

UNIVERSITY OF TARTU  
FACULTY OF SOCIAL SCIENCES

NARVA COLLEGE  
STUDY PROGRAM "INFORMATION TECHNOLOGY SYSTEMS DEVELOPMENT"

Konstantin Aleksandrov

**DEVELOPMENT OF A MOBILE GAME ABOUT SENGOKU ERA  
OF JAPANESE HISTORY FOR ANDROID PLATFORM**

Bachelor's Thesis

Supervisor: Andre Säask, M. Sc.

NARVA 2024

Olen koostanud töö iseseisvalt. Kõik töö koostamisel kasutatud teiste autorite tööd, põhimõttelised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

## **Non-exclusive license to reproduce thesis**

I, Konstantin Aleksandrov, 16.10.2000

1. herewith grant the University of Tartu a free permit (non-exclusive license) to reproduce, for the purpose of preservation, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright “Development of a mobile game about Sengoku era of Japanese history for Android platform” supervised by Andre Säask, M. Sc
2. I am aware of the fact that the author retains the right referred to in point 1.
3. This is to certify that granting the non-exclusive license does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Narva, **18.05.2024**

# TABLE OF CONTENTS

Table of Contents.....	4
Introduction.....	6
Problem.....	6
Solution.....	6
Goal.....	6
Tasks.....	7
Outline.....	7
1 Research and Design.....	8
1.1 Historical Background.....	8
1.2 Game Context.....	10
1.3 Existing games.....	10
1.4 Instruments.....	10
1.5 Map Design.....	11
1.6 Province Details Window Design.....	11
1.7 Gameplay.....	13
1.8 Goals and Events.....	13
2 Implementation of Core Components.....	15
2.1 Environment.....	15
2.2 Canvas.....	15
2.3 Camera Movement.....	15
2.4 Map Creation.....	16
2.5 Province.....	17
2.6 Country.....	18
2.7 Army.....	18
2.8 Battle.....	19
2.9 Goals.....	20
2.10 Events.....	21
2.11 Modifiers.....	21
2.12 Diplomacy.....	21
2.13 Coalitions (Alliances).....	22
2.14 Special Mechanics.....	22
3 Main Gameplay and Interface Scripts.....	23

3.1	Initialization.....	23
3.2	Turn Processing .....	23
3.3	Interface Scripts .....	24
3.4	Province UI .....	25
3.5	Province Details UI.....	25
3.6	Province and Clan Interactions .....	25
4	Post-Development Additions.....	27
4.1	Release on Google Play.....	27
4.2	Additional factions .....	27
4.3	Advertisements .....	27
4.4	In-app purchases .....	27
	Conclusion.....	28
	References.....	29

## **INTRODUCTION**

We live in a digital age and computers and smartphones are important parts of our lives. We use them for work, study, communication and relaxation. Video games are becoming increasingly popular for stress relief and sometimes they can even be helpful in studies. They can raise important questions or spark interest in learning more about various subjects. For example, puzzle games can improve problem-solving skills, or strategy games can improve planning skills.

### **Problem**

The strategy genre is quite popular among computer games, and there are quite a few computer games set in the Sengoku period. On the other hand, there aren't too many games on this topic on the mobile market at the moment and the existing games, from my observation, are not historically accurate enough. As I have seen as a fan of the strategy games from people commenting on other games, the demand for games of this genre is also quite high on the mobile market, so I decided to do my part and create an interesting game about this period of Japanese history on this platform.

### **Solution**

To satisfy this demand and popularize this historical period I will need to create a game that will provide not only entertainment value but will also provide relevant historical background to things happening in the game.

### **Goal**

I want to use the popularity of the mobile platform and the strategy genre in the PC game market and the unsatisfied demand for strategy games with cultural-historical content on the Android platform to create a game with at least one playable faction (Daimyo), that would provide not only entertainment value but will also provide relevant historic background to things happening in the game.

