

Projeto 2021/01 - Super Text Adventure

UTFPR/DACOM 2021-01

BCC35A-Linguagens de Programação

Text Adventure

- Jogo do gênero Adventure (puro) em modo texto
 - ▶ Click & Point Adventure games populares nos anos 80
 - ▶ Ênfase em narrativa, exploração e puzzles
- Mais sobre text-based games
 - ▶ https://en.wikipedia.org/wiki/Text-based_game
 - ▶ https://en.wikipedia.org/wiki/Interactive_fiction
- Jogos do gênero:
 - ▶ <http://textadventures.co.uk/>

Text Adventure: características gerais

- Jogo em modo **texto** com foco em narrativa.
 - ▶ Interação via linha/digitação de comandos:
 - ★ **use** - interagir com objetos da cena
 - ★ **use with** - usar objetos do inventário em objetos da cena
 - ★ **get** - obter objeto **coletável** da cena para o inventário
 - ★ **check** - descrever objeto (cena ou inventário)
 - ★ **inventory** - listar inventário
 - ★ **help** - exibir lista de comandos
 - ★ **save/load** - salvar/carregar estado do jogo
 - ★ **newgame** - iniciar novo jogo (após confirmação)

Cena 1: O Início

Você abre os olhos e está em quarto pequeno, com pouca iluminação, sentado em uma CADEIRA. Você vê um INTERRUPTOR na parede que está a sua frente. Ao lado oposto, há uma JANELA. Abaixo de sua cadeira há uma PEDRA.

```
/> use INTERRUPTOR  
O interruptor esta quebrado e sem efeito.
```

```
/> check JANELA  
Um dos vidros está rachado.
```

Text Adventure: características gerais

- Mundo composto por “cenas”
 - ▶ Cada cena possui:
 - ★ Descrição em texto
 - ★ Lista de objetos com os quais é possível interagir (alguns são **coletáveis**)
 - ▶ Cenas são interligadas
 - ★ Jogador avança nas cenas (pode haver “backtracking”)
- Jogo deve ter título, final e game over
 - ▶ Pode ter mais de um final
- Forte ênfase em narrativa: **sejam criativos!**
 - ▶ Podem haver múltiplos caminhos
 - ★ Alguns podem intersectar
- Gerenciamento simples de inventário
 - ▶ Obter objetos da cena
 - ▶ Usar objetos do inventário em objetos da cena
 - ★ Ex: use KEY with DOOR

Gameplay: cenas e objetos

- Cada cena apresenta título e descrição
- Os objetos da cena com os quais o jogador pode interagir são sempre descritos em letras MAIÚSCULAS
- A única forma de passar de uma cena para outra é através da interação com objetos da cena (SCENE_OBJECT)
 - ▶ Pelo uso direto de **objeto da cena**
 - ★ /> **use SCENE_OBJECT**
 - ★ Ex: use PORTA.
 - ★ Resultado: Você abriu a Porta e passou para o corredor.
 - ▶ Pelo uso de um **objeto do inventário** em um **objeto da cena**
 - ★ /> **use INVENTORY_OBJECT with SCENE_OBJECT**
 - ★ Ex: use PEDRA with JANELA
 - ★ Resultado: A pedra quebrou a janela e você passou por ela.
- Cena possui **Lista** de objetos. Quando um objeto **coletável** da cena é colocado no inventário, sai da **Lista** da cena e entra na **Lista** do inventário.
 - ▶ Objetos são usados para resolver pequenos quebra-cabeças (puzzles)

Gameplay: exemplo

Cena 1: O Início

Você abre os olhos e está em quarto pequeno, com pouca iluminação.

Você vê um INTERRUPTOR na parede que está a sua frente.

Ao lado oposto, há uma JANELA. Abaixo de sua cadeira há uma PEDRA.

```
/> use INTERRUPTOR
```

O interruptor esta quebrado e sem efeito.

```
/> use JANELA
```

Não há efeito.

```
/> check JANELA
```

Um dos vidros está rachado.

```
/> get PEDRA
```

A pedra está no inventário.

```
/> inventory
```

PEDRA, MANUAL

```
/> use PEDRA with JANELA
```

A pedra quebrou a janela e você passou por ela.

