4th Workshop "Software Engineering Education and Reverse Engineering"
Zagreb, 5 – 12 September 2004

The case study 'Encounter'

Section 'About case studies'

Živana Komlenov
Department of Mathematics and Informatics
Faculty of Science
University of Novi Sad

Encounter

- Potential additional case study for Joint Course in Software Engineering (for illustrating some of the basic principles and usage in practical assignments).
- Sources:
  - 'Software Engineering: An Object-Oriented Perspective' by Eric J. Braude (Wiley 2001)
  - 'Software Design: From Programming to Architecture' by Eric J. Braude (Wiley 2004)
- So far, we have game implementation and some documentation (the list will be given later).

Contents

- Software presentation
- Software Requirements Specification
- Version differences
- Differences with respect to IEEE Standard
- Software Design Document
- Source Code

3 Živana Komlenov: The case study 'Encounter'

Currently existing documentation

- Software Requirements Specification (SRS)
  - Version 1, Version 2, Final document
  - Version differences
  - Differences with respect to IEEE Standard
- Software Design Document (SDD)
- Source Code

- Some of these documents are just temporary, local files used in order to create final versions. Therefore, only final SRS and SDD documents shall be discussed.

4 Živana Komlenov: The case study 'Encounter'

Requirements Specification

- Version 1
  - Hal Furness and Karen Peters
- Version 2
  - 6/22/00 Update: Eric Braude and Tom van Court.
  - Reworked to reflect lessons learned by implementing previous version
- Final document based on version 2
- Version differences
- Differences with respect to IEEE Standard

5 Živana Komlenov: The case study 'Encounter'

Requirements Specification

- 1. Introduction
  - 1.1 Purpose
  - 1.2 Scope
  - 1.3 Definitions, acronyms, & abbreviations
  - 1.4 References
  - 1.5 Overview
- 2. Overall Description
  - 2.1 Product perspective
  - 2.2 Product functions
  - 2.3 User characteristics
  - 2.4 Constraints
  - 2.5 Assumptions and dependencies

6 Živana Komlenov: The case study 'Encounter'

Requirements Specification

- 2.6 Apportioning of requirements
- 3. Specific Requirements
  - 3.1 External interface requirements
  - 3.2 Specific requirements
  - 3.3 Performance requirements
  - 3.4 Design constraints
  - 3.5 Software system attributes
  - 3.6 Other requirements
- 4. Supporting information
  - 4.1 Table of contents and index
  - 4.2 Appendixes

SRS: 1. Introduction

- SRS document provides all of the requirements for release 1.0 of the Encounter video game.
- *Encounter* is to be a role-playing game which simulates all or part of the lifetime of the player's main character.
- Definitions, acronyms, & abbreviations:
  - *Alive*: a game character is said to be "alive" if it has at least one quality with non-zero value
  - *Encounter*: name of this application; also, a meeting between two game characters in an area (but not necessarily an engagement)
  - *Engagement*: an interaction between characters of the game, which typically affects the characters

SRS: 2. Overall Description

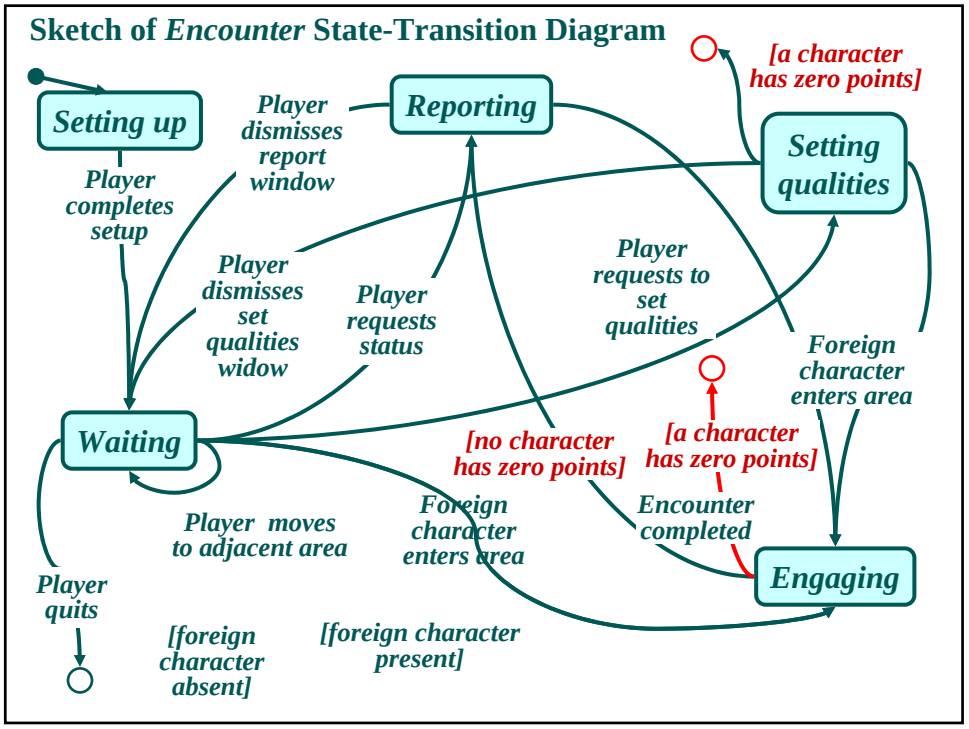
- Success in playing *Encounter* will be measured by the "life points" maximum attained by the player or by the ability of the player to live as long as possible.
- Some game characters are to be under the control of the player (one of them as “main”). The rest, called "foreign" characters, are to be under the application's control. In early versions of this game, there will be only one player-controlled character, and one foreign character.
- Game characters will have a fixed total number of points allocated among qualities: *strength*, *stamina*, *patience*, etc.
- Characters encounter each other when they are in the same area at the same time. The result of the engagement depends on the values of their qualities and on the environment in which the engagement takes place.

