COMP 4403 Assignment #3

Due Friday March 21 2008 at midnight.
Some diagrams will be due Friday March 14 at 4pm

Overview

This assignment is the third milestone for the larger system we are creating during the semester. In this assignment the focus is no longer on testing. Instead, in this assignment you will be creating the main game program that will use your classes from the first two assignments.

Submitting

Put your assignment in a folder named `Assign3_`*`youruserID`*. Put any resources used by your assignment and Eclipse workspace into this folder. Copy this folder to the submit drive in lab B215. Diagrams must be on paper with the use case name and author name, and submitted to my drop-box outside of B233.

Grading

The grade for this assignment will be broken down as follows:

Code Documentation	4%
Program Design	15%
Modeling (Diagrams)	21%
Features	60%

Diagrams

The following pages describe the various use cases your program must implement. Some of these use cases will also specify what type of diagram you need to create. Some diagrams will be due *before* the assignment as a whole.

Deliverables

The focus of this assignment is a console program (i.e., a class with a `static void main` method) that has the functionality specified in the following use case descriptions. This program may very well require the creation of several classes.

Use Case Name	**Program Start**	
Description	What happens when application begins	
Typical Course Of Events	**Actor Action**	**System Response**
1.	This use case is initiated when user starts program.	The system will display the program name, its version number, last modified date, and its authors.
2.		The system will then automatically execute the **New Game** use case.

Use Case Name	**Display Main Program Menu**
Description	System displays the main program menu. These are the choices for a user/player.
Typical Course Of Events	**Actor Action**
1.	Initiated after program starts as well as after each previous action has been processed.

	System Response
	The system will display the menu with the following list of actions (in this order). The choices are:
	`--- Turn # ---`
	`0 – Quit`
	`1 – Program Information`
	`2 – New Game`
	`3 – View Info`
	`4 – Display Action Menu`
	`5 – Display Map`
	`6 – End Turn`

Use Case Name	**Process Menu Choice**	
Description	System processes user’s menu choice. You must submit an activity diagram for this use case on March 14. You must submit a sequence diagram for this use case on March 14.	
Typical Course Of Events	**Actor Action**	**System Response**
1.	Initiated when user enters a menu choice, which must be a number between 0 and 6 (or whatever the highest value in the menu is).	The system verifies that the menu choice is correct. If not a valid number or is not a number between 0 and 6, then it will display an appropriate error message and then redisplay the menu. If the user’s choice is valid, then the appropriate use case will be executed.

Use Case Name	**Program Information**	
Description	This displays information about the program.	
Typical Course Of Events	**Actor Action**	**System Response**
1.	Initiated when user selects Program Information.	The system will display the same message as that shown in the **Program Start** use case.

Use Case Name	**Quit**	
Description	This exits the game.	
Typical Course Of Events	**Actor Action**	**System Response**
1.	Initiated when user selects Quit.	The system will display message `--- Quit ---`
2		Program execution will end.

Use Case Name	**View Info**	
Description	This displays the game statistics for the player. You must submit a sequence diagram for this use case on March 14.	
Typical Course Of Events	**Actor Action**	**System Response**
1.	Initiated when user selects Information.	The system will display the message `--- Player Information ---`
2.		The system will display the following information: Name Home Region Name Amount of Gold Score For each region display: Name The type of each army Type of each building/upgrade Number of Peasants Number of Resources)

Use Case Name	**New Game**	
Description	This will perform the processing necessary for a new game. You must submit an activity diagram for this use case on March 14.	
Typical Course Of Events	**Actor Action**	**System Response**
1.	Initiated when user selects New Game.	The system will display message `--- New Game ---`
2		System will ask user to enter a player name
3.	User enters name.	System will then ask user to choose the player's home region (enter either `1-Belgium, 2-England, 3-France, 4-Germany, 5-Ireland, 6-Italy, 7-Scotland, 8-Spain`). These are the regions in the game map.
5.	User enters between 1 and 8.	System will verify user entered one of these values. If not get user to re-enter. If valid, then system will generate the appropriate new player. A computer player will also have to be generated with a different home region.

