Regio Vinco™

Software Design Description

Author: Limeng Ruan

June, 2015

Version 1.0

Abstract: This document describes the software design for Regio Vinco, a casual and educational point and click game in development.

1 Introduction

This is the Software Design Description (SDD) for the Regio Vinco game application. Note that this document format is based on the IEEE Standard 1016-2009 recommendation for software design.

1.1 Purpose

This document is to serve as the blueprint for the construction of the Regio Vinco application. This design will use UML class diagrams to provide complete detail regarding all packages, classes, instance variables, class variables, and method signatures needed to build the application. In addition, UML Sequence diagrams will be used to specify object interactions post-initialization of the application, meaning in response to user interactions or timed events.

1.2 Scope

RegioVinco will be a point and click game that only requires users to click. Thus a Point and Click Game framework should be made or implemented so that for each screen and each region, there will not be much duplicated work. So, this design contains design descriptions for the development of both the framework and game. Note that Java is the target language for this software design.

1.3 Definitions, acronyms, and abbreviations

Class Diagram – A UML document format that describes classes graphically. Specifically, it describes their instance variables, method headers, and relationships to other classes.

IEEE – Institute of Electrical and Electronics Engineers, the "world's largest professional association for the advancement of technology".

Framework – In an object-oriented language, a collection of classes and interfaces that collectively provide a service for building applications or additional frameworks all with a common need.

Java – A high-level programming language that uses a virtual machine layer between the Java application and the hardware to provide program portability.

Point and Click Game – A game that requires only mouse clicks.

Point and Click Game Framework – A basic outline for a game that only requires mouse clicks. It includes methods for adding different components to the application, such as buttons, images. This Framework has been built.

Sequence Diagram – A UML document format that specifies how object methods interact with one another.

UML – Unified Modeling Language, a standard set of document formats for designing software graphically.

1.4 References

IEEE Std 830TM-1998 (R2009) – IEEE Standard for Information Technology – Systems Design – Software Design Descriptions

Regio Vinco™ SRS – Software Requirements Specification for the application.

1.5 Overview

This Software Design Description document provides a working design for the Regio Vinco software application as described in the Regio Vinco Software Requirements Specification. Note that all parties in the implementation stage must agree upon all connections between components before proceeding with the implementation stage. Section 2 of this document will provide the Package-Level Viewpoint, specifying the packages and frameworks to be designed. Section 3 will provide the Class-Level Viewpoint, using UML Class Diagrams to specify how the classes should be constructed. Section 4 will provide the Method-Level System Viewpoint, describing how methods will interact with one another. Section 5 provides deployment information like file structures and formats to use. Section 6 provides a Table of Contents, an Index, and References. Note that all UML Diagrams in this document were created using the VioletUML editor.

2 Package-Level Design Viewpoint

As mentioned, this design uses the Point and Click Game Framework that has been built. The Framework and the game heavily use Java API. Following are the components being used in the Framework and the game package that will be built.

2.1 Point and Click Game and Regio Vinco Overview

The Regio Vinco game will be built based on the Point and Click Game Framework. There are many APIs used in the Point and Click Game Framework, but they will not be

shown here since it's already designed and coded. There is also an Audio Manager package available for use that has some simple functions. At the same time, the data should be stored in a tree data structure so that it can be navigated.