9 Živana Komlenov: The case study ‘Encounter’

SRS: 2.1.1 Concept of operations

- *Encounter* can be in one of several states:
  - *Setting up*: the game is being set up by the player
  - *Reporting*: the system is displaying a window showing the status of the player's character
  - *Setting qualities*: equipping the player's character with qualities; can be performed as long as no foreign character is present.
  - *Engaging*: applies whenever a foreign character and the player's character are present in an area at the same time
  - *Waiting*: the player and the foreign character are not active

10 Živana Komlenov: The case study ‘Encounter’



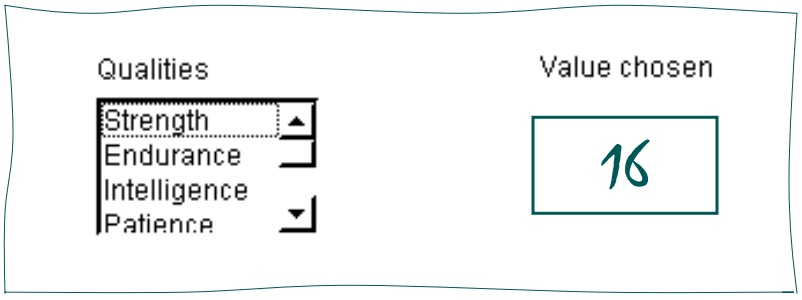
SRS: 2.1.2 User interface concepts [1]

- 2.1.2.1 *Area* user interface concept
  - The areas in which encounters take place shall have an appearance very roughly like that shown in the next figure.
- 2.1.2.2 User interface concept for setting quality values
  - When setting the values of game characters under his control, the player will retrieve an interface of the form sketched approximately in the figure following.
  - The scroll box is used to identify the quality to be set, and the text box is used for setting the value.

Preliminary *Encounter* Screen Shot

Graphics reproduced with permission from Corel.

Preliminary Sketch of User Interface for Setting Game Character Qualities

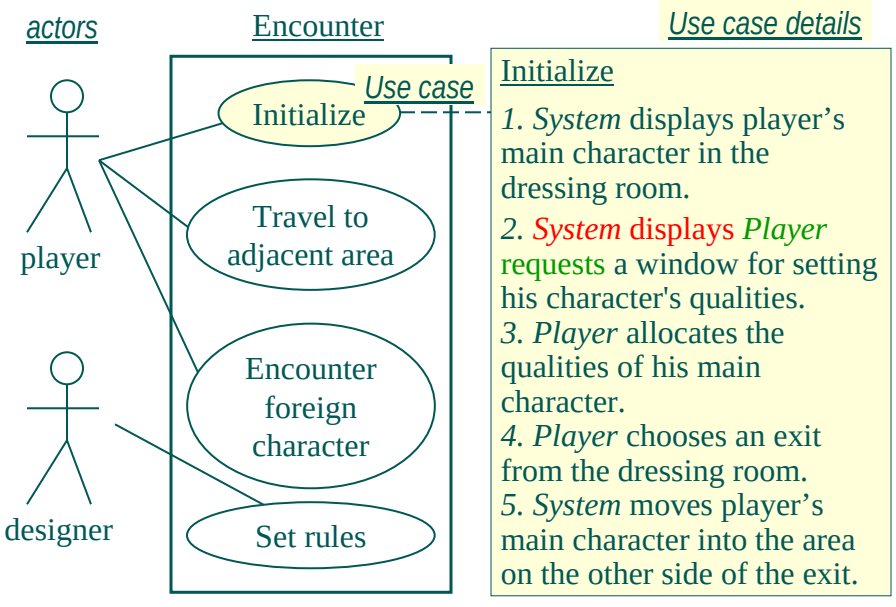


SRS: 2.2 Product functions

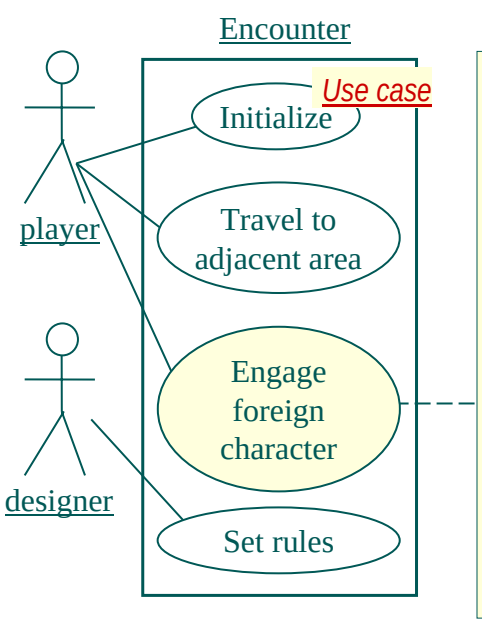
- 2.2.1 "Initialize" use case
  - Actor: player of *Encounter*
  - *Initialize* is the typical sequence users execute at the beginning of session.
- 2.2.2 "Travel to adjacent area" use case
  - Actor: player of *Encounter*
  - *Player* hits hyperlink connecting displayed area to adjacent area
  - *System* displays the indicated adjacent area containing player's character
- 2.2.3 "Encounter foreign character" use case
  - Actor: player of *Encounter*

15 Živana Komlenov: The case study 'Encounter'

Initialize Use Case for *Encounter*



Engage Foreign Character Use Case



Use case details

Engage Foreign Character

1. System displays the foreign character in the same area as the player's.

2. System exchanges quality values between the two characters.

3. System displays the results of the engagement.

4. If either of the characters has no points, the game terminates. Otherwise, *System* displays player's character in a random area.

