**ISCG7421**

**Advanced Programme Development  
Assignment 1B**

**Village Design Application**

**Prepared by**

**Lei Li**

**1494024**

**Due Date: 29/10/2018**

# Table of Contents

**[1. Introduction](#_Toc1698_WPSOffice_Level1)** **[1](#_Toc1698_WPSOffice_Level1)**

**[2. Annotated Screen Designs](#_Toc16539_WPSOffice_Level1)** **[2](#_Toc16539_WPSOffice_Level1)**

[2.1 View Form 1](#_Toc16539_WPSOffice_Level2) [2](#_Toc16539_WPSOffice_Level2)

[2.2 View Form 2](#_Toc9034_WPSOffice_Level2) [3](#_Toc9034_WPSOffice_Level2)

[2.3 View Form 3](#_Toc18010_WPSOffice_Level2) [5](#_Toc18010_WPSOffice_Level2)

**[3.UML Use Case Diagram](#_Toc9034_WPSOffice_Level1)** **[6](#_Toc9034_WPSOffice_Level1)**

**[4. Activity Diagrams](#_Toc18010_WPSOffice_Level1)** **[7](#_Toc18010_WPSOffice_Level1)**

**[5. Class Diagram](#_Toc25604_WPSOffice_Level1)** **[10](#_Toc25604_WPSOffice_Level1)**

**[6. Test Cases](#_Toc32229_WPSOffice_Level1)** **[11](#_Toc32229_WPSOffice_Level1)**

# Introduction

The origin of my idea is from a game that the player can build different buildings for protecting their hometown. My application is an easy version of this game. The player can add buildings in his village. The super class (Building) has three attributes. Each of the three sub-classes (Castle, Storage, Training) has one unique attribute with a mutually exclusive data type and has unique behaviour.

The application has three views.

In the first view, the user can view, add, modify or delete buildings in a text format.

In the second view, the user can view, add, modify or delete buildings in a graphic format. And the user can play game in this view.

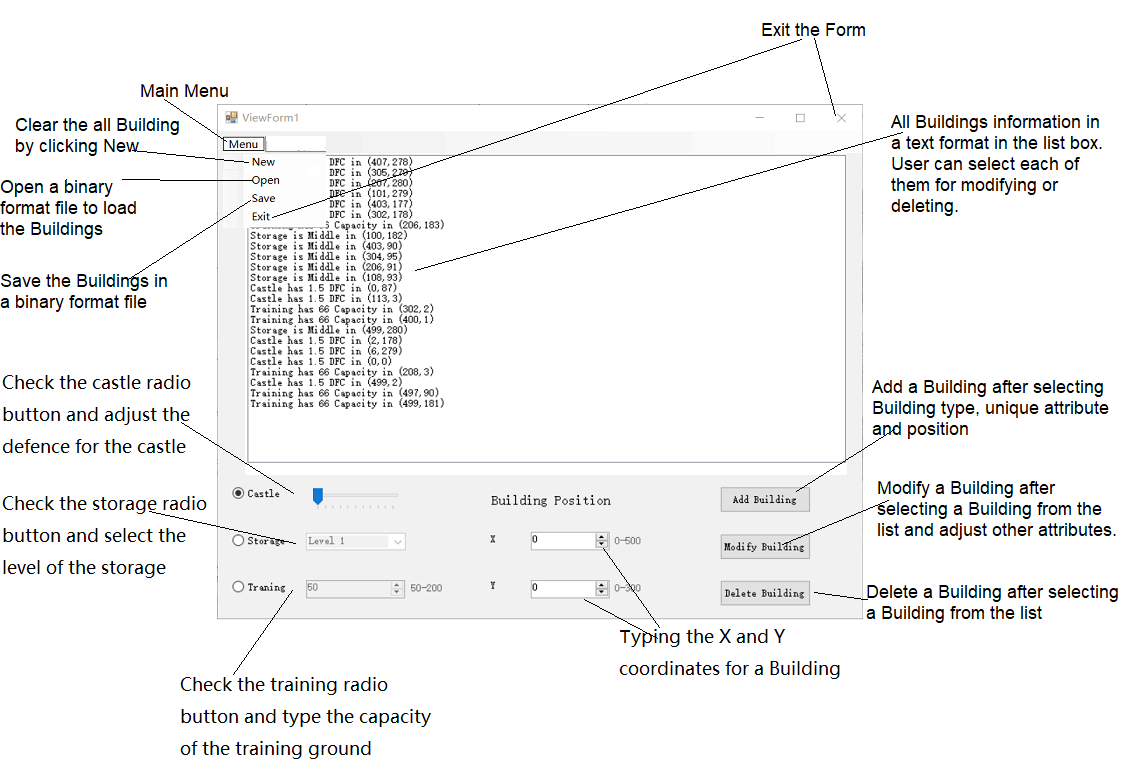
In the third view, the user can only view buildings and select to view each type of buildings.

