

Software Requirements Specification: *Encounter* Video Game

1. Introduction.....	4
1.1 Purpose.....	4
1.2 Scope.....	4
1.3 Definitions, acronyms, & abbreviations	4
1.4 Verification including Testing	5
2. Overall Description.....	5
2.1 Product perspective.....	6
2.1.1 Concept of operations	6
2.1.2 User interface concepts	7
2.1.2.1 <i>Area</i> user interface concept.....	7
2.1.2.2 User interface concept for setting quality values.....	8
2.1.3 Hardware interfaces	9
2.1.4 Software interfaces.....	9
2.1.5 Communications interfaces.....	9
2.1.6 Memory constraints	9
2.1.7 Operations	9
2.2 Product functions	9
2.2.1 "Initialize" use case.....	10
2.2.2 "Travel to adjacent area" use case.....	10
2.2.3 "Encounter foreign character" use case	11
2.3 User characteristics	11
2.4 Constraints	11
2.5 Assumptions and dependencies	11
2.6 Apportioning of requirements.....	11
3. Specific Requirements	13
3.1 External interface requirements	13
3.1.1 User interfaces	13
3.2 Specific requirements.....	14
3.2.1.1 <i>Initialize</i> use case	14
3.2.1.2 Travel to adjacent area use case.....	15
3.2.1.3 Engage foreign character use case	16
3.2.2 Classes for classification of specific requirements	17
3.2.AR <i>Areas</i>	18
3.2.AR.1 Attributes of areas	18
3.2.AR.1.1 <i>Area name</i>	18
3.2.AR.1.2 <i>Area image</i>	19
3.2.AR.1.3 <i>Area-specific qualities</i>	19
3.2.AR.2 Area entities.....	19
3.2.AR.2.1 <i>Courtyard area</i>	19
3.2.AR.2.2 <i>Dressing room area</i>	20
3.2.AR.2.3 <i>Dungeon area</i>	21
3.2.AR.2.4 <i>Kitchen area</i>	21
3.2.AR.2.5 <i>Living room area</i>	21
3.2.AR.2.6 <i>Study area</i>	22
3.2.AR.3 Area functionality.....	23
3.2.AR.4 Events pertaining to areas.....	23

Software Requirements Specification: *Encounter* Video Game

3.2.AR.4.1 Display on entry of player character	23
3.2.AR.4.2 Handling engagements.....	23
3.2.AR.4.7 Display on entry of foreign character.....	23
3.2.CH Connection hyperlinks between areas	23
3.2.CH.1 Attributes of connections hyperlinks.....	23
3.2.CH.1.1 Connection.....	23
3.2.CH.2 Connection hyperlink entities.....	24
3.2.CH.3 Functionality of connection hyperlinks.....	24
3.2.CH.4 Events pertaining to connection hyperlinks	24
3.2.CH.4.1 User clicks on a connection hyperlink	24
3.2.CO Connections between areas	24
3.2.CO.1 Attributes of connections between areas	24
3.2.CO.1.1 First and second areas	25
3.2.CO.2 Connections entities	25
3.2.CO.2.1 Dressing room - courtyard.....	25
3.2.CO.2.2 Dungeon - study.....	25
3.2.CO.2.3 Study - living room	25
3.2.CO.2.4 Courtyard - living room	25
3.2.CO.2.5 Dressing room - dungeon	25
3.2.CO.2.6 Courtyard - kitchen	25
3.2.CO.3 Functionality of area connections.....	25
3.2.CO.4 Events pertaining to area connections	25
3.2.CO.4.1 Moving a character through a connection	25
3.2.EC Encounter characters.....	26
3.2.EC.1 Attributes of Encounter characters	26
3.2.EC.1.1 Name of Encounter characters	26
3.2.EC.1.2 Qualities of Encounter characters.....	26
3.2.EC.1.3 Image of Encounter characters.....	26
3.2.EC.2 Encounter character entities.....	26
3.2.EC.3 Functionality of Encounter characters.....	26
3.2.EC.3.1 Living points	26
3.2.EC.3.2 Configurability of Encounter character quality values	27
3.2.ED Engagement displays	27
3.2.ED.4 Engagement display events	27
3.2.ED.4.1 Dismissing the display.....	27
3.2.EG The Encounter game.....	27
3.2.EG.1 Attributes of the Encounter game.....	28
3.2.EG.1.1 Duration.....	28
3.2.EG.1.2 Action buttons	28
3.2.EG.2 Entities of the Encounter game	28
3.2.EG.2.1 Single game.....	28
3.2.EG.4 Events the Encounter game	28
3.2.EG.4.4 Pressing the Set qualities button	28
3.2.EG.4.5 Pressing the End game button.....	28
3.2.EG.4.6 Pressing the Get status button.....	28
3.2.EN Engagements.....	29

Software Requirements Specification: *Encounter* Video Game

3.2.EN.1 Attributes of engagements	29
3.2.EN.2 Engagement entities.....	29
3.2.EN.3 Functionality of engagements.....	29
3.2.EN.3.1 <i>Engaging a foreign character</i>	29
3.2.EN.4 Events on engagements	30
3.2.EN.4.1 <i>Interrupting engagements</i>	30
3.2.FC Foreign characters	30
3.2.FC.1 Attributes of foreign characters.....	30
3.2.FC.2 Foreign character entities	30
3.2.FC.2.1 <i>Freddie foreign character</i>	30
3.2.FC.3 Functionality of foreign characters	31
3.2.FC.3.1 <i>Foreign character movement</i>	31
3.2.PC Player characters	31
3.2.PC.1 Attributes of player characters	31
3.2.PC.2 Player character entities	32
3.2.PC.2.1 <i>Player's main character</i>	32
3.2.PC.2.2 <i>Additional characters under the control of the player</i>	32
3.2.PC.3 Player character functionality	32
3.2.PC.3.1 <i>Configurability of the player character quality values</i>	32
3.2.PC.3.2 <i>Configurability of the player character images</i>	33
3.2.PC.3.3 <i>Aging of the player character images</i>	33
3.2.PQ The player quality window	33
3.2.PQ.1 Attributes of the player quality window	33
3.2.PQ.2 Player quality window entity	34
3.2.PQ.2.1 <i>Window for allocating qualities</i>	34
3.2.PQ.3 Player quality functionality.....	35
3.2.PQ.3.1 <i>Initiating the display</i>	35
3.2.PQ.4 Player quality window events	35
3.2.PQ.4.1 <i>Displaying the value of a quality</i>	35
3.2.PQ.4.2 <i>Setting the value of a quality</i>	35
3.2.PQ.4.3 <i>Dismissing the window</i>	35
3.2.PQ.4.4 <i>Interruption</i>	35
3.2.TH Thumbnail sketch of playing areas	35
3.2.TH.1 Attributes of the thumbnail sketch.....	35
3.2.TH.1.1 <i>Thumbnail image</i>	36
3.3 Performance requirements	36
3.4 Design constraints.....	36
3.5 Software system attributes	36
3.5.1 Reliability.....	37
3.5.2 Availability	37
3.5.3 Security	37
3.5.4 Maintainability	37
3.5.4.1 Changing characters and areas.....	37
3.5.4.2 Globally altering styles	37
3.5.4.3 Altering rules of engagement.....	37

1. Introduction

1.1 Purpose

This document provides all of the requirements for the *Encounter* video game. Parts one and two ["C-requirements"] are intended primarily for customers of the application, but will also be of interest to software engineers building or maintaining the software. Part three ["D-requirements"] is intended primarily for software engineers, but will also be of interest to customers.

1.2 Scope

This document covers the requirements for CA314 Iteration 1 and 2 releases of *Encounter*. Mention will be made throughout this document of selected probable features of future releases. The purpose of this is to guide developers in selecting a design that will be able to accommodate the full-scale application.

1.3 Definitions, acronyms, & abbreviations

Acronym or term	Definition
Alive	A game character is said to be "alive" if it has at least one quality with non-zero value.
C-requirement	Statement of the requirements for the application, expressed in a form clear to the customer.
D-requirement	Statement of the requirements for the application, given in a form detailed enough to be used by the developers for design and implementation. If possible, D-requirements should also be understandable to the customer.
<i>Encounter</i>	Name of this application; also, a meeting between two game characters in an area (but not necessarily an "engagement" -- see below)
Engagement	An interaction between characters of the game, which typically affect the characters
RPG	"Role-playing game": a game typically played on a computer in

Software Requirements Specification: *Encounter* Video Game

	which the players adopt character roles
Role-playing game	See RPG
Video game	A game played on a computer

1.4 Verification including Testing

It is a major part of the CA314 assignment that all the specified requirements be verified, mostly by testing (unit, integration, system) but also possibly by demonstration or analysis. In this document it is indicated, for some requirements, how and in what iteration they should be verified. However, if this is not indicated for a requirement it is up to each project team to decide when and how that requirement should be verified. Further specific requirements on what is to be submitted for iterations 1 and 2 are detailed in the "CA314 Statement of Work".

2. Overall Description

Encounter is to be a role-playing game which simulates all or part of the lifetime of the player's main character. It should be of interest to both men and women. The measure of "success" in playing *Encounter* is up to the player. Typically, success will be measured by the "life points" maximum attained by the player or by the ability of the player to live as long a life as possible.

Some game characters are to be under the control of the player. The rest, called "foreign" characters, are to be under the application's control. Game characters will have a fixed total number of points allocated among qualities such as *strength*, *stamina*, *patience* etc. Characters encounter each other when they are in the same area at the same time, and may then engage each other. The result of the engagement depends on the values of their qualities and on the environment in which the engagement takes place. Engagements are not necessarily violent or adversarial. Players have restricted opportunities to reallocate their qualities. One of the player-controlled characters will be referred to as the "main" player character.

Software Requirements Specification: *Encounter* Video Game

In early versions of this game, and particularly in Iteration 1 of the CA314 continuous assessment project, there will be only one player-controlled character, and one foreign character.

Encounter should eventually be highly customizable, so that users can either start with predefined games, substitute pre-designed characters and rules of engagement, or devise their own characters and rules of engagement.

The design should support expansion into a family of games, including Internet-based multiple player versions.

2.1 Product perspective

Encounter is intended to fulfill the need for gamers to have a greater influence over the contents of video games with and without programming. It is also intended for a somewhat mature clientele. *Encounter* is intended to appeal to both genders. The design and documentation for *Encounter* will make it convenient to expand and modify the game.