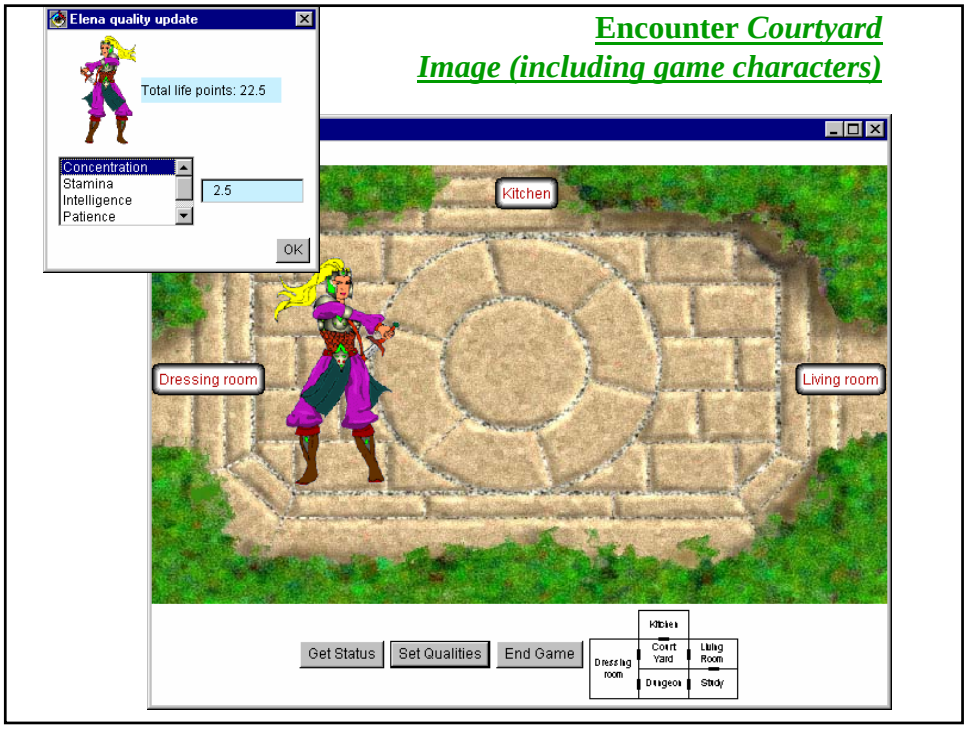
SRS: 3. Specific Requirements [1]

- 3.1 External interface requirements
  - 3.1.1 User interfaces
    - *Encounter* takes place in areas. Next figure shows a typical screen shot of the *courtyard* area, with a player-controlled character, and the results of an engagement.
    - Areas have connections to adjacent areas, labeled by hyperlinks. Clicking on one of these hyperlinks moves the player's character into the corresponding area.
- 3.2.1 Sequence diagrams
  - 3.2.1.1 *Initialize* use case
  - 3.2.1.2 *Travel to adjacent area* use case
  - 3.2.1.3 *Engage foreign character* use case

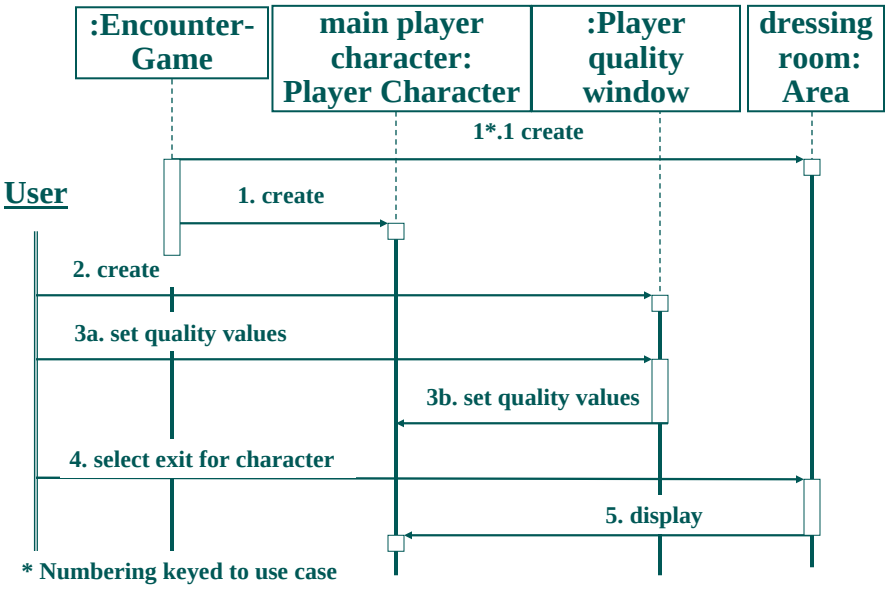
18 Živana Komlenov: The case study 'Encounter'

Encounter *Courtyard*

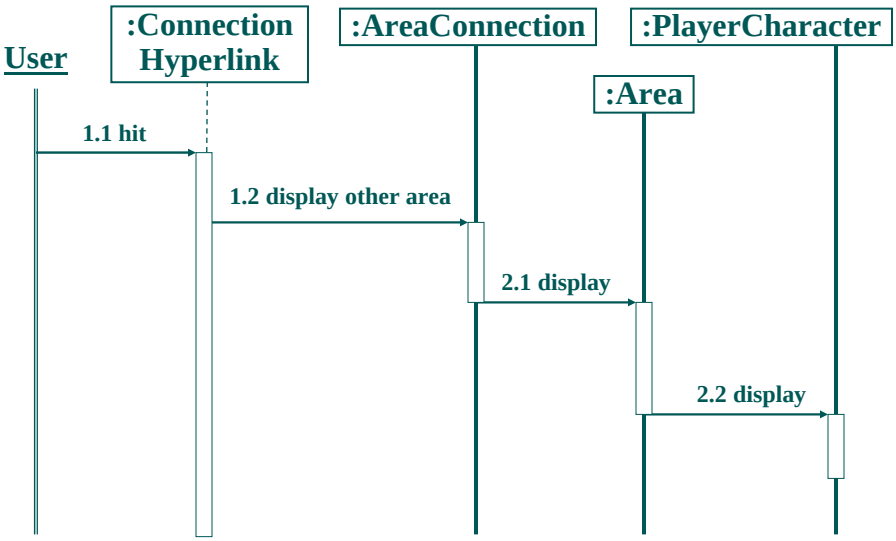
Image (including game characters)