## **Tasks**

To create content for this game I will have to research this period of history and decide on which game engine to use for this task. Furthermore, I will also need to prepare game assets needed to create game maps and required scripts for UI interactions. To make the game more engaging, new mechanics may also need to be added.

## **Outline**

In the first chapter, I will describe the historical context for this game and how the game will be designed in terms of gameplay and mechanics.

In the subsequent chapters, I will explain how different components are implemented.

The last chapter will describe final thoughts on the product, implementation and problems encountered, possible future developments and possibilities for improvements.

# 1 RESEARCH AND DESIGN

## 1.1 Historical Background

The Sengoku period was a period in Japan where various warlords struggled for power after the collapse of the shogunate.

In the Japanese feudal system of the time, the official ruler was the Emperor of Japan, but actual power was delegated to the Shogun. Nominally appointed by the Emperor, the title of Shogun was given to a military commander and eventually became hereditary. However, this led to the same succession problems as in European monarchies.

This caused the Onin War in 1467. Shogun Ashikaga Yoshimasa had no heir, so he convinced his brother Ashikaga Yoshimi to renounce the life of a monk and named him his heir. However, this changed with the birth of Yoshimasa's son. This led to a succession crisis that led to the Onin War and the collapse of the feudal system and the fall of Ashikaga power. [1][2]

Devastated by the Onin War, natural disasters, and the failures of the Ashikaga Shogunate, various daimyos (local lords) began to expand their influence through warfare and deception.

During this time, established powerful clans often fell to smaller but more capable clans. This social meritocratic way of “low conquering the high” became known as “gekokujo”.

One of these smaller clans was the Oda Clan. Oda, who was a vassal of the Shiba clan, occupied much of Owari province and defeated the great Imagawa clan under Oda Nobunaga and almost unified Japan.

Imagawa Yoshimoto was a long-time opponent of Oda Nobunaga's father, and had sought to expand his domain into Oda territory in Owari. In 1560, Imagawa Yoshimoto gathered an army of 25,000 men,[5] and marched toward the capital city of Kyoto, with the pretext of aiding the frail Ashikaga Shogunate. Against this, the Oda clan could rally an army of only 2,000 to 3,000 men.[6][7]

In June 1560, Nobunaga's scouts reported that Yoshimoto was resting at the narrow gorge of Dengaku-Kazama, ideal for a surprise attack and that the Imagawa army was

celebrating their victories over the Washizu and Marune fortresses. While Yoshimoto viewed victory ahead, Nobunaga's forces marched to the Atsuta Shrine, a fortified temple overlooking the Imagawa camp. Later, Nobunaga moved to Zensho-ji fort, set up a decoy army there, marched rapidly behind Yoshimoto's camp, and attacked after a terrific thunderstorm. Yoshimoto was killed by two Oda samurai.[8] With his victory in this battle, Oda Nobunaga gained greatly in prestige, and many samurai and warlords pledged fealty to him.

After conquering Owari and Mino, Nobunaga used the seal of "Tenka Fubu" to promote his vision with a view to unifying the country. "Tenka Fubu" means "spreading military power throughout the world", and it is the idea of ruling the country peacefully, in other words, ending the Warring States period and restoring peace and order.[9][10][11]

Starting with the matchlock gun, Nobunaga paved the way for unifying the country through a chain of innovations, from the development of military technology and new weapons to the international supply chain for importing raw materials for ammunition, to the development of domestic distribution networks, to the way the territory was governed.[12] Those innovations were supported by the financial resources obtained through his economic plans.[12]

Nobunaga made personnel decisions based on ability and results, not titles. Until then, the family culture of warlords had been one of respecting family lineage and passing down positions from generation to generation, but Nobunaga made a major shift to a personnel system based on merit. It is known that Hashiba Hideyoshi came from a poor peasant background in Owari, while Akechi Mitsuhide came from a samurai background in Mino, but had spent a long time as a poor ronin. These two men caught up with and eventually overtook Nobunaga's old vassals, including Shibata Katsuie, Niwa Nagahide, Sakuma Nobumori, and others. Among the Oda vassals, Mitsuhide was the first to become the lord of one province and one castle, and the second was Hideyoshi. This was unthinkable for other Sengoku Daimyō.[13][14]

Before he could finish his plans for unification he was betrayed and killed by his vassal Akechi Mitsuhide, who in turn was then killed by Nobunaga's other vassal Hashiba Hideyoshi, who later became Toyotomi Hideyoshi, and finished unification of Japan.