2.1.1 Concept of operations

Encounter can be in one of several states, as shown in Figure 2.1.1-1,

- *Setting up*: the state in which the game is being set up by the player
- *Reporting*: the system is displaying a window showing the status of the player's character(s)
- *Setting qualities*: equipping the player's character with qualities. This process consumes mandatory amounts of time, and can be performed as long as no foreign character is present.
- *Engaging*: the state which applies whenever a foreign character and the player's main character are both present in an area at the same time
- *Waiting*: The player and the foreign character(s) are not active

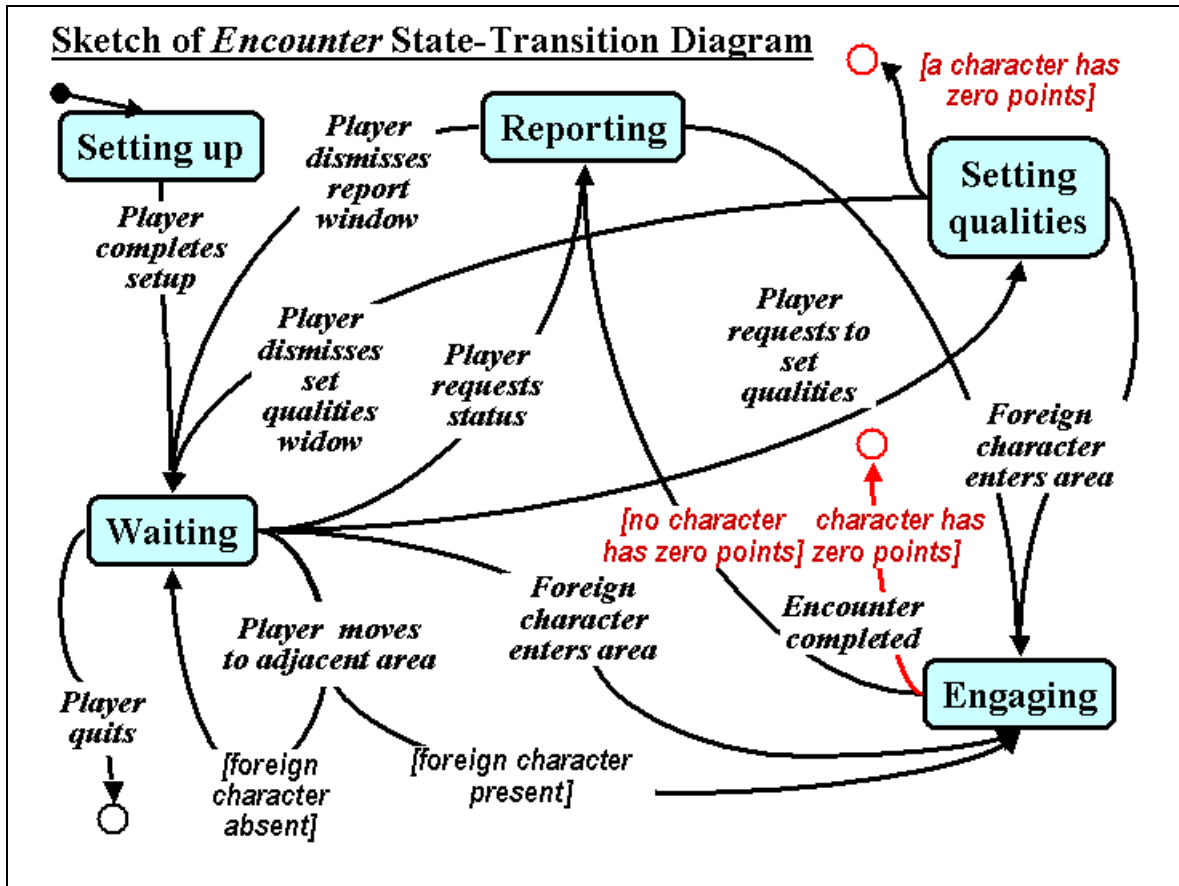


Figure 2.1.1-1: *Encounter* State-Transition Diagram

Testing task (Iteration 2): This state/transition shall be tested by an integration test.

2.1.2 User interface concepts

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 Figure 2.1.2.1-1:

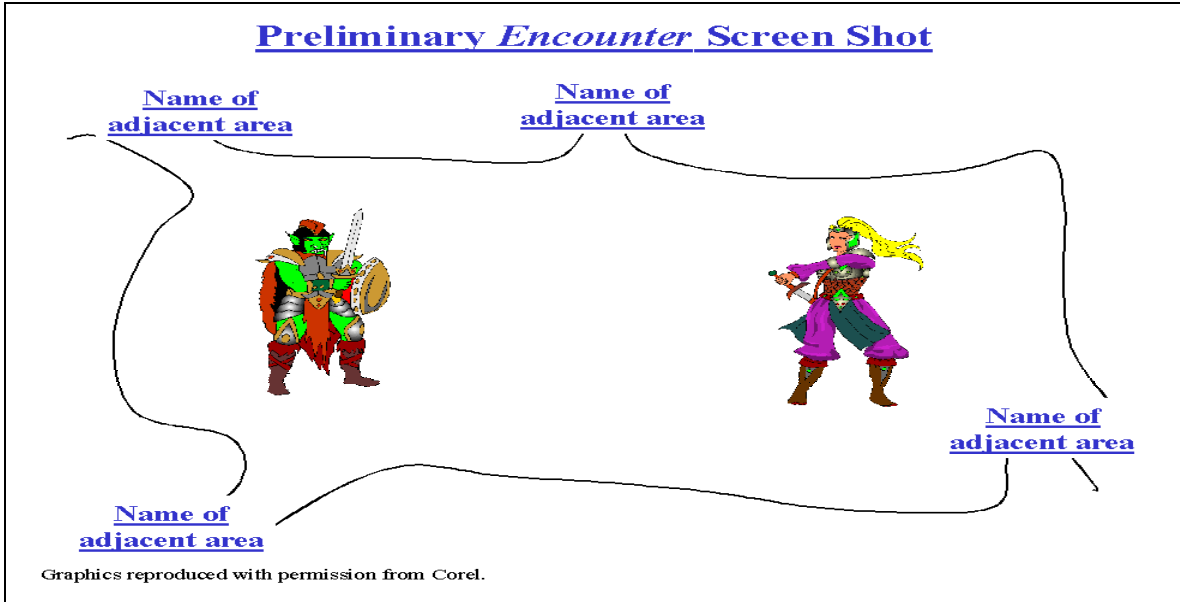


Figure 2.1.2.1-1: Preliminary *Encounter* Screen Shot

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 Figure 2.1.2.2-1. The scroll box is used to identify the quality to be set, and the text box is used for setting the value.

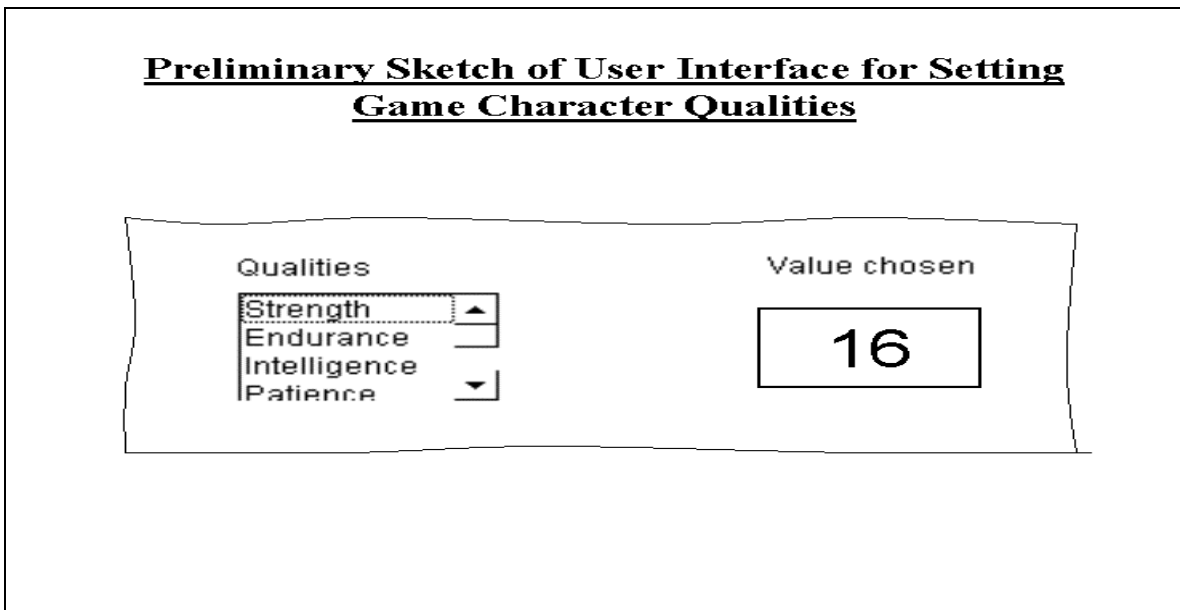


Figure 2.1.2.2-1: Preliminary Sketch of User Interface for Setting Game Character Qualities

Software Requirements Specification: *Encounter* Video Game

2.1.3 Hardware interfaces

Future releases will utilize a joystick – not applicable for CA314 assignment.

2.1.4 Software interfaces

Java virtual machine capable of executing Java 1.1 or higher.

Vetification task (non-test) (Iteration 1): Compliance with this shall be stated in the system documentation.

2.1.5 Communications interfaces

Iteration 1 version may be for a standalone machine but the Iteration 2 version should allow the game to be played remotely over the internet.

Vetification task (Iteration 2): Compliance with this requirement shall be demonstrated at system test level.

2.1.6 Memory constraints

Encounter shall not have specific memory constraints but its documentation should state how many MB of RAM and of secondary storage it requires for execution.

Vetification task (non-test) (Iteration 1): Compliance with this shall be verified by examination of the relevant part of the documentation.

2.1.7 Operations

[**Iteration 2**] It shall be possible to save and retrieve a game.

Vetification task (Iteration 2): Compliance with this requirement shall be demonstrated at system test level.

2.2 Product functions

This section specifies the required overall functionality of the application, but is not intended to provide the complete specifications. It was decided that use cases are an appropriate manner in which to specify the major overall functionality. Section 3 provides the requirements in complete detail.

Software Requirements Specification: *Encounter* Video Game

2.2.1 "Initialize" use case

Actor: player of *Encounter*

Use case: Figure 2.2.1-1 (use case diagram) gives the text of the *Initialize* use case. The use case is shown in context with the *Encounter foreign character* use case and the *Set rules* use cases. *Initialize* is the typical sequence users execute at the beginning of a session.

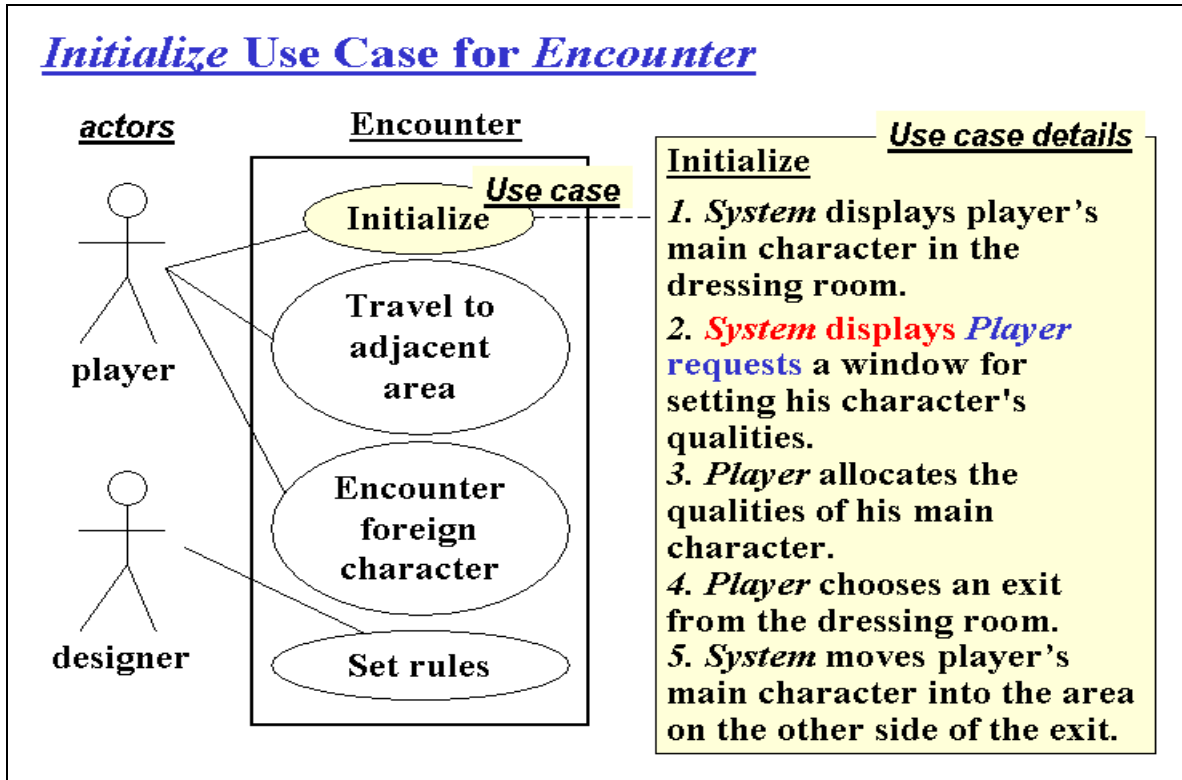


Figure 2.2.1-1: Initialize Use Case for *Encounter*

Testing task (Iteration 1): This use case shall be verified by testing at system level.