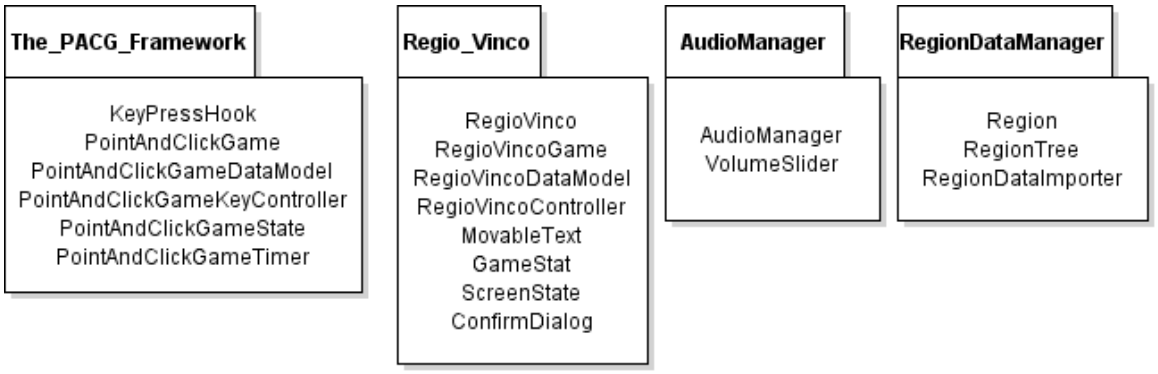

Figure 2.1 Design packages overview

2.2 Java API usage:

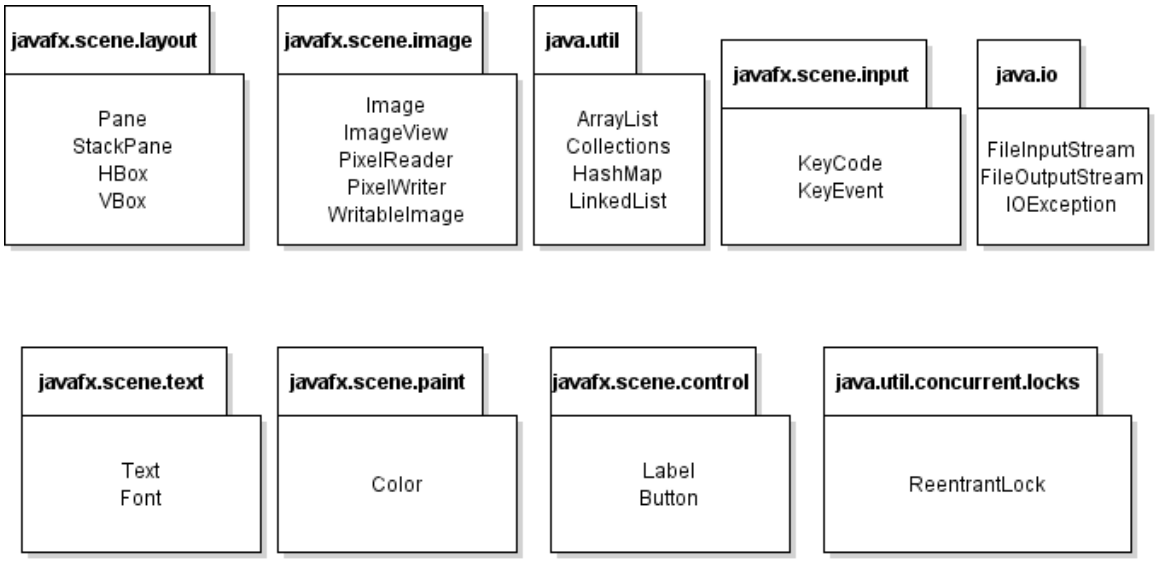

Figure 2.2 Java APIs and packages to be used. (XML format will be used for importing data, but it's not shown here in the construction of the game)

2.3 Java API usage description:

Table 2.1 to 2.8 below summarize how each of the classes will be used.

Table 2.1: use of javafx.scene.layout package

Class/Interface	Use
Pane	Create panes for details to be added, such as a new Pane for a different game.
StackPane	The whole game should be rendered using a StackPane, and add elements on top of it.
HBox	Use HBox for things that need to be aligned horizontally, such as in game stats.
VBox	Use VBox for things that need to be aligned vertically.

Table 2.2: use of javafx.scene.Image package

Class/Interface	Use
Image	Load existing images into the program.
ImageView	For displaying images
PixelReader	Read colors pixel by pixel
PixelWriter	Change colors pixel by pixel
WritableImage	Make the image that needs to be changed according to user input changeable.

Table 2.3: use of java.util package

Class/Interface	Use
ArrayList	Store the data in each pixel in an ArrayList
Collections	For convenient uses when needed, such as shuffle.
HashMap	In Region Name Game, store the color corresponding to a region in a HashMap

LinkedList	Use LinkedLists to store the stacks of regions names or flags according to game

Table 2.4: use of java.util.concurrent.locks package

Class/Interface	Use
ReentrantLock	Make sure when using data, there is no race condition.