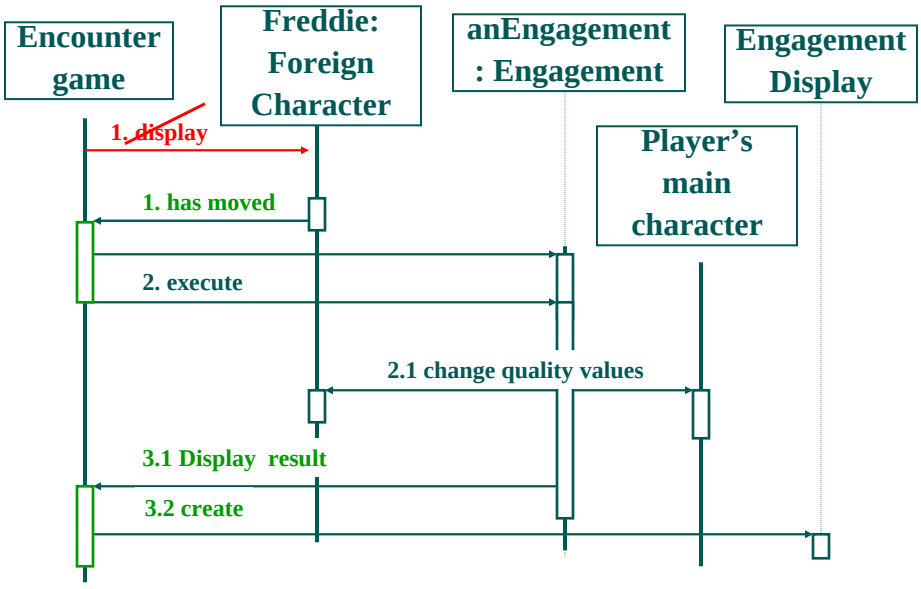
Conceptual Sequence Diagram for *Initialize* Use Case



Conceptual Sequence Diagram for *Travel to Adjacent Area* Use Case



Conceptual Sequence Diagram for *Encounter Foreign Character* Use Case

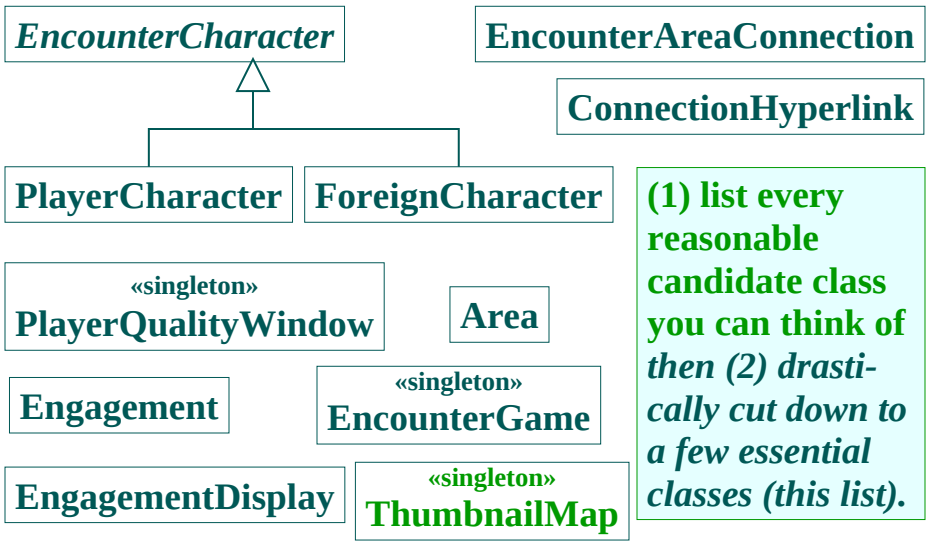


SRS: 3. Specific requirements [2]

- 3.2.2 Classes for classification of specific requirements
  - *Area*
  - *EncounterCharacter*
  - *EncounterGame*
  - *Engagement*
  - *EngagementDisplay*
  - *ForeignCharacter*
  - *PlayerCharacter*
  - *PlayerQualityWindow*
  - *ThumbnailMap*

23 Živana Komlenov: The case study 'Encounter'

Classes for *Encounter* Video Game, Showing Only Inheritance Relationships



SRS: 3.2.AR *Areas*

- An area is a place viewable on the monitor.
- All activities of *Encounter* (including engagements) take place in areas.
- Rooms, gardens and courtyards are examples of areas.
- 3.2.AR.1 Attributes of areas:
  - 3.2. AR.1.1 Area name
  - 3.2. AR.1.2 Area image
  - 3.2.AR.1.3 Area-specific qualities

    (Only some game character qualities shall be applicable in each area.)

25 Živana Komlenov: The case study 'Encounter'

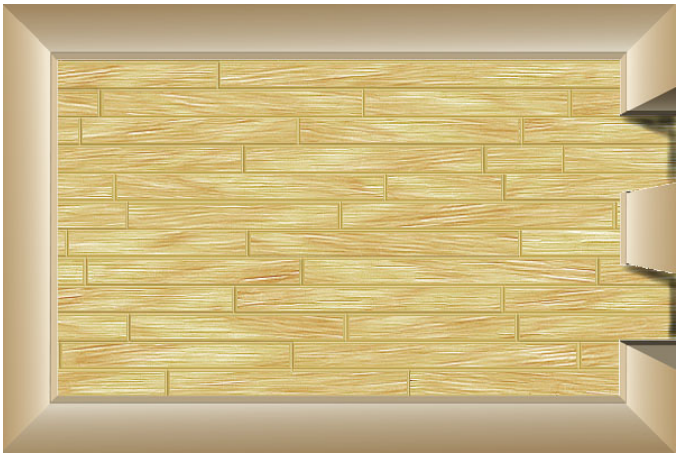
SRS: 3.2.AR.2.1 Courtyard area

- *Area* object with name "courtyard", requiring qualities *stamina*, and *strength*.

26 Živana Komlenov: The case study 'Encounter'

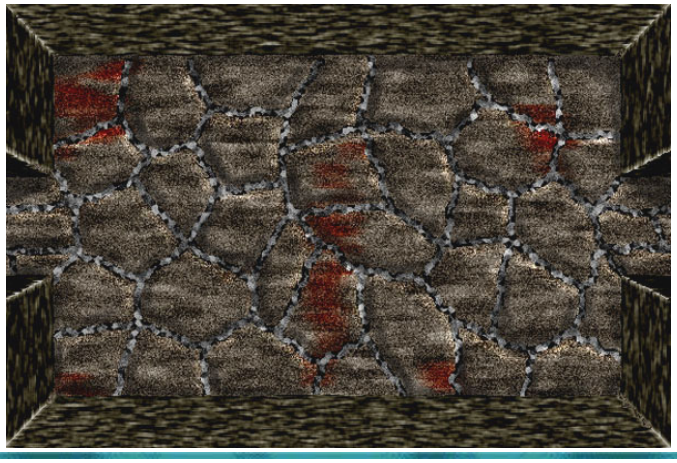
SRS: 3.2.AR.2.2 Dressing room area

- *Area* object with name "dressing room", requiring no qualities.