## **1.2 Game Context**

The game begins after the Battle of Okehazama, where Oda Nobunaga defeated Imagawa and is considered the starting point of Nobunaga's rise to power. In this game, the player controls one of the several clans and will have to guide these clans through to either a historical outcome or some other hypothetical outcome.

## **1.3 Existing games**

Most of the games on Google Play about the Sengoku period use it primarily for atmosphere and aesthetics, providing little in terms of historical accuracy. As an example, “Sengoku Fubu” focuses primarily on multiplayer experience. “IkemenSengoku Otome Anime Game” is a dating simulator. The only good competing games in this genre are “Shogun’s Empire: Hex Commander” which is port from PC and “Great Conqueror 2: Shogun” which came out after I released my game. Both were developed by big companies and are going to be hard to compete against. However, since release, my game came to close to them and is now among the top results for the “Sengoku” search term on Google Play.

## **1.4 Instruments**

For the task of creating this game two game engines have been considered. One of them is Unreal Engine, which is free and offers better graphics, and is more resource heavy as a result. The other one is Unity, which is also free, but has paid professional subscription, and is simpler than Unreal to learn. While Unreal Engine uses C++ for coding, Unity uses C#. This means that Unity is more accessible, while C++ is better suited for real-time 3D rendering and advanced game design.

For the purpose of this game, I decided to use Unity, since in strategy games there is no need for better 3D rendering, and I am more familiar with both Unity Engine and C# which it uses for scripting.

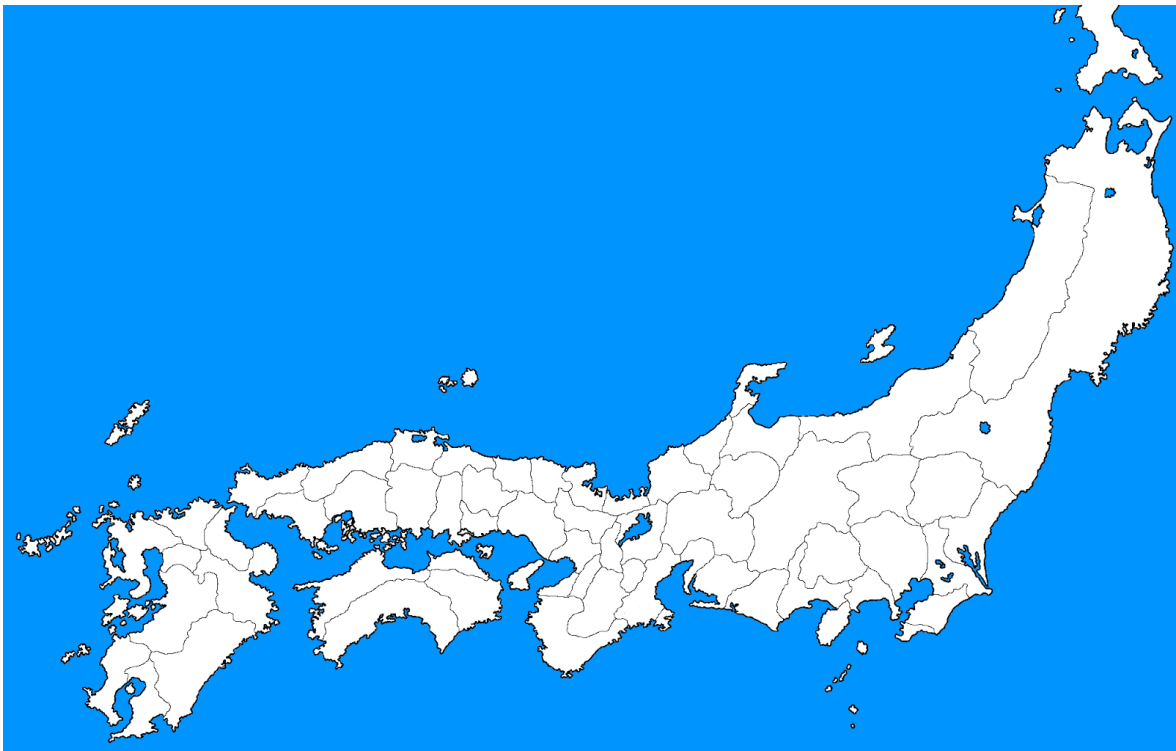
Since I am not using third-party assets, I am also using a software called Paint.NET for image processing to create province shapes and clan icons, which can then be placed on the map.

