

SOFTWARE DESIGN DESCRIPTIONS

Prepared by

Name	ID
Metin AKŐIT	1745728
Mustafa YÜKSEL	1819705
Emrah A YAZ	1448406

Date: 04.01.2015

CONTEXT

1. Overview	3
1.1 Purpose	3
1.2 Scope	3
1.3 Intended Audience	3
2. Definitions	3
2.1 References	4
3. Conceptual Model for Software Design Descriptions	4
3.1 Software Design Descriptions within the Lifecycle	4
3.2 Influences on SDD Preparation	4
4. Design Description Information Content	4
4.1 Design Languages	4
4.2 Design View	5
4.3 Design Rationale	5
5. Design Viewpoints	6
5.1 Introduction	6
5.2 Context Viewpoint	6
5.3 Use Case – User Interactions	8
5.4 Composition Viewpoint	31
5.5 Logical Viewpoint	33
5.6 Dependency Viewpoint	35
5.7 Information Viewpoint	36
5.8 Interface Viewpoint	36
5.9 Structure Viewpoint	44
5.10 Patterns Use Viewpoint	44
5.11 Interaction Viewpoint	45
5.12 State Dynamics Viewpoints	55
5.13 Algorithm Viewpoint	56
5.14 Resource Viewpoint	67

1. Overview

1.1. Purpose

The purpose of this document is to define design architecture of the system. This Software Design Descriptions report (SDD) describes the design viewpoints that are used to define data, components and interfaces for implementation phase. Software Requirements Specification Report is used as a reference that each requirement in the SRS is being traceable to design entities in the SDD.

1.2. Scope

This document defines the high level design and general explanations of the system. The use of each view is demonstrated by using UML diagrams such as Use Case, Sequence diagrams and so on. The sequence diagrams are used in the SDD in order to define event sequences between objects.

1.3. Intended Audience

This document is written for software developers, project consultants and team managers.

2. Definitions

Definitions of all terms, acronyms, and abbreviations are listed in the below table:

AI	Artificial Intelligence which determines decisions about the game and it provides game play.
Player	Person who plays and interacts the game.
Design View	A representation of one or more design elements relates to a specified design viewpoint.
Design Viewpoint	The specification of the elements available for using a design view.
Design Pattern	is a general solution to a widely occurring problem in software design.
Context Diagram	defines the boundary between the system, or part of a system, and its environment
C#	A multi-paradigm programming language
Component Diagram	depicts how components are wired together to form larger components and or software systems

Class Diagram	A type of diagram and part of a unified modeling language (UML) that defines and provides the overview and structure of a system in terms of classes, attributes and methods, and the relationships between different classes.
Package Diagram	Depicts the dependencies between the packages that make up a model.
Composite Structure Diagram	Composite structure diagram visualizes the internal structure of a class or collaboration.

Table 1 - Terms, Acronyms, Abbreviations

2.1. References

- IEEE Standard for Information Technology-Systems Design- Software Design Descriptions," IEEE STD 1016-2009
- SRS Document November 30, 2014
- Flowchart, <http://en.wikipedia.org/wiki/Flowchart>
- Composite Structure Diagram, <http://www.visual-paradigm.com/VPGallery/diagrams/CompositeStructureDiagram.html>

3. Conceptual model for software design descriptions

3.1. Software design descriptions within the life cycle

We use agile methodology, however we also prepare SRS and SDD documents in order to prepare requirement and design issues. After determining the requirements, specifications and design parts we started to implement code. Moreover, if there will any problem about implementation part, we can do maintenance of this project easily.

3.2. Influences on SDD preparation

The purpose of this SDD is to define and describe the use of each view, interaction between components and every dependency which system uses. As a result, because of the fact that software design description document is prepared with using our implementation process and methods, SDD preparation is influenced.

4. Design description information content

4.1. Design languages

In this project, Star UML 2.0 is used to draw all diagrams except flow charts and context diagram.

4.2. Design View

This part represents comprised of one or more design elements address one or more design concerns. In other words, this part demonstrates typical design concern include functionality, reliability, safety and performance.

Performance and Functionality

The application checks the charging state of the device. If the device is not in the charging state and the charge level of the phone is low, the application reduces its regular notification frequency so that the application optimizes battery usage. The application provides users with notification interval options.

The application uses low RAM. In order for the Android garbage collector to reclaim memory from the application, we will try to avoid introducing memory leaks which are usually caused by holding onto object references in global members.

CPU usage will be optimized by keeping the number of different materials per scene low. Also we will try to share as many materials between different objects as possible.

We will try to keep the game loading and action performing times short when designing our application so that the waiting time does not become unappealing for users.

The player will get informed when application crashes for some reason.

In our project is for mobile platforms, thus we design the user interfaces appropriate for mobile platforms.

Also, the application will have reasonably good graphics on 3.x-5.x inch diagonal displays, 7.x to 9.x inch diagonal mini tablets and 9.x to 12.x inch full size tablets. The text font size may need to be adjusted up (for high resolution screens) or down (for low resolution screens) so as to keep the text readable. The game will be played on landscape mode. Since we are designing it for Android, the game will be played on touch screen. The layout may need to be adjusted to increase or decrease the spacing between and around labels and widgets shown on the screen so as to prevent them from getting clustered together on high-res screens or spaced apart too much on low-res screens. Background images or background art will be provided in two different versions: a large size/high resolution version and a small size/low resolution version so that it properly fills the amount of physical space available on the screen on different devices with different sizes.

Safety and Reliability

After an interrupt occurs (phone calls, SMS, notification etc...) While the application is running so that it does not come in the way of the operating system's processing. Also, it does not result in any damage to the application's ability to function normally after the operating system finishes. The player will be able to choose to ignore or respond to an interrupt. After the user finishes handling the interrupt or after they choose to ignore the interrupt, the application resumes from the place where it was. The players get their game progress saved after playing the game.

4.3. Design Rationale

Our design consists of model classes and GUI classes which they interact with the user interface. The GUI classes manages user interface as player or AI desire. Moreover, GUI classes prepare the system in order to display actions on the user interface. On the other hand, model classes preform actions behind the scenes or user interfaces. They make decisions about the game, save or load components of the game and so on.

Furthermore, we use Unity 3D, thus our development environment is C# platform. We use serializing for save and load the game components and use singleton design pattern and subclassing in order to the fact that our code becomes more reusable and generic.

5. Design viewpoints

5.1. Introduction

This document is organized into six viewpoints. Context diagram is used to show the relationship that the system has with other external entities. To explain logical algorithm component, component diagram is used. To define event sequences interactions between objects, sequence diagrams are used. Logical and information viewpoints are used to show class structure of the system and to demonstrate database design respectively. Moreover, for making multi criteria decision, algorithm viewpoint is used.

5.2. Context viewpoint

In this view point, context and use case diagrams are used for showing relationships and interactions between system and its environment. Context and Use Case Diagrams are shown in below.

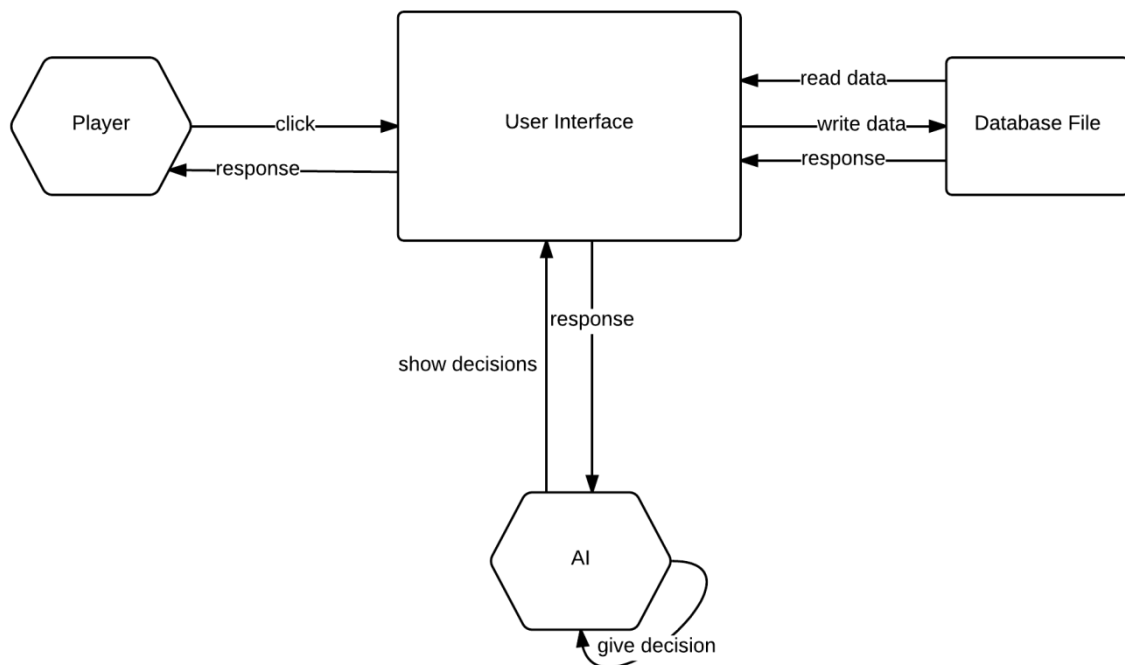


Figure 1: Context Diagram

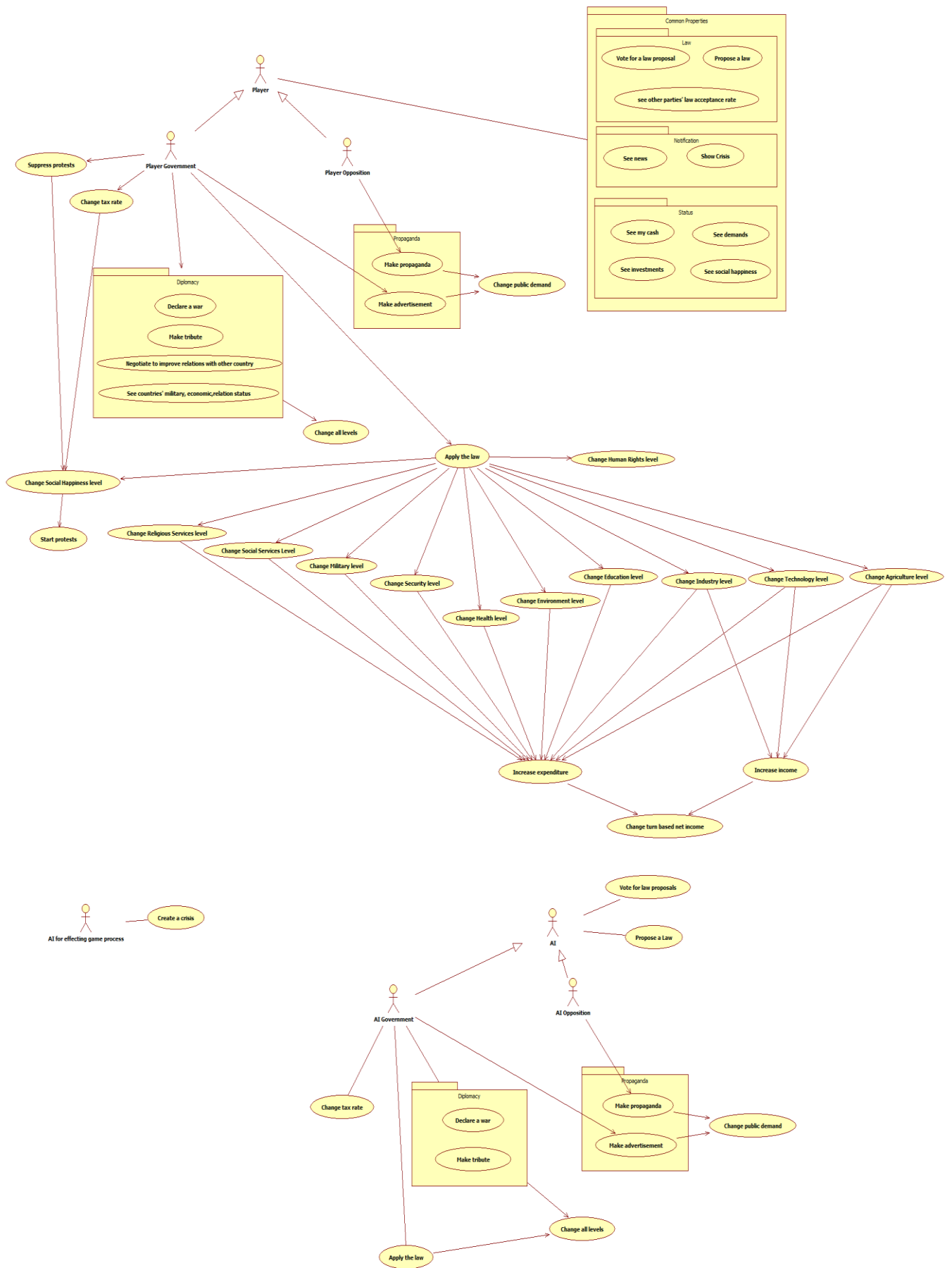


Figure 2: Use Case Diagram

5.3. Use Case – User Interactions

The relation between use cases and user interactions are shown below:

5.3.1. “Purpose a law” Use Case

Name	Purpose a law
Description	This use case describes the event of the player purposes a law.
Preconditions	None
Trigger	This use case is triggered when player want to change any level. “Purpose a law” screen opens automatically.
Basic Path	<ul style="list-style-type: none">- Player changes a level.- “Purpose a law” screen opens automatically.- Player can select “make propaganda” and/or “purpose a law” options to propose a law or make propaganda.
Post Conditions	<ul style="list-style-type: none">- Player is redirected to “Vote Result” screen in order to see law proposal’s acceptance percentage.
Error handling	None

5.3.2. “Vote for a law proposal” Use Case

Name	Vote for a law proposal
Description	This use case describes the event of the player votes for a law proposal.
Preconditions	There should be a law proposal that player can vote it.
Trigger	This use case is triggered when player passes a new turn. “Vote” screen opens automatically.
Basic Path	<ul style="list-style-type: none">- Player passes a new turn.- There is a law proposal to vote.- “Vote” screen opens automatically.- Player selects “yes” or “no” options in order to give its vote for this law proposal.
Post Conditions	<ul style="list-style-type: none">- Voting option of a law proposal is stored and this situation affects law proposal’s acceptance percentage.- Player is redirected to “Vote Result” screen in order to see law proposal’s acceptance percentage.
Error handling	None

5.3.3. "Show crisis" Use Case

Name	Show crisis
Description	This use case describes the event of showing crisis to player.
Preconditions	None
Trigger	This use case is triggered when player passes a new turn. "Show crisis" pop up opens automatically.
Basic Path	- Player passes a new turn. - "Show crisis" pop up opens automatically.
Post Conditions	Player is redirected "Main" screen after closing this popup.
Error handling	None

5.3.4. "See other parties' law acceptance rate" Use Case

Name	See other parties' law acceptance rate
Description	This use case describes the event of showing other parties' law acceptance rate to player.
Preconditions	None
Trigger	This use case is triggered when player proposes a law. "Other parties' law acceptance rate" popup opens automatically.
Basic Path	- Player proposes a law - Popup opens automatically.
Post Conditions	Player is redirected "DemandsSupplies" screen after closing this popup.
Error handling	None

5.3.5. "See news" Use Case

Name	See news
Description	This use case describes the event of the player see news about everything related to game.
Preconditions	None
Trigger	This use case is triggered when player passes a new turn. After that, player can see news on "Main" screen.
Basic Path	<ul style="list-style-type: none">- Player passes a new turn.- "Main" screen opens automatically.- Player can see news about everything related to game.
Post Conditions	None
Error handling	None

5.3.6. "See my cash" Use case

Name	See my cash
Description	This use case describes the event of the player see its cash.
Preconditions	None
Trigger	None
Basic Path	<ul style="list-style-type: none">- Player can see its cash on "Main" screen.
Post Conditions	None
Error handling	None

5.3.7. "See investments" Use Case

Name	See investments
Description	This use case describes the event of the player see investments.
Preconditions	None
Trigger	This use case is triggered when player presses "Demands & Investments" button. "Demands and Investments" screen opens automatically.
Basic Path	- Player presses "Demands & Investments" to see demands -"Demands and Investments" screen opens automatically
Post Conditions	None
Error handling	None

5.3.8. "See demands" Use Case

Name	See demands
Description	This use case describes the event of the player see demands.
Preconditions	None
Trigger	This use case is triggered when player presses "Demands & Investments" button. "Demands and Investments" screen opens automatically.
Basic Path	- Player presses "Demands & Investments" to see demands -"Demands and Investments" screen opens automatically
Post Conditions	None
Error handling	None

5.3.9. “See social happiness” Use Case

Name	See social happiness
Description	This use case describes the event of the player see social happiness rate.
Preconditions	None
Trigger	None
Basic Path	- Player can see social happiness rate on “Main” screen.
Post Conditions	None
Error handling	None

5.3.10. “See countries’ military, economic and relation status” Use Case

Name	See countries’ military and economic status
Description	This use case describes the event of the player see countries’ military, economic and relation status.
Preconditions	None
Trigger	This use case is triggered when player presses “Diplomacy” button. “Diplomacy” screen opens automatically.
Basic Path	- Player presses “Diplomacy” to see countries’ properties. -“Diplomacy” screen opens automatically.
Post Conditions	None
Error handling	None

5.3.11. “Change tax rate” Use Case

Name	Change tax rate
Description	This use case describes the event of the player government changes tax rate.
Preconditions	None
Trigger	This use case is triggered when player government presses “Tax” button. “Tax” screen opens automatically.
Basic Path	<ul style="list-style-type: none">- Player presses “Tax” to change tax rate.- “Tax” screen opens automatically.- Player can increase or decrease tax rate on “Tax” screen.
Post Conditions	None
Error handling	None

5.3.12. “Suppress protests” Use Case

Name	Suppress protests
Description	This use case describes the event of the player government suppress protests
Preconditions	There should be a protest against player government.
Trigger	This use case is triggered when player government passes a new turn. “Suppress protests” popup opens automatically.
Basic Path	<ul style="list-style-type: none">- There is a protest against player government.- “Suppress protests” popup opens automatically.- Player governments can select whether “Suppress protests” or not.
Post Conditions	Player government is redirected to “Main” screen after closing this popup.
Error handling	None