2.2.2 "Travel to adjacent area" use case

Actor: player of *Encounter*

Use case:

1. Player hits hyperlink connecting displayed area to adjacent area
2. System displays the indicated adjacent area containing player's character

Testing task (Iteration 1): This use case shall be verified by testing at system level.

Software Requirements Specification: *Encounter* Video Game

2.2.3 "Encounter foreign character" use case

Actor: player of *Encounter*

Use case:

1. *System* moves a foreign game character into the area occupied by the player.

Or

Player moves into an area containing a foreign character.

2. *System* causes the two characters to engage

3. *System* displays the result of the engagement

4. If either the player's character or the foreign character has no points, the game terminates, otherwise

5. *System* moves the player's character to a random area different from that in which the encounter took place, and displays it there.

Testing task (Iteration 1): This use case shall be verified by testing at system level.

2.3 User characteristics

Not specified for CA314 assignment

2.4 Constraints

Java shall be the implementation language.

2.5 Assumptions and dependencies

None

2.6 Apportioning of requirements

The requirements described in sections 1 and 2 of this document, are referred to as "C-requirements"; those in section 3 are referred to as "D-requirements". The primary audience for C-requirements is the customer community, and the secondary audience is the developer community. The reverse is true for the D-requirements. These two levels of requirements are intended to be consistent. Inconsistencies are to be logged as defects. In

Software Requirements Specification: *Encounter* Video Game

the event a requirement is stated within both the C-requirements and the D-requirements, the application shall be built from the D-requirement version, since it is more detailed.

Each specific requirement is identified as [**Iteration 1**], [**Iteration 2**] or [**Desirable**] meaning that it shall be implemented in either Iteration 1 or Iteration 2, or that it is a desirable but not essential requirement.

3. Specific Requirements

3.1 External interface requirements

3.1.1 User interfaces

Encounter takes place in areas. Figure 3.1.1-1 shows a typical screen shot of the *courtyard* area. In addition, though not depicted here, a player-controlled character and the results of an engagement should appear.

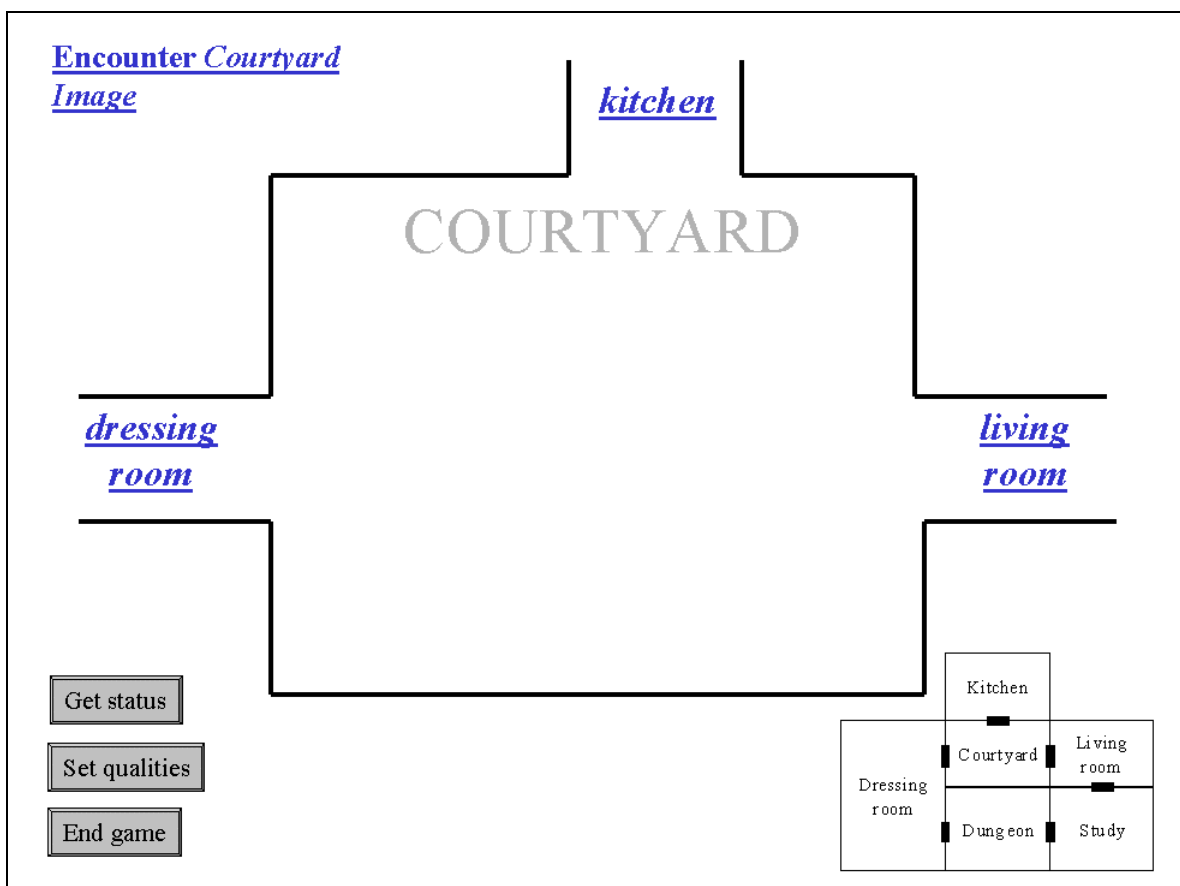


Figure 4.54 *Encounter Courtyard* Image (excluding Game Characters)

Note that each part of this figure is specified separately below. This interface takes up the entire window. Areas have connections to adjacent areas, labeled by hyperlinks. Clicking on one of these hyperlinks moves the player's character into the corresponding area.

The entire set of interfaces is as follows:

- a. One image for each area, specified in section 3.2AR below.

Software Requirements Specification: *Encounter* Video Game

- b. A user interface for setting the quality values of the player's character, specified in section 3.2.PQ.
- c. A user interface for displaying the results of an engagement, specified in section 3.2.ED. The same user interface is used to show the status of the player's character.
- d. A window announcing the start of an engagement.

An interface of type *a* above will always be present on the monitor. When called for by these requirements, interfaces of type *b* or *c* will be superimposed.

Testing task (Iteration 1): This “user interface” requirement shall be verified by testing at system level.

3.2 Specific requirements

Notes for CA314:

1. In this project, the requirement specification is written in terms of objects. Therefore the specification has already identified some of the classes for the system. However, there is still plenty of work to be done!

The section first describes the sequence diagrams required to express the use cases of section 2.2 (section 3.2.1).; the test requirements for the use cases are already stated The classes required to express these use cases are then used to classify the detailed requirements (section 3.2.2).

2. There may be some departures from the UML notation defined in lectures. This is inevitable given the preferences of different authors.

3.2.1 Sequence diagrams

3.2.1.1 *Initialize* use case

The sequence diagram for the *Initialize* use case is shown in Figure 3.2.1.1-1:

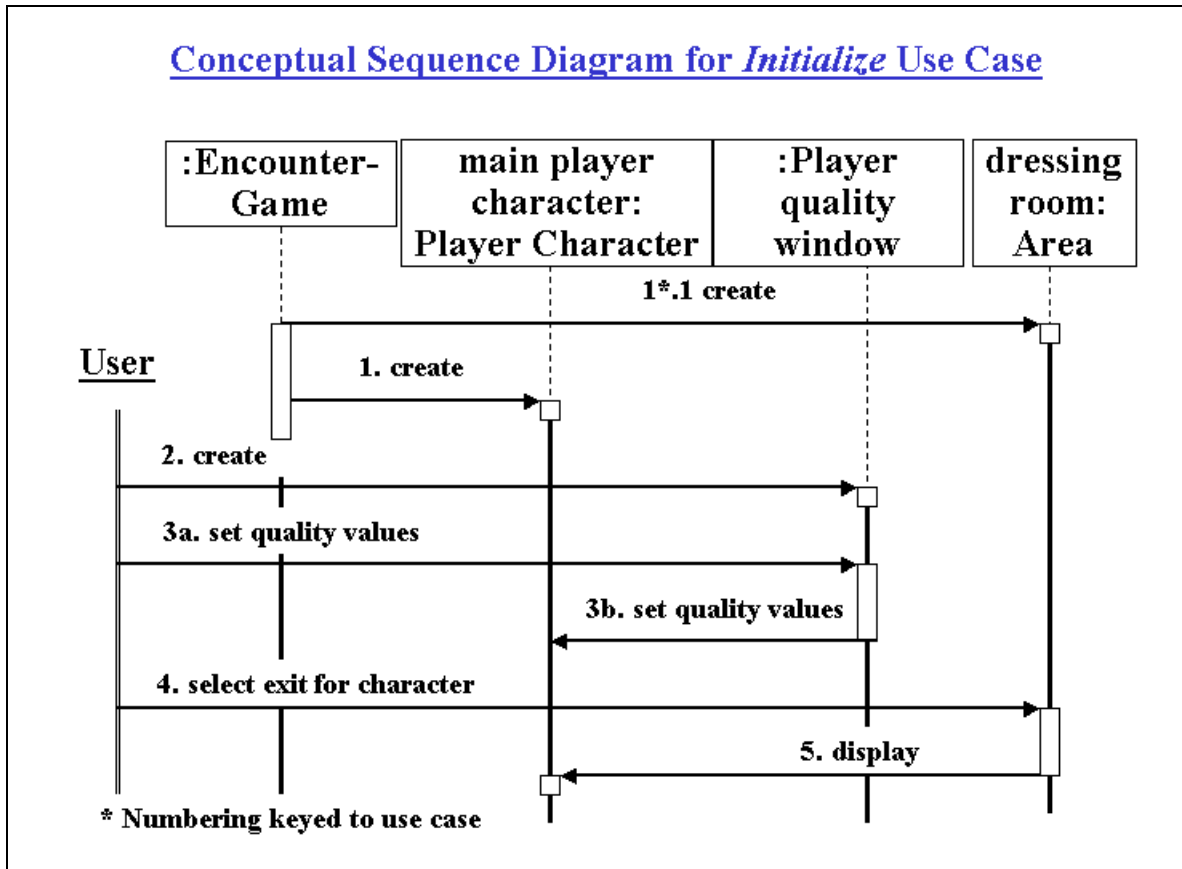


Figure 3.2.1.1-1: Sequence Diagram for *Initialize* Use Case

This use case requires classes *EncounterGame* (of which there is only one instance), *PlayerCharacter* (with instance main player character), *PlayerQualityWindow* (of which there is only one instance), and *Area* (with instance dressing room).