Table 2.5: use of javafx.scene.text package

Class/Interface	Use
Text	Used for creating texts contents
Font	Used for styling the texts.

Table 2.6: use of javafx.scene.paint package

Class/Interface	Use
Color	Make colors or get colors or any related customization.

Table 2.7: use of javafx.scene.control package

Class/Interface	Use
Label	Used for creating texts with background.
Button	Used for creating buttons for user to interact with.

Table 2.8: use of javafx.scene.input package

Class/Interface	Use
KeyEvent	Uses for detecting keys being pressed.
KeyCode	Receives the keys being pressed and call certain functions such as pause, cheat.

3. Class-level design viewpoint

As mentioned, Regio Vinco will be constructed with the Point and Click Game Framework. The following UML briefly reflects the relationships between classes. Due to the complexity of the project, we present the class designs using a series of diagrams going from overview diagrams down to detailed ones.

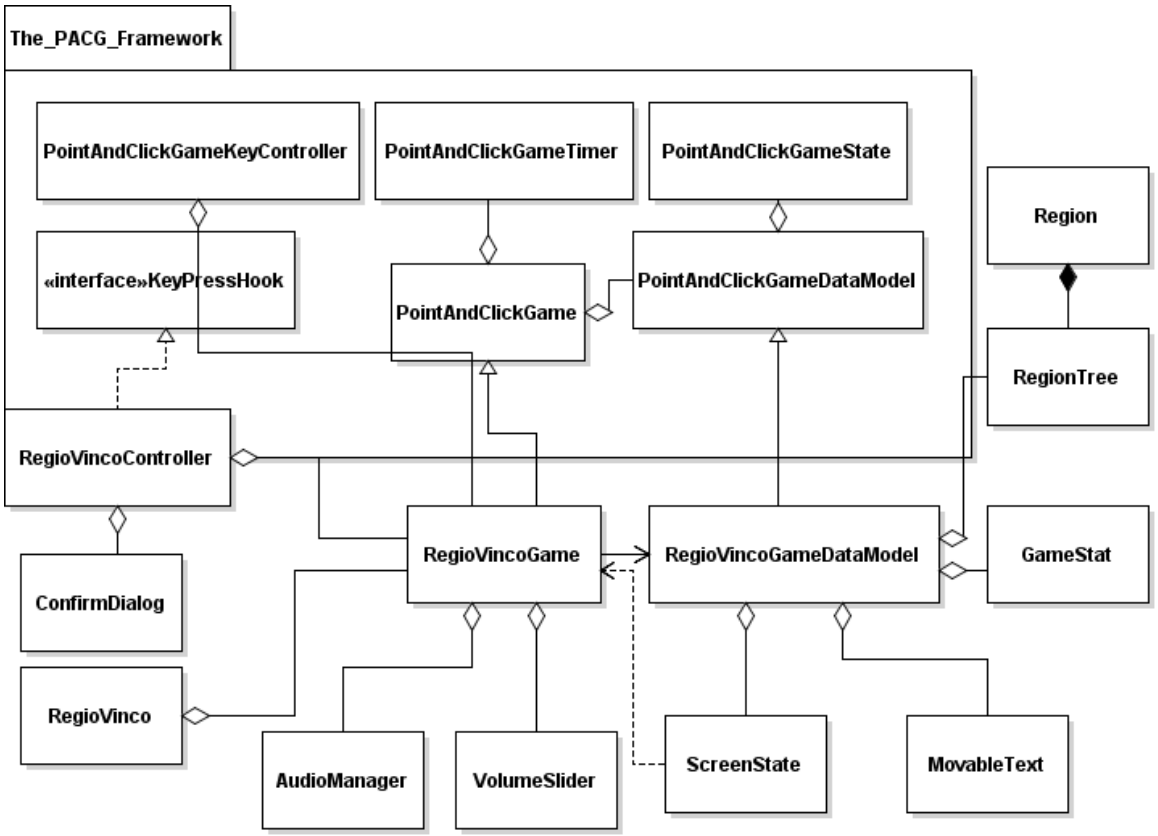

Figure 3.1 RegioVinco and Point and Click Game Framework Overview.