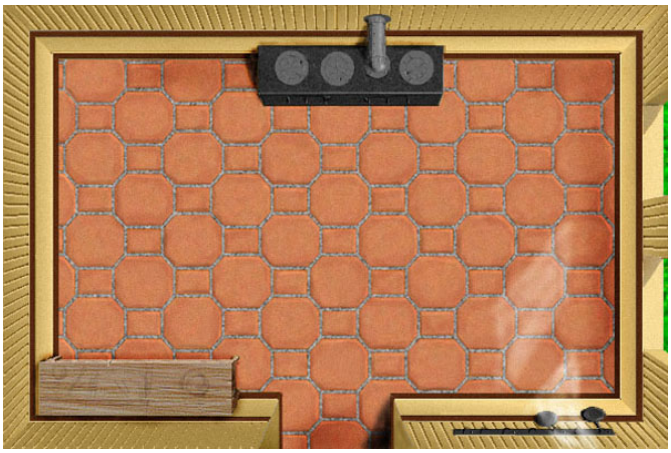
SRS: 3.2.AR.2.3 Dungeon area

- *Area* object with name "dungeon", requiring qualities *stamina*, and *patience*.



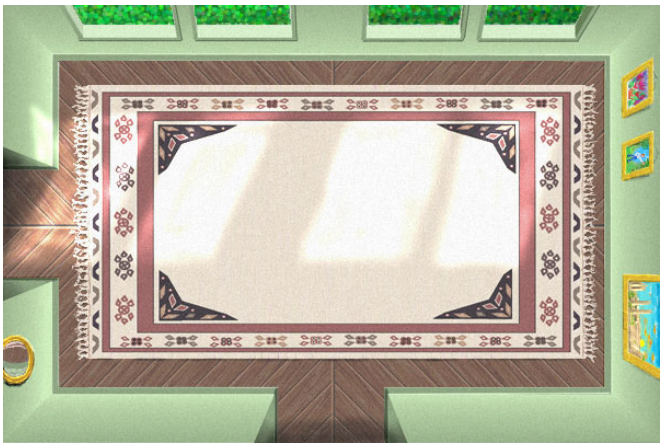
SRS: 3.2.AR.2.4 Kitchen area

- *Area* object named "kitchen", requiring the quality *concentration*.



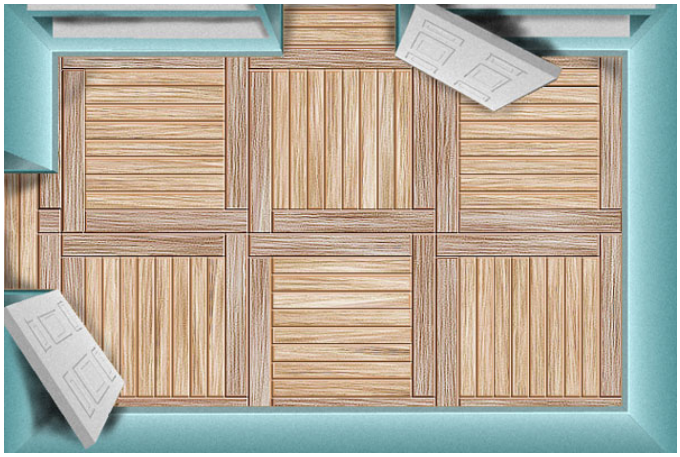
SRS: 3.2.AR.2.5 Living room area

- *Area* object with name "living room", requiring qualities *concentration*, and *stamina*.



SRS: 3.2.AR.2.6 Study area

- *Area* object with name "study", requiring quality *concentration.*



SRS: 3.2.AR.4 Events pertaining to areas

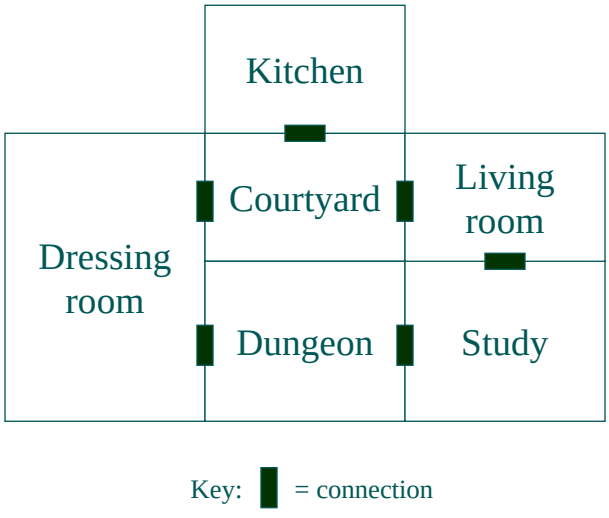
- 3.2.AR.4.1 Display on entry of player character
  - Whenever the player's main character enters an area, that area and the characters in it shall be displayed.
- 3.2.AR.4.2 Handling engagements
  - When a foreign game character enters an area containing the player's main character, or vice versa, they engage each other.
- 3.2.AR.4.3 Display on entry of foreign character
  - Whenever the foreign character enters the area in which the player is present, that area and the characters in it shall be displayed.

SRS: Connections between areas

- 3.2.CH *Connection hyperlinks between areas*
  - Connection hyperlinks are hyperlinks placed at each area exit, showing the area to which it is connected.
- 3.2.CO *Connections between areas*
  - Characters travel from area to adjacent area by means of connections. Each of these connects two areas. Next figure shows the required connections among the areas.
  - Connections are displayed as hyperlinks at the borders of areas whenever the player's character is in the area.
  - When the user clicks such a hyperlink, the linked area is displayed, with the character in this area.