3.2.1.2 Travel to adjacent area use case

The sequence diagram for the *Travel to adjacent area* use case is shown in Figure 3.2.1.2-1:

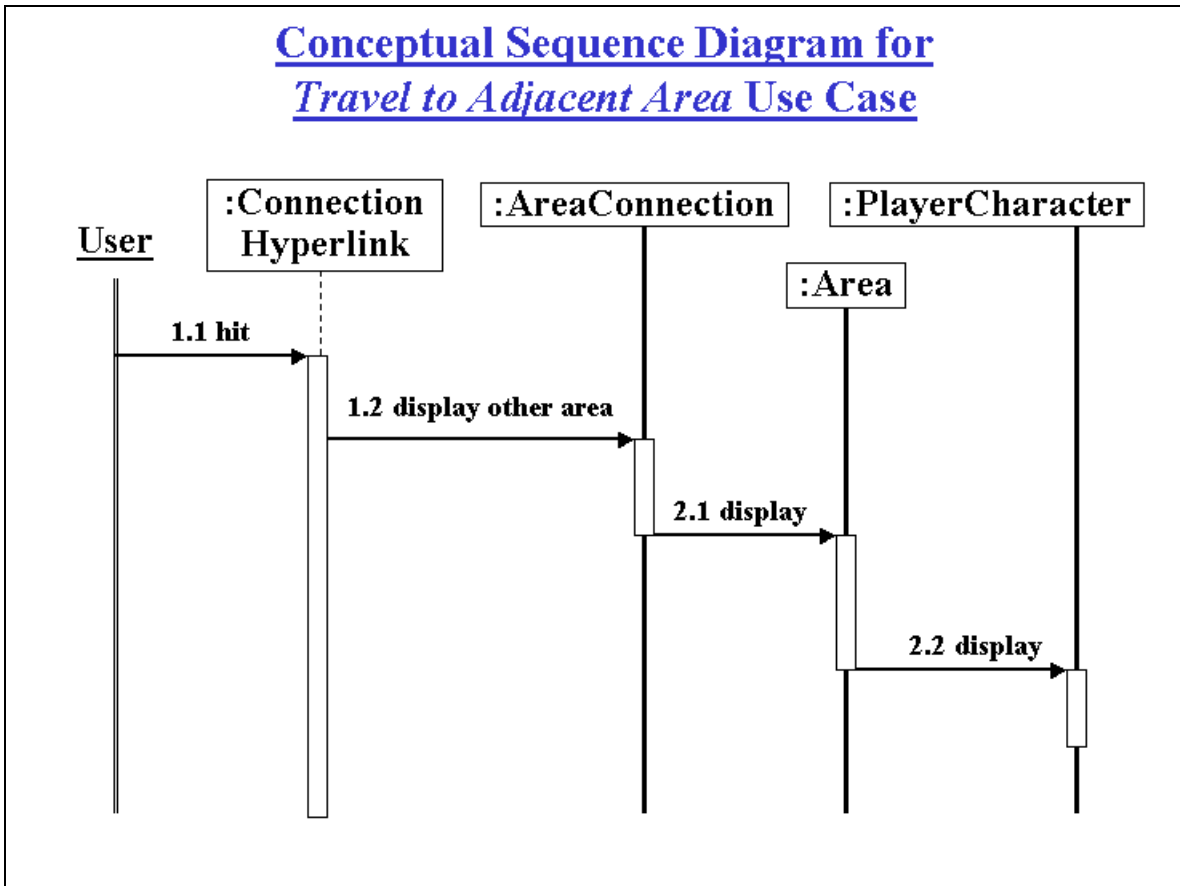


Figure 3.2.1.2-1: Sequence Diagram for Tavel to Adjacent Area Use Case

This use case requires classes *ConnectionHyperlink*, *AreaConnection*, *Area*, and *PlayerCharacter*.

3.2.1.3 Engage foreign character use case

The sequence diagram for the *Engage foreign character* use case is shown in Figure 3.2.1.3-1:

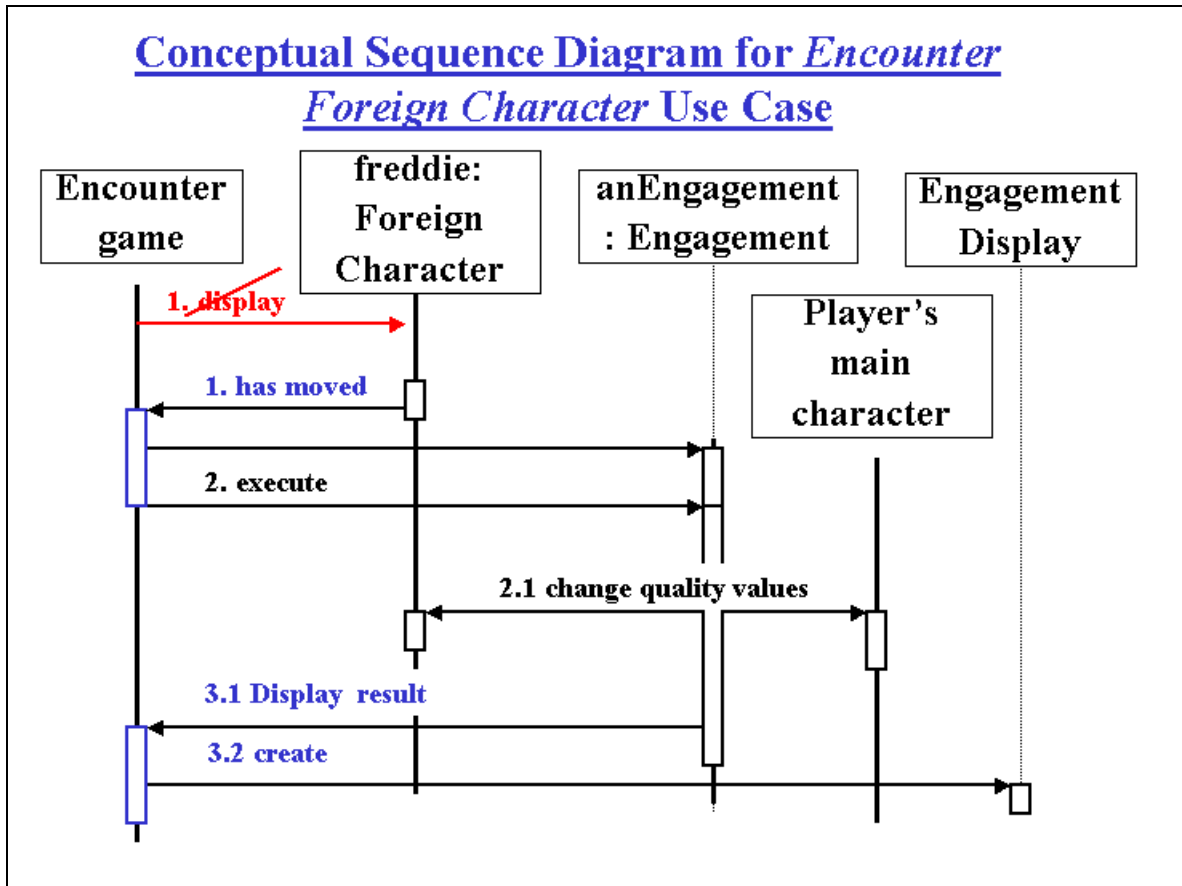


Figure 3.2.1.3-1: Sequence Diagram for *Encounter Foreign Character Use Case*

This use case requires classes *EncounterGame* (of which there is only one instance), *ForeignCharacter* (with instance *freddie*), *Engagement*, *PlayerCharacter*, *PlayerQualityWindow*, and *EngagementDisplay*.

3.2.2 Classes for classification of specific requirements

The classes for the *Encounter* video game sufficient for expressing the requirements are *Area*, *EncounterCharacter*, *EncounterGame*, *Engagement*, *EngagementDisplay*, *ForeignCharacter*, *PlayerCharacter*, *PlayerQualityWindow*, and *ThumbnailMap*. These are shown in the object model of Figure 3.2.2-1. It is emphasized that these are unlikely to be all the classes used by the application.

Software Requirements Specification: *Encounter* Video Game

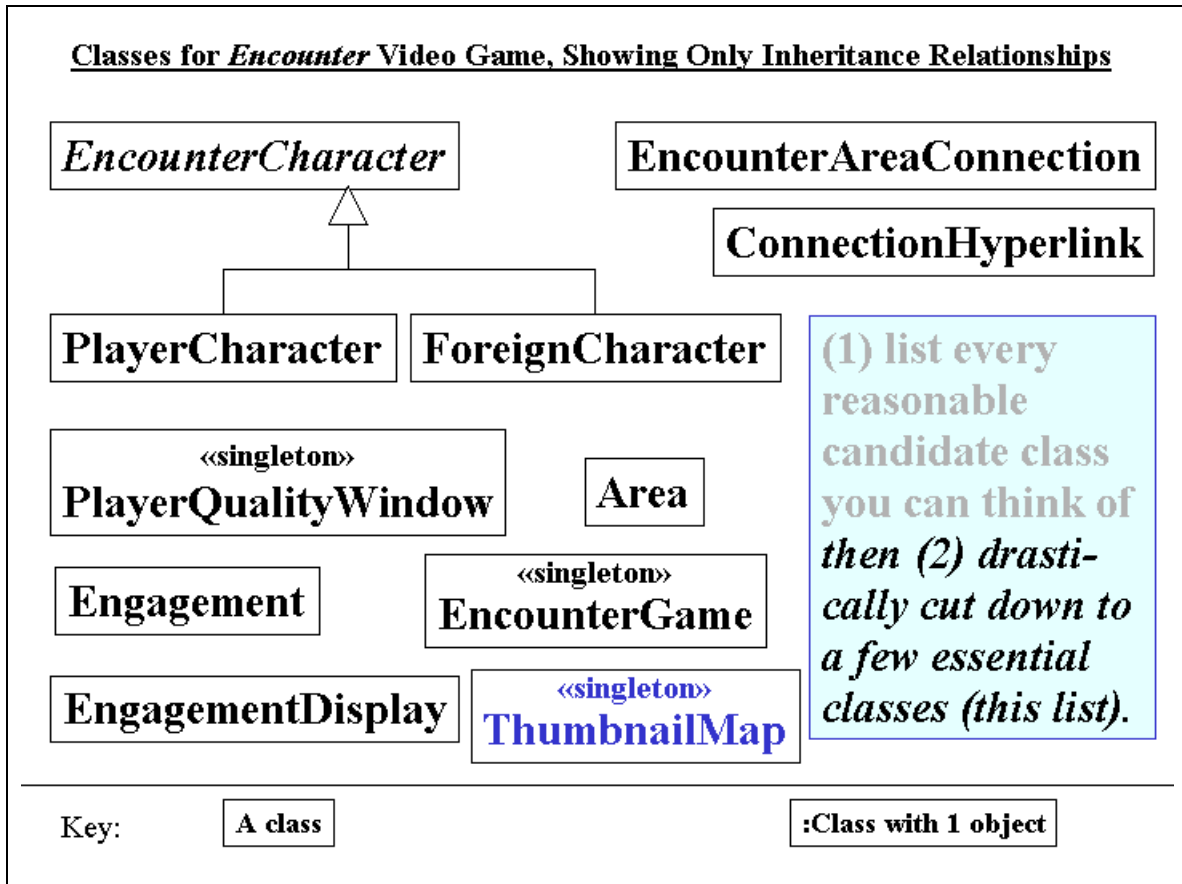


Figure 3.2.2-1: Classes for *Encounter* Video Game, Showing Only Inheritance Relationship

Note: In the following, the sub-section numbering used "3.2.Area.N.N..." etc. is intended to make it easier to insert, remove, and locate requirements by organizing alphabetically the classes that contain them.