## 1.5 Map Design

The main component with which the user interacts is a map, which has province shapes and clan icons on it. To interact with the province and its owner user has to tap clan icon. This opens a province details window, which is an integral part of the game which is used the most after the map itself. To build a map I am using a province map outline that contains Japanese provinces as they were at the time.

## 1.6 Province Details Window Design

In this window users can see various details about selected province, some information about the owner and player data as well. On the same window there is an attack button, a button to switch to battle information, another button to switch to diplomatic interactions and a slider to adjust army investment. Details also contain information about the owner's allies and how consolidated this province is. This screen and battle screen also contain a list of modifiers that affect an owner's combat ability or economy.



*Figure 1. Map Outline*

Some information might be specifically omitted due to various reasons. For example, some details about army capabilities like specific attack and defense values are hidden so as to not clutter the screen and to realistically depict uncertainty of military capabilities until actual battle happens. Players can't also see enemy army's training level, their numerical limit and will need to estimate their strength. Modifiers change specific values to represent changes that happen during the period or certain advantages or strong points some clans had. For example, Oda clan was proficient in firearms and, thus, has a "Teppo" (tr. "Iron Cannon", the Japanese word for guns) modifier that increases army attack and defense.

The main view has three buttons. One switches to diplomatic interactions, where a player can: sign a treaty to improve relations, enter alliance or end it, demand vassalage or annexation. The second button allows the player to attack this province. The third switches between the main view and battle view.

The army investment slider allows players to control how much of the clan income goes to strengthening the army. This affects final income and thus a number of troops that can sustain their combat readiness, training, survival as wounded and morale. It also affects clan's ability to integrate conquered provinces as increased army investment leaves less resources for local garrisons. Lower income also affects economic development. Thus, while in short term larger army investment may help survive crises it will slow down combat potential increase in the long term due to economic stagnation.

