlog. Os principais suspeitos são seus rivais políticos, sabendo disso Victor sabe que ele está na elite da polícia.

Muito bem. Sua jornada começa agora mesmo. Boa sorte.

Pressione qualquer tecla para continuar. . .



Requisitos: 7 - Usar a biblioteca gráfica Allegro

- ▶ Esta biblioteca pode ser baixada gratuitamente no site: <http://liballeg.org>
- ▶ Você pode encontrar diversos tutoriais espalhados pela internet.

Estrutura do jogo... Como implementar?

- ▶ Função main() pequena se comparada ao código todo.
- ▶ Uso extensivo de funções.
- ▶ Duas estruturas são mais comuns na hora de implementar um jogo:
 - ▶ Com laço de repetição.
 - ▶ Sem laço de repetição.

Exemplo com laço de repetição:

- ▶ Função main() grande.
- ▶ Porém mais legível.

Fica parecida com um “resumo” do jogo

```
int main()
{
    CarregarJogo();
    while(fimJogo == 0)
    {
        switch(fase)
        {
            case 1:
            {
                CarregarJogo();
            } break;
            case 2:
            {
                CarregarJogo();
            } break;
        }
    }
}
```

Exemplo sem laço:

- ▶ Função main() pequena.
- ▶ Porém menos legível.

Para entender o código é necessário seguir o fluxo das chamadas.

```
int main()
{
    ExibirDialogoInicial();
    CriarPersonagem();
    IniciarJogo();
}
```

Função que conterá as inicializações

Opcional:

- ▶ Lista encadeada.
- ▶ Busca e ordenação.



Horários da Monitoria:

Segunda-feira -> 11 às 13h (Sala 3)
Terça-feira -> 11 às 13h (Laboratório)
Quinta-feira -> 11 às 13h (Sala 3)

E-mail para contato:

victor2663@hotmail.com