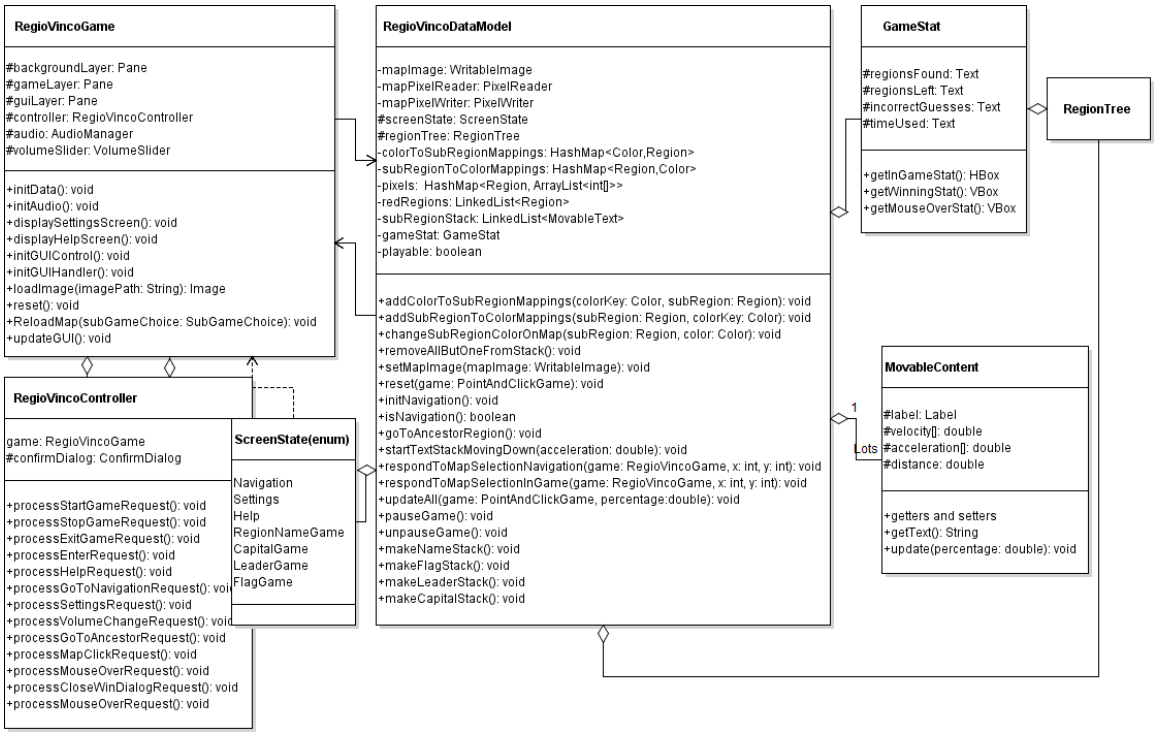

Figure 3.2 Detailed Regio Vinco Class Diagram and related components

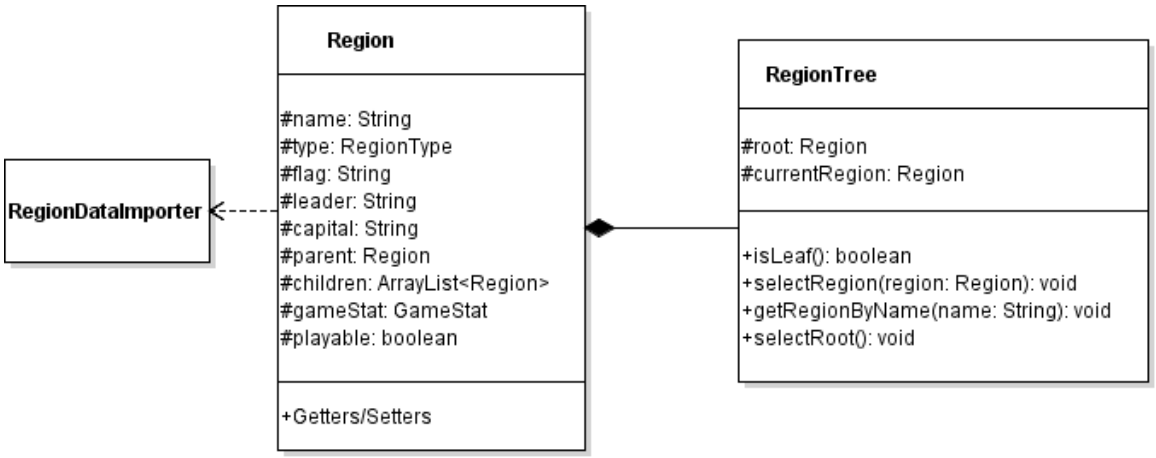

Figure 3.3 Class Diagram Overview of data structure for regions.

4 Method-Level Design Viewpoint

Now that the general architecture of the classes has been determined, it is time to specify how data