Afan Rasool ; Richmond Frimpong ;Syed

Group #:

Abstract

Implementation of the game Zork in Eiffel Programming Language

eecs 3311

Project : ZORK

Contents

[BRIEF DESCRIPTION 3](#_Toc457605818)

[USER DOCUMENTATION 🡪 FOR PLAYERS 3](#_Toc457605819)

[CLASS DOCUMENTATION 4](#_Toc457605820)

[CLASS INVENTORY 4](#_Toc457605821)

[DESCRIPTION OF FEATURES: 4](#_Toc457605822)

[DESIGN CONTRACTS 4](#_Toc457605823)

[Design Pattern 5](#_Toc457605824)

# BRIEF DESCRIPTION

This is an implementation of the game Zork. This is a text game in which a player ventures through different mazes in search of wealth and treasure (anything valuable the player can find while exploring the maze). The maze is considered to be dangerous and the ultimate goal of this game is to come out of the maze alive with all the treasures needed to complete the maze. Scoring depends on the number of moves you make before collecting all the treasures from the maze.

The game is played using various text commands consisting of, although not limited to, various verb and noun words such as “NORTH”, “SOUTH”, “OPEN DOOR” etc. Complete description of the commands available is explained later in this report.

# USER DOCUMENTATION 🡪 FOR PLAYERS

< ALL THE HELP DOCS FOR THE USERS COME HERE.. DESCRIPTION OF EACH AND EVERY COMMAND AVAILABLE TO THEM AND WHAT DOES IT DO.. KIND OF SIMILAR TO THE HELP.TXT FILE>

# CLASS DOCUMENTATION

## CLASS INVENTORY

### DESCRIPTION OF FEATURES:

>list: ARRAYED\_LIST[ITEM[T]]

-Holds the list of treasures available

>count: INTEGER

-Number of items in the list.

>add\_inventory(item:T):BOOLEAN

- Adds an item to the list of treasures (inventory)

>get\_inventory:ARRAYED\_LIST[T]

-Returns a list of inventory items

>remove\_inventory\_item(item:T):BOOLEAN

-Removes item:T from the list of inventory

### DESIGN CONTRACTS

Class Invariant: list/=void & count >= 0

>add\_inventory(item:T):BOOLEAN

- require: item /= void

- ensure: count = old count + 1 & list.count = count

Explanation: The pre-condition we chose makes sure that item being added to the list is not void. The post-condition we chose makes sure that the list has been incremented by one making sure that a new item has been added.

>get\_inventory:ARRAYED\_LIST[T]

require: <no precondition required>

ensure: result /= void

Explanation: There is no pre condition because our design allows the players to have no inventory (an empty list). The post condition we chose makes sure that this function we have a valid result and not void.

> remove\_inventory\_item(item:T):BOOLEAN

require: item /= void

ensure: old count - 1 & list.count = count

Explanation: The pre-condition we have chosen ensures that the item we are trying to remove exits in the list and isn’t void. The post-condition we chose makes sure that after the function has been performed, our list has one lesser item.

# Design Pattern

1. Singleton
2. MVC (Model, View, Controller)

* SINGLETON

< description OF SINGLETON COMES HERE>

* MVC

MVC design pattern used to implement the divide and conquer approach. Different tasks being handled by each one of the Model, View and Controller simplifies communication problems between tasks.

Model:

Model does data manipulation and processing if needed. <in our case I think model isn’t being used>

Controller:

Controller is the “middleman” between model and view. View and Model cannot interact directly. View mostly interacts with the controller, reacts to the actions performed by view and instructing it to get changed accordingly. In the context of Zork , it receives user input collected from view and updates the view accordingly depending on what action is performed by the user.

View:

This is the graphical part of the program, the screen seen by the User. In our context this is the command line and game text visible to the user is handled here in View.e class of our program.

Figure 1

User enters “NORTH”

Updates View with what can be seen in the NORTH

Figure 1 shows an example of the communication between the controller.e and view.e classes of our Zork prototype.