In first and second view, the buildings are Serializable that user can save/open their information in a Binary Format.

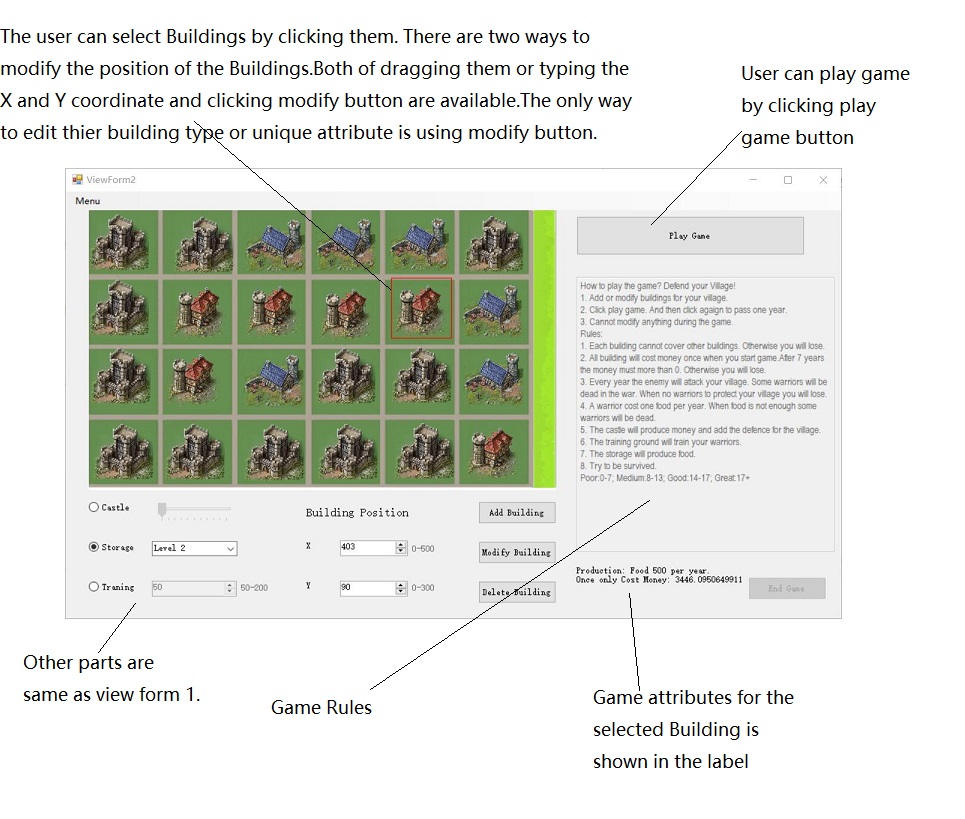
The Multi-threading is used in the game. It plays a GIF in the panel using while(true) loop.