5.3.13. “Declare a war” Use Case

Name	Declare a war
Description	This use case describes the event of the player government declares a war.
Preconditions	None
Trigger	This use case is triggered when player government presses “Declare war” button on “Diplomacy” screen.
Basic Path	- Player government presses “Declare war” button to declare a war.
Post Conditions	Player government is redirected to “War result” screen and it can see whether win or lose war.
Error handling	None

5.3.14. “Make Tribute” Use Case

Name	Make Tribute
Description	This use case describes the event of the player government initiate negotiation.
Preconditions	None
Trigger	This use case is triggered when player government presses “Negotiate” button on “Diplomacy” screen.
Basic Path	- Player government presses “Make Tribute” button to declare a war.
Post Conditions	Player government is redirected to “Make Tribute result” screen and it can see whether its demands are provided or not.
Error handling	None

5.3.15. “Negotiate to improve relations with other country” Use Case

Name	Negotiate to improve relations with other country
Description	This use case describes the event of the player government improves relations with other country.
Preconditions	None
Trigger	This use case is triggered when player government presses “Improve relation” button on “Diplomacy” screen.
Basic Path	- Player government presses “Improve relation” button to improve relation with other country. - Opens “Improve relation” screen automatically and player government can select other country’s level that will increase in order to improve relation between them.
Post Conditions	Player government is redirected to “Improve relation result” screen and it can see how much increase their relationship level.
Error handling	None

5.3.16. “Change all levels” Use Case

Name	Change all levels
Description	This use case describes the event of changing all levels.
Preconditions	There should be a war, negotiation or improve relation situations.
Trigger	This use case is triggered when player government declare a war, initiate negotiation or improve relation.
Basic Path	- Player government’s country’s levels are changed after some situations were happened.
Post Conditions	None
Error handling	None

5.3.17. “Make advertisement” Use Case

Name	Make advertisement
Description	This use case describes the event of the player government makes advertisement to change public demand.
Preconditions	There should be a law proposal.
Trigger	This use case is triggered when player government proposes a law proposal.
Basic Path	-Player government can make advertisement on “Purpose a law” screen.
Post Conditions	Player government can make whether advertisement or not from this screen and “Vote” screen opens automatically.
Error handling	None

5.3.18. “Make propaganda” Use Case

Name	Make propaganda
Description	This use case describes the event of the player opposition makes propaganda to change public demand.
Preconditions	There should be a law proposal.
Trigger	This use case is triggered when player opposition proposes a law proposal.
Basic Path	-Player opposition can make propaganda on “Purpose a law” screen.
Post Conditions	Player opposition can make whether propaganda or not from this screen and “Vote” screen opens automatically.
Error handling	None

5.3.19. “Change public demand” Use Case

Name	Change public demand
Description	This use case describes the event of changing public demand.
Preconditions	There should be an advertisement or a propaganda situation.
Trigger	This use case is triggered when player government makes an advertisement or player opposition make propaganda.
Basic Path	-Player government’s or player opposition’s demands are changed after some situations were happened.
Post Conditions	None
Error handling	None

5.3.20. “Apply the law” Use Case

Name	Apply the law
Description	This use case describes the event of applying a law.
Preconditions	There should be an accepted law proposal.
Trigger	This use case is triggered when a law proposal is accepted.
Basic Path	-A law proposal is accepted. -Player government applies this law.
Post Conditions	Player government’s country’s levels could be changed.
Error handling	None

5.3.21. “Change Religious Services level” Use Case

Name	Change Religious Services level
Description	This use case describes the event of changing religious services level.
Preconditions	There should be a law that is applied.
Trigger	This use case is triggered when a law is applied.
Basic Path	-A law is applied. -Religious service level is changed.
Post Conditions	Expenditure is increased by this situation.
Error handling	None

5.3.22. “Change Social Services level” Use Case

Name	Change Social Services level
Description	This use case describes the event of changing social services level.
Preconditions	There should be a law that is applied.
Trigger	This use case is triggered when a law is applied.
Basic Path	-A law is applied. -Social service level is changed.
Post Conditions	Expenditure is increased by this situation.
Error handling	None

5.3.23. “Change Military level” Use Case

Name	Change Military level
Description	This use case describes the event of changing military level.
Preconditions	There should be a law that is applied.
Trigger	This use case is triggered when a law is applied.
Basic Path	-A law is applied. -Military level is changed.
Post Conditions	Expenditure is increased by this situation.
Error handling	None

5.3.24. “Change Security level” Use Case

Name	Change Security level
Description	This use case describes the event of changing security level.
Preconditions	There should be a law that is applied.
Trigger	This use case is triggered when a law is applied.
Basic Path	-A law is applied. -Security level is changed.
Post Conditions	Expenditure is increased by this situation.
Error handling	None

5.3.25. “Change Health level” Use Case

Name	Change Health level
Description	This use case describes the event of changing health level.
Preconditions	There should be a law that is applied.
Trigger	This use case is triggered when a law is applied.
Basic Path	-A law is applied. -Health level is changed.
Post Conditions	Expenditure is increased by this situation.
Error handling	None

5.3.26. “Change Environment level” Use Case

Name	Change Environment level
Description	This use case describes the event of changing environment level.
Preconditions	There should be a law that is applied.
Trigger	This use case is triggered when a law is applied.
Basic Path	-A law is applied. -Environment level is changed.
Post Conditions	Expenditure is increased by this situation.
Error handling	None

5.3.27. “Change Education level” Use Case

Name	Change Education level
Description	This use case describes the event of changing education level.
Preconditions	There should be a law that is applied.
Trigger	This use case is triggered when a law is applied.
Basic Path	-A law is applied. -Education level is changed.
Post Conditions	Expenditure is increased by this situation.
Error handling	None

5.3.28. “Change Industry level” Use Case

Name	Change Industry level
Description	This use case describes the event of changing industry level.
Preconditions	There should be a law that is applied.
Trigger	This use case is triggered when a law is applied.
Basic Path	-A law is applied. -Industry level is changed.
Post Conditions	Expenditure and also income are increased by this situation.
Error handling	None

5.3.29. “Change Technology level” Use Case

Name	Change Technology level
Description	This use case describes the event of changing technology level.
Preconditions	There should be a law that is applied.
Trigger	This use case is triggered when a law is applied.
Basic Path	-A law is applied. -Technology level is changed.
Post Conditions	Expenditure and also income are increased by this situation.
Error handling	None

5.3.30. “Change Agriculture level” Use Case

Name	Change Agriculture level
Description	This use case describes the event of changing agriculture level.
Preconditions	There should be a law that is applied.
Trigger	This use case is triggered when a law is applied.
Basic Path	-A law is applied. -Agriculture level is changed.
Post Conditions	Expenditure and also income are increased by this situation.
Error handling	None

5.3.31. “Change Human Rights level” Use Case

Name	Change Human Rights level
Description	This use case describes the event of changing human rights level.
Preconditions	There should be a law that is applied.
Trigger	This use case is triggered when a law is applied.
Basic Path	-A law is applied. -Human rights level is changed.
Post Conditions	None
Error handling	None

5.3.32. “Increase expenditure” Use Case

Name	Increase expenditure
Description	This use case describes the event of increasing expenditure.
Preconditions	There should be a level changing situation.
Trigger	This use case is triggered when a level is changed.
Basic Path	-A level is changed. -Expenditure increases.
Post Conditions	Turn based net income is changed by this situation.
Error handling	None

5.3.33. "Increase income" Use Case

Name	Increase income
Description	This use case describes the event of increasing income.
Preconditions	There should be a level changing situation.
Trigger	This use case is triggered when a level is changed.
Basic Path	-A level is changed. -Income increases.
Post Conditions	Turn based net income is changed by this situation.
Error handling	None

5.3.34. "Change turn based net income" Use Case

Name	Change turn based net income
Description	This use case describes the event of changing turn based net income.
Preconditions	There should be a situation that is increased expenditure and/or increased income.
Trigger	This use case is triggered when a level is changed.
Basic Path	-Expenditure is increased and/or income is increased. -Turn based net income changes.
Post Conditions	None
Error handling	None

5.3.35. “Change Social Happiness level” Use Case

Name	Change Social Happiness level
Description	This use case describes the event of changing social happiness level.
Preconditions	There should be a situation that is suppressed protests and/or changed tax rate and/or applied the law.
Trigger	This use case is triggered when suppressed protests and/or changed tax rate and/or applied the law situations were happened.
Basic Path	-Some situations were happened. -Social happiness level changes.
Post Conditions	Protests start after this situation happens.
Error handling	None

5.3.36. “Start protests” Use Case

Name	Start protests
Description	This use case describes the event of starting protests.
Preconditions	There should be social happiness level changings.
Trigger	This use case is triggered when social happiness level is changed.
Basic Path	-Social happiness level is changed. -Protests start.
Post Conditions	None
Error handling	None

5.3.37. “Create a crisis” Use Case

Name	Create a crisis
Description	This use case describes the event of creating a crisis by AI automatically.
Preconditions	None
Trigger	None
Basic Path	-AI determines to create a crisis.
Post Conditions	Some harmful situation will happen in country of player.
Error handling	

5.3.38. “Vote for a law proposal” Use Case

Name	Vote for a law proposal
Description	This use case describes the event of the AI votes for a law proposal.
Preconditions	There should be a law proposal that AI can vote it.
Trigger	This use case is triggered when player passes a new turn.
Basic Path	- Player passes a new turn. - There is a law proposal to vote. - AI selects “yes” or “no” options in order to give its vote for this law proposal.
Post Conditions	- Voting option of a law proposal is stored and this situation affects law proposal’s acceptance percentage.
Error handling	None

5.3.39. "Purpose a law" Use Case

Name	Purpose a law
Description	This use case describes the event of the AI purposes a law.
Preconditions	None
Trigger	This use case is triggered when AI want to change any level.
Basic Path	- AI changes a level. - AI can make propaganda and/or purpose a law
Post Conditions	None
Error handling	None

5.3.40. "Change tax rate" Use Case

Name	Change tax rate
Description	This use case describes the event of the AI government changes tax rate.
Preconditions	None
Trigger	None
Basic Path	- AI government can increase or decrease tax rate.
Post Conditions	None
Error handling	

5.3.41. “Declare a war” Use Case

Name	Declare a war
Description	This use case describes the event of the AI government declares a war.
Preconditions	None
Trigger	None
Basic Path	- AI government declares a war.
Post Conditions	None
Error handling	None

5.3.42. “Initiate negotiation” Use Case

Name	Initiate negotiation
Description	This use case describes the event of the AI government initiate negotiation.
Preconditions	None
Trigger	None
Basic Path	- AI government initiates negotiation.
Post Conditions	None
Error handling	None

5.3.43. “Apply the law” Use Case

Name	Apply the law
Description	This use case describes the event of applying a law.
Preconditions	There should be an accepted law proposal.
Trigger	This use case is triggered when a law proposal is accepted.
Basic Path	-A law proposal is accepted. -AI government applies this law.
Post Conditions	AI government’s country’s levels could be changed.
Error handling	None

5.3.44. “Change all levels” Use Case

Name	Change all levels
Description	This use case describes the event of changing all levels.
Preconditions	There should be a war, negotiation or improve relation situations.
Trigger	This use case is triggered when AI government declare a war, initiate negotiation or apply the law.
Basic Path	- AI government’s country’s levels are changed after some situations were happened.
Post Conditions	None
Error handling	None

5.3.45. “Make advertisement” Use Case

Name	Make advertisement
Description	This use case describes the event of the AI government makes advertisement to change public demand.
Preconditions	There should be a law proposal.
Trigger	This use case is triggered when AI government proposes a law proposal.
Basic Path	-AI government can make advertisement.
Post Conditions	None
Error handling	None

5.3.46. “Make propaganda” Use Case

Name	Make propaganda
Description	This use case describes the event of the AI opposition makes propaganda to change public demand.
Preconditions	There should be a law proposal.
Trigger	This use case is triggered when AI opposition proposes a law proposal.
Basic Path	-AI opposition can make propaganda.
Post Conditions	None
Error handling	None

5.3.47. “Change public demand” Use Case

Name	Change public demand
Description	This use case describes the event of changing public demand.
Preconditions	There should be an advertisement or a propaganda situation.
Trigger	This use case is triggered when AI government makes an advertisement or AI opposition make propaganda.
Basic Path	-AI government’s or AI opposition’s demands are changed after some situations were happened.
Post Conditions	None
Error handling	None

5.3.1. Design concerns

Design concern of the context viewpoint is to identify design subjects offered services, its users and other stakeholders to establish the system boundary.

5.3.2. Design elements

- Player interacts with UserInterface and UserInterface shows everything about vision of the game to the player.
- AI gives decisions about game play.
- Database File store saved game instances.

5.4 Composition viewpoint

The composition viewpoint describes how the framework is hierarchically structured into its constituent components by using component diagram. The following diagram addresses the design concern and describes the design elements involved and any implementation notes. Logical decomposition is demonstrated as component diagram in below.

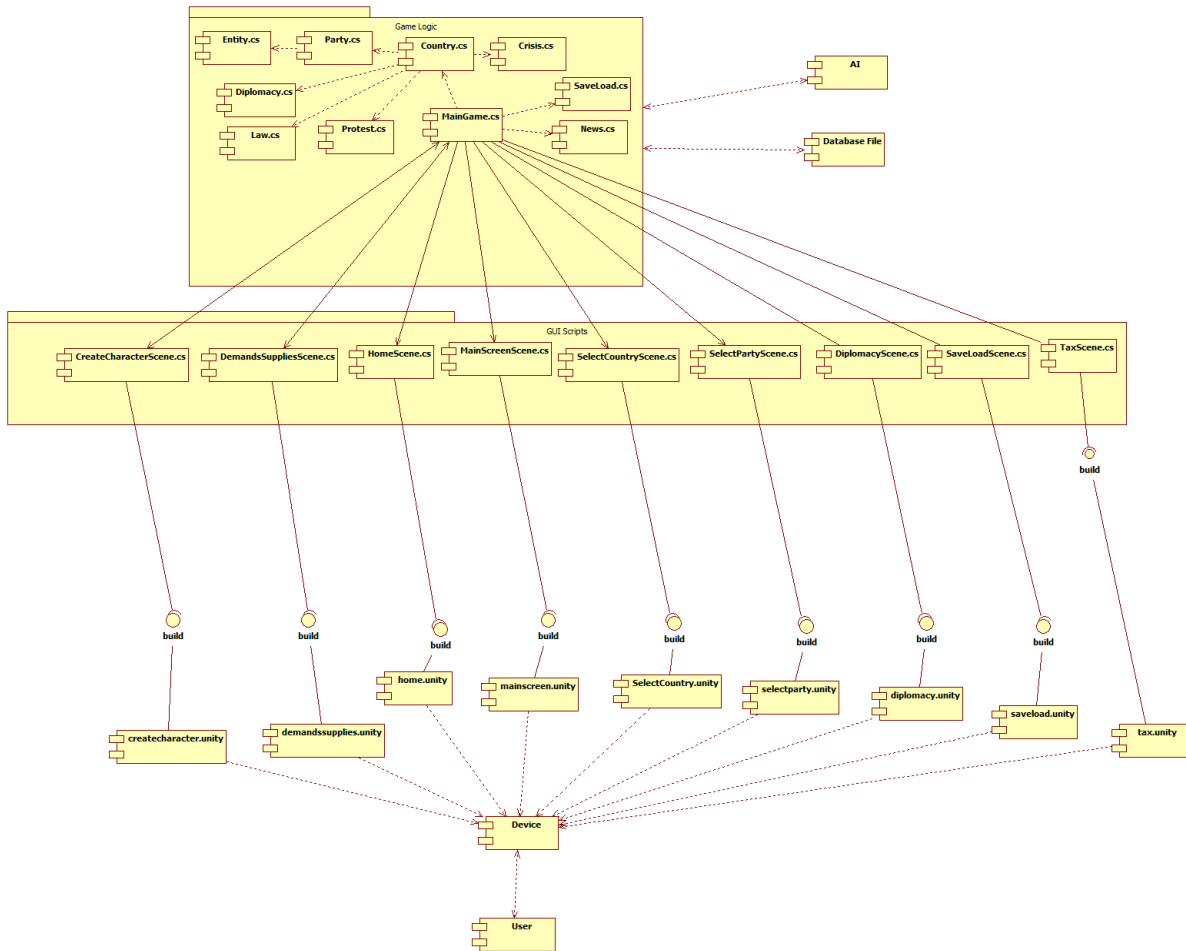


Figure 3: Component Diagram

5.4.1 Design concerns

Its design concern is:

- The composition of the system in terms of subsystems and components

5.4.2 Design elements

- In a C# script file named MainGame.cs includes a singleton class. Country.cs, SaveLoad.cs and News.cs have a dependency to it. Moreover, MainGame.cs has an association with all scene scripts.
- Party.cs, Law.cs, Protest.cs, Diplomacy.cs and Crisis.cs all have a dependency to Country.cs. It has relationships with important components of the system.
- Files having .unity extensions are scene components and user interacts them with using its device in this system. Scene scripts build .unity files and thus, them. Unity files are pushed to Device component.