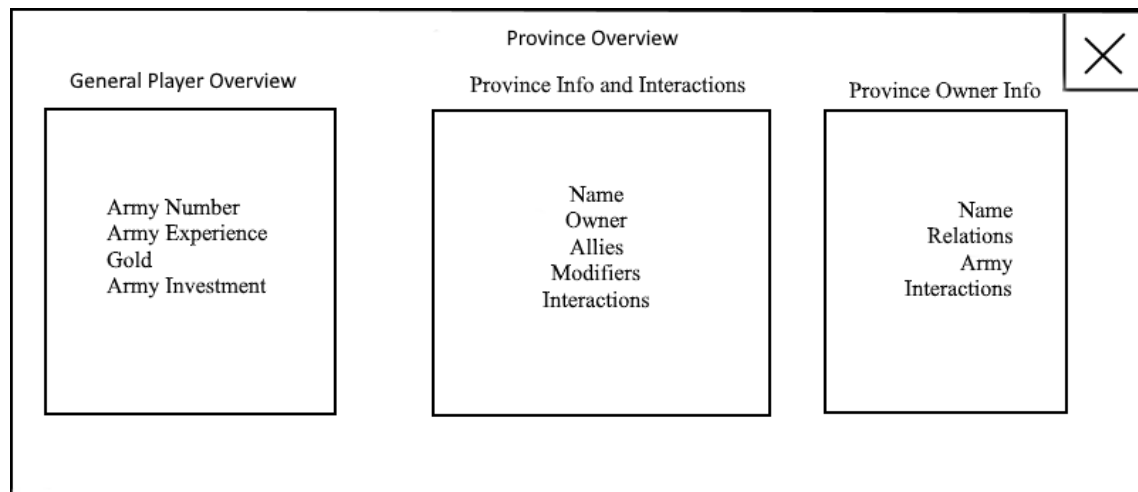


Figure 9. Province Details Diagram

## 1.7 Gameplay

Basic gameplay consists of conquering provinces with the final goal of controlling all provinces directly or through allies.

To that end, the player can use army to conquer province or improve relations with the owner and enter an alliance.

To make it more engaging this gameplay loop is supplemented by various game mechanics including events, goals, and bonuses and maluses (modifiers).

## 1.8 Goals and Events

These mechanics are meant to direct game flow and react to changes in strategies. Game can follow historical direction and reach the historical outcome or depending on player's or other clan's actions take other route and reach different outcomes. Non-player clans are by default meant to follow a more or less historical route, meaning they will generally avoid acting too aggressively unless provoked. It is important since most clans were not aiming for domination over an entire realm, but their own respective regions. More specifically, clans that were aiming for unification are very few and Oda clan is the most prominent example.

Two mechanics used to achieve this are Goals and Events.

Goals are aimed to help a player follow the historical path but can diverge at certain points. To complete these goals a clan has to fulfill certain objectives, for example ally

a clan or conquer a province. After completion, the clan can receive reward and move on to the next goal.

Events are more situational. They are generally triggered by certain conditions. These conditions can be more general, for example a certain year, or more precise state of the map or clan. Some events have choices that can affect clans or change situations on the map in different ways. They can also have long lasting effects on future events and lead to different outcomes.

While historical accuracy is important it is also important to allow a certain degree of freedom and unpredictability to make it more engaging for a player in the long run and not restrict him by the same exact scenario over and over again. These mechanics allow the player to choose different strategies based on situation while also maintaining ability to follow the historical course if player wants it.

## 2 IMPLEMENTATION OF CORE COMPONENTS

### 2.1 Environment

I chose Unity to develop this game because it is one of the most popular engines, easier to learn and I have previous experience with it. Unity also has integration with Microsoft Visual Studio which is used to edit C# scripts.

### 2.2 Canvas

The main part of the user interface is *Canvas*. *Canvas* is built with an *EventSystem* that handles user input, contains scripts required for processing clicks or taps and handles input, raycasting, and sending events.

The *EventSystem* is responsible for processing and handling events in a Unity scene. A scene should only contain one *EventSystem*. The *EventSystem* works in conjunction with a number of modules and mostly just holds state and delegates functionality to specific, overridable components.

When the *EventSystem* is started it searches for any *BaseInputModule* attached to the same *GameObject* and adds them to an internal list. On update each attached module receives an *UpdateModules* call, where the module can modify internal state. After each module has been Updated the active module has the *Process* call executed. This is where custom module processing can take place. [3]

A *BaseInputModule* designed for mouse/keyboard/controller input. [4]

### 2.3 Camera Movement

Although the input module can also handle touch, it needs additional scripting to handle map movement and zooming. In this document, I often mention clicking and scrolling, but this also includes touch screen actions.

Since this game is a map-based 2D game, I'm using an orthographic camera, so I just have to resize it to zoom in or out.

Unity has a component called *Event Trigger* that handles user interaction with game objects. Although this can be used to handle camera movement, it is better to handle the camera in a dedicated script for two reasons. First, the drag trigger handles the drag

as a whole, so until you release the mouse button, the camera does not move with the mouse, and when you release, it jumps to the mouse position at the end of the drag. So, you need to calculate the mouse movement in each frame until you release the mouse button, this will give a smooth movement. Second, the trigger can handle touch events along with mouse events, but you still need a script that zooms the camera along with the finger gesture.