Use Case Name	**Display Action Menu**
Description	System displays the action menu. These are the action choices for a user/player.
Typical Course Of Events	**Actor Action**
1.	Initiated after user selects the display action menu item.

	System Response
	The system will display the menu with the following list of actions (in this order). The choices are:
	`0 – Return to Previous Menu`
	`1 – Levy Taxes`
	`2 – Recruit Army`
	`3 – Recruit Special Unit`
	`4 – Move Army`
	`5 – Move Special Unit`
	`6 – Display Build Action Menu`

	Additional Requirements: In a given turn, the user/player can only choose 5 actions. The program must ensure that the user cannot choose more than 5 actions in a turn. You can use whichever user interface method you'd like to achieve this requirement, but in general it is better to prevent an error from happening in the first place rather than let an error occur and then display an error message.

Use Case Name	**Display Map**	
Description	This displays the game map	
Typical Course Of Events	**Actor Action**	**System Response**
1.	Initiated when user selects Display Map.	The system will display message `--- Display Map` ---, followed by a list of each region that shows the name of the region, its owner, the army units in that region, and the special units (and their owner) in that region.

Use Case Name	**Display Build Action Menu**
Description	System displays the build action menu. These are the build action choices for a user/player.
Typical Course Of Events	**Actor Action**
1.	Initiated after user selects the display action menu item.

	System Response
	The system will display the menu with the following list of actions (in this order). The choices are:
	`0 – Return to Previous Menu`
	`1 – Build Farm`
	`2 – Build Intermediate Culture`
	`3 – Build Advanced Culture`
	`4 – Build Fort`
	`5 – Build Keep`
	`6 – Build Castle`
	`7 – Build Improved Resources`

Use Case Name	**Levy Taxes Action**	
Description	Used to levy taxes against a region. You must submit a sequence diagram for this use case on March 14.	
Typical Course Of Events	**Actor Action**	**System Response**
1.	Initiated when user selects Levy Taxes from Display Action Menu.	If the user hasn’t already chosen above the maximum number of actions, then system will display message `--- Levying Taxes ---`
2.		The system will display a menu asking the user to select the region (from the player’s regions) to levy taxes in (e.g., `1=France, 2=Germany`)
3.	The user will enter a number.	If valid, the system will levy the taxes and provide the user with a message whether the action succeeded or not for the region, e.g., `Taxes levied successfully for France`

Use Case Name	**Recruit Army Action**	
Description	Used to recruit an army in a region. You must submit a sequence diagram for this use case on March 21.	
Typical Course Of Events	**Actor Action**	**System Response**
1.	Initiated when user selects Recruit Army from Display Action Menu.	If the user hasn't already chosen above the maximum number of actions, then system will display message `--- Recruit Army ---`
2.		The system will display a menu asking the user to select the region (from the player's regions) to raise army in (e.g., `1=France, 2=Germany`)
3.	The user will enter a number.	The system will display a menu asking the user to select the type of army to raise (e.g., `1=Militia, 2=Spearmen, etc`)

4.	The user will enter a number.	If valid, the system will try to raise the army and provide the user with a message whether the action succeeded or not for the region, e.g.,
		`Militia raised successfully for France`
		`Militia was not raised for France since not enough gold`

Use Case Name	**Build Farm Action**	
Description	Used to build a farm in a region.	
Typical Course Of Events	**Actor Action**	**System Response**
1.	Initiated when user selects Build Farm from Build Action Menu.	If the user hasn't already chosen above the maximum number of actions, then system will display message `--- Build Farm ---`
2.		The system will display a menu asking the user to select the region to build farm in.
3.	The user will enter a number.	If valid, the system will build the farm and provide the user with a message whether the action succeeded or not for the region.