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

Software Requirements Specification: *Encounter* Video Game

[**Iteration 1**] Every area shall have a unique case-insensitive name consisting of 1 to 15 characters. Acceptable characters shall consist of any digits and characters as specified by the ISO Unicode standard.

Testing task: This requirement shall be verified by unit test.

Note: Each attribute-type requirement maps to a pair of *get*- and *set*- functions.

3.2.AR.1.2 Area image

[**Iteration 1**] There shall be an image to display each *Area* object on the part of the window not occupied by the required buttons or the thumbnail map.

Verification task: Project team to decide how to verify this requirement.

3.2.AR.1.3 Area-specific qualities

[**Iteration 1**] Only some game character qualities shall be applicable in each area. The specific qualities required for each area are specified in section 3.2.AR.2.

Verification task: Project team to decide how to verify these requirements.

3.2.AR.2 Area entities

3.2.AR.2.1 Courtyard area

[**Iteration 1**] There shall be an *Area* object with name "courtyard", requiring qualities *stamina*, and *strength*. The courtyard image is shown in Figure 3.2.AR.2-1:

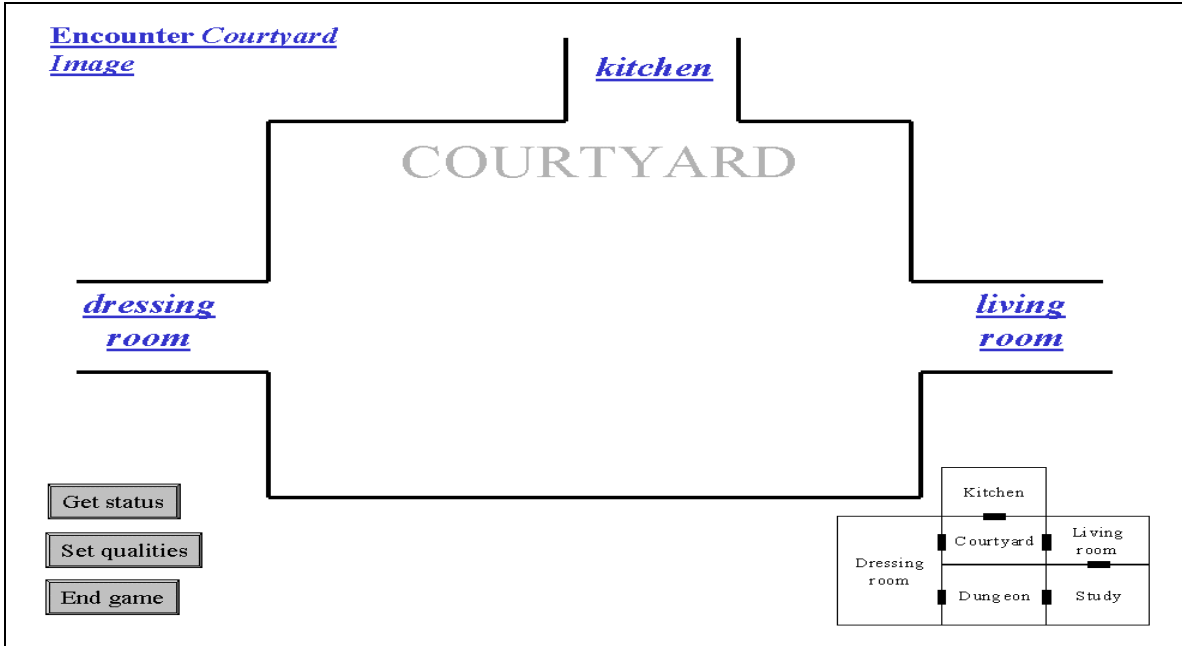


Figure 3.2.AR.2-1: Encounter Courtyard Image

3.2.AR.2.2 Dressing room area

[Iteration 1] There shall be an area with name "dressing room", requiring no qualities.
The image is shown in the Figure 3.2.AR.2-2:

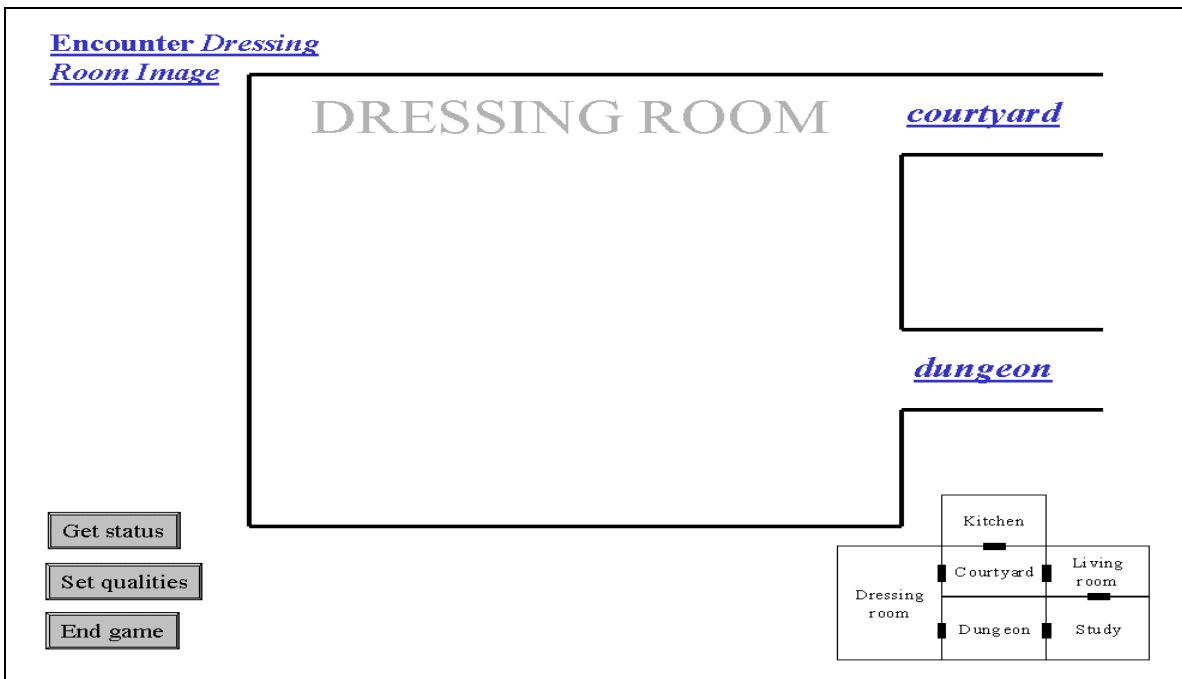


Figure 3.2.AR.2-2: Encounter Dressing Room Image

Software Requirements Specification: *Encounter Video Game*

3.2.AR.2.3 Dungeon area

[Iteration 1] There shall be an area with name "dungeon", requiring qualities *stamina*, and *patience*. Its image is shown in Figure 3.2.AR.2-3:

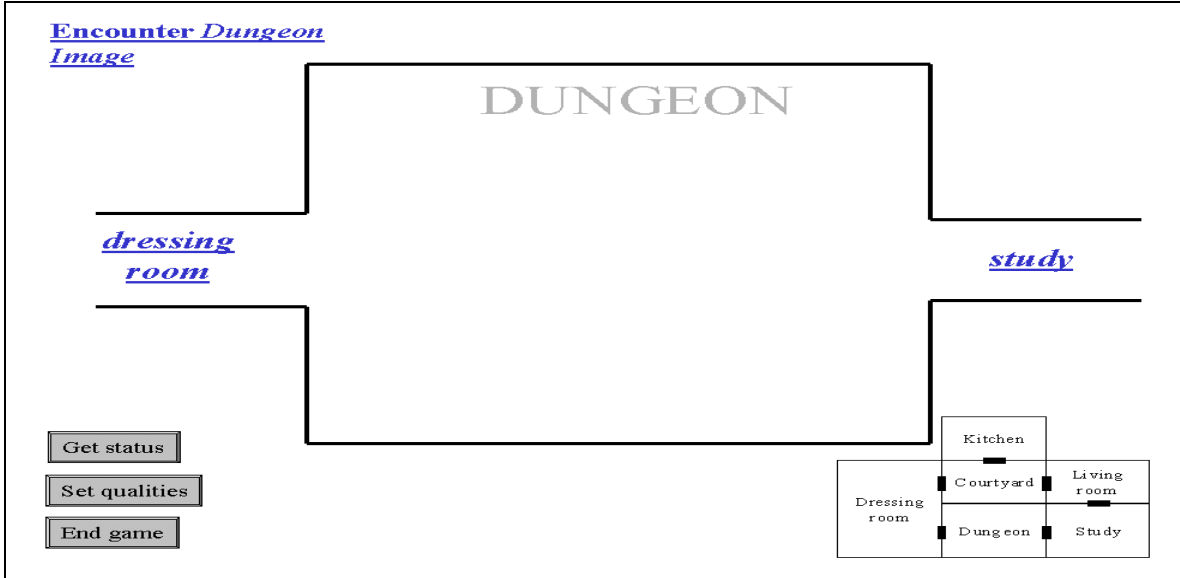


Figure 3.2.AR.2-3: *Encounter Dungeon Image*

3.2.AR.2.4 Kitchen area

[Iteration 1] There shall be an area with name "kitchen", requiring the quality *concentration*. The kitchen image is shown in Figure 3.2.AR.2-4:

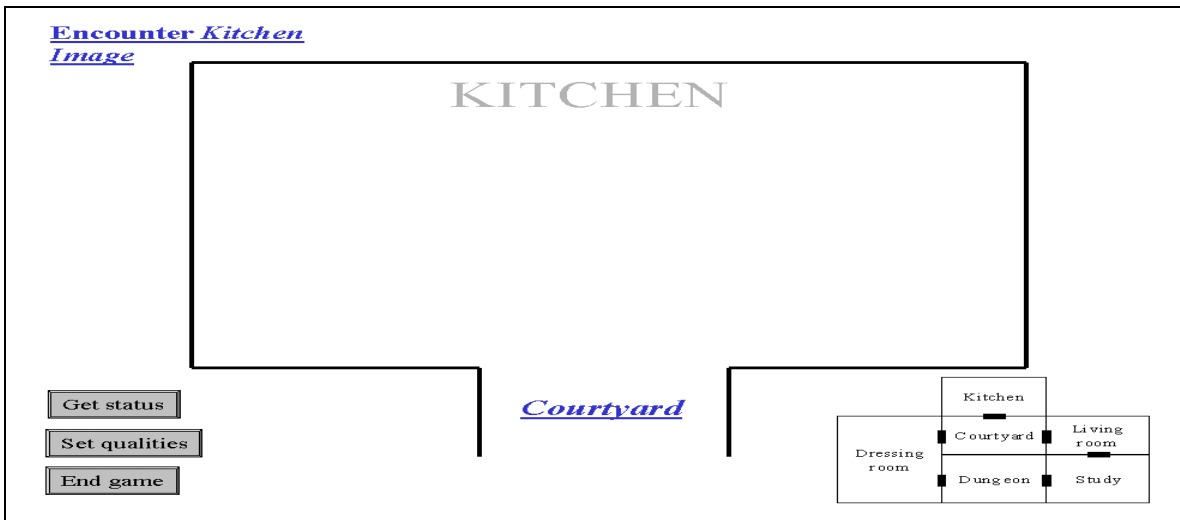


Figure 3.2.AR.2-4: *Encounter Kitchen Image*

3.2.AR.2.5 Living room area

Software Requirements Specification: *Encounter* Video Game

[Iteration 1] There shall be an area with name "living room", requiring qualities *concentration*, and *stamina*. Its image is shown in Figure 3.2.AR.2-5:

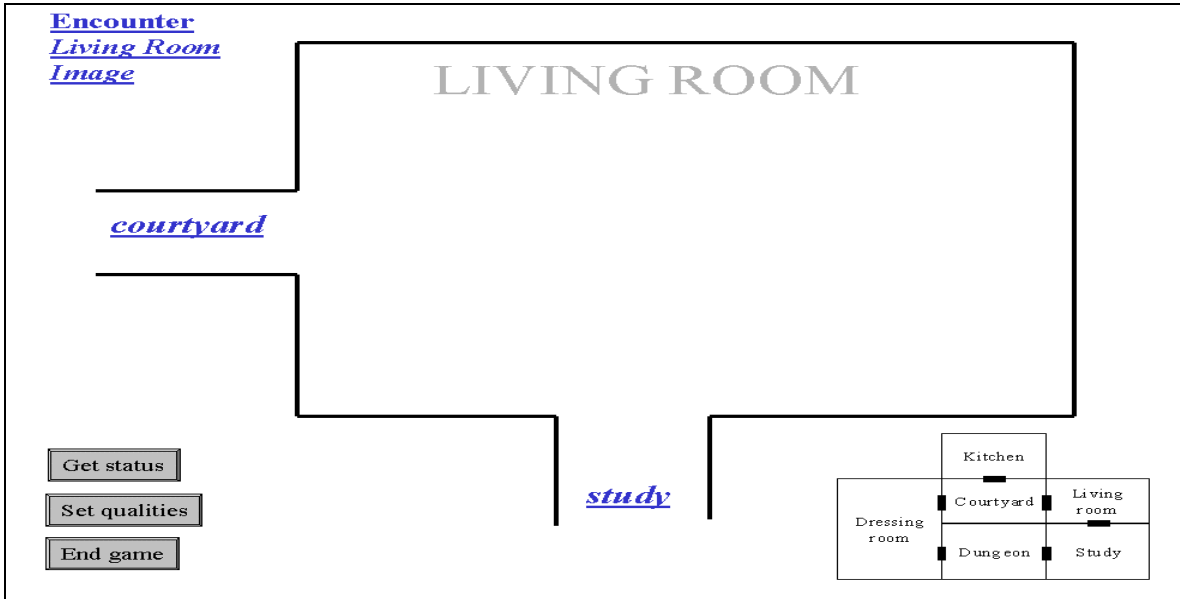


Figure 3.2.AR.2-5: Encounter Living Room Image

3.2.AR.2.6 Study area

[Iteration 1] There shall be an area with name "study", requiring quality *concentration*. Its image is shown in Figure 3.2.AR.2-6:

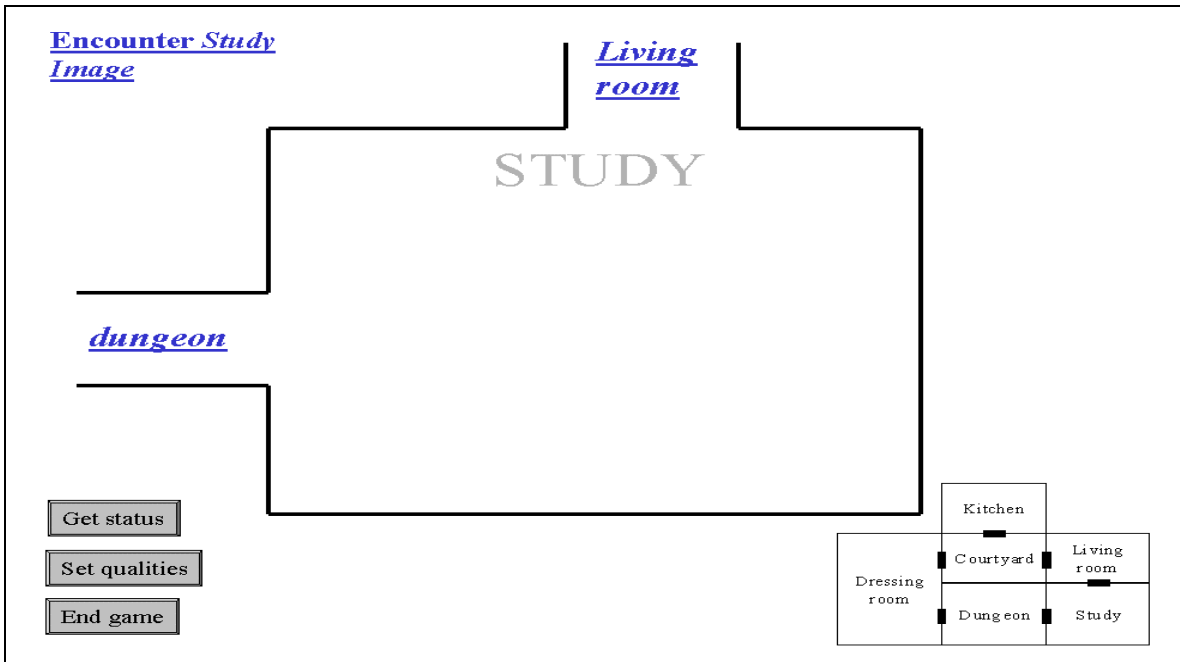


Figure 3.2.AR.2-6: Encounter Study Image

Software Requirements Specification: *Encounter* Video Game

3.2.AR.3 Area functionality

none

3.2.AR.4 Events pertaining to areas

Note: An event is an action that occurs to the application, and which is instigated from outside of it.

Verification task: Project team to decide how to verify these requirements (by test).

3.2.AR.4.1 Display on entry of player character

[**Iteration 1**] Whenever the player's main character enters an area, that area and the characters in it shall be displayed on the monitor, filling the monitor.

3.2.AR.4.2 Handling engagements

[**Iteration 1**] When a foreign game character enters an area containing the player's main character, or vice versa, they engage each other.

Note: Contents of Revision 1 sections 3.2.AR.4.3-6 either deleted or moved. This note is included just to show that it can often be preferable not to re-number sections when something is deleted in order to avoid cross-referencing problems.

3.2.AR.4.7 Display on entry of foreign character

[**Iteration 1**] Whenever the foreign character enters the area in which the player is present, that area and the characters in it shall be displayed.

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.CH.1 Attributes of connections hyperlinks

3.2.CH.1.1 Connection

[**Iteration 1**] Each connection hyperlink corresponds to an area connection.

Software Requirements Specification: *Encounter* Video Game

3.2.CH.2 Connection hyperlink entities

[Iteration 1] There are two connection hyperlinks corresponding to each area connection, one in each area of the connection.

3.2.CH.3 Functionality of connection hyperlinks

none

3.2.CH.4 Events pertaining to connection hyperlinks

3.2.CH.4.1 User clicks on a connection hyperlink

The effect of clicking a connection hyperlink is that the player's character is displayed in the area on the other side of the area connection.

3.2.CO Connections between areas

Characters travel from area to adjacent area by means of connections. Each of these connects two areas. Figure 3.2.CO-1 shows the required connections among the areas.

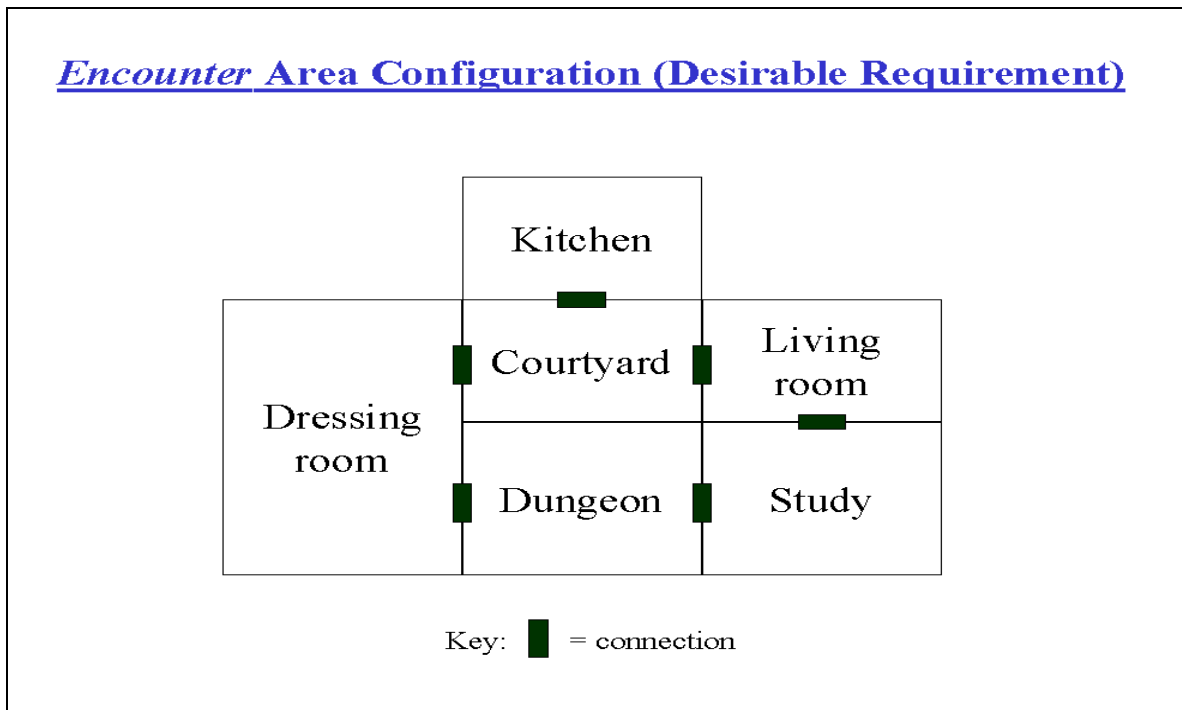


Figure 3.2.CO-1: *Encounter* Area Configuration (Desirable Requirement)

3.2.CO.1 Attributes of connections between areas

Software Requirements Specification: *Encounter* Video Game

3.2.CO.1.1 First and second areas

[**Iteration 1**] Each connection will connect a pair of areas, which we will call the "first" and "second".

3.2.CO.2 Connections entities

3.2.CO.2.1 Dressing room - courtyard

[**Iteration 1**] There will be a connection between the dressing room and the courtyard.

3.2.CO.2.2 Dungeon - study

[**Iteration 1**] There will be a connection between the dungeon and the courtyard.

3.2.CO.2.3 Study - living room

[**Iteration 1**] There will be a connection between the dungeon and the living room.

3.2.CO.2.4 Courtyard - living room

[**Iteration 1**] There will be a connection between the living room and the courtyard.

3.2.CO.2.5 Dressing room - dungeon

[**Iteration 1**] There will be a connection between the dressing room and the dungeon.

3.2.CO.2.6 Courtyard - kitchen

[**Iteration 1**] There will be a connection between the kitchen and the courtyard.

3.2.CO.3 Functionality of area connections

none

3.2.CO.4 Events pertaining to area connections

3.2.CO.4.1 Moving a character through a connection

Software Requirements Specification: *Encounter* Video Game

[Iteration 1] 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.