## 2.4 Map Creation

The map consists of a background province outline map image, a province shape image, and icons. The map shows the boundaries of the province. Province shapes are then placed on this map. Province shapes are measured and placed to fit within the boundaries. The province shapes allow me to change the color of the province depending on the status. For example, if the province is controlled by the enemy, then it is red, by the ally - blue, by the player or their vassal - green, by ally of an ally - light blue, by overlord - dark green, and otherwise white. If a battle is currently taking place in a province, the color will turn yellow, regardless of the above conditions.

After that I need to place the icons over the provinces. While province shapes have to be manually placed and measured, icons are easier in this regard. Instead of going through and resizing all the icons one by one, it is possible to add all the necessary components to one of them, resize it properly and make enough copies. Then just rename them and change the source image.

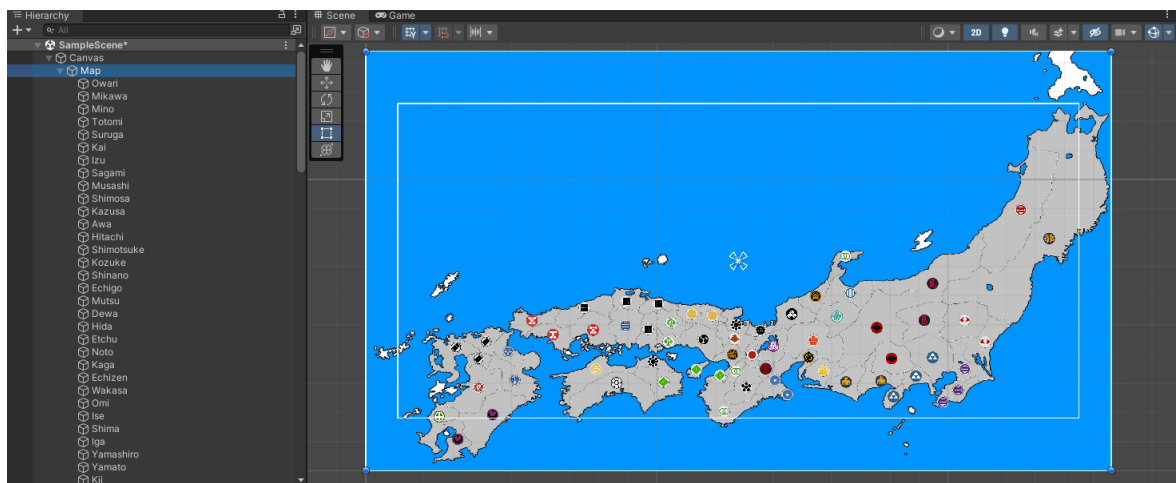


Figure 17. Game Map

## 2.5 Province

One of the most important components is Province. Province has important properties such as Name, Owner, Income, Consolidation and Modifiers. Some of them are interconnected and thus in the getters are calculated based on other properties using the following formula:

$$income * (1 + \sum modifiers) * consolidation.$$

Owner property has an OnPropertyChanged method that interacts and modifies owner's properties and handles change of province owner and consequences.

Additionally, there are several auxiliary methods that help interacting between other components and access data from the context of the province.

When a Consolidation value drops to 0 a rebellion method is fired that spawns a rebellion inside a province based on certain conditions. If the province is a home province of another clan that does not exist on the map, then a battle starts, and the winner gains the province. Home provinces are also called cores. When a non-core province is conquered, and an Owner property changes Consolidation is dropped to 20%. Consolidation negatively affects Income and is affected by Army Investment. The less Army Investment is the faster province consolidates. This is one of the several things that discourages a player from increasing Army Investment too much for too long. When a province is attacked, it also decreases Consolidation, so if the province is attacked and Consolidation is low enough, it can spawn a rebellion, in this case in form of modifier that decreases income. This modifier is lost when a province is fully consolidated.

