

Instituto de Computação
Bacharelado em Sistemas de Informação
Disciplina: Projeto de Software
Trabalho (Versão IVs) - 2019.2

Uma Ficção Interativa (*Interactive Fiction*) [1] é um tipo de *software* onde o jogador utiliza comandos textuais para controlar personagens, explorar e influenciar o ambiente de um jogo. Um termo muito comum utilizado para descrever tais jogos é *Text Adventure*, onde a interface é considerada puramente textual.

Abaixo vemos uma tela como o jogo Zork [2], um dos mais famosos *Texture Adventure* já criados.

```
West of House                               Score: 0  Moves: 3

ZORK I: The Great Underground Empire
Copyright (c) 1981, 1982, 1983 Infocom, Inc. All rights
reserved.
ZORK is a registered trademark of Infocom, Inc.
Revision 88 / Serial number 840726

West of House
You are standing in an open field west of a white house, with
a boarded front door.
There is a small mailbox here.

> examine mailbox
The small mailbox is closed.

> open mailbox
Opening the small mailbox reveals a leaflet.

> get leaflet
Taken.

> |
```

Existem vários componentes que formam um *Texture Adventure*:

- Narrativa
- Interpretação de comandos textuais
- Navegação em um ambiente formado por regiões interconectadas
- Alteração do ambiente
- Controle do estado do jogador
- Gerenciamento de objetos coletados (*inventory*)
- Gerenciamento de eventos e objetivos
- Quebra-cabeças (Puzzles)

Narrativa

A narrativa é um dos componentes principais e muito importante para manter o jogador imerso no ambiente do jogo. É ela que permite descrever em detalhes o ambiente em que a história se passa para que o jogador possa explorar e interagir de forma adequada com o ambiente. Como um *Texture Adventure* não possui gráficos, a qualidade da narrativa, o nível de detalhes a ser fornecido por ela e sua consistência, é fundamental para que a jogabilidade do jogo seja apropriada.

Comandos

Um *Texture Adventure* se limita a comandos definidos por textos na linha de comandos. Logo, é necessário que eles sejam expressivos o suficiente. Formas mais complexas envolvem o desenvolvimento de um parser (intérprete). Entretanto, para os casos mais simples uma estrutura de comandos formada por “verbo + objeto” ou “verbo + objeto direto + objeto indireto” é suficiente para a interação com o ambiente.

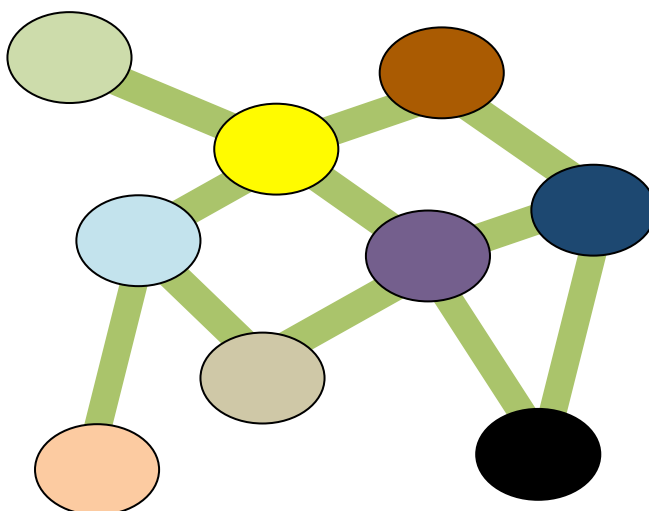
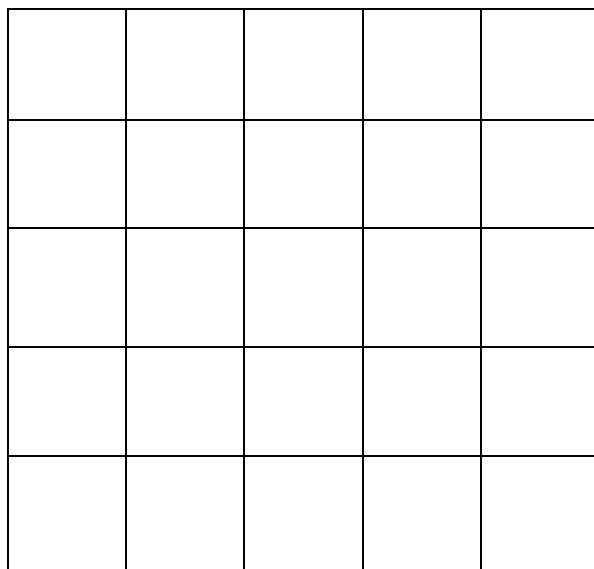
Exemplos de comandos típicos são:

- Ver
- Analisar + objeto
- Pegar + objeto
- Atirar + objeto [+ objeto alvo]
- Descartar + objeto
- Ir + direção
- Saltar + objeto
- Subir [+ objeto]
- Descer [+ objeto]
- Mostrar objetos armazenados (*inventory*)

Para tornar o jogo mais realista, os comandos devem poder admitir sinônimos tanto para os verbos quanto para os objetos. Respostas diferenciadas e inusitadas para comandos impossíveis de se realizar podem tornar o jogo mais atraente.