3.2.EC Encounter characters

3.2.EC.1 Attributes of Encounter characters

3.2.EC.1.1 Name of Encounter characters

[Iteration 1] Every game character in the *Encounter* video game shall have a unique name. The specifications for names shall be the same as those for Area names, specified in 3.2.AR.1

3.2.EC.1.2 Qualities of Encounter characters

[Iteration 1] Every game character has the same set of qualities. Each quality shall be a non-negative floating point number with at least one decimal of precision. These are all initialized equally so that the sum of their values is 100. The value of a quality cannot be both greater than zero and less than 0.5.

For Iteration 1 the qualities shall be *concentration*, *intelligence*, *patience*, *stamina*, and *strength*. They shall be the same for Iteration 2 unless a team decides to modify them.

3.2.EC.1.3 Image of Encounter characters

[Iteration 1] Every game character shall have an image.

3.2.EC.2 *Encounter* character entities

The characters of the game are described among the types of *Encounter* characters.

3.2.EC.3 Functionality of *Encounter* characters

3.2.EC.3.1 Living points

[Iteration 1] The *Encounter* game shall be able to produce the sum of the values of any character's qualities, called its *living points*.

Software Requirements Specification: *Encounter* Video Game

3.2.EC.3.2 Configurability of *Encounter* character quality values

[Iteration 1] 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.

3.2.ED Engagement displays

[Iteration 1] There shall be a window displaying the result of engagements. The format is shown in Figure 3.2.ED-1:

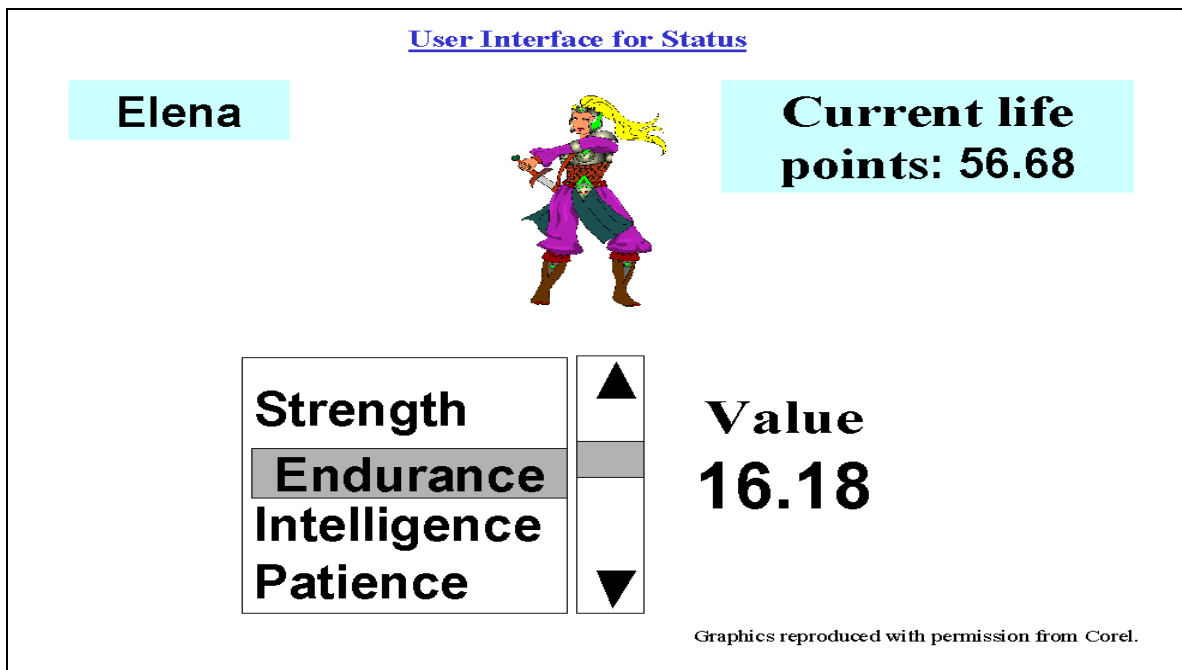


Figure 3.2.ED-1: User Interface for Showing Status

3.2.ED.4 Engagement display events

3.2.ED.4.1 Dismissing the display

[Iteration 1] When the user hits OK, the display disappears.

3.2.EG The *Encounter* game

The requirements in this section pertain to the game as a whole.

Software Requirements Specification: *Encounter* Video Game

3.2.EG.1 Attributes of the *Encounter* game

3.2.EG.1.1 *Duration*

[**Iteration 2**] A record will be kept of the duration of each game, timed from when the player begins the game.

3.2.EG.1.2 *Action buttons*

[**Iteration 1**] Every game window shall show a "Get status" button, a "Set qualities" button and an "End game" button in the lower left corner.

3.2.EG.2 Entities of the *Encounter* game

3.2.EG.2.1 *Single game*

[**Iteration 1**] There shall be a single game.

[*Note*: future releases might allow several versions of the game to run at the same time. It is up to each project team to decide whether they would provide this capability in **Iteration 2** – it is not mandatory.]

3.2.EG.4 Events the *Encounter* game

3.2.EG.4.4 *Pressing the Set qualities button*

[**Iteration 1**] When the user presses the *set qualities* button, and provided that there is no foreign character in the area, a window for setting the values of qualities appears, superimposed on the area. See 3.2.PQ for the specifications of this window.

3.2. EG.4.5 *Pressing the End game button*

[**Iteration 1**] When the user presses the *end game* button, the game terminates. No additional screens appear.

3.2. EG.4.6 *Pressing the Get status button*

[**Iteration 2**] When the user presses the *get status* button, an engagement display window appears showing the status of the player's character before and after the last engagement.

3.2.EN Engagements

An *engagement* is the interaction between a game character controlled by the player and a foreign character.

3.2.EN.1 Attributes of engagements

none

3.2.EN.2 Engagement entities

There are no permanent engagement entities.

3.2.EN.3 Functionality of engagements

3.2.EN.3.1 *Engaging a foreign character*

[**Iteration 1**] 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.

No transfer of points takes place if no character is stronger.

After the value re-allocations are made, if either character has no points, the game ends.

If the game does not end, the player's character is moved to a random area, and the results of the engagement are displayed.

Example of the value re-allocations: suppose that the player engages a foreign character in an area preferring *stamina* and *concentration*. If p_s is the value of the player's stamina etc., and assuming $p_s + p_c > f_s + f_c$, we would have $p_s' = p_s + f_s/2$, $p_c' = p_c + f_c/2$, $f_s' = f_s/2$, and $f_c' = f_c/2$ where x' is the value of x after the transaction.

To take a numerical example of an engagement in this area: if the player's stamina value is 7, and concentration value is 19, and Freddie the foreigner's stamina is 11 and concentration 0.6, then the player is stronger. The result of the engagement would be:

Player: stamina $7 + 11/2 = 12.5$; concentration $19 + (0.6)/2 = 19.3$

Freddie: stamina $11/2 = 5.5$; concentration 0 because $(0.6)/2$ is less than 0.5

3.2.EN.4 Events on engagements

3.2.EN.4.1 Interrupting engagements

[**Iteration 1**] Players are able to interrupt engagements on a random basis. On average, the player can stop one of every ten engagements, by executing the procedure to set qualities. The user tries to interrupt an engagement by attempting to set the player's qualities. If the game does not allow this, no indication is given: the game proceeds as if the attempt had not been made.

3.2.FC Foreign characters

A foreign character is an *Encounter* character not under the player's control.

3.2.FC.1 Attributes of foreign characters

[**Iteration 1**] See *Encounter* character requirements. These are initialized to be equal.

[In future releases, foreign characters may mutate into new forms. It is up to each project team to decide whether they wish to implement this capability in **Iteration 2**.]

3.2.FC.2 Foreign character entities

Note: this section indicates that there is only one foreign character.

3.2.FC.2.1 Freddie foreign character

[**Iteration 1**] There shall be a foreign character named "Freddie", whose image is shown Figure 3.2.FC-1 – in fact, project teams may decide on an image of their choice. This character shall initially have a total of 100 points, distributed equally among its qualities.

Foreign Character Freddie's Image



Graphics reproduced with permission from Corel.

Figure 3.2.FC-1: Image for Freddie Foreign Character

3.2.FC.3 Functionality of foreign characters

3.2.FC.3.1 Foreign character movement

[Iteration 1] 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, such that the value of these qualities are as close to equal as possible.

3.2.PC Player characters

These are *Encounter* characters under the control of the player.

3.2.PC.1 Attributes of player characters

See *Encounter* character attributes. Player character images can be selected from one of the images in Figure 3.2.PC-1 – in fact, project teams may use images of their own choice.

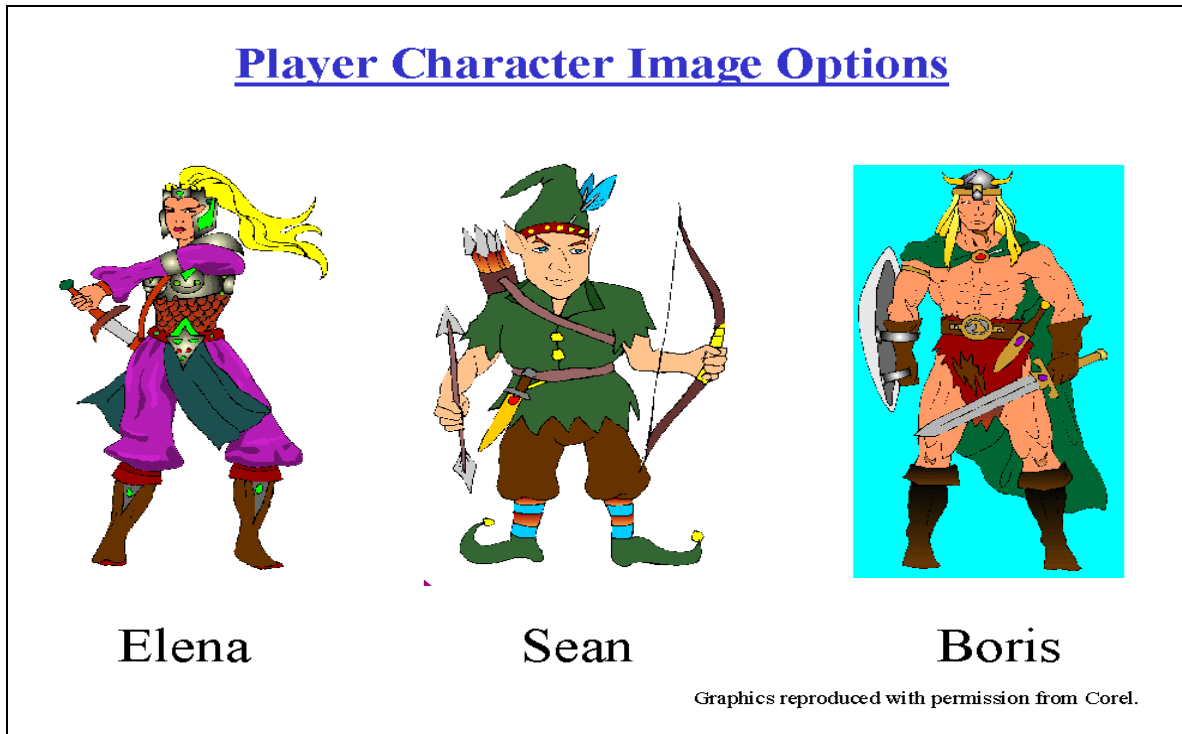


Figure 3.2.PC-1: Player character image options

3.2.PC.2 Player character entities

3.2.PC.2.1 Player's main character

[Iteration 1] The player shall have control over a particular game character called the "main" character. The nature of this control is subject to the restrictions specified in the remaining requirements. This character shall initially have a total of 100 points, distributed equally among its qualities.