5.5 Logical viewpoint

Logical viewpoint is used to show classes, interfaces and their relationships between each other. Class diagram is demonstrated below:

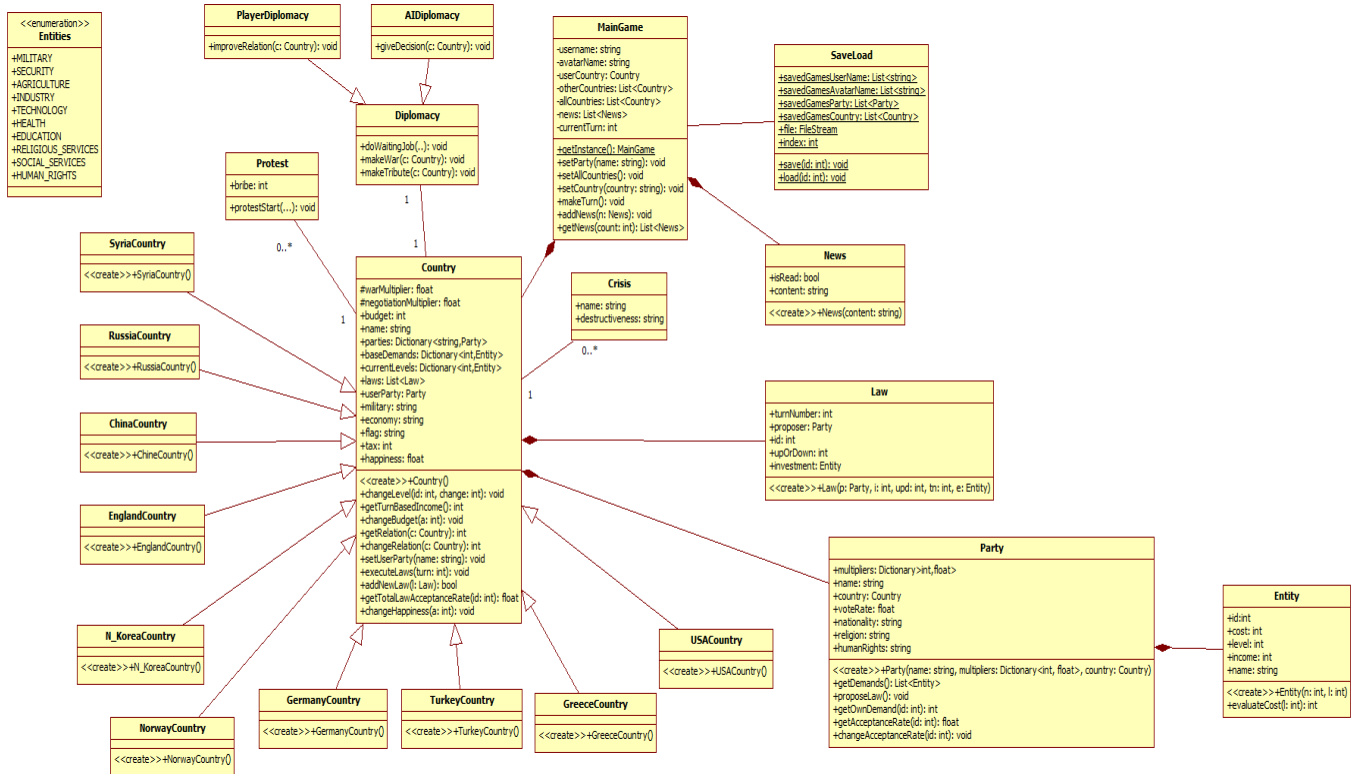


Figure 4: Class Diagram

5.5.1 Design concerns

Its design concerns are:

- Classes, interfaces and their relationships
- Reuse of types and implementations

5.5.2 Design elements

In Software Requirements Specification document, there is a detailed explanation about design elements of the Class Diagram. Apart from this, singleton design pattern is used in order to make code more efficient, reusable and understandable. MainGame class is a singleton class and it has static getInstance() method.

In this project, there are several objects such as Country, Political Party, Diplomacy, and News and so on. Let's analyze them one by one with their properties and capabilities.

The Game Object: This is the main object which manipulates all the game structure above from everything else. It has methods like getInstance, makeTurn, addNews, setCountry and so on. This object will be created automatically by Unity Game Engine and will be manipulated by the developers.

The Country Object: This is the most important object in this game design. Every action in the game will take place either in Country object (laws, news and so on) or between two Country objects (diplomacy).

It has political parties, investments, budget, news system, tax rate and so on. It has methods to manipulate these values.

Every country has some unique properties called warMultiplier and negotiationMultiplier which implicates the country's desire to make wars with other countries or want tributes from other countries respectively. It has also a property called happiness which is basically important to all political issues in that country. baseDemands describes that the real people's desire of investment levels in that country. It is invisible to any party and used only to calculate political parties' own demand visions.

Entity: This is an object that describes an investment type in a country. It has a level, an increasing level cost while going through top (100 level is top at the moment), and an income value which implies that the money that will be added or subtracted from budget in every turn.

There are currently 10 different types of investments. The Agriculture, Industry and Technology have positive income rates. The Human Rights has no effect on income and the other 6 type has negative income rates, meaning that they are draining budget every turn.

All increase on these investments is done through the law system while decrease can be done by the government for money needs or result of a war or a crisis.

Party: This object describes the political party in a country. Currently there are 6 parties in each country with initially different vote rates: The Nationalist Party, The Republican Party, The Liberal Party, The Conservative Party, The Socialist Party and The Environmental Party.

Each party has its own public investment demand vision calculated according to party ideologies. These ideology coefficients are stored in Multipliers section.

A party can propose (offer) a law to increase an investment level according to its own demand vision and budget and vote for a law proposal to pass or not.

If the party is the government, it should struggle to develop country, to make public happy, to keep itself in power etc. In order to do that, it should pass laws, make a good diplomacy with other countries, be careful about protests and arrange tax rates. If the party is an opposition, it should struggle to become the government.

Election: The elections hold in the game regularly unless there is a run for early election caused by a protest. This object redistributes vote rates for each party, thus there may be a change in the government.

Protest: If the citizens of the country is not satisfied the management, their happiness will decrease according to several reasons such as non-supply of the investment demands, high tax rates and so on; there may be a protest against the government. In this step, the government may suppress it if it has the enough bribes (money), or the protest may cause an early election.

Diplomacy: This object shows a country's relations and actions with other countries. Only the governments decide what will be done, but all the country would be affected by its consequences. There are several options for actions: making war, demanding tribute, improving relations (only for human player). The results of an action will be determined in the next turn, doWaitingJob is responsible from that.

Crisis: There are crises in the game which may happen rarely, but its results will be devastating. Some of them have reasons (pollution is a result of highly ignorance of environmental investments); some of them have not (like natural disasters).

According to its destructiveness, it may take down the government or just break its plans to attack another country for a while.

News: This object is designed for human players to keep them aware of what is going on in the game. It has different type of news ranging from accepted laws to war results, from election results to crises. In the main screen of the game, the latest news will be shown to human player.

SaveLoad: This object is designed for save or load game objects in order to player can play a saved game easily with using save and load methods.

5.6 Dependency viewpoint

Dependency viewpoint provides overall picture of the design subject in order to assess the impact of the requirements or design changes with package diagram and component diagram. Component diagram is already shown in part 5.3 which is Composition Viewpoint. Thus, it is no need to show it again in this part, however package diagram is demonstrated below.

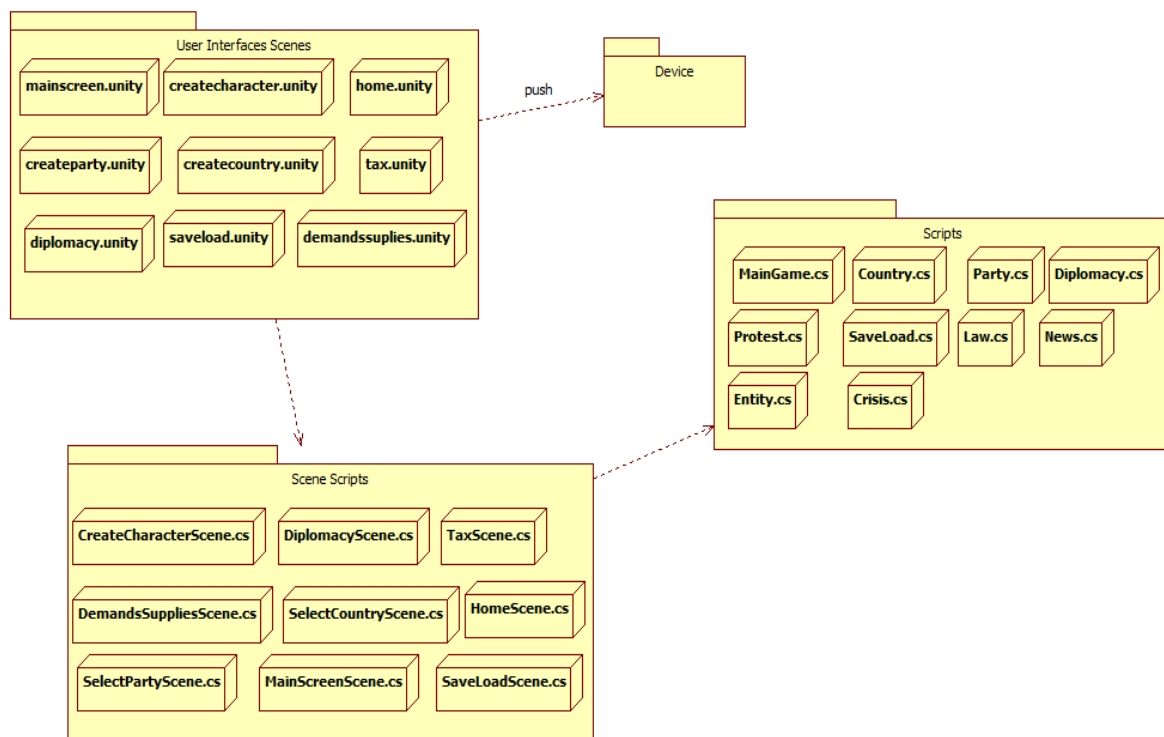


Figure 5: Package Diagram

5.6.1 Design Concerns

Its design concerns are:

- Interconnection, sharing and parameterization
- Isolate system failures

5.6.2 Design elements

- In Package Diagram, there are 4 packages that are categorized to UserInterfacesScenes, Device, Scripts, SceneScripts and all of the components are stored in these packages.

5.7 Information viewpoint

In this project, actually there is no database. There are only a few database files that store saved game instances and these files are in application data path, therefore user cannot access and/or change them. If user wants to load a saved game, the system loads this saved game by interacting with database file. We use serializing for doing this operation. We serialize some game objects such as user country, user party, user name or user avatar and so on in order to save or load these objects. If user wants to open a new game, user should save its game from SaveScene and Game objects are saved in SaveScene.cs script. Also, if user want to load a saved game (pressing load game button from home screen), LoadScene.cs class load the game. As a result, there is no ER diagram to show how database works.

5.7.1 Design Concerns

Its design concerns are:

- Persistent data structure
- Data content
- Data access schemes
- Definition of metadata

5.8 Interface viewpoint

Interface viewpoint provides information that how to correctly use the services provided by the design subject. This viewpoint can be supported by component diagram which is in section 5.3.

About the User Interface

The first screen is “home screen” and user can select new game to create a new game, load game to load saved game, settings, credits and quit from this screen. If user presses new game button, user is redirected to “create character screen”. In this screen, user can select his or her avatar and enter his or her nickname. User selects its country and party in “country screen” and “party screen” respectively. After user selects its country and party as his or her desires, user encounters “save screen”. In this scene, user can save his or her game. Then, “main screen” opens automatically. This screen includes user’s nickname, user’s avatar, user’s party, user’s country’s money, social happiness rate, news parts. In news part, user can see all the news about game process. Moreover, there are 4 buttons in this screen. These are demands & investments, tax, diplomacy, quit and next turn buttons. User can change tax rate with using tax buttons and “tax screen” opens automatically. With pressing demands & investments button, “demands screen” opens and in this screen, user can see demands and investment and also compare to them. Furthermore, from this screen, user can increase 1 point of one investment and thanks to it, propose a law. If user presses increase button for proposing a law, popup opens and user can see other parties accepted rate for this propose. On the other hand, if user presses diplomacy button, “diplomacy screen” opens and in this screen, user can declare a war, initiate negotiation or improve relations (if user is government) to another country. If user wants to propose a law, “propaganda screen” opens automatically. “propaganda screen” consists of parties’ demands about proposed law and user can make propaganda with pressing make propaganda button if he or she wants. User can see vote results from “vote result screen”. Besides, any proposed laws can be voted and it is seen from “vote screen”. This screen may be seen both cases that are other parties’ proposals and user’s proposals. Also, user can adjust game settings from “settings screen”. CreateCharacter scene is managed by CreateCharacterScene.cs class. In this class, avatar image and inputted user name from input field are stored in game objects. SelectCountry scene’s script is SelectCountryScene.cs and in this script, countries flags and names are

changed dynamically and user's selected country is stored in a game object. Likewise, in SelectPartyScene.cs controls the SelectPartyScene that user's selected party is stored and parties properties are changed dynamically as well. Game objects are saved in SaveScene.cs script. Also, if user wants to load a saved game, LoadScene.cs class load the game. In Mainscren, by using Mainscren.cs, "Tax", "Diplomacy", "Demands&Supplies", "next Turn" and "quit" buttons can be worked. Every next turn, game objects are saved automatically. With using DemandsSuppliesScene.cs, other parties accepted rates for a propose are fetched and it helps to user sees the country's levels. Furthermore, TaxScene.cs changes tax rate as user's desires.