33 Živana Komlenov: The case study 'Encounter'

Encounter Area Configuration (Desirable Requirement)



SRS: 3.2.EC Encounter characters

- 3.2.EC.1 Attributes of Encounter characters
  - 3.2.EC.1.1 Name of Encounter characters
  - 3.2.EC.1.2 Qualities of Encounter characters
    - Every game character has the same set of qualities.
    - Non-negative floating point numbers with at least one decimal of precision.
    - Initialized equally so that the sum of their values is 100.
    - Quality value cannot be both greater than zero and less than 0.5.
    - First release qualities: *concentration*, *intelligence*, *patience*, *stamina*, and *strength*.
  - 3.2.EC.1.3 Image of *Encounter* characters

SRS: 3.2.EC.3 Functionality of *Encounter* characters

- 3.2.EC.3.1 Living points
  - The *Encounter* game shall be able to produce the sum of the values of any character's qualities, called its *living points*.
- 3.2.EC.3.2 Configurability of *Encounter* character quality values
  - Whenever an *Encounter* character is alone in an area, the value of any of its qualities may be set.
  - The value chosen must be less than or equal to the sum of the quality values.
  - The values of the remaining qualities are automatically adjusted so as to maintain their mutual proportions, except for resulting quantities less than one, which are replaced by zero quality values.

SRS: 3.2.ED *Engagement displays*

- There shall be a window displaying the result of engagements.
- The format is shown in the following figure.
- 3.2.ED.4 Engagement display events
  - 3.2.ED.4.1 Dismissing the display
    - When the user hits OK, the display disappears.

User Interface for Showing Status

Elena

Current life points: 56.68

Strength
Endurance
Intelligence
Patience

Value
16.18

SRS: 3.2.EN *Engagements*

- 3.2.EN.3.1 Engaging a foreign character
  - When an engagement takes place, the "stronger" of the two characters, is the one whose values of area-specific qualities sum to the greater amount.
  - The system transfers to the stronger, half the values of each area-specific quality of the weaker. If either character loses all points, the game ends.
- 3.2.EN.4.1 Interrupting engagements
  - Players are able to interrupt engagements on a random basis. On average, the player can stop one of every ten engagements, by executing procedure to set qualities.
  - If the game does not allow this, no indication is given: the game proceeds as if the attempt had not been made.

39 Živana Komlenov: The case study 'Encounter'

SRS: 3.2.FC Foreign characters

- 3.2.FC.2.1 Freddie foreign character
  - There shall be a foreign character named "Freddie". This character shall initially have a total of 100 points, distributed equally among its qualities.
- 3.2.FC.3.1 Foreign character movement
  - As long as it is alive, a foreign character should move from area to adjacent area at random intervals averaging two seconds.
  - After being present in an area for a random amount of time averaging one second, all of the player's life points are divided among the qualities relevant to the area.

40 Živana Komlenov: The case study 'Encounter'

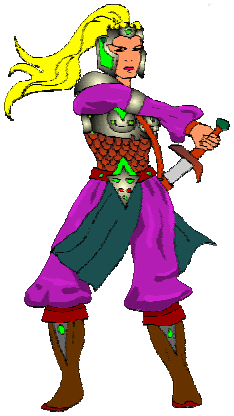
Foreign Character Freddie's Image

Graphics reproduced with permission from Corel.

SRS: 3.2.PC Player characters

- Player character images can be selected from one of the images in the following figure (not yet implemented).
- 3.2.PC.2.1 Player's main character
  - This character shall initially have a total of 100 points, distributed equally among its qualities.
- 3.2.PC.3.1 Configurability of the player character quality values
  - Whenever all foreign players are absent from the area containing the player's main character, the player may set the value of any quality of the character using the *PlayerQualityWindow*.

Player Character Image Options



Elena

Sean

Boris

Graphics reproduced with permission from Corel.

SRS: 3.2.PQ The player quality window

- The value chosen must be less than or equal to the sum of the quality values.
- The values of the remaining qualities are automatically adjusted so as to maintain their mutual proportions
- Resulting quantities less than 0.5 are replaced by zero quality values.
- The window for setting the qualities of a player character in *Encounter* is shown by means of a typical example in the following figure.

User Interface for Setting Quality Values

Sean

Current life points: 100.0

Choose the quality you wish to set

Strength
Endurance
Intelligence
Patience

Image

Choose the value of the quality selected

16.3

Explanation

The values of the qualities not specifically chosen remain in the same proportion to each other. Values less than 0.5 are counted as zero. E.g.,

***before*: strength = 10.0, endurance = 60.0, intelligence = 30.0, patience = 0.0**

(current life points 10.0 + 60.0 + 30.0 + 0.0 = 100.0)

***change*: strength from 10.0 to 20.0**

***after*: strength = 20.0, endurance = 53.3, intelligence = 26.7**

OK

Graphics reproduced with permission from Corel.

Requirements Specification

- Version 1
- Version 2
- Final document based on version 2
- Version differences
- Differences with respect to IEEE Standard

46

Živana Komlenov: The case study 'Encounter'

SRS: Version differences

- Some diagrams differ from their previous versions (due to slightly changed requirements).
- Visual impression is completely different in Version 2, since almost all images are new.

- Other differences are not drastic, but are very significant because they directly show the stages of game creating process. They are the direct consequence of the atypical way of creating this SRS (final version after implementation of the first one).
- All differences are highlighted, explained, with extra notes for students, comparison of certain issues to IEEE standard recommendations, etc.
- Therefore, this material can be very useful to students, since they can see the complexity of writing a proper documentation on a relatively simple project, but still good enough to give them important lessons for similar real-life situations.

47 Živana Komlenov: The case study 'Encounter'

Requirements Specification

- Version 1
- Version 2
- Final document based on version 2
- Version differences
- Differences with respect to IEEE Standard

48 Živana Komlenov: The case study 'Encounter'

SRS: Differences with respect to IEEE Standard [1]

- Encounter SRS complies to the IEEE Std 830-1998 (Revision of IEEE Std 830-1993).
- It modifies the standard by omitting some less important sections and by adding sections on concept of operations and use cases.
- Section *2.1 Product perspective* was partly changed.
  - Encounter is here compared with other related or competing products, which provides perspective on the application.
  - According to the standard, there should be subheading *2.1.1 System interfaces*, listing each system interface and identifying functionality of the software.
  - It has been changed to *2.1.1 Concept of operations* in order to accommodate "concept of operations".