Use Case Name	**Recruit Special Unit Action**	
Description	Used to recruit a special unit in a region	
Typical Course Of Events	**Actor Action**	**System Response**
1.	Initiated when user selects Recruit Special Unit from Display Action Menu.	If the user hasn’t already chosen above the maximum number of actions, then system will display message `--- Recruit Special Unit ---`
2.		The system will display a menu asking the user to select the region to raise army in.
3.	The user will enter a number.	The system will display a menu asking the user to select the type of army to raise (e.g., `1=Diplomat, 2=Priest, etc`)

4.	The user will enter a number.	If valid, the system will try to raise the army and provide the user with a message whether the action succeeded or not for the region, e.g.,
		`Merchant raised successfully for France`
		`Priest was not raised for France since not enough gold`

Use Case Name	**Build Intermediate or Advanced Culture Action**	
Description	Used to build an intermediate or advanced culture in a region.	
Typical Course Of Events	**Actor Action**	**System Response**
1.	Initiated when user selects Build Intermediate or Build Advanced Culture from Build Action Menu.	If the user hasn’t already chosen above the maximum number of actions, then system will display appropriate message, e.g., `--- Build Intermediate Culture ---`
2.		The system will display a menu asking the user to select the region to build culture in.
3.	The user will enter a number.	If valid, the system will attempt to build the culture and provide the user with a detailed message whether the action succeeded or not for the region.

Use Case Name	**Build Fort, Keep, or Castle Action**	
Description	Used to build a fort in a region.	
Typical Course Of Events	**Actor Action**	**System Response**
1.	Initiated when user selects Build Fort or Build Keep or Build Castle from Build Action Menu.	If the user hasn’t already chosen above the maximum number of actions, then system will display appropriate message, e.g., `--- Build Fort ---`
2.		The system will display a menu asking the user to select the region to build fort/keep/castle in.
3.	The user will enter a number.	If valid, the system will attempt to build the fort/keep/castle and provide the user with a detailed message whether the action succeeded or not for the region.

Use Case Name	**Build Improved Resources Action**	
Description	Used to build improved resources in a region.	
Typical Course Of Events	**Actor Action**	**System Response**
1.	Initiated when user selects Build Improved Resources from Build Action Menu.	If the user hasn’t already chosen above the maximum number of actions, then system will display message `--- Build Improved Resources ---`
2.		The system will display a menu asking the user to select the region to build improved resources in.
3.	The user will enter a number.	If valid, the system will attempt to build the improved resources and provide the user with a detailed message whether the action succeeded or not for the region.

Use Case Name	**Move Army Action**	
Description	Used to move an army from one region to another region. You must submit a sequence diagram for this use case on March 21.	
Typical Course Of Events	**Actor Action**	**System Response**
1.	Initiated when user selects Move Army from Display Action Menu.	If the user hasn't already chosen above the maximum number of actions, then system will display message `--- Move Army ---`
2.		The system will display a menu asking the user to select the source region that contains the army.
3.	The user will enter a number.	If valid, the system will display a menu asking the user to select the destination region for the army.
4.	The user will enter a number.	If valid, the system will attempt to move the army and provide the user with a detailed message whether the action succeeded or not. If the destination region contains no enemy army units (the other player's units), then the region will be added to the player's region collection at the end of the turn. This newly conquered region will start with a control percent of 20%. If the destination region does contain enemy army units, then a battle must be fought at the end of the turn.

Use Case Name	**Move Special Unit Action**	
Description	Used to move a special unit from one region to another region	
Typical Course Of Events	**Actor Action**	**System Response**
1.	Initiated when user selects Move Special Unit from Display Action Menu.	If the user hasn’t already chosen above the maximum number of actions, then system will display message `--- Move Special Unit ---`
2.		The system will display a menu asking the user to select the source region that contains the special unit.
3.	The user will enter a number.	If valid, the system will display a menu asking the user to select the destination region for the special unit.
4.	The user will enter a number.	If valid, the system will attempt to move the special unit and provide the user with a detailed message whether the action succeeded or not.