Figure 6 – Home Scene



Figure 7 – CreateCharacter Scene



Figure 8 – SelectCountry Scene



Figure 9 – SelectParty Scene



Figure 10 – Save Scene



Figure 11 – MainScene Scene

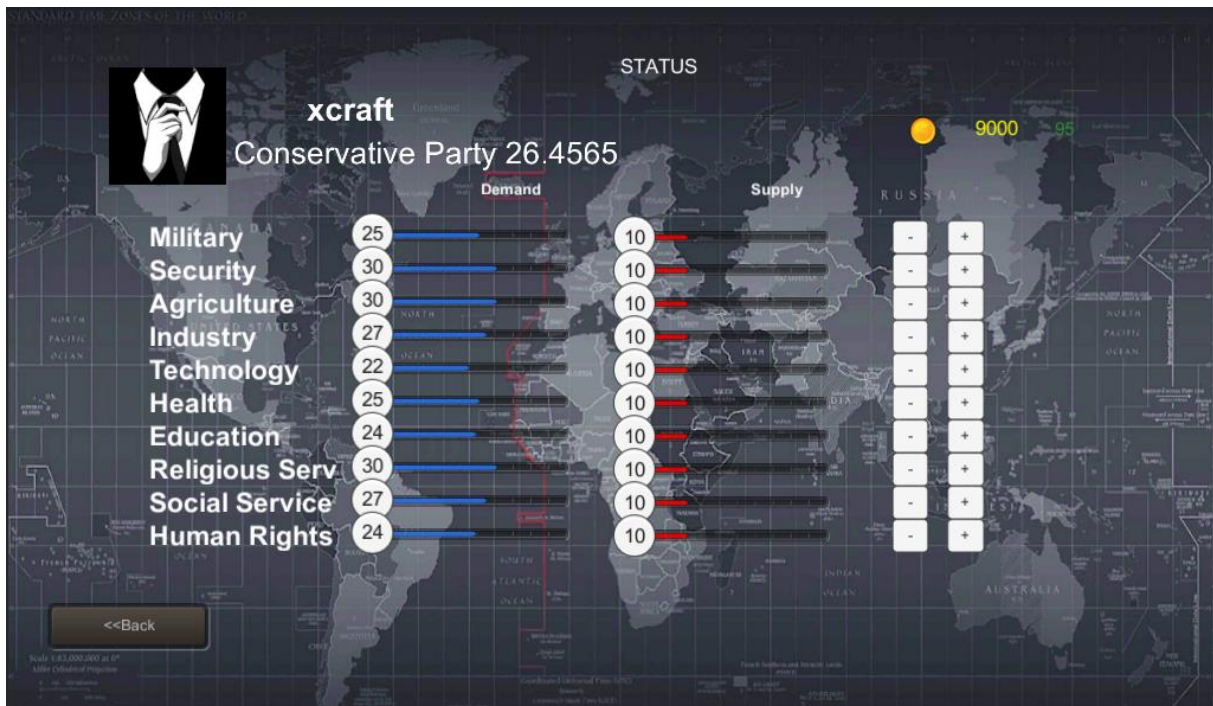


Figure 12 – Demands&Supplies Scene

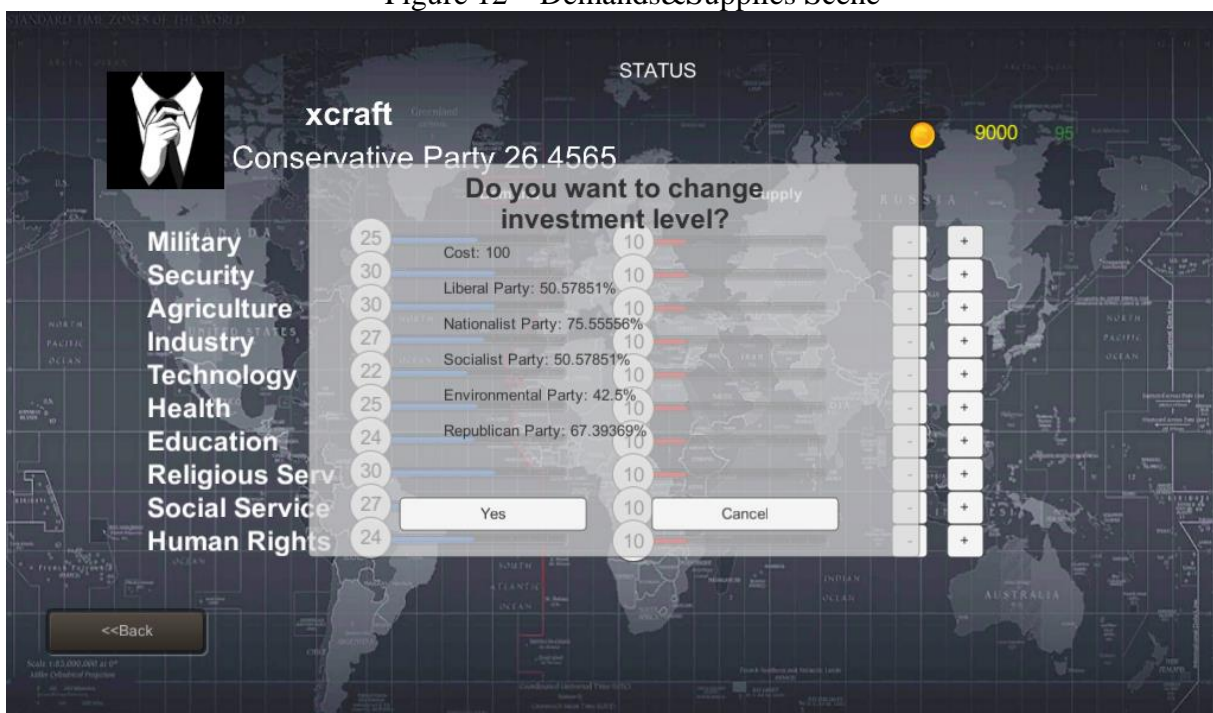


Figure 13 – Other parties' law acceptance rate popup



Figure 14 – News panel on the MainScreen Scene



Figure 15 – Tax Scene



Figure 16 – Quit Scene

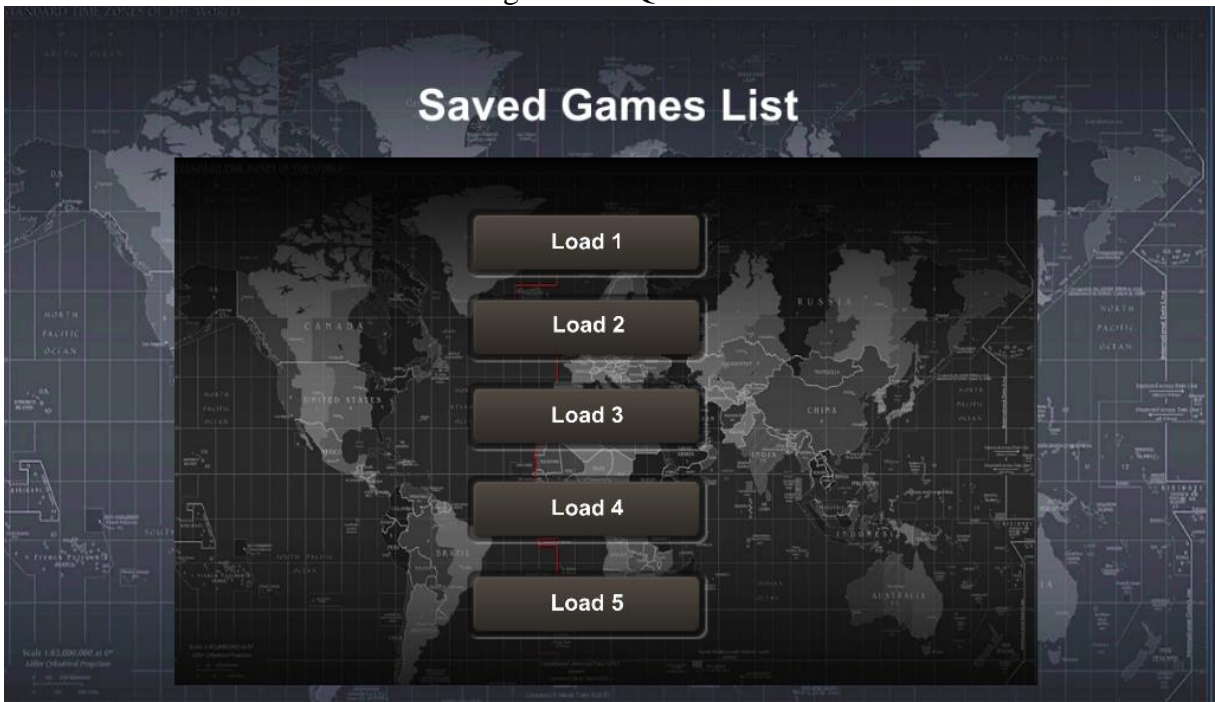


Figure 17 – Load Scene

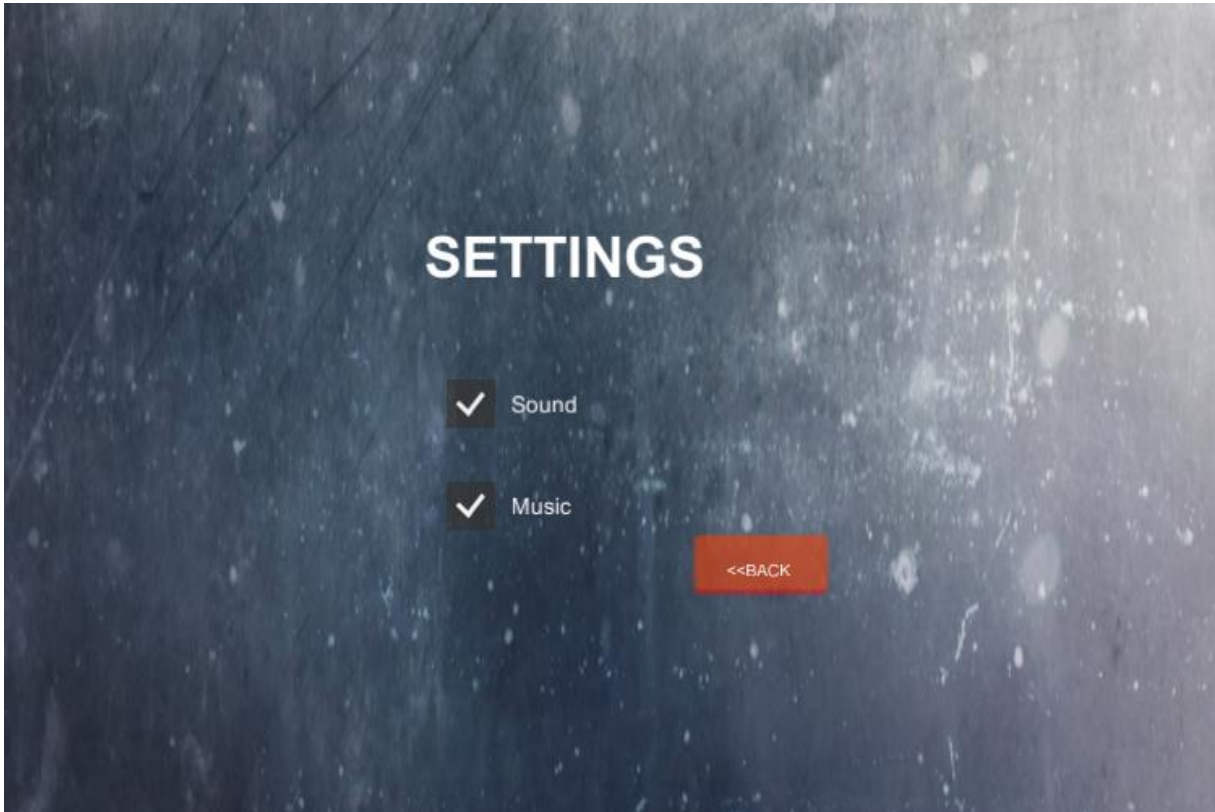


Figure 18 – Settings Scene



Figure 19 – Diplomacy Scene

5.8.1 Design concerns

Its design concerns are:

- Service definition
- Service access

5.8.2 Design elements

Design elements are also given in section 5.3.2.

5.9 Structure viewpoint

Composite structure diagram is shown in below.

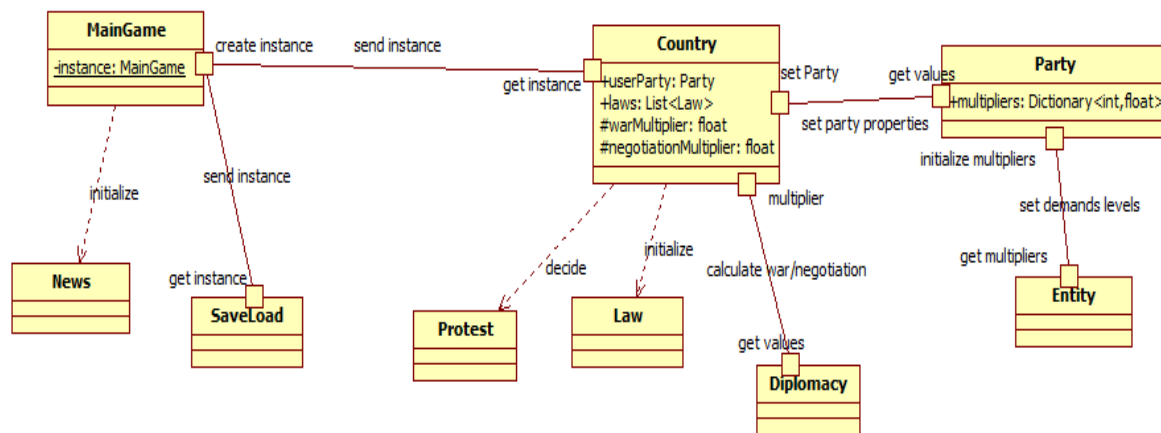


Figure 20: Composite Structure Diagram

5.9.1 Design concerns

Its design concerns are:

- Service definition
- Service access

5.9.2 Design elements

- MainGame is a component that includes initialized variables and instances. It is the most important part and some game variables, properties are accessed with using MainGame component.
- Country component has a multiplier and it sends calculate war/negotiation knowledge to Diplomacy component.
- Diplomacy component gets values, properties about war or negotiation.
- Country component also set party properties and Party component has a relationship with entity. Entity component consists of multipliers about all kinds of levels.

5.10 Patterns Use viewpoint

This viewpoint's aim is to address design ideas as collaboration patterns involving abstracted roles and connectors. Composite structure diagram is used for explain to this aim and this diagram is already detailed in Structure viewpoint part 5.9.

5.10.1 Design concerns

Its design concerns are:

- Reuse of patterns
- Available Framework template

5.10.2 Design elements

Design elements are also given in section 5.9.2.

5.11 Interaction viewpoint

This part defines strategies for interaction among entities. This viewpoint is supported by sequence diagrams. Sequence diagrams are categorized by using use cases. Moreover, sequence diagrams are drawn in order to show interaction between system player, AI, scenes and objects. There are 20 sequence diagrams to show these interactions. Sequence diagrams are demonstrated as below:

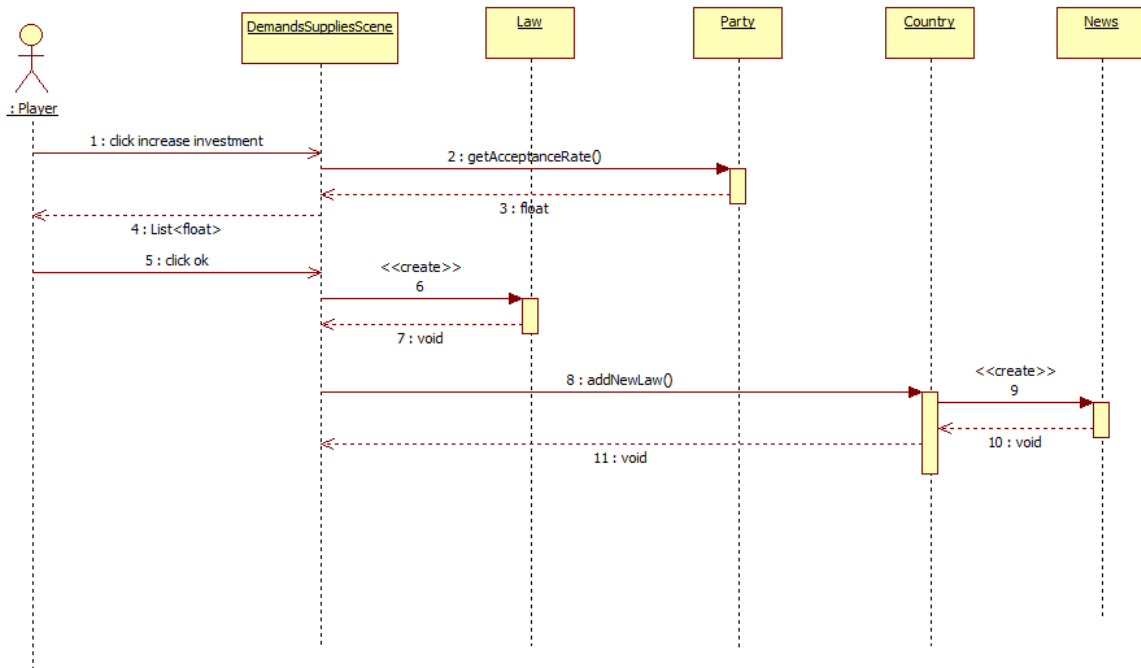


Figure 21: Player - Purpose a law Sequence Diagram

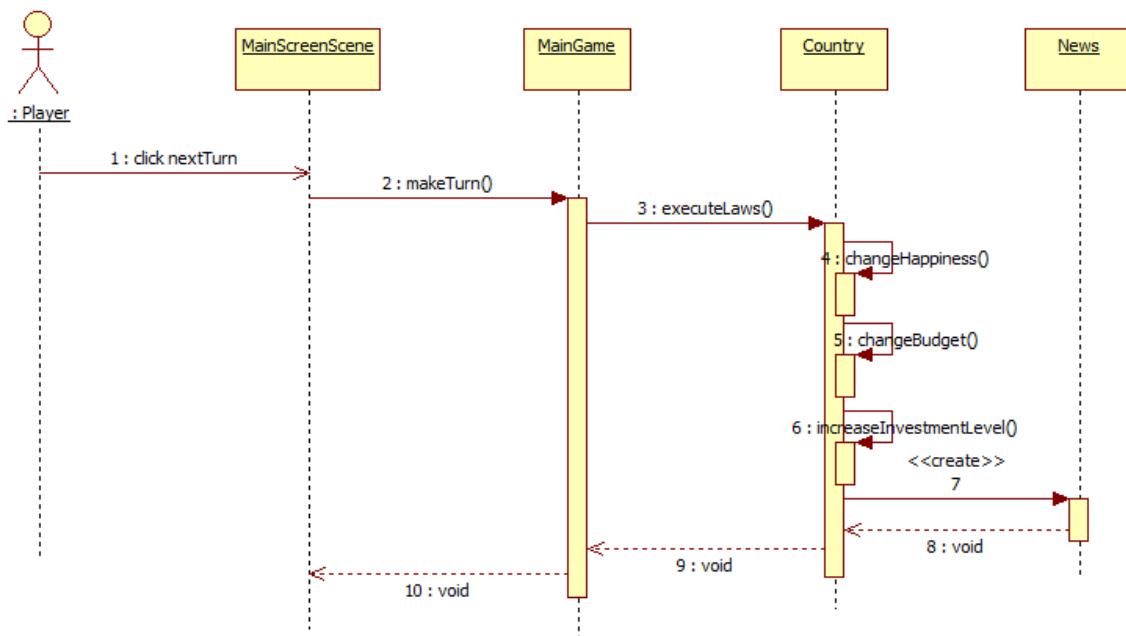


Figure 22: Pass a law Sequence Diagram

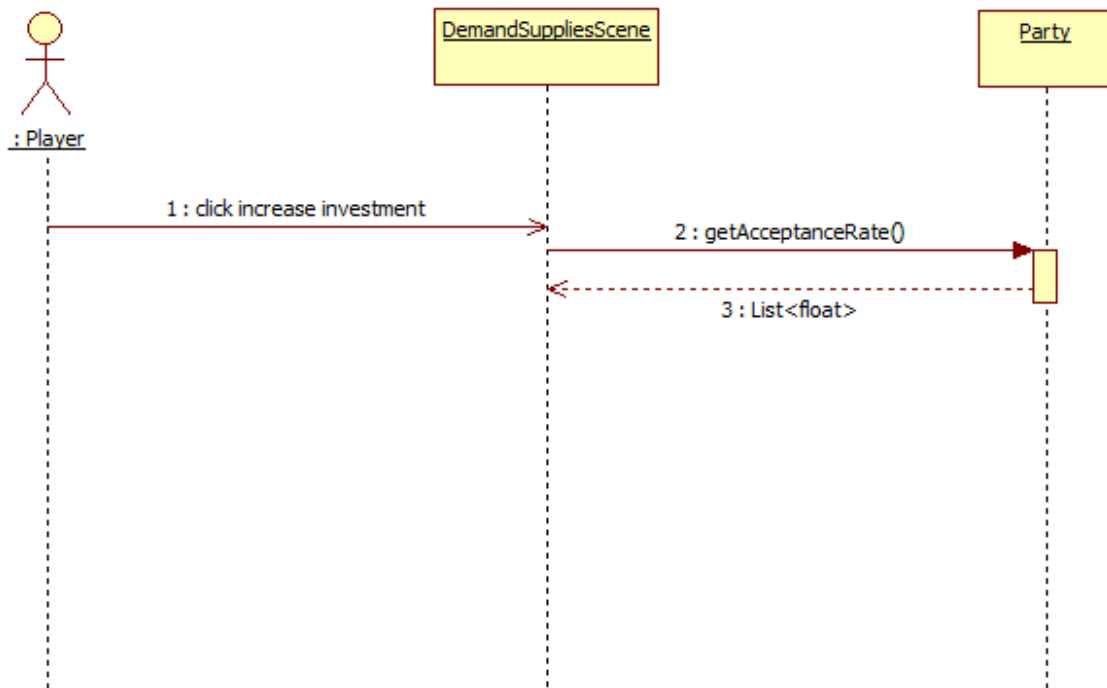


Figure 23: Player - See other parties' law acceptance rate Sequence Diagram

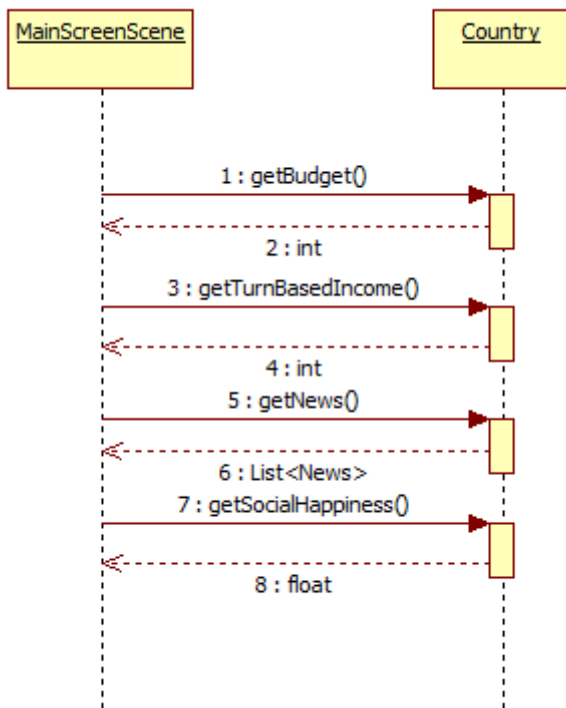


Figure 24: Player – See news, cash, social happiness Sequence Diagram

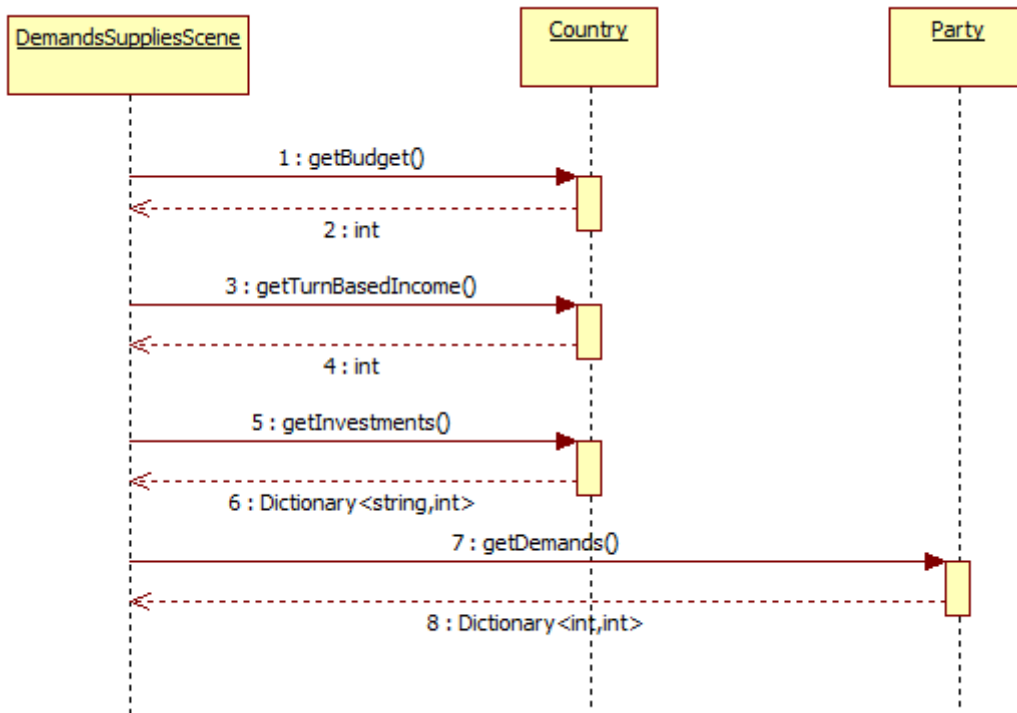


Figure 25: Player – See demands and investments Sequence Diagram

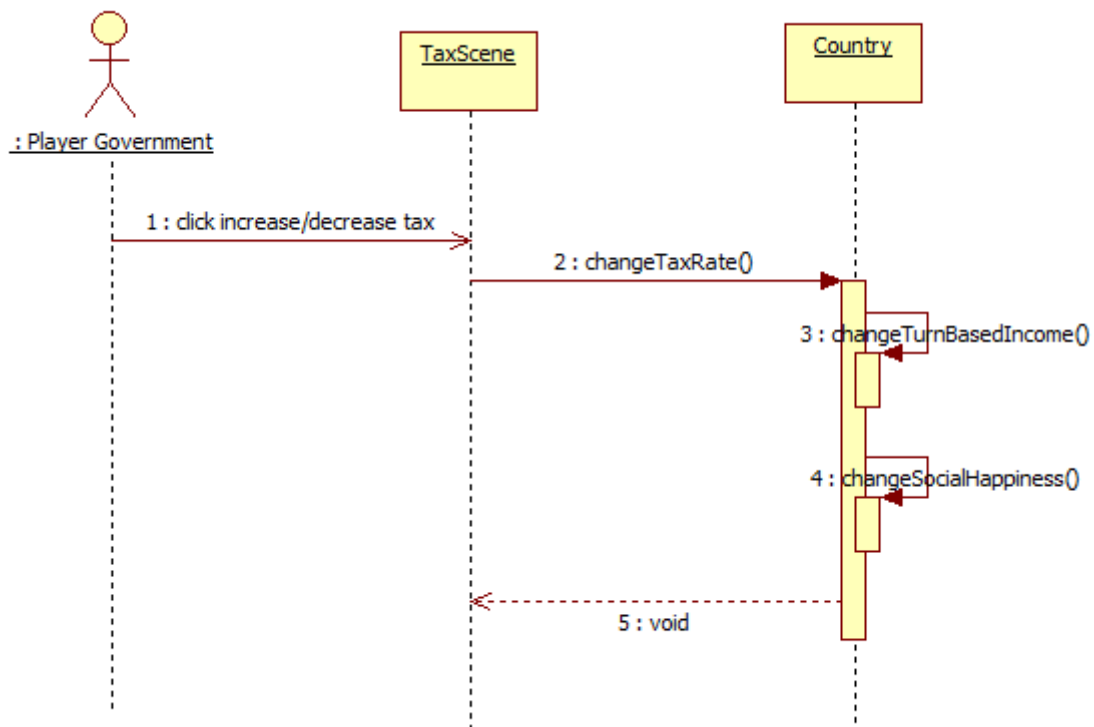


Figure 26: Player Government – Change tax rate Sequence Diagram

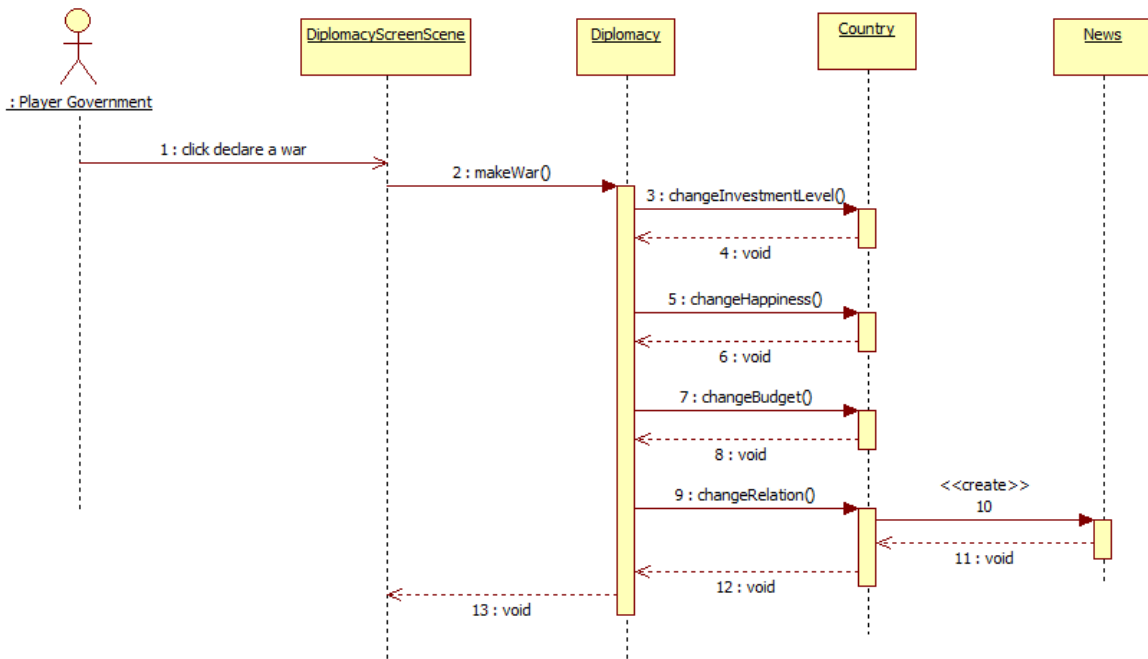


Figure 27: Player Government – Declare a war Sequence Diagram

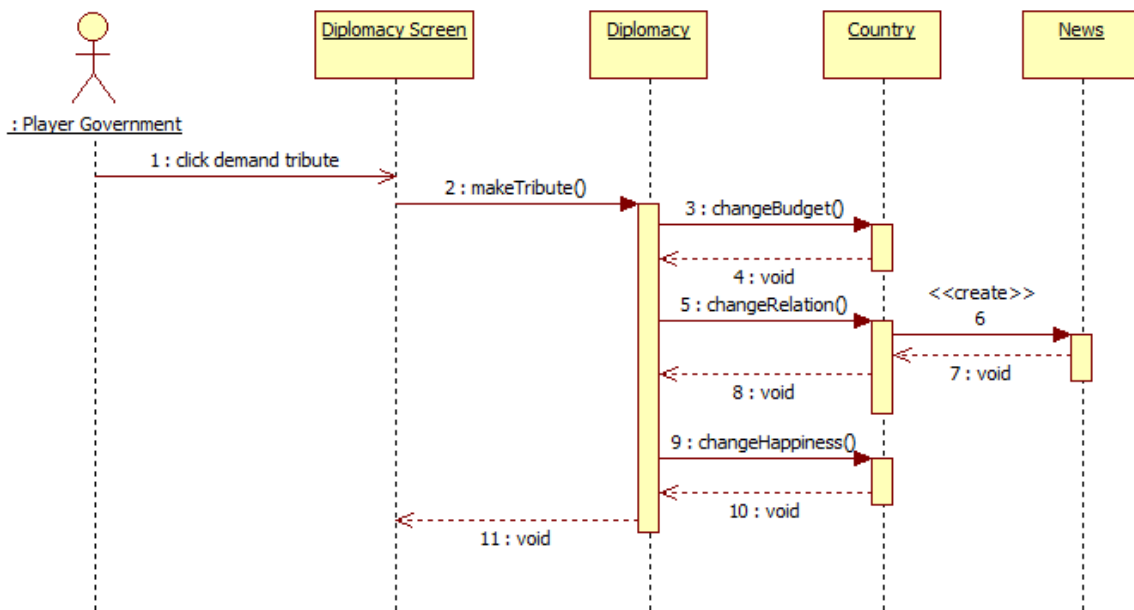


Figure 28: Player Government – Make tribute Sequence Diagram

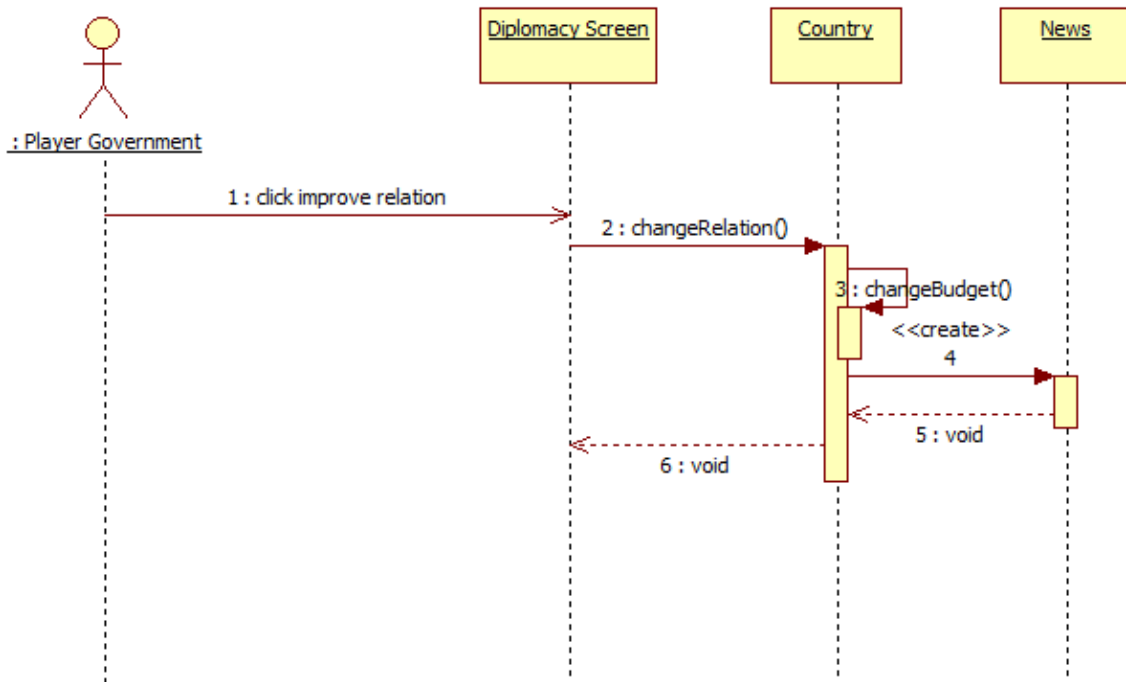


Figure 29: Player Government – Improve relation Sequence Diagram

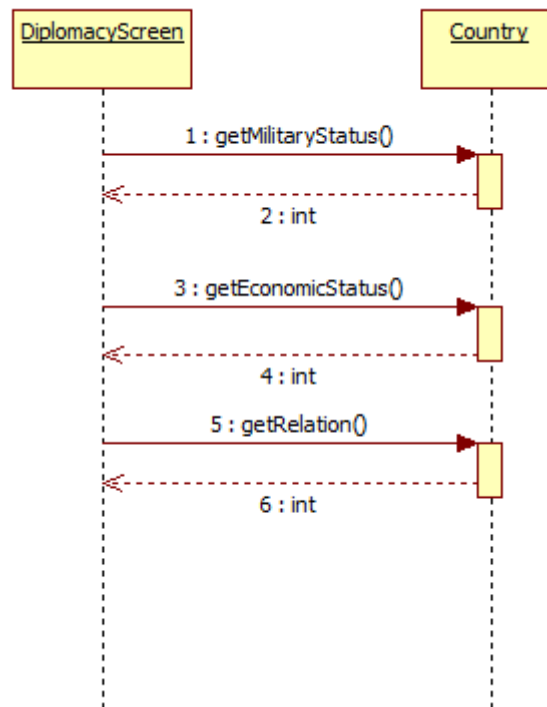


Figure 30: Player Government – See countries' military, economic and relation status Sequence Diagram