Each turn province gains a certain amount income. This is called development. Unless province owner is a vassal, province developments increase base income by a certain multiplier, which is currently a flat 1%. Increase in development affects Army number limit and recruitment, representing an economic capability to support troops and recruit more. If the owner is a vassal, income is halved, and the other half goes to the overlord and only 1% of the remaining half develops. Development is calculated using the following formula:

$$income * (1 + multiplier) + \sum VassalProvinceIncome * \left(\frac{multiplier}{2}\right).$$

## 2.6 Country

Country represents a clan that can be on the map or off-map, which is determined by exists method that returns a Boolean value. If a Country has more than 0 provinces or is flagged as off-map, it returns True. Each country has a Name, Gold, Army, Consolidation Rate and Modifiers for it, modifiers to subjugation acceptance and aggression, Army Investment, Sprite (clan icon), and Overlord. Additionally, countries have lists of Allies, Core Provinces, Claims and Goals.

As with Provinces, Country class also has auxiliary methods to interact with both Provinces and other components as well as get data from the context of the Country. Some methods also help with minor or major tasks, like ending alliance or collapsing a country.

Country also calculates income based on certain other values. Namely, it calculates sum of individual province incomes, which already have values calculated from Province component based on provincial modifiers, and decreases this sum based on global Country values such as Army Investment and number of diplomatic treaties, which for now cost 1% of income each. Income is calculated by the following formula:

$$\sum ProvinceIncomes * 0.99 * (1 - ArmyInvestment) * (1 - 0.01 * \#ActiveIncreaseRelationTreaties)$$

To compare clans, they also have a Power property which is calculated using the following formula:

$$(ArmyNumber + 1) * ArmyAttack * ArmyDefence * ArmyMorale + OverlordPower$$

## 2.7 Army

Each Country has its own army. Instead of only being a number of troops, an army is a class as well and has values such as attack, defense, morale, experience, number, and modifiers for attack, defense, and morale. It also has a certain number of possible attacks and can have wounded soldiers.

Country's Army Investment and battles can affect these values. If a country wins battles, morale increases and if loses, morale decreases. Morale, number of troops and

experience also change based on Army Investment. Morale affects attack and defense and the number of soldiers that can be saved as wounded.

With experience Army's combat abilities will increase. Better attack and defense and more attacks per turn will increase Country's ability to conquer several provinces at once. However, this may still be dangerous as stretching army on several fronts may leave it open to attacks.

Both Attack and Defense are calculated by a same formula:  $value * (1 + \sum modifiers)$ , with value being Attack or Defense and modifiers related to them.

Morale is calculated by the same formula but is limited to values  $\geq 0$ .

## 2.8 Battle

Battle is a component that handles ongoing battles. It has attacker, defender, place, reinforcements from both sides and wounded from both sides.

When a battle starts and the defending side is a puppet, overlord sends their army as reinforcements automatically. It represents protective relations of an overlord due to the vassal territory being a part of an overlord's realm.

Each turn of the game a battle is evaluated and the damage is done to both sides based on their stats. Prolonged battles also affect morale. If one side consistently does more proportional damage, the enemy will eventually lose morale and lose battle. Depending on the morale and army investment some damage done may be converted to wounded.

Battle can end in several ways. Either one side loses all morale or troops or it ends when one Army gives up due to low morale.

At the end of the battle reinforcements return home, except if they are the overlord of a defender. Overlord does not send reinforcements and temporarily lose them from their own army but defends it as if it was their own territory. The winner of the battle gets the province and morale. Wounded return to each side and if defending country ceases to exist remaining forces will be split among allies and winner.

Battles also have auxiliary methods that check for participation in battle, like, if ally or self participates.

Damage is calculated as follows:

First raw damage is calculated for both attacker and defender. It is calculated by randomly selecting number within a range calculated by following formula for attacker for minimum

$$\text{AttackerArmyNumber} * (\text{AttackerArmyAttack} - \text{DefenderArmyDefense})$$

and following formula for maximum

$$\text{AttackerArmyNumber} * (\text{AttackerArmyAttack} - \text{AttackerArmyDefense} + \text{AttackerArmyMorale}).$$

For defender formula is the same with Attacker being replaced by Defender and vice versa.