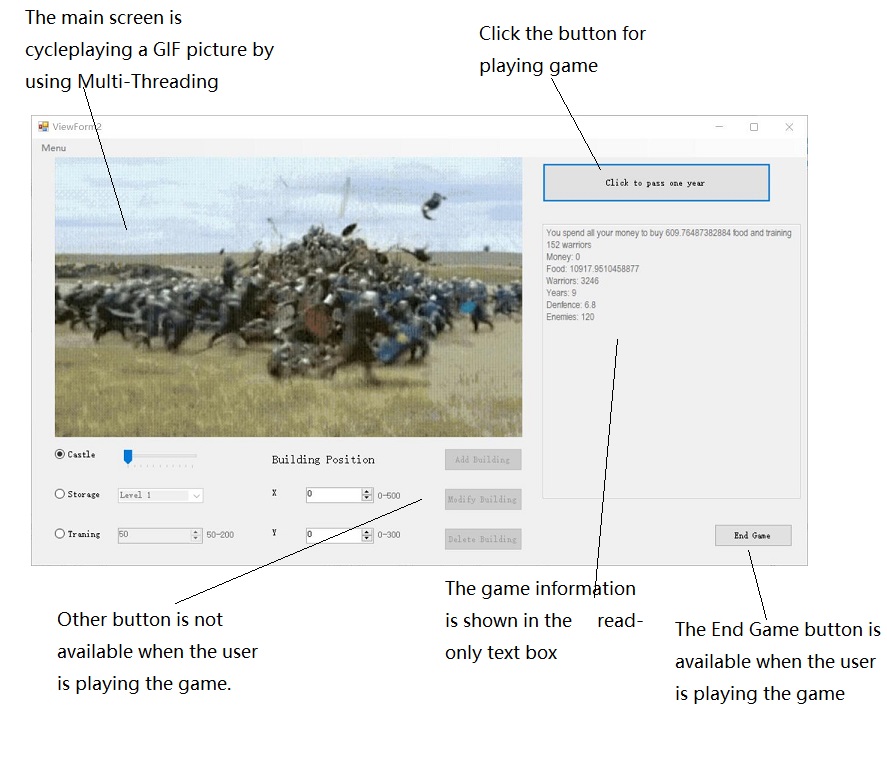
# Annotated Screen Designs

## 2.1 View Form 1

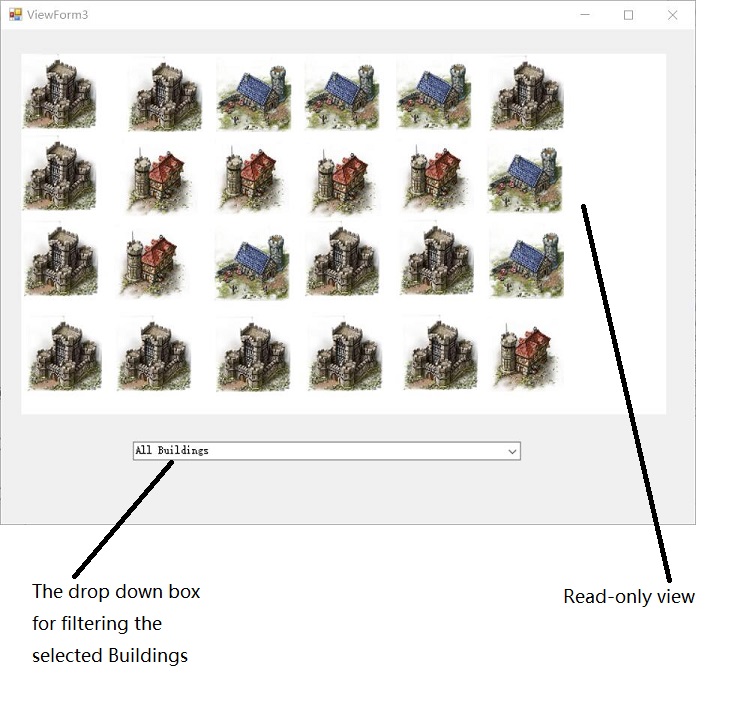


## 2.2 View Form 2

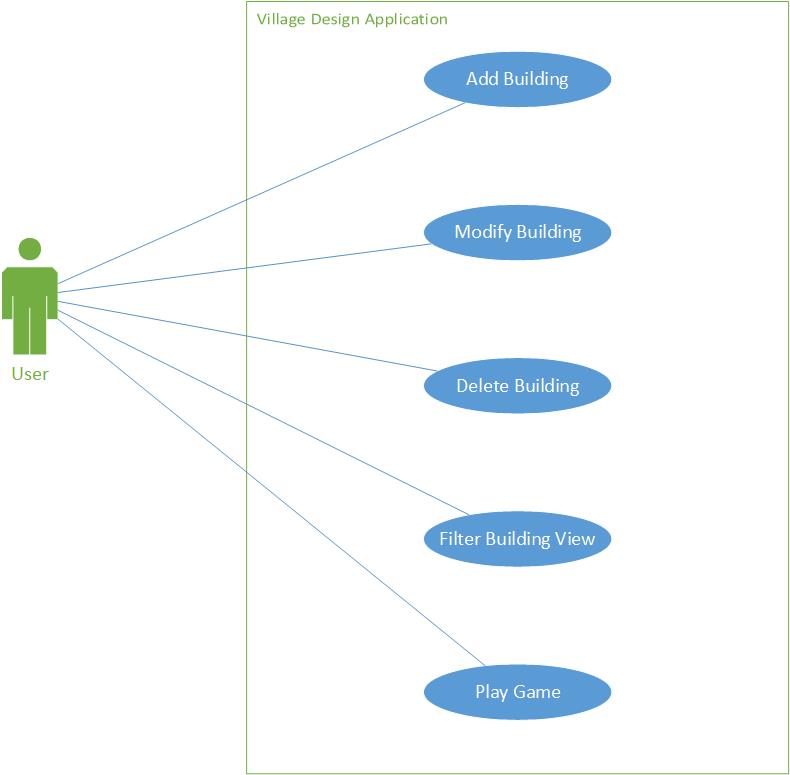




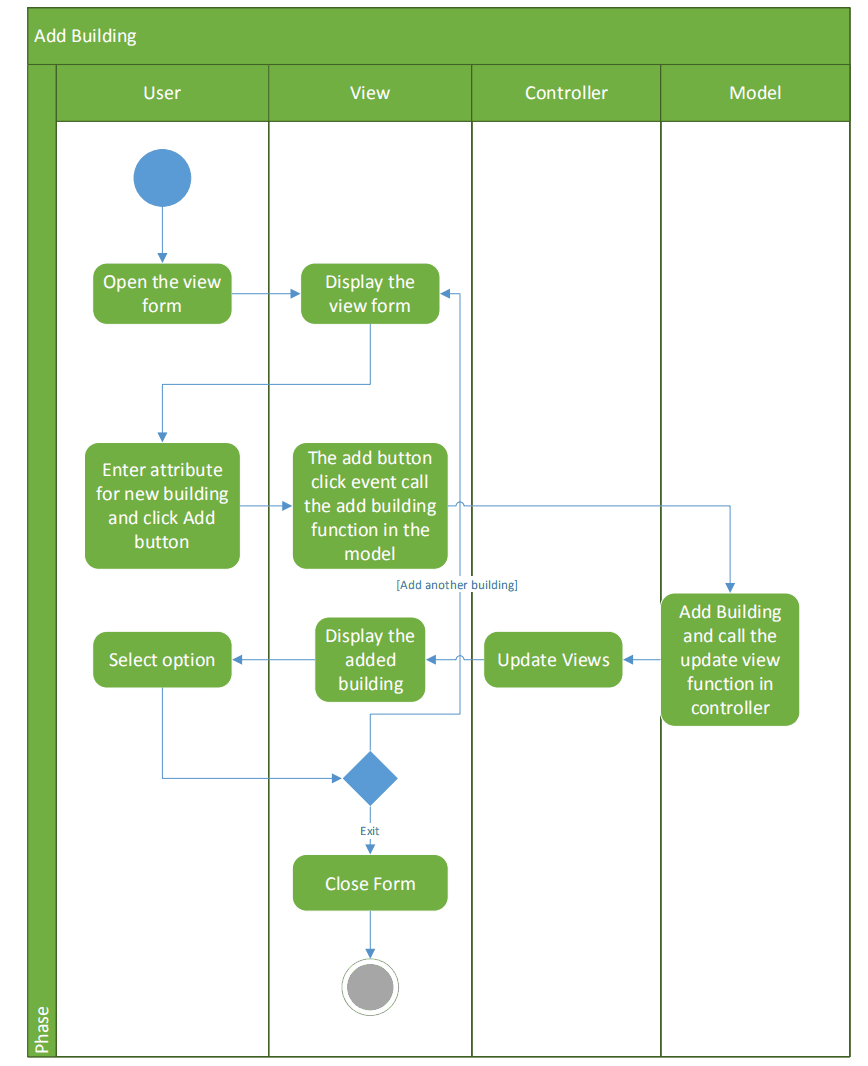
## 2.3 View Form 3

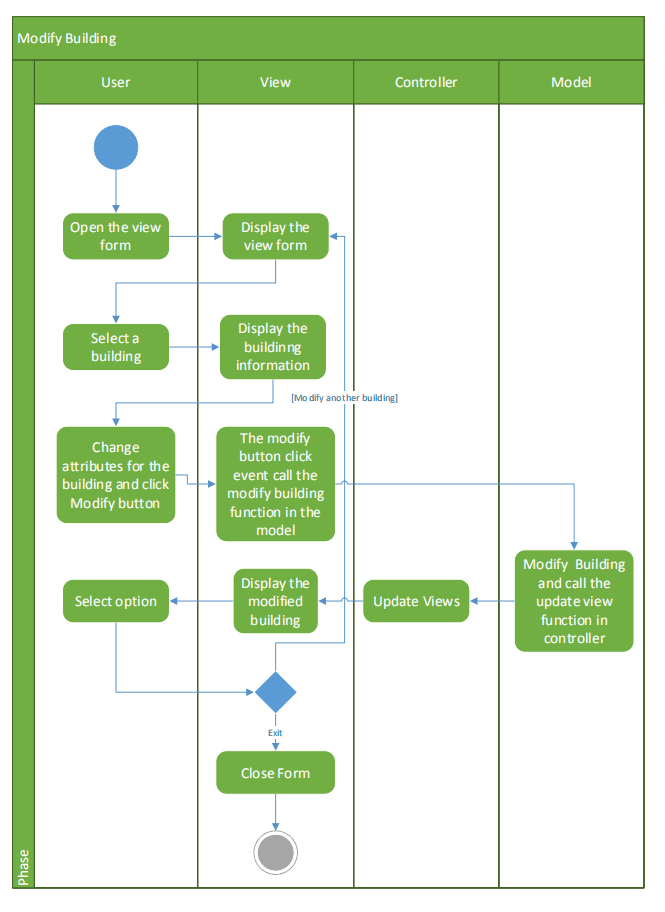


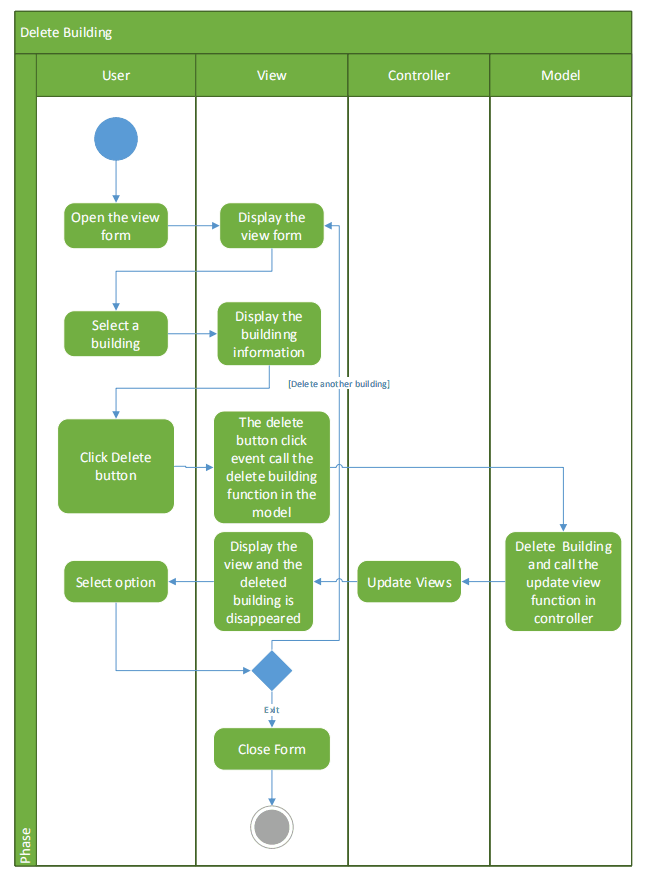
# 3.UML Use Case Diagram



# 4. Activity Diagrams







# 5. Class Diagram

## Drawing2

# Test Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement to test** | **Test Data Input** | **Expected Outcomes** | **Actual Outcomes** |
| 1.1 Add Building in view form 1 | Select Castle  Enter “300” in X  Enter “200” in Y  Click Add building button | The Castle is shown in the three views | As the expected outcomes.  IMG_256  Pass |
| 1.2 Modify Building in view form 1 | Click the Castle  Select Storage  Enter “200” in X  Enter “100” in Y  Click Modify building button | The Castle is modified as a Storage which is shown in the three views | As the expected outcomes.  Pass |