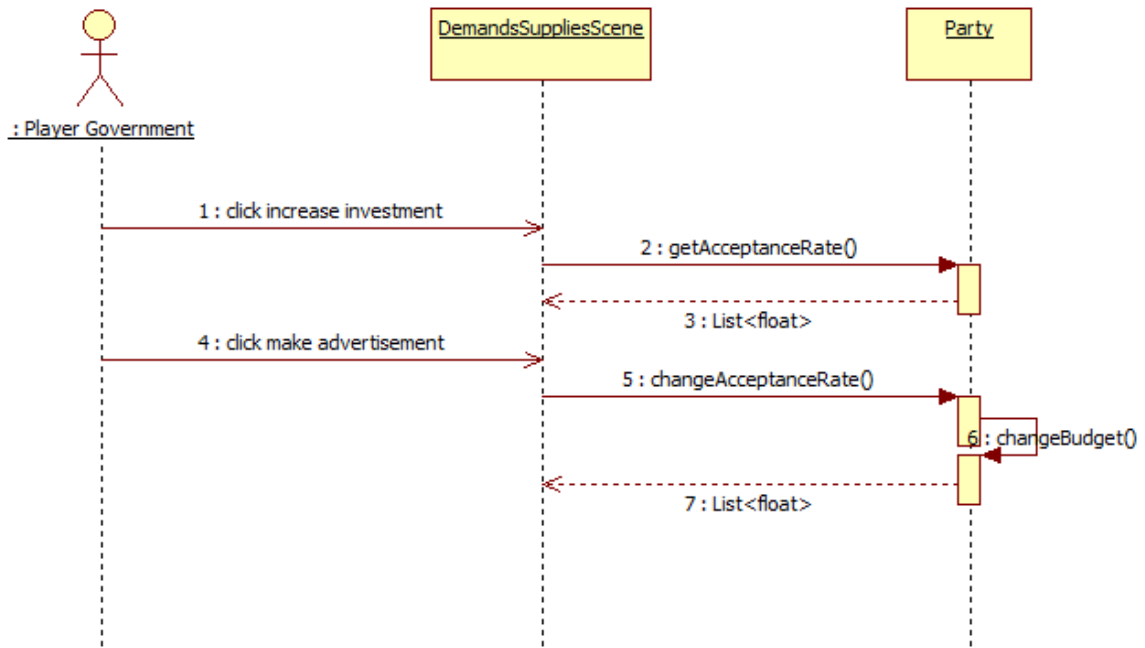


Figure 31: Player Government – Make advertisement Sequence Diagram

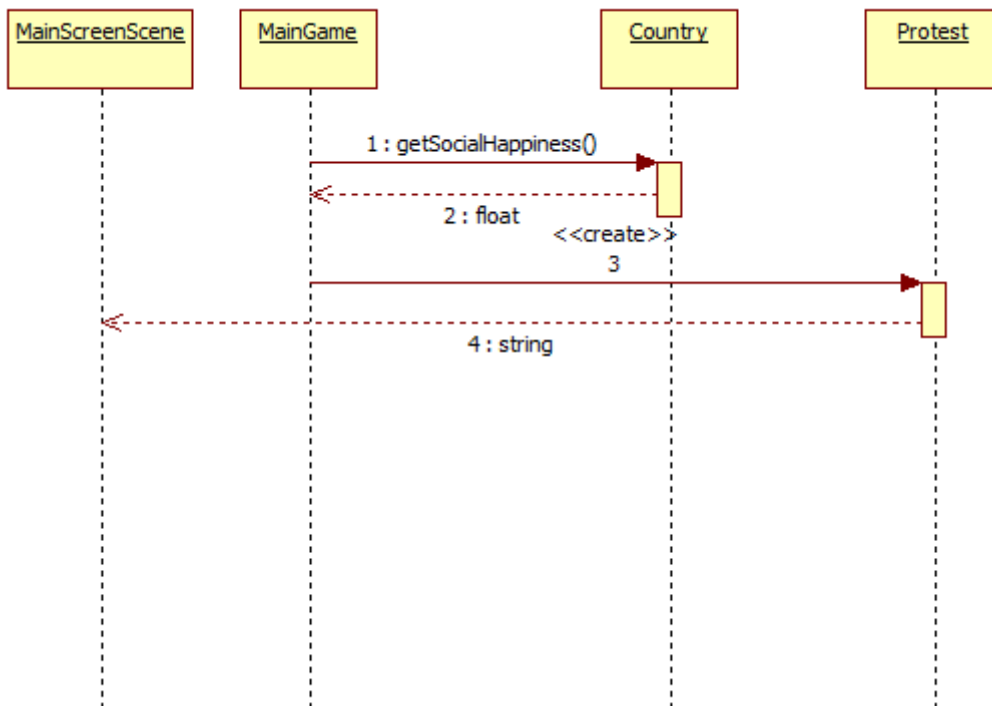


Figure 32: Player Government – Start protest Sequence Diagram

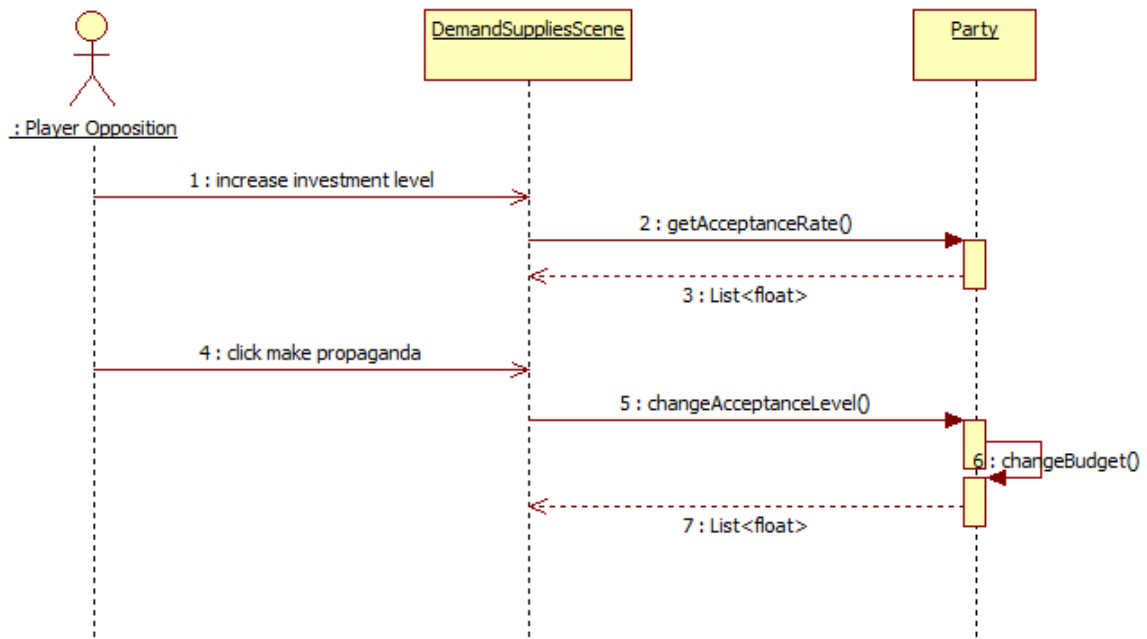


Figure 33: Player Opposition – Make propaganda Sequence Diagram

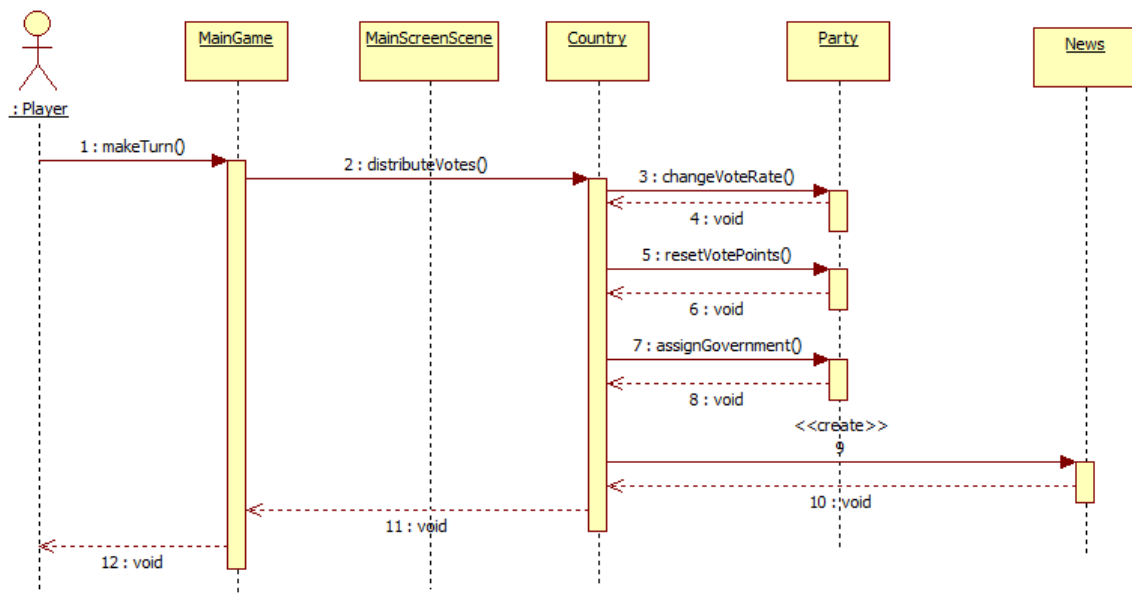


Figure 34: Player – Election Result Sequence Diagram

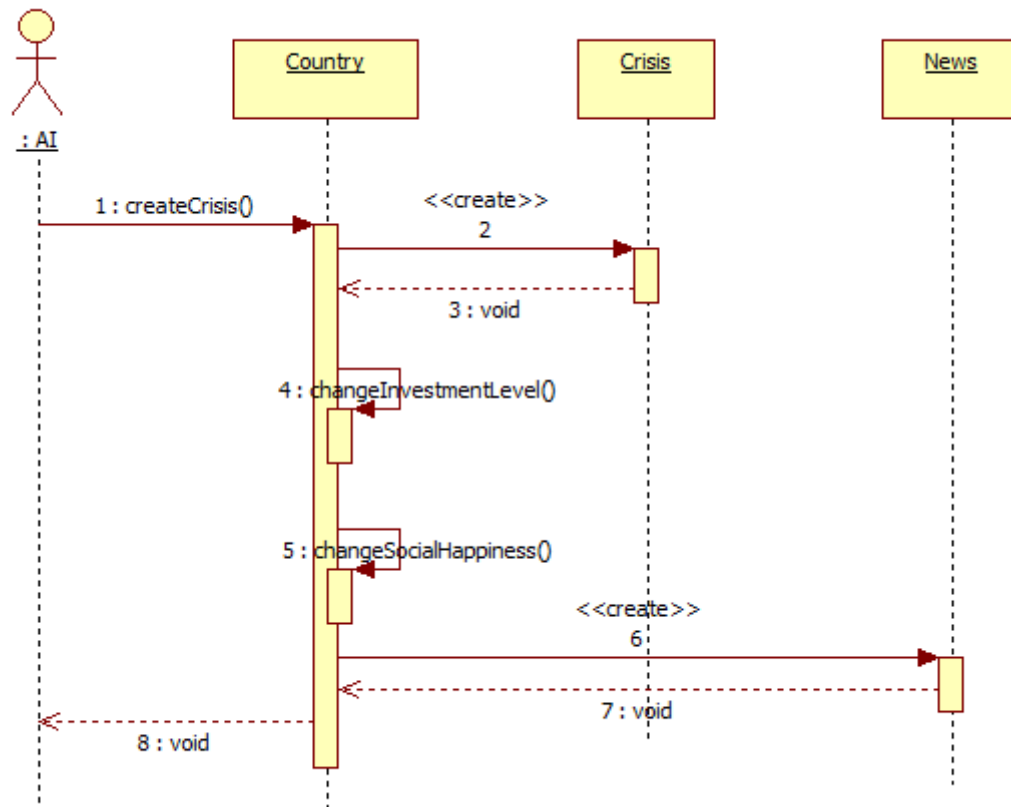


Figure 35: Crisis Sequence Diagram

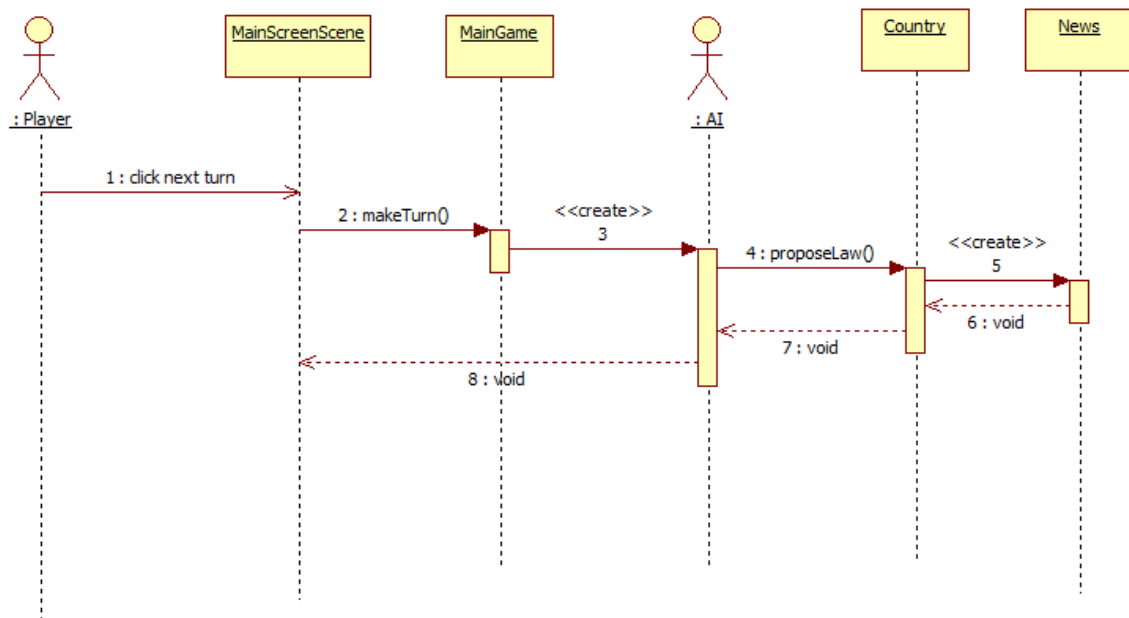


Figure 36: AI – Propose a law Sequence Diagram

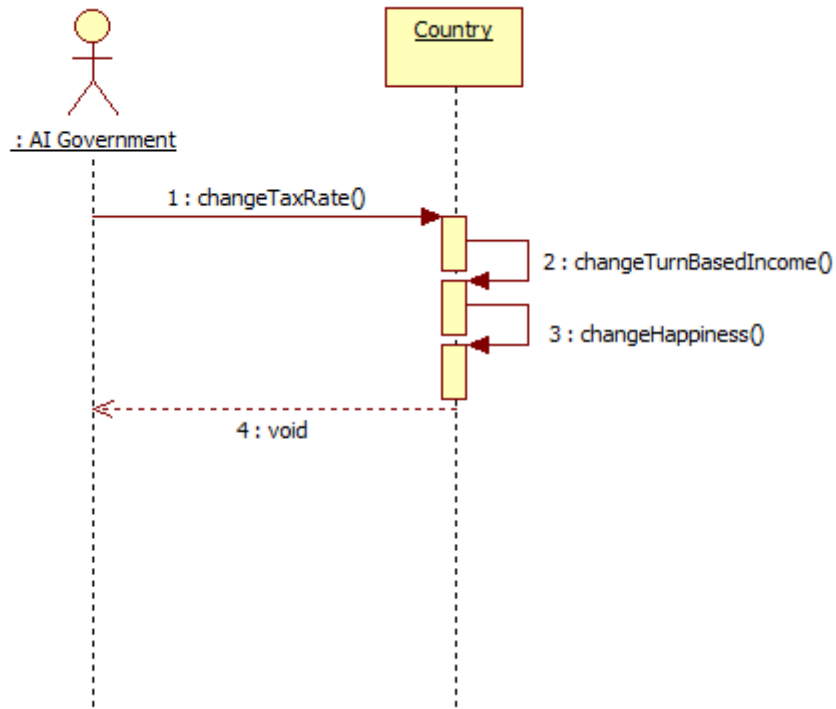


Figure 37: AI Government – Change tax rate Sequence Diagram

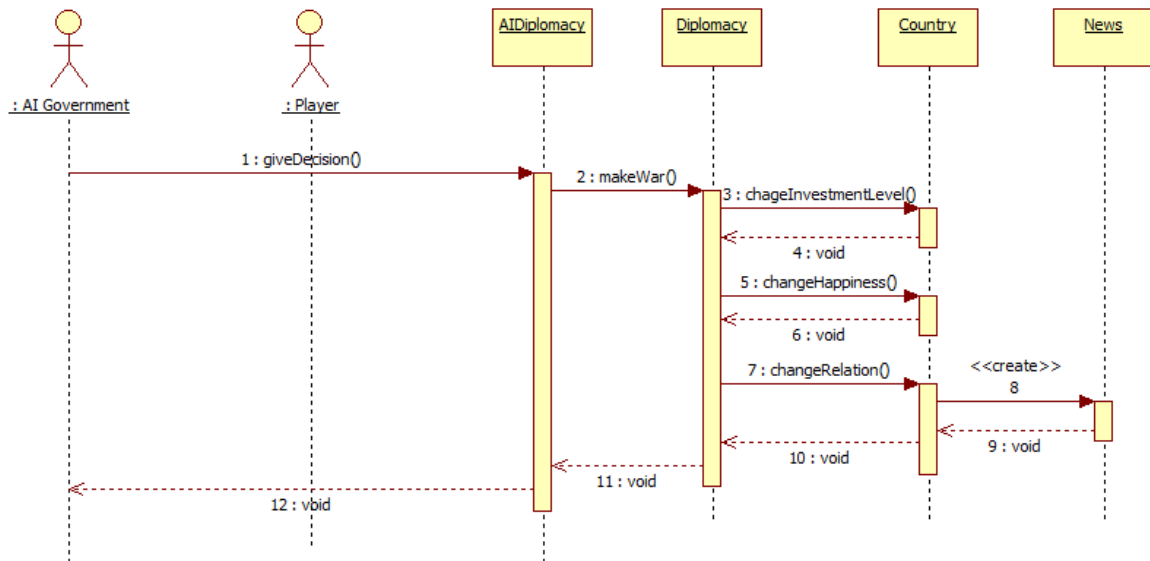


Figure 38: AI Government – Declare a war Sequence Diagram

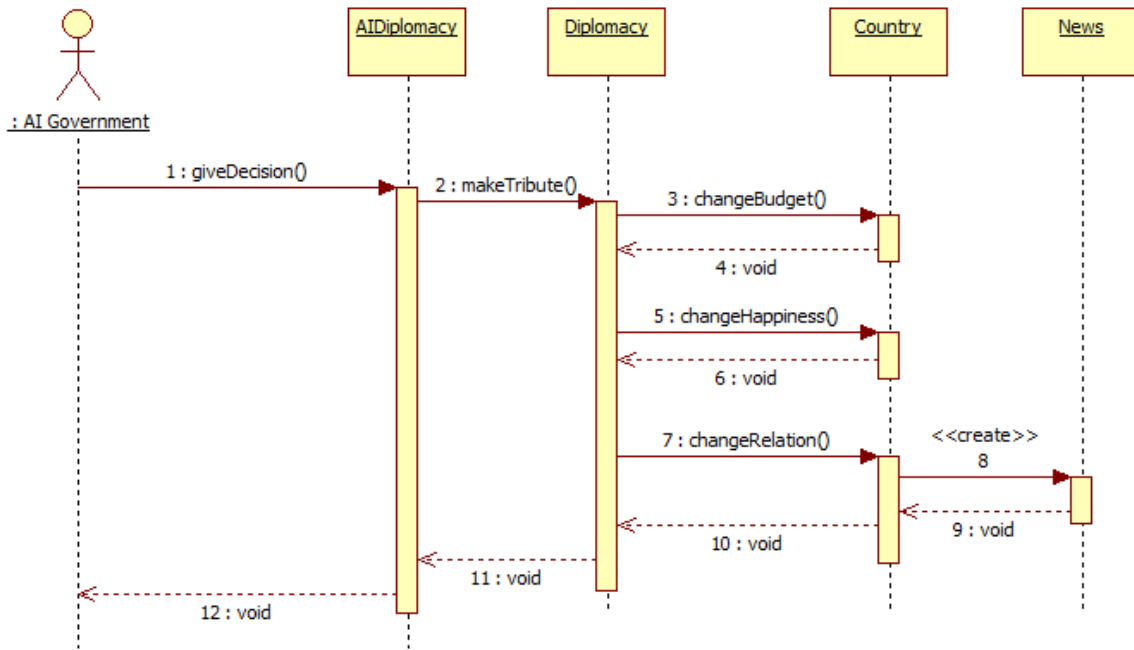
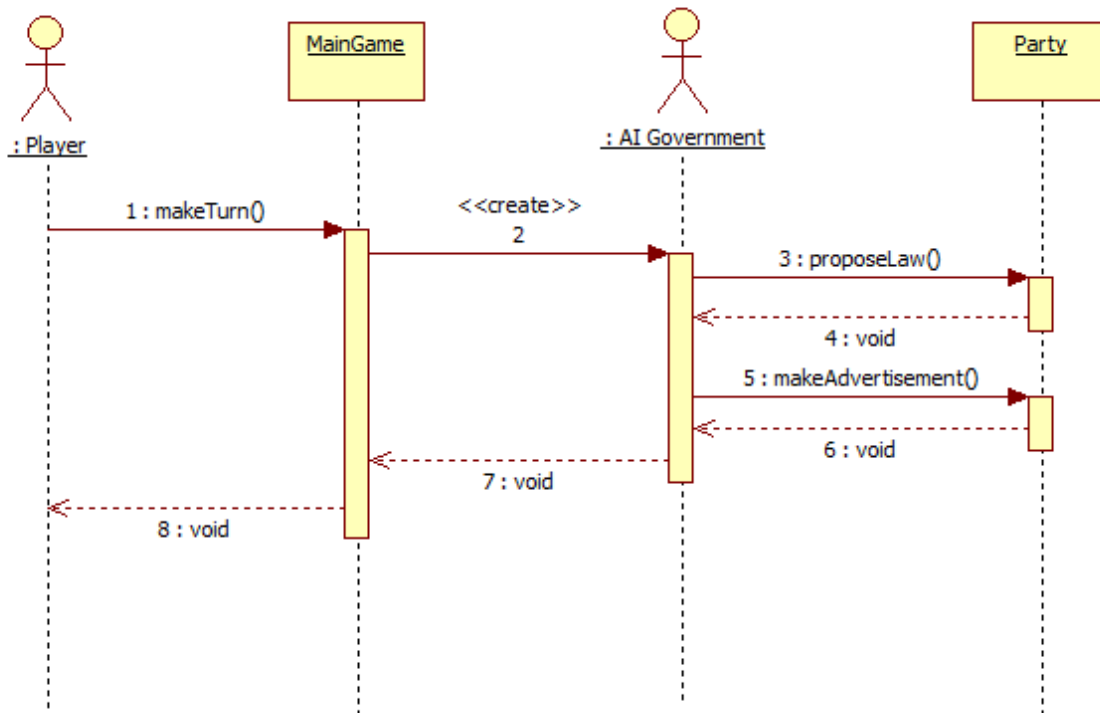


Figure 39: AI Government – Make tribute Sequence Diagram



AI Figure 40: Government – Make advertisement Sequence Diagram

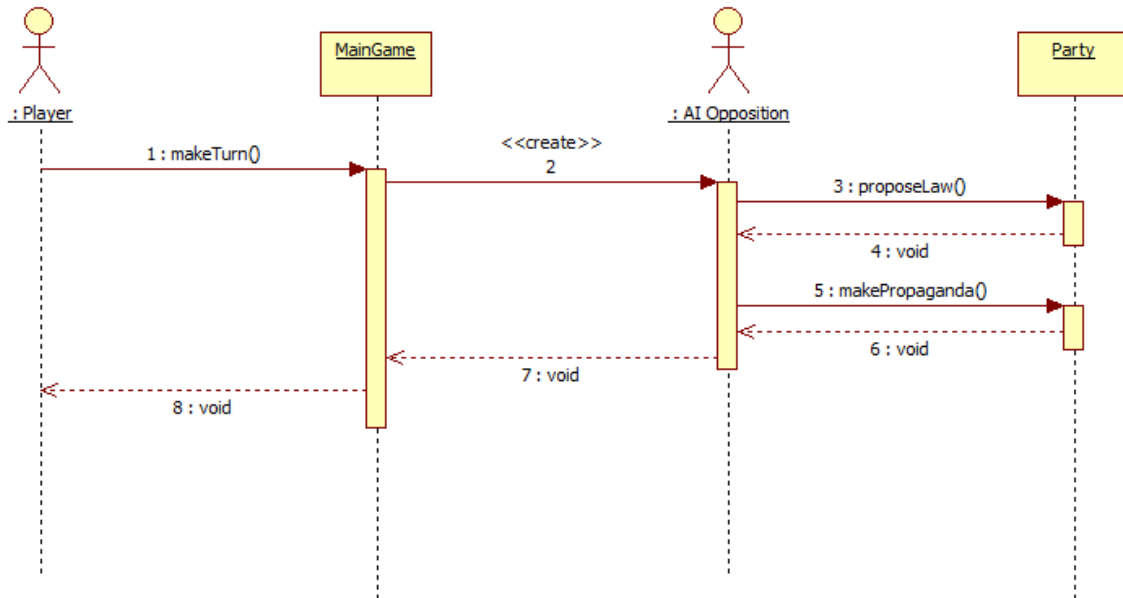


Figure 41: AI Opposition – Make propaganda Sequence Diagram

5.11.1 Design concerns

- Object communication
- Messaging

5.11.2 Design elements

- These sequence diagrams show the interaction between Player, Player Government, Player Opposition, AI, AI Government, AI Opposition and related classes, scenes.
- Sequence diagrams are drawn according to use cases. In other words, these sequence diagrams provide all the basic use cases.