(deve mudar de cena, mostrando o novo texto)

Comandos do Jogo

- Lista conteúdo do inventário
 - ▶ **/> inventory**
- Interage com algo (abrir, entrar, acionar)
 - ▶ **/> use SCENE_OBJECT**
 - ★ Objeto que está na Lista da cena
 - ▶ Mostra resposta e troca (ou não) de cena
- Usa objeto que está na Lista do inventário com objeto da cena
 - ▶ **/> use INVENTORY_OBJECT with SCENE_OBJECT**
 - ★ Checar se objeto está no inventário
- Descreve objeto (tanto da cena quando do inventario)
 - ▶ **/> check OBJECT**
- Obtêm objeto da cena para o inventário
 - ▶ **/> get SCENE_OBJECT**
 - ★ Somente objeto marcado como **coletável**
 - ▶ Remove objeto da Lista de cena e o coloca na Lista do inventário

Comandos do sistema

- Instruções e Ajuda de comandos
 - ▶ **/> help**
 - ▶ Fornece instruções gerais do jogo
 - ▶ Fornece lista e descrições dos comandos
- Salva estado atual do jogo
 - ▶ **/> save NOME_SAVE**
 - ▶ Salva em arquivo: cena corrente, lista de objetos do inventário e listas dos objetos das cenas
 - ▶ NOME_SAVE indica nome do arquivo
- Carregar estado do jogo
 - ▶ **/> load NOME_SAVE**
 - ▶ Carrega de arquivo os dados salvos
 - ▶ NOME_SAVE indica nome do arquivo
- Reiniciar jogo
 - ▶ **/> newgame**
 - ▶ Pede confirmação ao usuário (S/N)

Requisitos e Critérios

Linguagens e implementação

- Será necessário **instalar compilador/interpretador** no computador
 - ▶ UI em **modo texto**
 - ▶ leitura e escrita de arquivos
 - ▶ requisições http para server **A DEFINIR**
 - ▶ possibilidade: tocar sons via terminal
- Linguagens Compiladas
 - ▶ Todas disponíveis em **repl.it**
 - ▶ Go
 - ▶ C#
 - ▶ Rust
 - ▶ Swift
 - ▶ Kotlin
 - ▶ Nim

Requisitos

- Mínimo

- ▶ 15 salas
- ▶ Número variado de objetos por salas
- ▶ Salas devem ser montadas em arquivo(s) texto
 - ★ Puro, JSON ou XML

- Valorização

- ▶ Esmero com interface e mensagens do jogo
- ▶ Complexidade e qualidade da história
- ▶ **NOVAS FUNCIONALIDADES**
 - ★ Sejam criativos

Avaliação e Questões Gerais

- **Cópias:** Qualquer tipo de cópia de código-fonte (trabalhos de colegas, internet, etc) anulará o trabalho
 - ▶ Seja por porções de código ou pelo trabalho completo
- **Entrega:** código fonte pelo Moodle
- **Apresentação:** síncrona, por sessão agendada
 - ▶ Serão feitas (várias) perguntas sobre o código e funcionalidades
- **Data:** informada na página do Moodle
- **CrITÉrios:**
 - ▶ Cumprimento dos requisitos
 - ▶ Qualidade e acabamento gerais
 - ▶ Qualidade da implementação
 - ▶ Diferenciais em relação aos requisitos e aos demais trabalhos

Projeto e Implementação

Estrutura do jogo

- O mundo do jogo é composto de lista de Cenas
- Cada cena possui uma lista de objetos
- Quando um objeto é coletado (comando **get**), sai da lista de cena e vai para a lista do inventário

Classe Jogo

```
cenar: Lista de Cena  
cena_atual: Chave (Inteiro/Texto)
```

Classe Cena

```
key: Chave (Inteiro/Texto)  
titulo: Texto  
descricao: Texto  
objetos: Lista de Objetos
```

Classe Inventario

```
itens: Lista de Objetos
```

Estrutura do jogo

Classe Objeto

```
id: Chave (Inteiro/Texto)
tipo: SCENE (SCENE_OBJECT) ou COLLECTABLE (INVENTORY_OBJECT)
nome: Texto
descricao: Texto
resultado_positivo: Texto // exibido após uso do comando correto
resultado_negativo: Texto // indica o uso de comando incorreto
// guarda o comando correto que o "soluciona".
// Ex: use INTERRUPTOR; get PEDRA; use PEDRA with JANELA
comando_correto: Texto (pode ser Lista de Texto, caso mais de uma solução)
// chave da cena para qual este objeto leva caso o comando_correto for
// usado ou NONE (não muda cena)
cena_alvo: Chave (Inteiro/Texto) (ou NONE)
// indica se já houve comando anterior que o resolveu
// (para navegar em salas já visitadas/solucionadas)
// Ex: uma porta que precisada de chave e isso já foi resolvido
// Logo, se voltar a interagir, basta usar "use SCENE_OBJECT"
resolvido: Booleano
// indica se objeto já foi obtida da cena, isto é, está no inventário
obtido: Booleano // opcional
```