49 Živana Komlenov: The case study 'Encounter'

SRS: Differences with respect to IEEE Standard [2]

  - Requirements developers decided that state/transitions best convey the overall concept of the application.
- Section *3. Specific Requirements* was partly changed. (Encounter SRS uses object-oriented style.)
  - The biggest difference is in section *3.2 Classes/Objects* (this is the correct headline according to the standard). OO style of Encounter SRS expects that detailed requirements are classified by classes.
  - Our section is titled *3.2 Specific requirements*. It takes some liberties with the IEEE standards in order to account for use cases.
  - First it describes sequence diagrams required to express use cases of section 2.2 of the SRS. Classes required to express these use cases are then used to classify detailed requirements.

50 Živana Komlenov: The case study 'Encounter'

Currently existing documentation

- Software Requirements Specification (SRS)
- Software Design Document (SDD)
  - Version 1, Version 2, Final document
  - Version differences
  - Differences with respect to IEEE Standard
- Source Code

Software Design

- Version 1
  - Hal Furness and Karen Peters
- Version 2
  - 6/22/00 Update: Eric Braude and Tom van Court.
  - Reworked to reflect lessons learned by implementing previous version.
- Final document based on version 2
- Version differences
- Differences with respect to IEEE Standard

Software Design Document

- Two designs are described:
  - I. Role-Playing Game Architecture Framework
  - II. Architecture of Encounter Role-Playing Game
- Each of those documents are designed properly, consisting of two main parts covering:
  - Architectural design
  - Concrete detailed design.
- The dependence of Encounter on the framework is specified in the Encounter case study.

Software Design Document

I. Role-Playing Game Architecture Framework
- 1. Introduction
  - 1.1 Purpose
  - 1.2 Scope
  - 1.3 Definitions, acronyms and abbreviations
- 2. References
- 3. Decomposition description
  - 3.1 Module decomposition
  - 3.2 Concurrent process decomposition
- 4.0 Dependency description
- 5.0 Interface description

II. Architecture of Encounter Role-Playing Game
- 1. Introduction

Software Design Document

- 1.1 Purpose
- 1.2 Scope
- 1.3 Definitions, acronyms and abbreviations

- 2. References
- 3. Decomposition description
  - 3.1 Module decomposition (object model)
  - 3.2 Concurrent process decomposition
  - 3.3 Data decomposition
  - 3.4 State model decomposition
  - 3.5 Use case model decomposition
- 4.0 Dependency description
  - 4.1 Inter-module dependencies (object model)
  - 4.2 Inter-process dependencies

55 Živana Komlenov: The case study 'Encounter'

Software Design Document

- 4.3 Data dependencies
- 4.4 State dependencies
- 4.5 Layer dependencies

- 5. Interface Description
  - 5.1 Module interfaces
  - 5.2 Process interface

I. Detailed design of Role-Playing Game Framework

- 6. Detailed design of Role-Playing Game Framework
  - 6.1 Module detailed design

II. Detailed design of Encounter

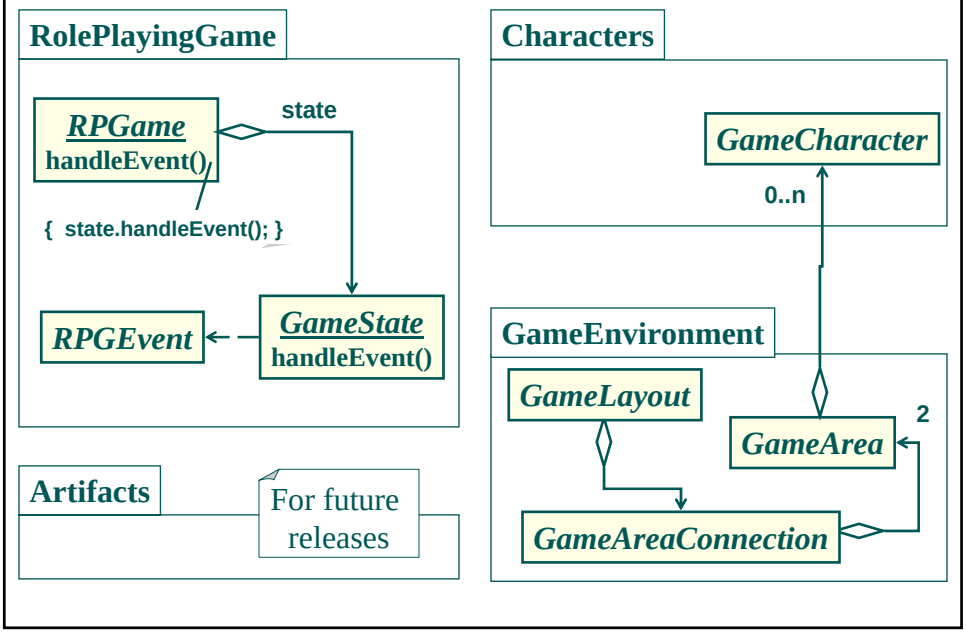
- 6.Detailed design for Encounter
  - 6.1 Module detailed design for Encounter
  - 6.2 Data detailed design

56 Živana Komlenov: The case study 'Encounter'

SDD: I. Role-Playing Game Architecture Framework

- This framework covers essentials of role-playing game classes.
- 3.1 Module decomposition
  - 3.1.1 RolePlayingGame package
    - This package is designed as a state-transition machine. Makes it possible to describe states of the game, and the actions that can take place in response to events.
  - 3.1.2 Characters package
    - Contains GameCharacter class, which describes the characters of the game.
  - 3.1.3 GameEnvironment package
    - Describes the physical environment of the game.

