txtEngine Documentation

Language Specification

Introduction

Utilising the advantages of XML we are able to suitably appropriate the framework to interpret instruction directly into object-orientated items. There are several objects that are critical to the framework design. These are World, Area, AreaCommand, Item, ItemCommand , and StateDescriptor. These will be discussed in depth below.

DTD

There is a DTD in the dtd folder, read that for further clarification on xml formatting.

XML Version Tag

Specifies the type of encoding used in the game file.

```
<?xml version ="1.0"?>
```

World

The World element contains everything.

Attributes:

- **initialarea** - the start area for the game
- **language** - the language spec used
- **author** - who wrote the game

Example:

```
<world initialarea="start_location" language="txtEngine" author="Reginald">
        …...
</world>
```

Area

Area elements are the places the player can ‘move’ to in the game. They contain the Items initially.

Attributes:

- **id** - identifier for the area
- **initialdescription** - the id of the state descriptor to use when the area is first visited.

Example:

```
 <area id="start_chamber" initialdescription="maze_start" >
        …...
</area>
```

Area Command

Attributes:

- **name** - the name of the areacommand
- **area** - the area that this command takes you to
- **depends** - specifies whether you need a certain item in order to use this command
- **synonyms** - alternative names for the area command

Example:

```
<areacommand name="east" area="furnace_room" depends="lamp" synonyms=”up”>
          The smell of coal fills your nostrils and you feel like you are walking into the sun.
</areacommand>
```

Item

Things throughout the world which the player can interact with. If collectable, the item can be moved to the players inventory.

Attributes:

id - a name that uniquely identifies the item
initialdescription - the description of an item when first
collectable - true or false indicating whether the item can be picked up or not
synonyms - alternative names to refer to the item

Example:

```
<item id="lamp" initialdescription="onpodium" collectable ="true"
synonyms="torch,light,flashlight">
 ….....
</item>
```

Item Command

The itemcommand element is placed inside an item element and defines the interactions that the player can have with the item.

Attributes:

name - a name that uniquely identifies the item command
statedescription - the state of the item changes to this when the command is executed
changecollectable - true if this command changes the collectable tag of the item when called. false if it does not
collectabledependent - true if this command can only be called when the item is collecatble
areachange - the area the item changes to when this command is executed
synonyms - alternative names to refer to the item command

Example:

```
<itemcommand name="get" synonyms="fetch,grab,pickup" statedescription="onplayer"
changecollectable="true" collectabledependent="true" areachange="inventory">
        You pick up the lamp the light is warm and comforting.
</itemcommand>
```

StateDescriptor

The statedescriptor element is used inside Items and Areas to describe the Items and Areas in different states during the game.

Attributes:

id - the unique id of the statedescriptor

Example:

```
<statedescriptor id="dropped">
        A glowing lamp lying amongst the rubble.
</statedescriptor>
```

Overview of Game:

```
<world>
        <area>
                <statedescriptor>
                </statedescriptor>
                <areacommand>
                </areacommand>
                <item>
                        <statedescriptor>
                        </statedescriptor>
                        <itemcommand>
                        </itemcommand>
                </item>
        </area>
</world>
```

Installing the Game

Any standard C++ compiler can be used to compile txtEngine.
Alternatively for:

Linux

A Makefile is included which builds the txtEngine configuration for Linux operating systems. GNU Make and G++ Compiler are needed to use this feature.
Simply navigate to the txtengine folder and run ‘make’.

Playing the Game

New Game

To start a new game, from the build directory type:

```
./txtgame [name of game file]
```

Loading a Game

To load a previously saved game:

```
./txtgame [name of game file] [name of save file]
```

Saving a Game

When playing a game, simply type the command ‘save’ and press enter.
You will be prompted to enter a file name.
This file is now saved. To load a previously saved file see Loading a game.

Inbuilt Commands

inventory - Lists the items in the players inventory in the order they were picked up
help - Displays a help string which is defined in the Constants.h file
quit - Exits the game
go north - also: n, north - Move the character to the north.
go south - also: s, south - Move the character to the south.

go east - also: e, east - Move the character to the east.
go west - also: w, west - Move the character to the west.

Bugs

- Duplicate synonyms across area commands, item commands, and items are not checked for and will be implemented in the next version of the game.

Future Features and To Do

- A graphical user interface for writing games.
- Documentation written in Latex.
- Comment code and generate proper Doxygen Documentation.