5.12 State dynamics viewpoint

This viewpoint demonstrates systems' internal behavior with state machine diagram.

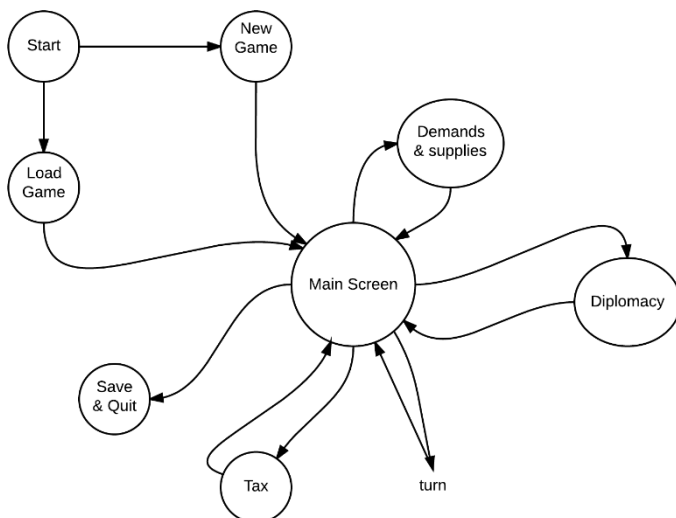


Figure 42: State Machine Diagram

5.12.1 Design concerns

Its design concern is:

- Dynamic state transformation

5.12.2 Design elements

- The State Machine Diagram shows the main lifecycle of the project. The player can choose New Game or Load Game from the start point. The mainscreen is used to main stage of the game. All other stages are used for certain tasks and the player always returns to mainscreen eventually except the Quit option.

5.13 Algorithm viewpoint

AI is used to generate intelligent behaviors for other countries in this project. For this AI, firstly we decided to use Bayesian Network. However, later we understand that it is not appropriate for our project. The detailed explanation about this is in the SRS document. Therefore, we decide to not use any known algorithmic model for AI. All the structure, design and implementation of the project belong to us. We calculate an AI decisions with using below flow charts. All AI decisions are calculated with using mathematical formulas which are already determined.

Firstly, explain the names in the diagram to understand diplomacy diagrams easily.

Military: This is the level of a certain investment named military.

Economy: This is a point which is calculated using a country's certain investments named Agriculture, Industry and High Technology.

Relation: This is the level of relation between these two countries.

Money: This is the cash money of a country. The other name of this is budget.

War Tolerance: When 2 countries made a war, there will be unfavorable consequences for the loser one. In order to keep it from another attack and to let it to develop again, it cannot be attacked by any other country for certain turns. War Tolerance implicates that.

Decision: It is the mathematical calculation which determines what will be the action against another country on this turn. There are 3 options: war, wanting tribute and keep peace (no action).

Determining an Action: The situation to determine whether AI takes any action to another country or not, is calculated in the way below. The human government player can declare war or want tribute from another country anytime he/she wants.

For attack possibility: This case is prepared for estimating Action possibility of country A against country B. We are assuming in this case, Country A is trying to determine to make a move against Country B or not.

Equation1 = $\text{Military level}_A / (\text{Military level}_A + \text{Military level}_B)$

Equation2 = $\text{Economy level}_A / (\text{Economy level}_A + \text{Economy level}_B)$

Equation3 = $1 - (\text{Relationship level} / 100)$

Equation3 is different from Equation1 and Equation2 because Relationship level has opposite influence on Action possibility result, meaning that the higher level of relationship, less possibility to declare war etc.

c1,c2 and c3 are coefficients that affects Action possibility result. Relationship level between countries and Military levels affect Action possibility result rather than Economy levels. This difference is provided by using these coefficients.

$$\text{Action possibility} = (\text{Equation1} * c1 + \text{Equation2} * c2 + \text{Equation3} * c3) / (c1 + c2 + c3)$$

For Decision flow:

Every country has 2 unique coefficients named aggressiveness and tributeness that affect War, Negotiation and Peace possibilities.

w: aggressiveness coefficient

n: tributeness coefficient where $w + n < 1$

$$\text{War possibility} = w * \text{Action possibility} * 100$$

$$\text{Negotiation possibility} = n * \text{Action possibility} * 100$$

$$\Sigma = \text{War possibility} + \text{Negotiation possibility}$$

$$\text{Peace possibility} = 100 - \Sigma$$

These moves (war, tribute, peace) cannot be chosen at the same time. So, in this step, there will be a random choice (ranging 0 to 100) and only 1 of them will be chosen at the end.

Making War: If the decision of country A is to declare a war against country B, there will be a calculation to determine which one will win. These 2 equations below will be used again:

$$m_1 = \text{Military level}_A / (\text{Military level}_A + \text{Military level}_B)$$

$$m_2 = \text{Military level}_B / (\text{Military level}_A + \text{Military level}_B)$$

$$e_1 = \text{Economy level}_A / (\text{Economy level}_A + \text{Economy level}_B)$$

$$e_2 = \text{Economy level}_B / (\text{Economy level}_A + \text{Economy level}_B)$$

$$(m_1 * c1 + e_1 * c2) / (c1 + c2) = S_1 \text{ which is the possibility of country A to win the war.}$$

$$(m_2 * c1 + e_2 * c2) / (c1 + c2) = S_2 \text{ which is the possibility of country B to win the war.}$$

Here the c1 and c2 are the same coefficients in the Decision part.

Then there will be 2 random calls ranging 1 to 3 as below:

$$\text{Random}(S_1, 3S_1) = rS_1$$

$$\text{Random}(S_2, 3S_2) = rS_2$$

If $rS_1 > rS_2$, country A wins the war; if $rS_2 > rS_1$, country B wins the war; if they are equal, there will be a draw.

War Consequences: If one country beats the other, some 4 of the defeated country's investments levels plus military level will decrease 2 levels and the money comes from that (when a level is decreased from investment levels, half of the money spent to increase that 1 specific level come out) will be taken by the winner country as contraband of war.

If there is a draw, both country's military level and some 2 investment levels will be decreased 2 levels and the money come out from there will be lost.

In any case, the relationship between these 2 countries will decrease and their war tolerance becomes active for 5 turns. Because the relationship goes to level 50 as medium 1 step per turn and because there will be a war tolerance, this decrease of relationship level won't cause unavoidable continuous wars.

Tribute: If the decision will want tribute, the demander country's AI will determine some money to want and send a request to the target country. Then the target country will determine whether it will pay the tribute. In this decision, determine war equations will be held and the target country will be aware of if it denies paying the tribute, this decision will increase the war possibility in future turns.

If the target country accepts to pay the tribute, the cost will be decreased from its money and will be added to the demander country's money. The relation between them will be increased.

If the target country declines to pay the tribute, the relation between these two countries will be decreased.

Peace: There will be no action in this case. But, because the relationship tend to close 1 step in each turn, peace will be beneficial for those who have bad relationships and will be unfavorable for those who have good relationships.

Improve Relation: This action can only be done by the human player. If the player wants to improve relationship level with another country for some reason (such as avoiding possible war), he/she will send some money to the target country. The cost will be calculated through this equation:

$$\text{Cost} = c * l$$

c : necessary money to increase relationship 1 level

l : level which wanted to be increased

Secondly, explain the names in the diagram to understand law diagrams easily.

Current Levels: This implicates that the investment levels in the country on that specific moment.

Demand: This is the vision of a political party. The party thinks that the public wants the current level of investments in these levels. "Demand" has the same structure as "current levels". Each party has its own vision of public demand and makes decisions according to that. The real base public demand is not reachable by parties; it is used only to calculate parties' visions.

Money: This is the country’s budget at that moment. It can be spent by passing laws to increase investment levels, by sending some tribute to other countries, by distributing opposition parties to make propaganda etc.

Offer: When a party wants to increase some investment, offers it as a law to be voted. There is only 1 offer for each party in each turn.

Vote: When some party offers some law, every party vote for it.

Acceptance: Acceptance of a law.

Vote Increase: The vote rate change of parties. It will be explained further.

Current Vote Levels: The public vote distribution of parties.

Offering a Law: At the beginning of each turn, every party will call a random[0,1] function. The parties which get 0 from that cannot offer any laws for that turn.

The party AI’s examine the current levels, demands and budget; determines which law it wants to offer in this turn. Then it gives the offer and goes to the propaganda step. In here, it will see other parties’ acceptance possibility of that offer. By making propaganda, it can increase other parties’ demand visions which will change the acceptance possibility. Propaganda takes a huge cost (decreased from that party’s propaganda budget) and can be applied several times. When the offering party determines it is enough to pass this law, it will continue to vote step.

Vote for a Law: When there is an offering, other parties’ will vote for it as yes or no. In this decision, each party will analyze the offering and compare with its own demand vision. There will be always a little leakage in each party. The human player is free to choose yes/no.

Acceptance of Laws: After voting, laws won’t be accepted instantly. They will be ordered as their “Yes vote count” and be waited until the end of the turn. After the human player push the “next turn” button, these ordered laws will began to be accepted one by one until the budget come to an end. The remaining laws will be deleted from that list.

Changing Vote Distribution: There will be vote increase points for each party. At the beginning of each election, these points will be set to 0.

	Proper for Party Demand	Not Proper for Party Demand	Vote Point Change for a Party
Offer a Law	x		+1
Vote Yes to a Law	x		+1
Vote No to a Law		x	+1
Offer a Law		x	-1
Vote Yes to a Law		x	-1
Vote No to a Law	x		-1
Acceptance of an Offered Law of This Party			+1

The points will be calculated as this table shows. At the begging of each election, every party will have a total point. The average of these total points will be calculated; the ones whose point is above the average will get vote increase and the ones whose point is below the

average will get vote decrease. The increase/decrease amounts will be calculated proportional to the points.

In this way, an opposition party may increase its votes just by offering right laws and voting right for laws. The AI won't give wrong decisions easily. But budget will be its limit and "not proper for party demand" cases are usually for human players.

For protest diagram:

Each country in the game has a property named Public Happiness. This value is affected by several reasons such as non-supply of demands, high tax rate etc.

In every turn, "Protest Probability" is calculated using public happiness. The less value of public happiness, the more possibility of a protest.

There is a Protest Tolerance value which blocks protests happen even if the public happiness is very low and the game decided for a protest. This protest tolerance will be set to 10 turn in the election and gives the new government a chance to increase public happiness by supplying the demands.

If there is a protest, the government may choose to suppress it with a cost. Then, the protest won't give any harm, but if the government won't change its attitude about public happiness, there may be another protest in near future.

If there is no money at that time the protest happens, the country will run for an early election. Because the cause of the protest is probably the non-supply of demands, the chance of government's change is high.

For election diagram:

In the game, elections will be held regularly unless there is an early election caused by a protest. In elections, there will be several changes: public votes for each party will be increased or decreased according to party points, the party points will be resetted, and there may be a government change and so on.

For news diagram:

Each country has a news part in order to keep the human players be aware of what is going on in the country. It has several resources for news and show it them in the main screen of the game.

Resources:

- Our country has declared a war to another country
- Another country has declared a war to ours
- Some country has just take some tribute from us
- The relations with another country improved/deteriorated
- War results (in the next turn of declaration)
- Law accepted, X improved (such as health level)
- Tax rate increased/decreased
- Public happiness is in critic level
- X won the election and become the new government
- A crisis hit our country and give small/huge damage (such as earthquake)

- Income is decreased to negative value (in every turn there will be a money loss from budget)
- Income is increased to positive value

For crisis diagram:

This is a mostly random, devastating, mostly uncontrollable action. Happens automatically and rarely, some of them cannot be avoided.

Crises may not happen even if the circumstances hold (except economic crisis), but when they happen, they will decrease the public happiness so sharp, a protest and thus early election (probably taking down government) chance will peak.

Economic Crisis: If the budget could not fulfill the turn based expenses, the economic crisis happens. It will automatically decrease some of the investment levels having turn based expenses until the budget can meet them.

Plague: If the Health level is far more low than it should be, there may be a plague disaster (there is a randomness here). This will cause to decrease public happiness.

Ignorance: If the Education level is far lower than it should be, this may cause an ignorance crisis and increase protest probability by decreasing public happiness.

Famine: If the Agriculture level is far lower than it should be, it may cause a shortage of food. Again this will decrease public happiness sharply.

Pollution: If the Environment level is far lower than it should be, it may cause a pollution and decrease public happiness.

Natural Disaster: There is no reason for that. It is totally random. When it happens, it gives damage to the investments and decreases their levels. It is designed for two reasons: giving a challenge to super developed countries, include some fun and surprise into the game.

Other Explanations:

All functions are designed and implemented in a very generic way so that they are used both the human player and the AI player mutually. We tried to make them common as much as possible in order not to repeat ourselves. This attitude also increased reusability a lot. So, we do not have to refactor our implementations.

We also used Object Oriented approach for two main reasons. Firstly, it's worldwide commonly used. The other reason is that we are familiar most to it, so that our possibility of making errors is minimized and the time required to fix an error is reduced exponentially.

The rest of the details are explained further in the SRS.

Below the diagrams which are used to design algorithms are shown.