will flow through the system. The following UML Sequence Diagrams describe the methods called within the code to be developed in order to provide the appropriate event responses.

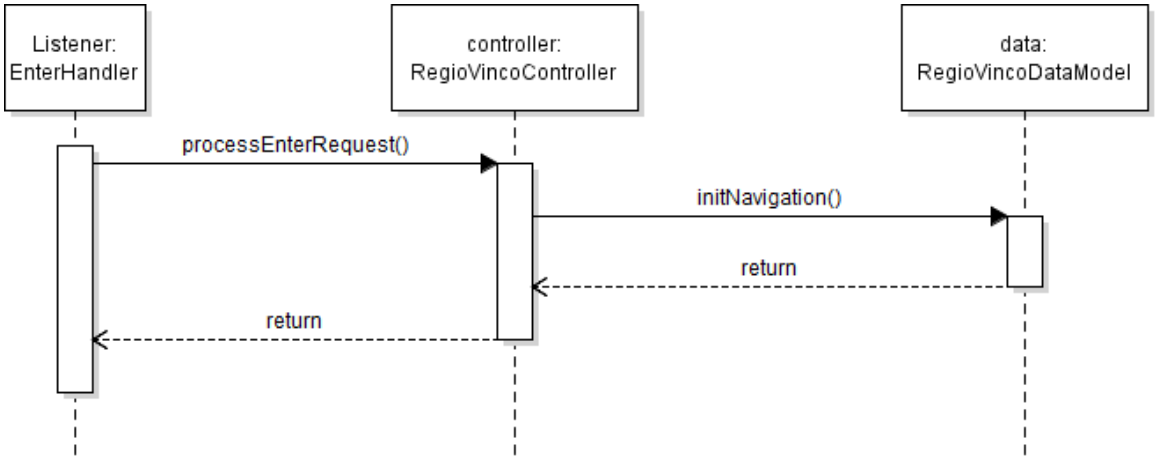

Figure 4.1 Enter Game Event Sequence Diagram.

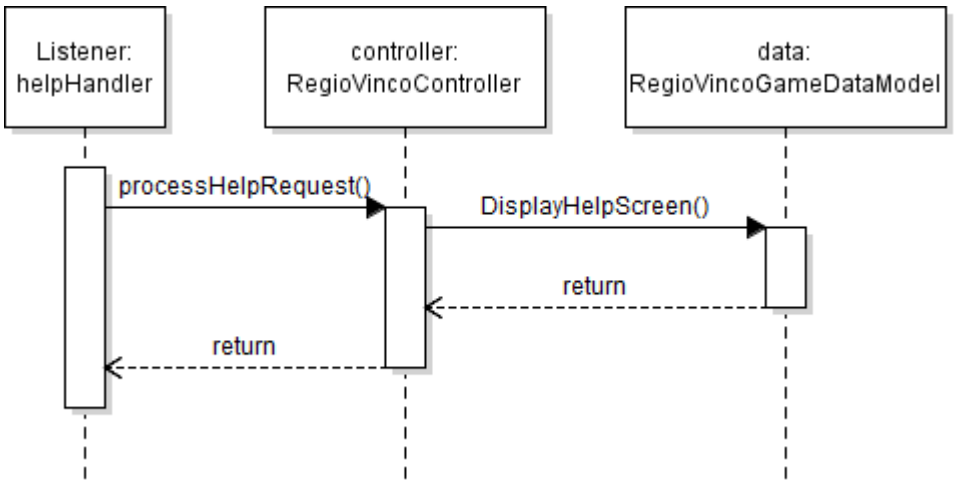

Figure 4.2 Help Event Sequence Diagram.

