

*ALGORITMOS E LINGUAGENS DE PROGRAMAÇÃO**PROFESSOR JEFFERSON CHAVES**jefferson.chaves@ifpr.edu.br**ATIVIDADE AVALIATIVA***Codificando seu próprio jogo de aventura
em modo texto**

Na década de 1980, a computação gráfica ainda estava em sua infância, com personagens de jogos e uma paleta de cores limitada para trabalhar. Era muito comum os jogos de aventura e role-playing serem completamente baseados em texto (também conhecidos como Interactive Fiction), com o jogador usando sua imaginação para criar visões do mundo do jogo.

Neste tipo de jogo toda a interação ocorre por meio de palavras na tela. Apesar de suas origens primitivas nascidas de limitações de hardware, esse tipo de aventura ainda está vivo hoje e é agradável em dispositivos modernos.

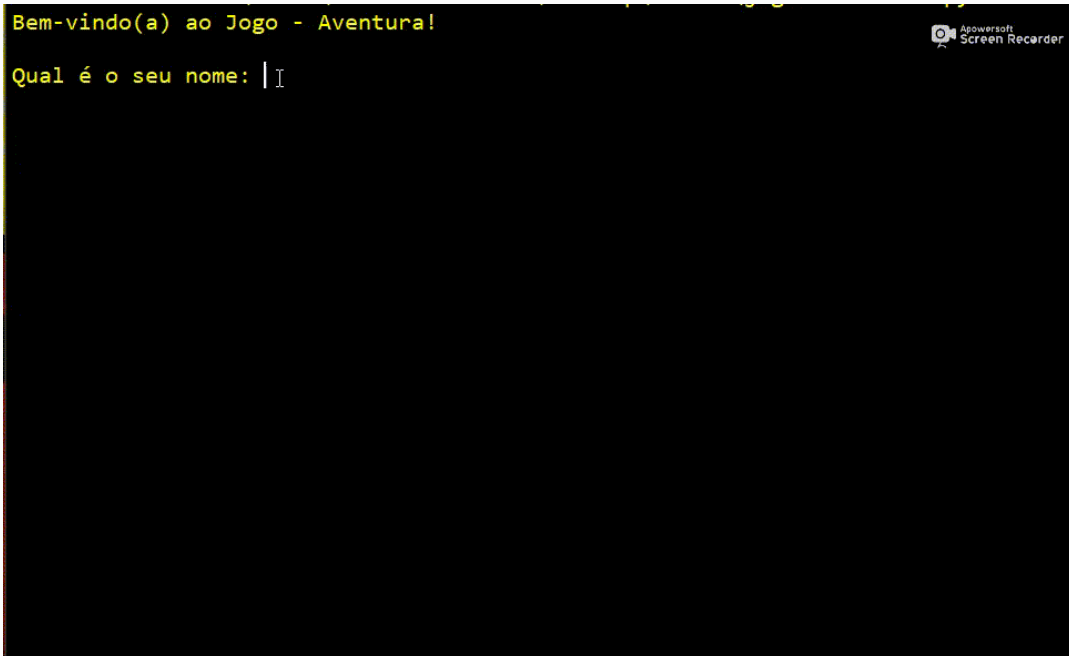
Jogos como Zork criaram mundos ricos, com histórias e personagens atraentes, mesmo com poucos gráficos para ilustrar os ambientes. Isso só mudou no final dos anos 80, início dos anos 90, devido a um trabalho fantástico da LucasArts, que criou uma coleção de jogos clássicos de aventura gráfica como Loom, Monkey Island e Full Throttle.



CONSTRUINDO SEU JOGO

Este projeto consiste em construir um jogo de aventura baseado em texto. “Ahhhhhhhhhhh, como eu vou fazer um jogo?!”

Após saber que deveria fazer um jogo, talvez você tenha se perguntado como, mas fique tranquilo, fazer um jogo pode ser mais simples do que você pensa. **Um exemplo:**



```
Bem-vindo(a) ao Jogo - Aventura!  
Qual é o seu nome: |
```

1 - Criando uma narrativa

Nosso jogo precisa de três coisas: uma história para manter o jogador entretido, lógica e programação para controlar como a história se desenrola. Para a nossa história, podemos criar um mundo de {???}, onde magia e aventura estão em cada esquina. Podemos criar um herói que deve viajar por este novo mundo,

visitando novas aldeias e assentamentos, e conversando com as pessoas para aprender mais. Seu herói pode ser uma lagartixa ou o Velho da Havan. Sinta-se à vontade para tornar a história sua. Este é seu jogo.

2. Use variáveis

Nossa história se desdobra através de blocos de texto que formam nossa narrativa, e você verá que cada bloco se parece com isso:

```
cadeia chapter1 = "Foi uma noite fria, e a chuva veio do oeste com uma ferocidade conhecida apenas pelos deuses."
```

Essas são chamadas de variáveis e permitem armazenar muitos textos ou números. Nós os usamos para conter nossa história e, quando queremos usá-los, usamos a função de impressão assim:

```
escreva(chapter1)
```

A função de impressão olha dentro da variável e imprime seu conteúdo na tela, o que é realmente útil e significa que nós só temos que escrever a história uma vez e podemos reutilizá-la quantas vezes quisermos...

3. Crie funções

Você pode agrupar vários códigos em uma função. Para criar uma, usamos **funcao nome_da_funcao () { comandos }** - isso cria uma função chamada nome_da_funcao, e podemos chamar a função como quisermos.



4. Faça escolhas

Declarações condicionais são escolhas lógicas; nós os usamos para controlar o fluxo do jogo - por exemplo, em qual direção você vai?

Nós armazenamos a direção que o jogador deseja ir como uma variável chamada, por exemplo, de **direcao**.

Então compare o valor armazenado na variável com a tecla que pedimos ao usuário para pressionar. Se os dois corresponderem, o código relacionado será executado.

5. Apresente pessoas

Nosso jogo precisa de pessoas interessantes para falar, então é necessário criar ao menos uma em seu jogo.

6. Escolha um número

Números aleatórios ajudarão nossa aventura a ser completamente única. Para selecionar aleatoriamente esses números, usamos o módulo aleatório, em particular uma função chamada **sorteia** disponível na biblioteca **Util**.

Para usar o **sorteia**, devemos informar qual o intervalo de números com o qual trabalhar, e isso é feito da seguinte maneira:

sorteia (1,10)

Isso cria um intervalo entre 1 e 10 e solicita que o **sorteia** escolha um número aleatório nesse intervalo.

7. tenha uma batalha épica

E quanto ao combate? Aventura que se preze tem que possuir uma treta. Inimigos podem ser criados, podem ser usados temas como saúde e magia (HP e MP em linguagem de aventura), sua criatividade quem define.

Recomendações

A principal recomendação é: **no princípio você faça o mínimo.** depois que este “mínimo” funcionar, prossiga com o desenvolvimento de seu jogo.

Não comece uma nova parte do jogo, sem antes ter testado as outras partes.

Separa seu código em funções. Dividir um grande problema em partes menores, permitirá focar em um tema mais específico por vez, trará melhor organização do seu código, além de possibilitar o reuso, quando necessário.

Seja criativo:

<https://patorjk.com/software/taag/#p=display&f=Doh&t=TDS>

<https://manytools.org/hacker-tools/convert-images-to-ascii-art/>