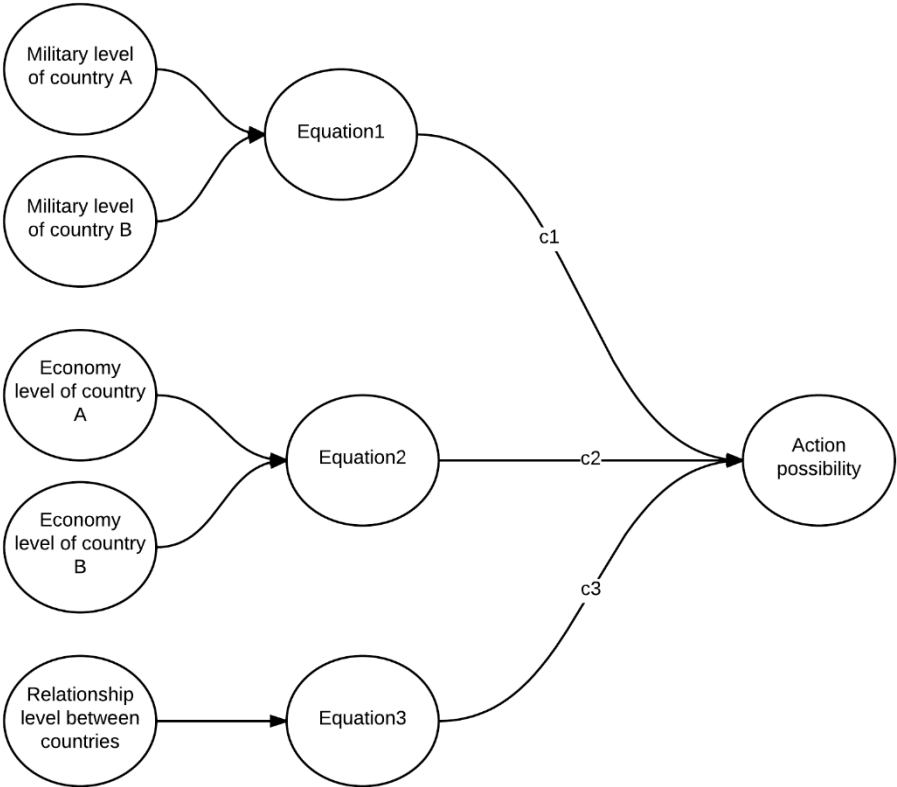


Figure 43: Attack possibility calculation for war

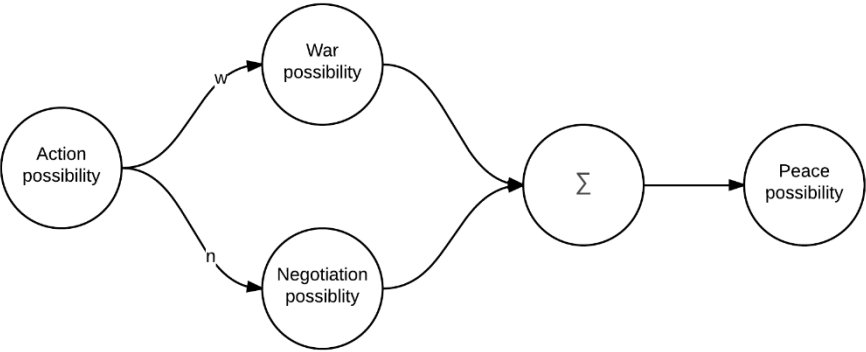


Figure 44: Decision flow for diplomacy

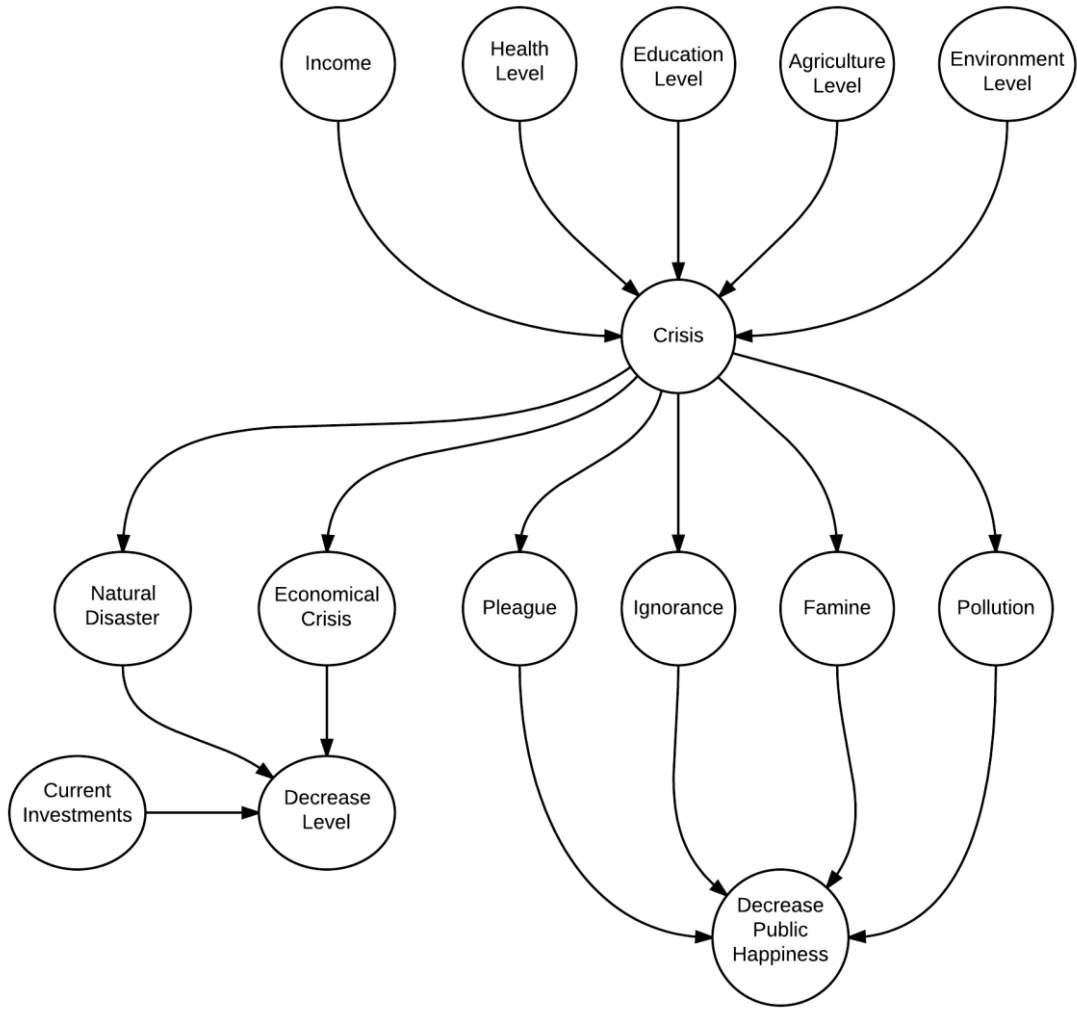


Figure 45: Crisis system

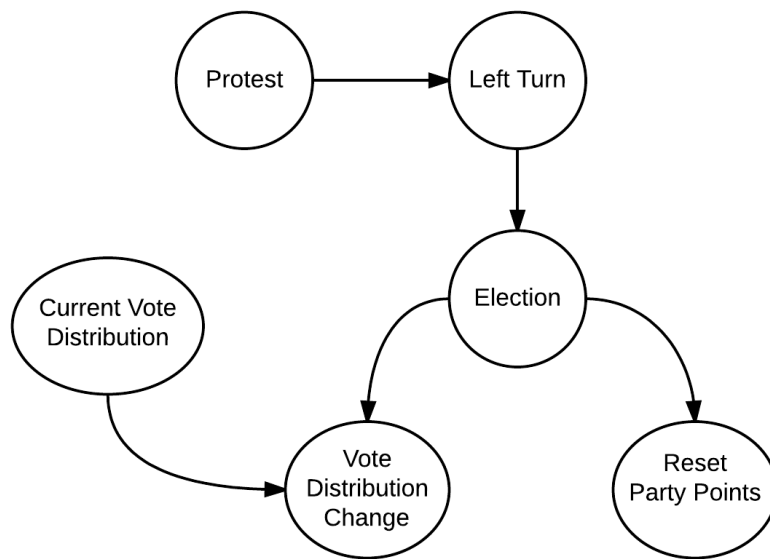


Figure 46: Protest System

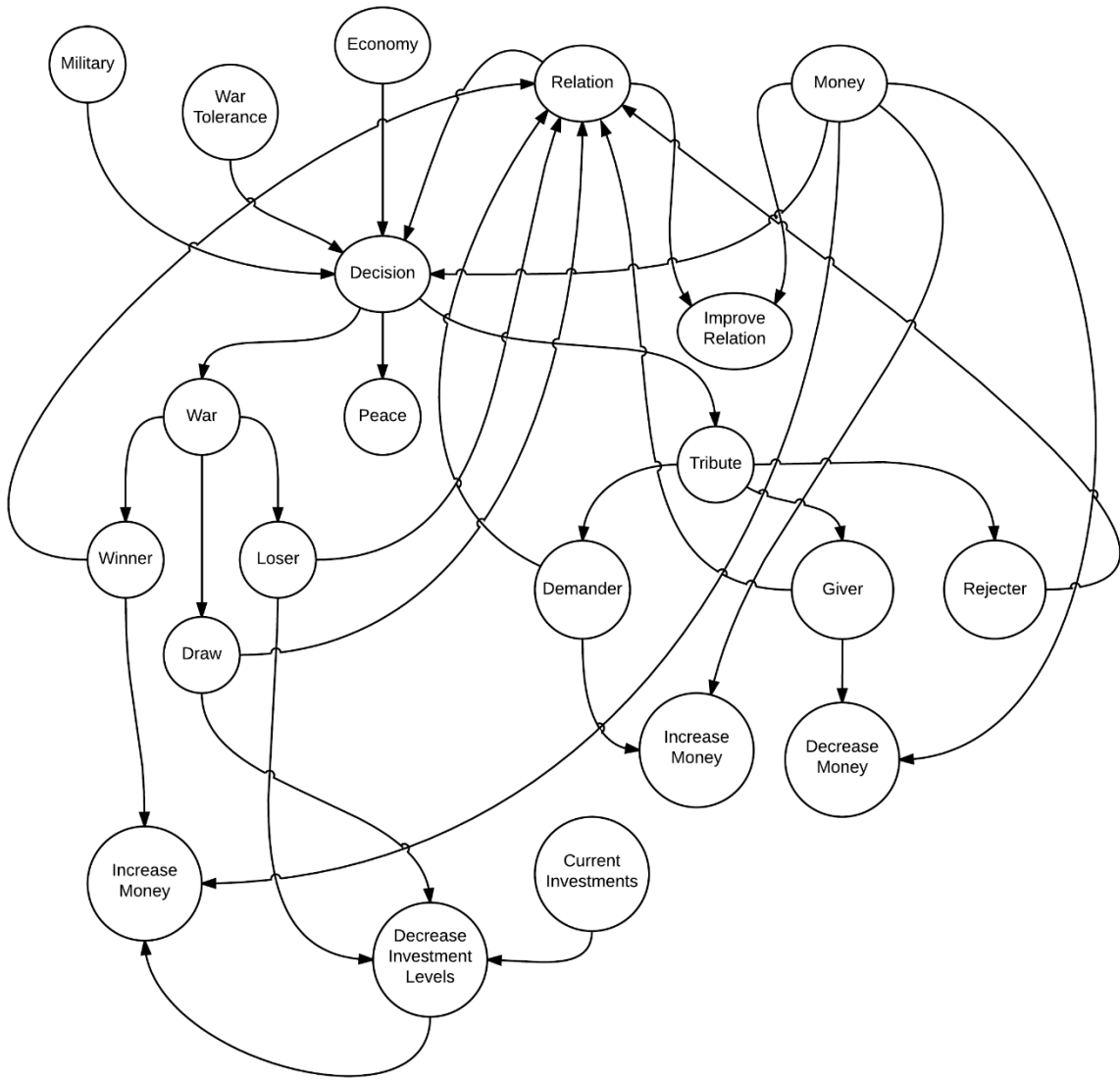


Figure 47: Complete Diplomacy System

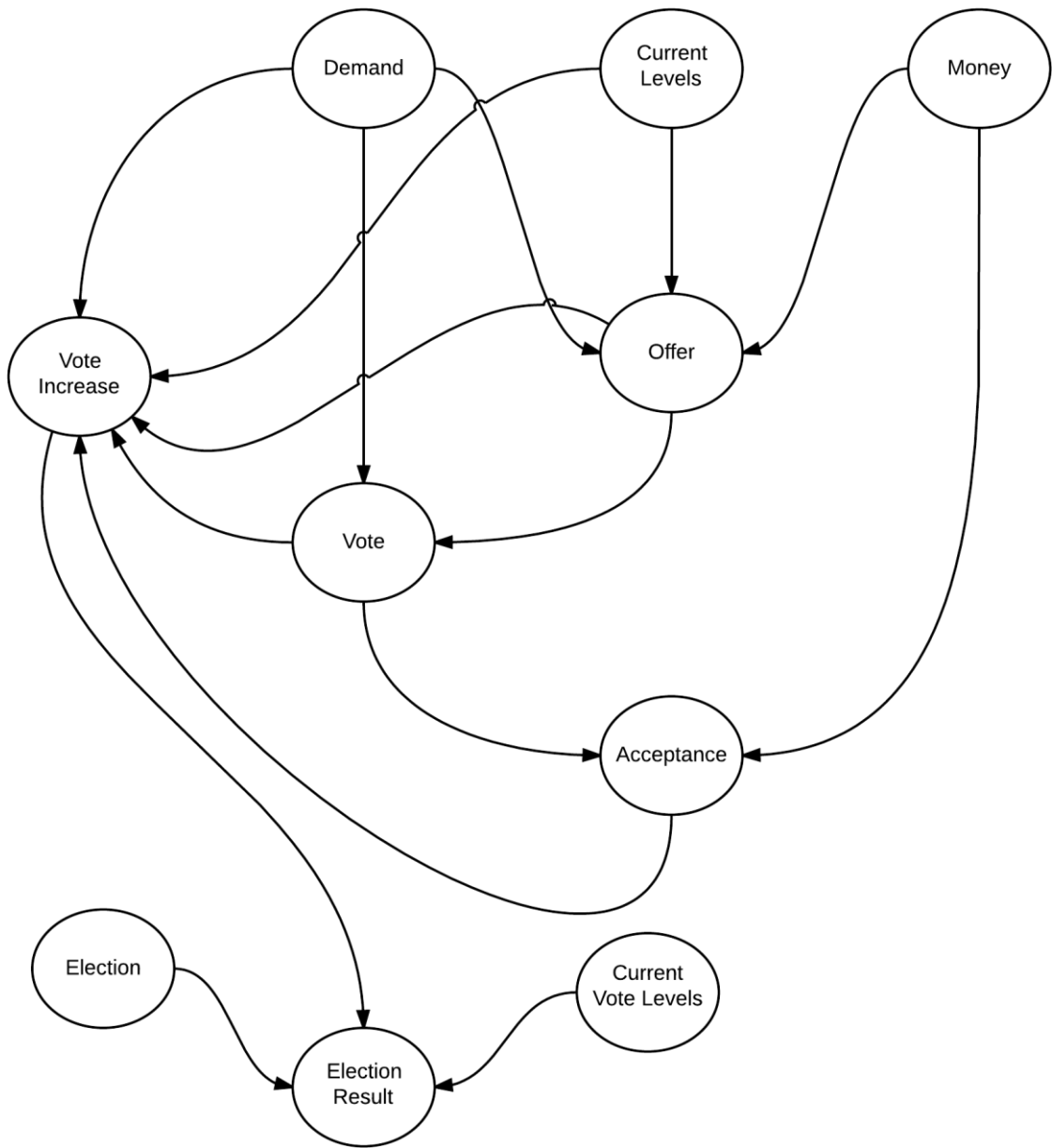


Figure 48: Complete Law System

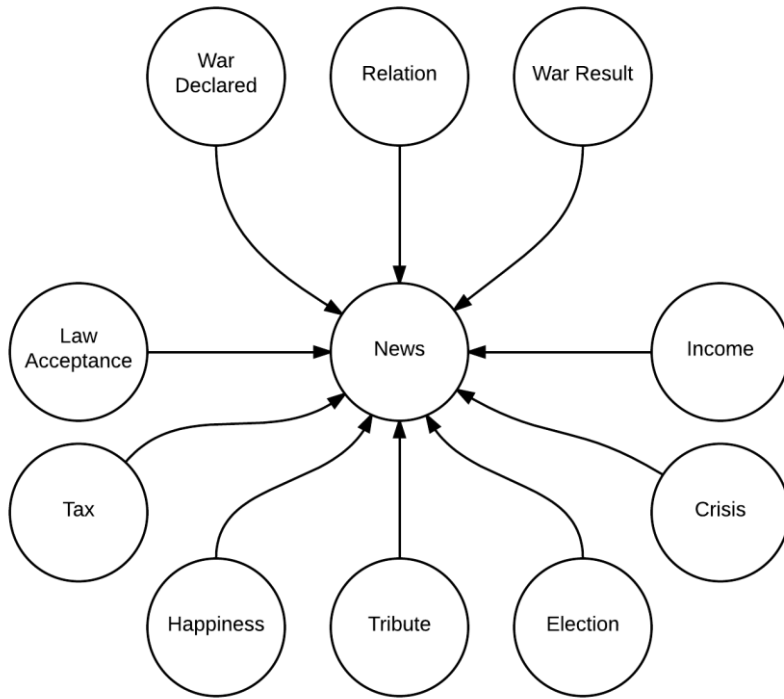


Figure 49: News System

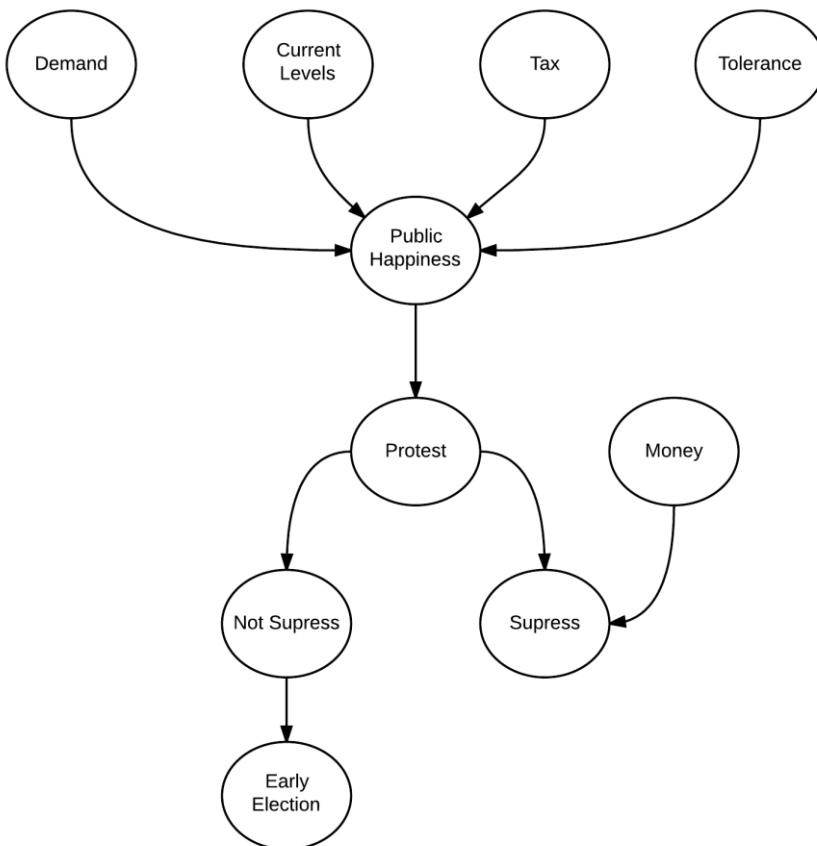


Figure 50: Protest System

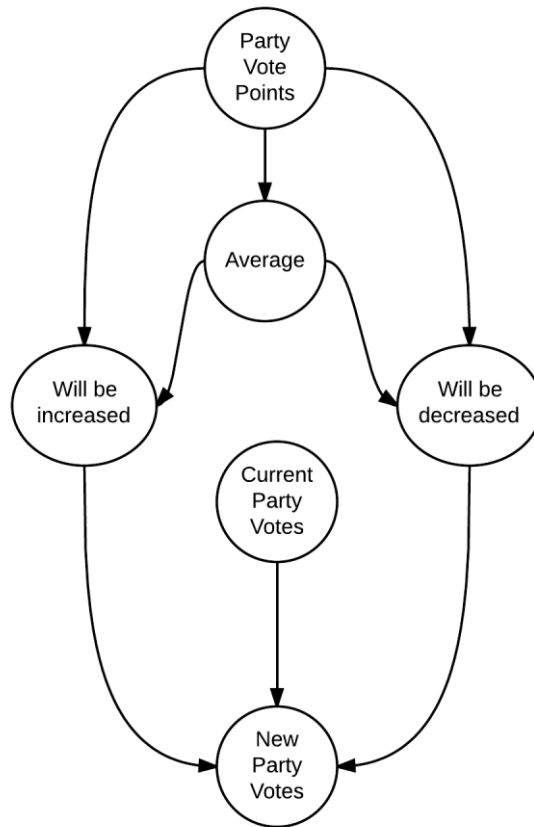


Figure 51: Election System

5.14 Resource viewpoint

This viewpoint's purpose is to model the characteristics of resources in a design subject. In this viewpoint, class diagram is used to show this purpose. Class diagram is shown in 5.4 logical viewpoint part.

5.14.1 Design Concerns

Its design concern is:

- Resource utilization

5.14.2 Design elements

In chapter 5.4.2, design elements of the class diagram have been explained.