3.2.PC.2.2 Additional characters under the control of the player

[Iteration 2 – very desirable] The player shall be able to introduce characters he controls other than the main player. Details to be decided by each project team.

3.2.PC.3 Player character functionality

3.2.PC.3.1 Configurability of the player character quality values

[Iteration 1] 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

Software Requirements Specification: *Encounter* Video Game

the *PlayerQualityWindow* shown in Figure 3.2.PQ-1. 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 0.5, which are replaced by zero quality values.

3.2.PC.3.2 Configurability of the player character images

[Iteration 2] The player shall have the option to choose the image representing his or her main character from at least two images. These options are shown in the figure.

3.2.PC.3.3 Aging of the player character images

[Iteration 2] ("Player aging") The main player character shall automatically increase each quality by a percentage for the first half of his or her life, then decrease them by the same percentage for the second half. Details are to be decided by each project team.

3.2.PQ The player quality window

This is a window from which the player may allocate the values of his or her characters.

3.2.PQ.1 Attributes of the player quality window

[Iteration 1] The window for setting the qualities of a player character in *Encounter* is shown by means of a typical example in Figure 3.2.PQ-1:

Software Requirements Specification: *Encounter* Video Game

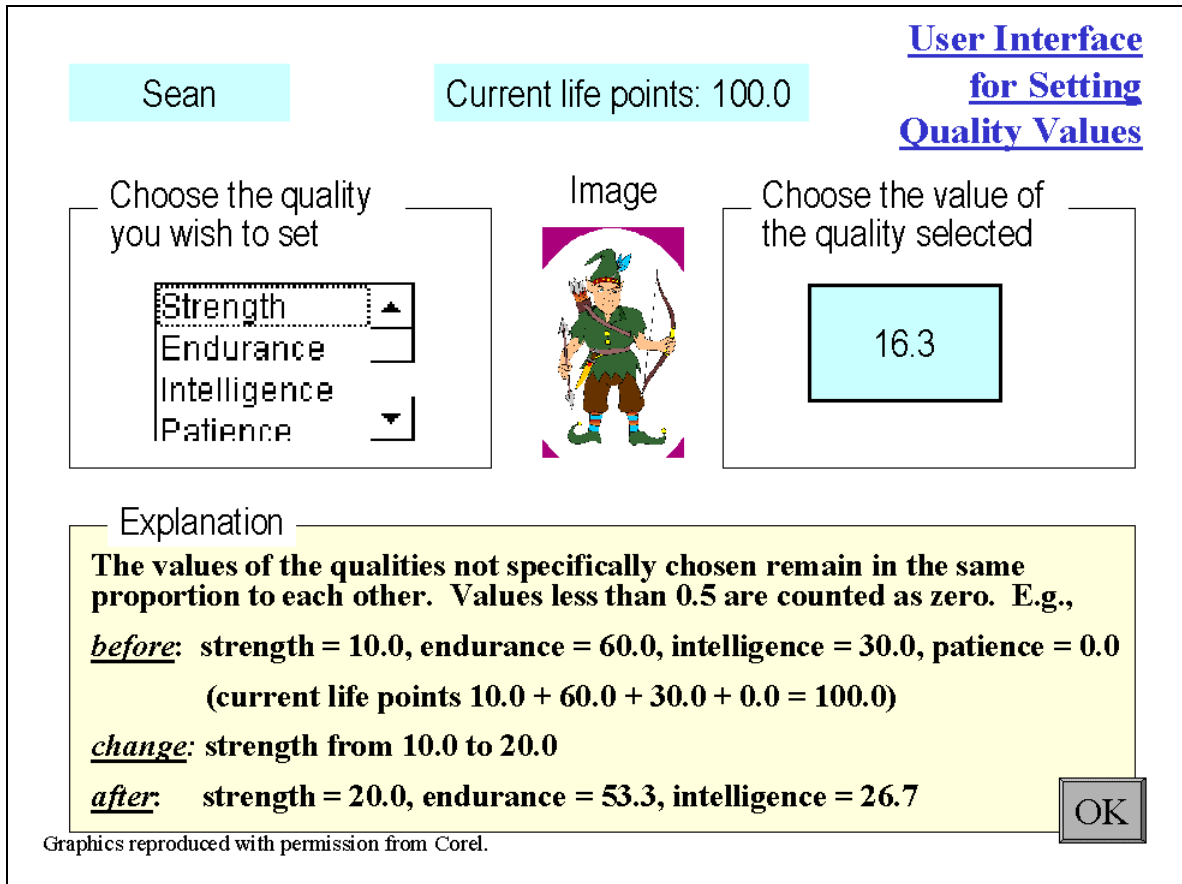


Figure 3.2.PQ-1: User Interface for Setting Values

The game character icon appears in the center and its name at the left top. The character's life points appear in the right top corner. On the left center is a list box of the qualities displaying four of them at a time. Clicking on one of these allows the player to select a value for it in the text box on the right. An explanation of how the arithmetic is performed is shown in a pale yellow box at lower center. Color backgrounds for the name, life points and value boxes are to be pale turquoise.

The method for performing quality value computations is described in section 3.2 below.

3.2.PQ.2 Player quality window entity

3.2.PQ.2.1 Window for allocating qualities

[Iteration 1] A window shall be available under the conditions described above to allocate the values of the player character. The window shall have the appearance of the GUI shown in section 3.1.1.2 of this specification.

Software Requirements Specification: *Encounter* Video Game

3.2.PQ.3 Player quality functionality

3.2.PQ.3.1 Initiating the display

[Iteration 1] The player quality menu shall be able to display itself.

3.2.PQ.4 Player quality window events

3.2.PQ.4.1 Displaying the value of a quality

[Iteration 1] When the player clicks on a quality in the list box on the left, the value of that quality shall be displayed in the text box on the right.

3.2.PQ.4.2 Setting the value of a quality

[Iteration 1] When the user enters a legitimate value for a quality, and hits the "enter" button, the value of that quality may be set to the amount entered, subject to requirement 3.2.PQ.4.3. If the value is invalid, an error message shall appear.

3.2.PQ.4.3 Dismissing the window

[Iteration 1] When the user hits the OK button, a time of four seconds elapses, after which the window disappears. At the end of this time period (i.e., if there are no interruptions), the value allocations made.

3.2.PQ.4.4 Interruption

[Iteration 1] On interruption of the quality value window being displayed, the window vanishes.

Note that interruptions will be caused by a foreign character entering the area. Note also that the quality values are not changed, and an engagement takes place.

3.2.TH Thumbnail sketch of playing areas

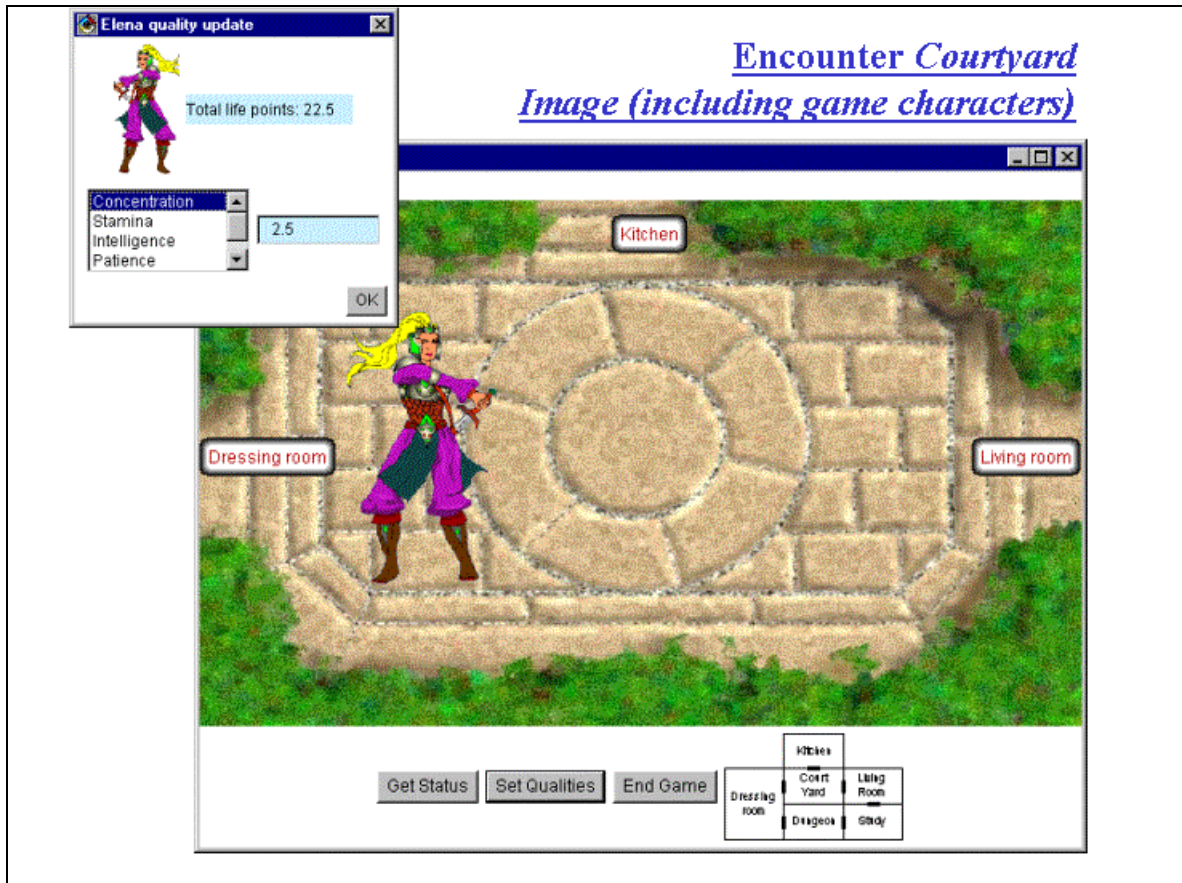
This is a figure showing the relationship among the areas in the immediate environment of the player.

3.2.TH.1 Attributes of the thumbnail sketch

Software Requirements Specification: *Encounter* Video Game

3.2.TH.1.1 Thumbnail image

[Iteration 1] The thumbnail image shall be as shown below:



3.3 Performance requirements

The application shall load and display the initial image in less than a minute.

Engagements should execute in less than one second.

3.4 Design constraints

Encounter shall be designed using UML and object-oriented design.

It shall be implemented in Java.

It shall be designed in a way that makes it relatively easy to change the rules under which the game operates so that others can customize the game.

3.5 Software system attributes

Software Requirements Specification: *Encounter* Video Game

3.5.1 Reliability

Encounter shall fail not more than once in 1000 encounters.

3.5.2 Availability

No specific requirement.

3.5.3 Security

[**Iteration 2**] Allow access to saved games only with a password,

3.5.4 Maintainability

3.5.4.1 Changing characters and areas

[**Iteration 1**] It shall be relatively straightforward to change characters and areas.

3.5.4.2 Globally altering styles

[**Iteration 2, desirable**] It shall be straightforward to alter globally the style of the areas and connections. (Style changes reflect different levels of game play in the same environment.)

3.5.4.3 Altering rules of engagement

[**Iteration 2, desirable**] rules of engagement should be relatively easy to change