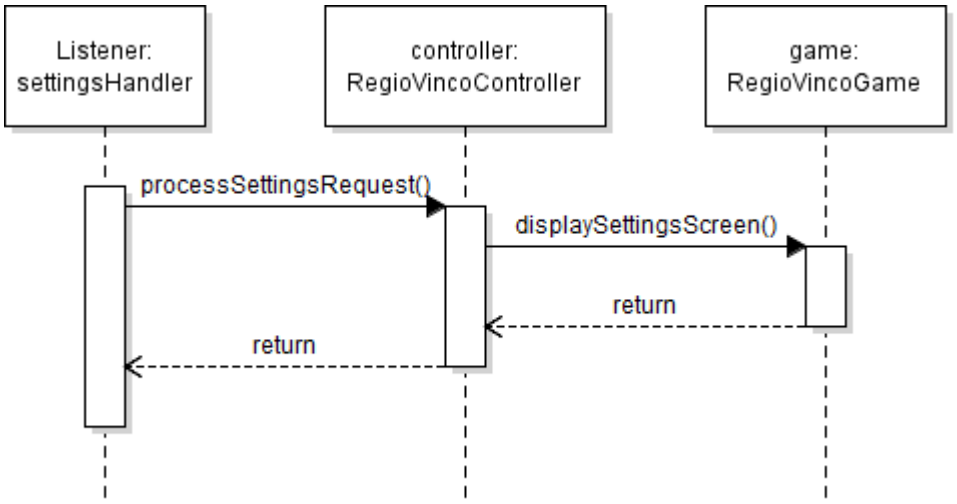

Figure 4.3 Settings Event Sequence Diagram

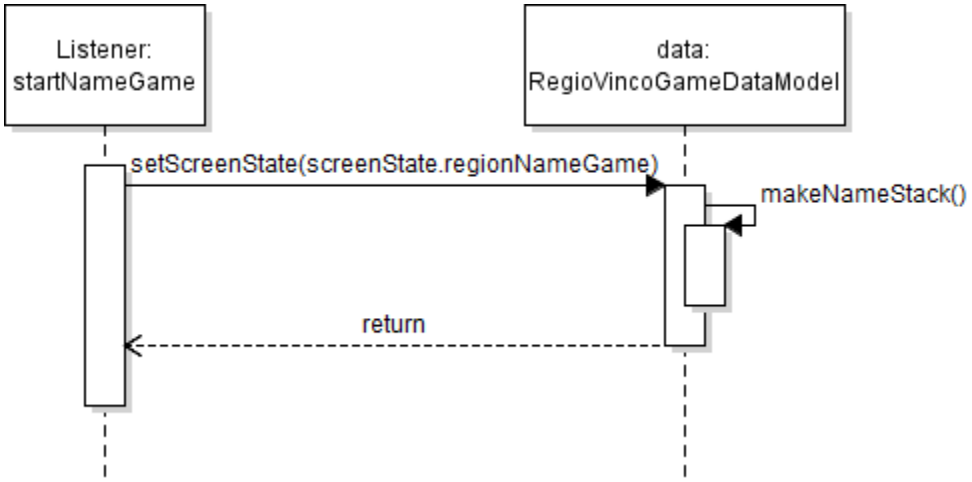

Figure 4.4 Start a Game Sequence Diagram. (Note: the other games are similar except for the stacks, if any other game is selected, a different Stack will be made.)