RPG Framework for Role-Playing Video Games

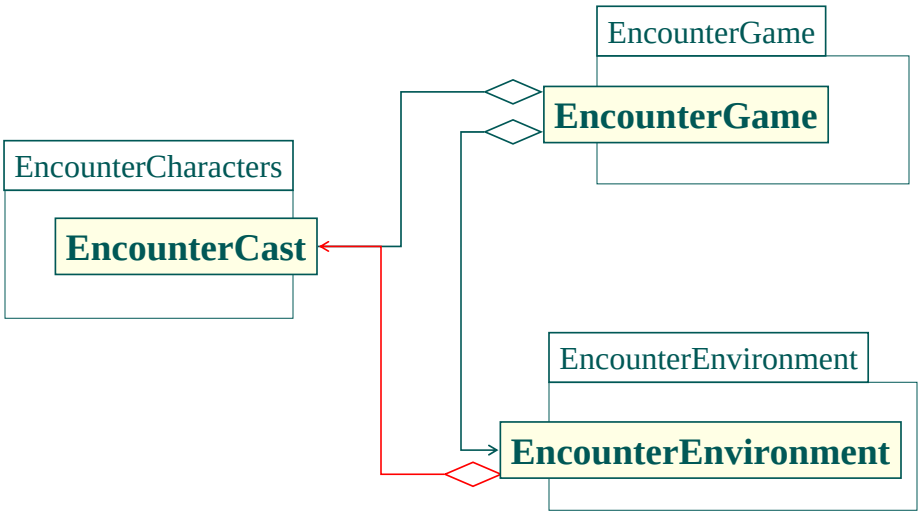


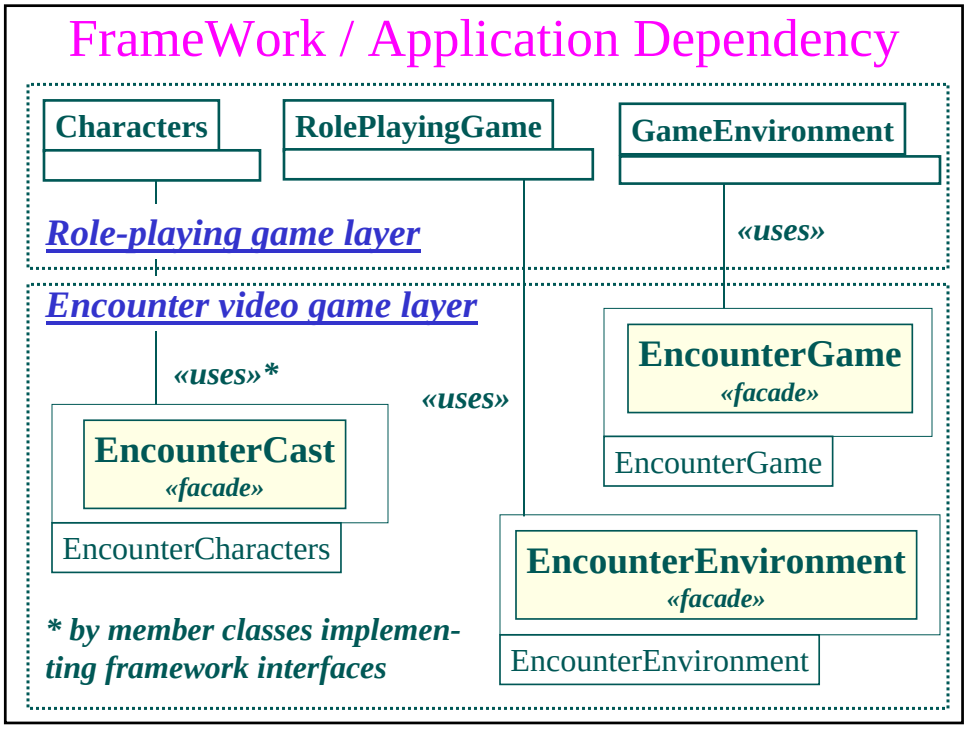
SDD: II. Architecture of Encounter Role-Playing Game

- This document describes the design of the Encounter role-playing game.
- 3. Decomposition description
  - The Encounter application is provided using three models:
    - Use case (already presented in SRS)
    - Class (object) model
    - State (already presented in SRS).
  - In addition, the relationship between the domain packages of Encounter and the framework will be shown.

59 Živana Komlenov: The case study 'Encounter'

Architecture / Modularization





Software Design

- Version 1
- Version 2
- Final document based on version 2
- Version differences
  - Slight differences, too specific to go into details
- Differences with respect to IEEE Standard

Software Design

- Version 1
- Version 2
- Final document based on version 2
- Version differences
- Differences with respect to IEEE Standard
  - Comparison of Encounter SDD to IEEE Std 1016-1998 (Revision of IEEE Std 1016-1987) Recommended Practice for Software Design Descriptions.
  - Almost no differences have been found.

Currently existing documentation

- Software Requirements Specification (SRS)
- Software Design Document (SDD)
- Source Code
  - Reorganized and functional
  - No installation
  - No unnecessary files
  - High-quality comments
  - Module list with descriptions

Summary and future work [1]

- In order for any case study to become a successful part of SE lessons standard tasks have to be fulfilled:
  - ✓ Find a problem of reasonably large size an complexity (for example from textbooks, real projects or educational projects)
  - ✓ Develop requirements specification
  - ✓ Develop a full class diagram as the basis of object-oriented analysis
  - ✓ Develop accompanying diagrams for the dynamic view of object-oriented analysis: state automata (object life cycle), activity diagrams, collaboration diagram, sequence diagram

65 Živana Komlenov: The case study 'Encounter'

Summary and future work [2]

  - – Develop a data-flow diagram for a significant part of requirements
  - – Perform the structured analysis of the system: develop a hierarchy of data flow diagrams for a significant part of the requirements
  - – Do a cost estimation
  - ✓ Implement the case study
  - – Write parts of user manual
- Encounter could (probably) be used as a suitable case study for topics:
  - – Implementation
  - – Testing
  - – Software Metrics...

66 Živana Komlenov: The case study 'Encounter'

Conclusions

- Encounter is recommended to be used as a supplementary case study, not as the main one, since it is smaller (compared to the current main case study) and illustrates just some of the required concepts.
- Therefore, it is very suitable for practical assignments.
- This case study is generally developed well enough to start being used immediately, even as the main case study in shorter SE course version without structural analysis.
- There is no cost estimation, since it is generally difficult to produce one for this type of problems.