| 1.3 Delete Building in view form 1 | Click the Storage  Click Delete building button. | The Storage is deleted which is disappeared in the three views | As the expected outcomes.  Pass |
| 1.4 Add Building in view form 1 with unique attribute | Select Training  Enter “120” in Capacity  Enter “100” in X  Enter “100” in Y  Click Add building button | The Training with unique attribute is shown in the three views. | As the expected outcomes.  Pass |
| 2.1 Add Building in view form 2 | Select Training  Enter “250” in X  Enter “220” in Y  Click Add building button | The Training is shown in the three views | As the expected outcomes.  Pass |
| 2.2 Modify Building in view form 2 | Click the Training image  Select Storage  Enter “90” in X  Enter “0” in Y  Click Modify building button | The Training is modified as a Storage which is shown in the three views | As the expected outcomes.  Pass |
| 2.3 Delete Building in view form 2 | Click the Storage  Click Delete building button. | The Storage is deleted which is disappeared in the three views | As the expected outcomes.  Pass |
| 2.4 Add Building in view form 2 with unique attribute | Select Storage  Select “Level 3” in Level  Enter “100” in X  Enter “10” in Y  Click Add building button | The Storage with unique attribute is shown in the three views. | As the expected outcomes.  Pass |
| 2.5 Modify Building by dragging in view form 2 | Click the Storage image  Drag and Drop the image | The Storage x and y attribute is changed and this is shown in the three views | As the expected outcomes.  Pass |
| 3.1 Filter the buildings in view form 3 | Select all castles in the drop down box | All castles is shown in the view form 3.  No change in other forms. | As the expected outcomes.  Pass |
| 4.1 Save Serialization in view form 1 | Add some buildings in the view form 1.  Select and click save button in the menu.  The file name is default as “Village.dat”.  Click “OK” in the save dialogue. | The file is saved in the correct path. | As the expected outcomes.  Pass |
| 4.2 New button in view form 1 | Click new button in the menu | All buildings are clear in the three views. | As the expected outcomes.  Pass |
| 4.3 Open Serialization in view form 1 | Select and click Open button in the menu.  The file name is default as “Village.dat”.  Click “OK” in the Open dialogue. | The all buildings are loaded correctly in the three views. | As the expected outcomes.  Pass |
| 4.4 Save Serialization in view form 2 | Add some buildings in the view form 2.  Select and click save button in the menu.  The file name is default as “Village.dat”.  Click “OK” in the save dialogue. | The file is saved in the correct path. | As the expected outcomes.  Pass |
| 4.5 New button in view form 2 | Click new button in the menu. | All buildings are clear in the three views. | As the expected outcomes.  Pass |
| 4.6 Open Serialization in view form 2 | Select and click Open button in the menu.  The file name is default as “Village.dat”.  Click “OK” in the Open dialogue. | The all buildings are loaded correctly in the three views. | As the expected outcomes.  Pass |
| 5.1 Play game (Multi-threading testing) | Add some buildings in the view form 2.  Add click Play Game.  The GIF images is played in while(true) loop as well.  Click End game the thread is Abort(). | The admission status is shown as paid.  And it is recorded in text file | As the expected outcomes.  Pass |

## 