Use Case Name	**End Turn**	
Description	Used to signal the end of the player’s turn.	
Typical Course Of Events	**Actor Action**	**System Response**
1.	Initiated when user selects End Turn from the Main Program menu.	The system will display message `--- End of Turn # ---`
2.		The system will perform the end of turn actions, including fighting any battles and incrementing the turn number. For battles, see the Fight Battle case.

Use Case Name	**Fight Battle (Optional for groups of 2)**
Description	Used to determine the winner of a battle.
Typical Course Of Events	
1.	A battle consists of multiple combat rounds. In each round, the attacker (the army that initiated the combat by moving army units into the region) fights first. If there are any defending units left after that, then the defender fights back. This process continues until one side is defeated.
2.	The algorithm for a combat round is as follows: Each attacking unit in the region determines its *combat round attack number*, which is the sum of its unit attack value plus a D6 (a random number between 1..6). Similarly, each defender determines its *combat round defense number*, which is the sum of its unit defense value plus a D6 plus the fortification value (0=none,1=fort,2=keep,3=castle). If any combat round attack numbers are greater than any of the combat round defense numbers, then remove as many defending units as there are combat round attack numbers that are larger than combat round defense numbers. If there are any defenders left, then they get to attack back. For each defending unit in the region, determine its *combat round attack number*, which is the sum of its unit attack value plus a D6 (a random number between 1..6). Similarly, each attacker determines its *combat round defense number*, which is the sum of its unit defense value plus a D6. If any combat round attack numbers are greater than any of the combat round defense numbers, then remove as many attacking units as there are combat round attack numbers that are larger than combat round defense numbers.

Note:

In order to test this use case, the computer player's home region will have to be a given one or more army units.

Summary: Due March 14

Process Menu Choice activity diagram
Process Menu Choice sequence diagram
View Info sequence diagram
New Game activity diagram
Levy Taxes sequence diagram

Summary: Due March 21

Recruit Army Action sequence diagram
Move Army Action sequence diagram
Working program

Other Changes

In order to make this game work, we will have to change some of the gold and labour unit requirements for the different buildings. The new values are as follows:

Build Fort: previously required 50 labour units, now requires 5 labour units and 5 gold

Build Keep: previously required 100 labour units, now requires 7 labour units and 7 gold

Build Castle: previously required 150 labour units, now requires 10 labour units and 10 gold

Build Intermediate Culture: previously required 60 gold, now requires 10 gold

Build Advanced Culture: previously required 135 gold, now requires 15 gold

Build Improved Resources: previously required 50 gold, now requires 8 gold

We are also going to make some changes to the army units and special units:

Militia: Gold 2, Attack 1, Defense 1
Spearman: Gold 4, Attack 1, Defense 2
Light Calvary: Gold 6, Attack 2, Defense 1
Swordsman: Gold 8, Attack 4, Defense 2
Heavy Calvary: Gold 10, Attack 5, Defense 3
Catapult: Gold 15, Attack 6, Defense 0
Priest: Gold 10
Diplomat: Gold 5
Merchant: Gold 15

Example Combat

On the next page is a sample combat split over two combat rounds.

Combat Round 1

Attacker

Swordsman
Attack: 4
Defense: 2

Combat Round Attack Value: 4 + 3 (random #)

Militia
Attack: 1
Defense: 1

Combat Round Attack Value: 1 + 1

Defender

Spearman
Attack: 1
Defense: 2

Combat Round Defense Value: 2 + 5 + 2 (Keep)

Result: No defenders destroyed

Defender

Spearman
Attack: 1
Defense: 2

Combat Round Attack Value: 1 + 4 (random #)

Attacker

Swordsman
Attack: 4
Defense: 2

Combat Round Defense Value: 2 +6

Militia
Attack: 1
Defense: 1

Combat Round Defense Value: 1 + 2

Result: Militia is destroyed

Attacker

Swordsman
Attack: 4
Defense: 2

Combat Round Attack Value: 4 + 4 (random #)

Defender

Spearman
Attack: 1
Defense: 2

Combat Round Defense Value: 2 + 2 (random) + 2 (keep)

Result: Spearman is destroyed, the attacker has conquered the region