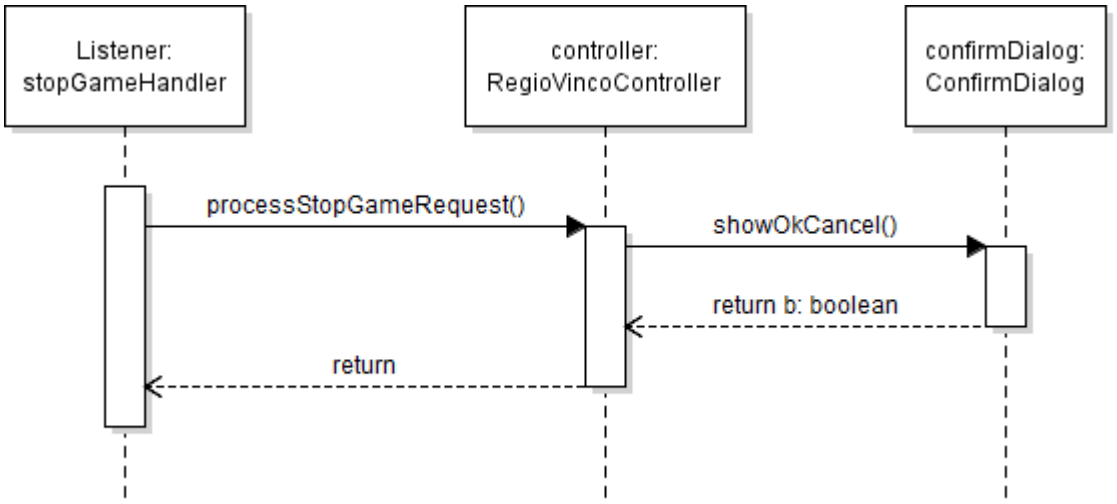

Figure 4.5 Stop Game Event Sequence Diagram

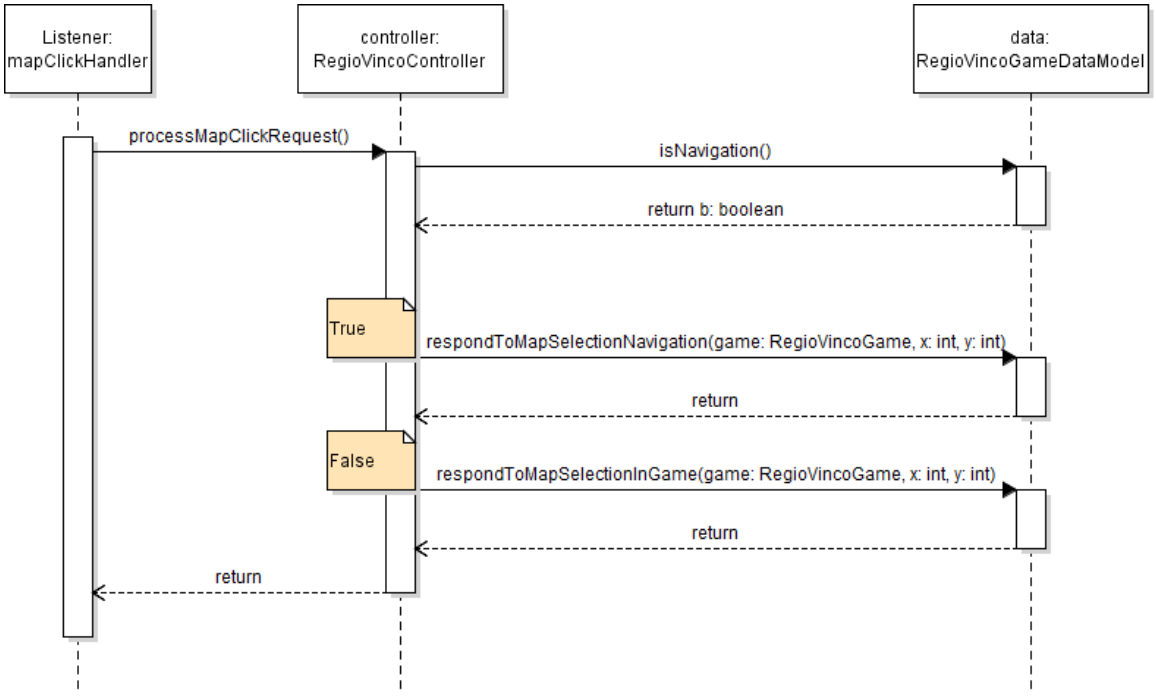

Figure 4.6 Map Click Event Sequence Diagram

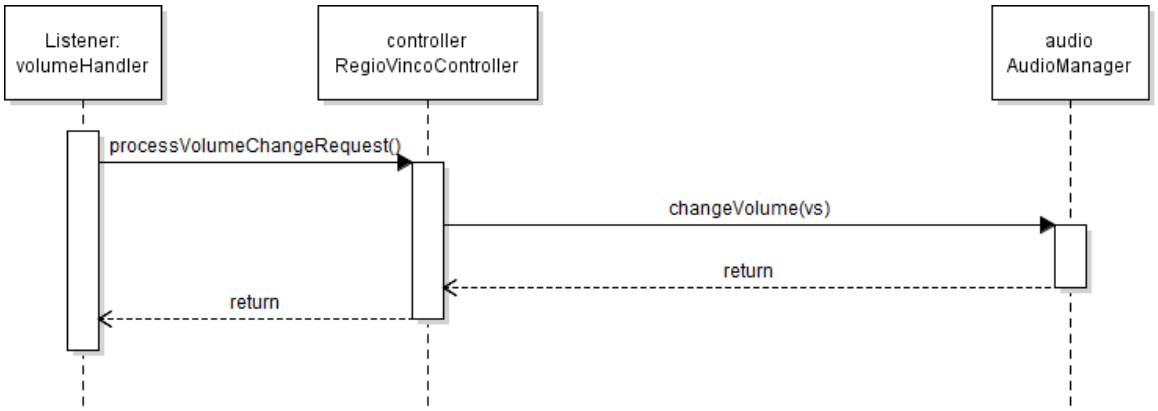

Figure 4.7 Volume Change Event Sequence Diagram

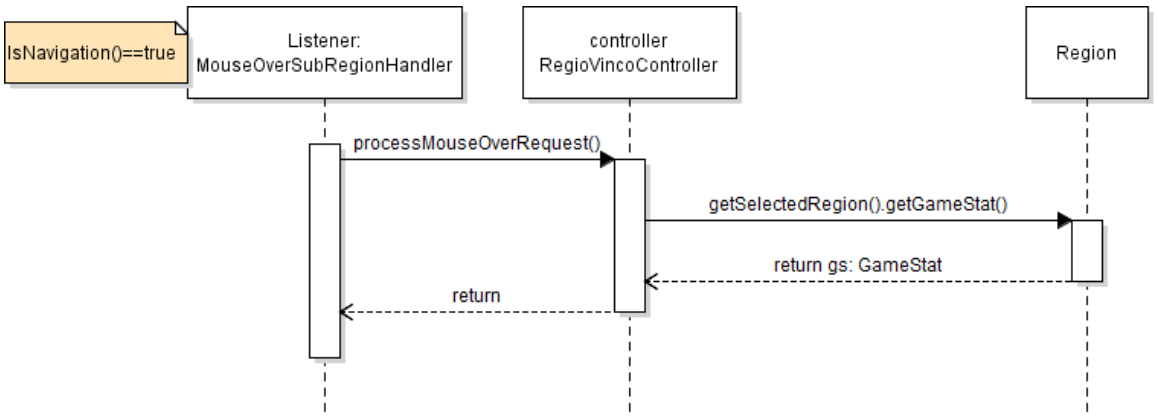

Figure 4.8 Mouse-Over Event Sequence Diagram

5. File Structure and Formats

Note that the game records will be stored in a txt file whenever a game is finished.

All the data that makes this game will be imported from an XML file and images.

The XML format will be in as the following image.

```
▼<region name="The World">
   <sub_region red="225" green="225" blue="225" name="Africa"/>
   <sub_region red="200" green="200" blue="200" name="Antarctica"/>
   <sub_region red="175" green="175" blue="175" name="Asia"/>
   <sub_region red="150" green="150" blue="150" name="Europe"/>
   <sub_region red="125" green="125" blue="125" name="North America"/>
   <sub_region red="100" green="100" blue="100" name="Oceania"/>
   <sub_region red="75" green="75" blue="75" name="South America"/>
 </region>
```

6. Support Information

This is a document for the project, look for information yourself, a table is not provided.