Navegação e exploração do ambiente

O ambiente de um jogo do tipo *Texture Adventure* é composto de várias regiões (alguns jogos podem ter até centenas) interconectadas através de uma estrutura topológica, normalmente na forma de um grafo ou até mesmo, de modo mais simples, via um reticulado.



Em uma estrutura de reticulado, o jogador navega pelo ambiente podendo se dirigir às direções norte, sul, leste e oeste. Em alguns dos casos, (ver a seção de eventos) a movimentação só terá sucesso se algum evento tiver sido disparado ou um sub-objetivo realizado, por exemplo, obter uma chave que abre um portão que dá acesso a direção norte, ou então, obter uma corda que permite o jogador descer por um desfiladeiro e chegar a região a oeste.

Alteração do ambiente

Conforme o jogador interage com o ambiente ele pode alterá-lo, ou alterar sua percepção do mesmo. Por exemplo, ele pode coletar objetos, alterar a o nível de iluminação do ambiente (acendendo uma luz, vela ou tocha),quebrar um objeto, etc. Estas ações disparam eventos, alguns dos quais necessários para concluir um determinado objetivo em uma região, caso ele tenha sido definido.

Gerenciamento de eventos e objetivos

Um dos grandes desafios de um *Texture Adventure* é gerenciar os eventos que podem ser causados pela interação do jogador e como uma sequência de eventos evolui para a conclusão de um determinado objetivo. A sequência de eventos pode ser linear ou não , sendo esta última mais difícil de gerenciar. Um exemplo de uma sequência linear seria um personagem encontrar uma vela, uma caixa de fósforos, em seguida acender a vela, observar o ambiente, encontrar um baú, encontrar a chave em um armário, abrir o baú, encontrar um machado e arrombar uma porta travada. Neste caso, arrombar a porta para sair da sala seria um objetivo local à região, neste caso.

Controle do estado do jogador

O jogador pode ter vidas ou energia, habilidades e outras características, podendo estas serem estáticas ou dinâmicas, isto é, podendo evoluir conforme a aventura se desenrola. Isto levaria um *Texture Adventure* a compartilhar características com um RPG. O controle do estado do jogador em relação ao ambiente e suas ações é um componente importante de um *Texture Adventure*.

Quebra-cabeças ou puzzles

Quebra-cabeças são sub-jogos que podem enriquecer o desafio de um *Texture Adventure*. Alguns deles podem ser apenas opcionais e outros podem ser parte de eventos que levam o jogador a alcançar um objetivo.

Especificação do trabalho

- 1) O trabalho consiste em desenvolver um *Texture Adventure* em uma linguagem orientada a objetos buscando atender os seguintes requisitos:
 - O jogo deve possuir todos os componentes principais do jogo: narrativa, gerenciamento de eventos e objetivos, interpretação de comandos, gerenciamento do objetos armazenados pelo jogador, navegação, controle do ambiente e do estados do jogo.
 - O sistema deve ser extensível (novas regiões, eventos, objetos e comandos devem poder ser adicionados sem maiores alterações no código)
 - Todos os componentes do jogo devem poder ser instanciados sem criar problemas de manutenção do código.
 - Deve ser possível adicionar novos eventos, objetos e comandos válidos sem alterar substancialmente a estrutura e definição destes elementos
 - O núcleo do controle do jogo deve ser uniforme para todas as regiões. Procure não gerar código específico para cada região.

Para atingir os objetivos, atente para o uso correto dos princípios de projeto O.O e uso de técnicas de reuso como padrões e *frameworks*.

Projete a arquitetura do sistema procurando identificar cada uma de suas camadas.

- 1) Projete a arquitetura do sistema procurando identificar cada uma de suas camadas. Utilize as visões da arquitetura que julgar necessário para a apresentação do projeto.
- 2) A documentação deve conter um Diagrama de Arquitetura, Diagrama de Casos de Uso e Diagrama de Classes. Utilize Diagramas de Sequência e de Estados se julgar apropriado.

3) Implemente o seu sistema observando os seguintes aspectos e tendo as seguintes diretivas em mente:

- a) Obedeça sempre que possível os princípios de projeto O.O apresentados no curso.
- b) Procure minimizar as congeneridades e manter um sistema com classes coesas.
- c) Identifique os domínios a que pertencem as classes do sistema e evite classes mistas.
- d) Utiliza sempre que possível os padrões apropriados para as situações que podem surgir durante o desenvolvimento.
- e) Procure desenvolver seu sistema prevendo possíveis extensões e mudanças.

Avaliação

O trabalho será avaliado em duas partes: documentação e implementação.

A implementação será avaliada em função do uso correto dos princípios de projeto O.O e da satisfação dos requisitos especificados anteriormente. Procure utilizar pelo menos 3 padrões de projeto em sua implementação e justifique o seu uso.

Data de entrega

O trabalho deve ser entregue em 03 de dezembro de 2019

Referências

[1] http://en.wikipedia.org/wiki/Interactive_fiction

[2] <http://en.wikipedia.org/wiki/Zork>