Damage is constrained between 0 and the opposite side Army size

From this damage some soldiers become wounded based on the formula for attacker

$$\text{Damage} * \min(\text{AttackerArmyMorale} + \min(\text{AttackerArmyInvestment}, 1) - 1, 1).$$

Defender has the same formula, but with values for Defender instead of Attacker.

## 2.9 Goals

Goals are an integral part of the game that directs player and other clans in historical direction.

Goals have title, description, short description, completion status, and completion function. Goals are static objects and at initialization major clans are given their first goal which is added to the list of their goals. After each goal is completed, the completion status is changed, and a next goal is added by a completion function.

Title and description are self-explanatory, short description is shown at the top of the map view for player to keep track of the current goal.

Completion function evaluates completion condition, makes necessary post completion changes like giving modifiers or changing provinces, and returns Boolean value depending on the outcome which is then assigned as a completion status. After that it is no longer being evaluated. Each turn every clan has only its not completed goals evaluated.

## **2.10 Events**

Initially events were triggered effects that would fire before goal evaluation but were reworked and now are their own mechanic. They are similar to goals, they are also static objects, but they fire only under certain conditions. They have list of target countries that will receive this event, id, title, description, and Boolean value that may prevent or allow event to fire multiple times. They also have conditions and one or more choices, and an AI choice factor. This factor represents the choice which non-player countries are more inclined to make. Similar to goals events can affect campaign and even change the path to more hypothetical non-historical outcomes.

Additionally, events also serve as a tool for teaching some important events of the time like Honno-ji Incident or Battle of Sekigahara.

## **2.11 Modifiers**

Modifiers create certain advantages to some factions, either at the start of the game or as the game progresses via events and goals. They are primarily based on historical advantages or disadvantages, but in ahistorical paths may represent hypothetical bonuses or maluses based on general context. For example, non-historical path of Oriental Crusade for Otomo will create a modifier for Europeans that will slow down Consolidation Rate representing Japanese people's resistance to foreign occupation and aggressive spread of foreign religion. Modifiers at first were just a dictionary with name and value of the modifier. Now they are separate objects that can have more properties, such as duration. New modifier mechanic also made UI script simpler, because there is no need to add modifiers separately for each type. Some additional methods creating, assigning and removing modifiers were also added.

## **2.12 Diplomacy**

Diplomacy mechanic is an expanded version of clan interactions. Initially, on release, options were limited to improving relations and alliance. Later it was expanded to include subjugation and annexation. It allows more interactions with an ability to add more diplomatic option later in development. It also more varied and precise interactions with longer lasting effects. For example, improve relations is now

continuous, and adds opinion every turn while costing 1% of income to maintain. There is also now a possibility to add ability to create broader coalitions manually.

### **2.13 Coalitions (Alliances)**

This mechanic was unique to Imagawa, as a way to represent their Triple Alliance, however it was expanded and some other factions are now also able to create Coalitions or Alliances similar to Triple Alliance. The main premise of these broader Alliances is that, unlike ordinary alliances, all members are interallied to each other. Alliances have a leader and a modifier to Army Morale that depends on the number of factions in it with leader's modifier being doubled. Only leader can disband Alliance. Members leave alliance by breaking alliance with the leader. Additionally, Alliances have a benefit when Alliance member conquers the Capital. All members are converted to vassals of the one that controls the Capital and will either submit or leave alliance.

### **2.14 Special Mechanics**

Some factions also have some special mechanics that add variety to the gameplay. For example, Otomo clan, famous for adopting Christianity, may at some point start spreading Christianity. Imagawa clan had a unique alliance set up where three major clans interallied to each other by marriage. During development this mechanic was expanded to some other factions as well.

## **3 MAIN GAMEPLAY AND INTERFACE SCRIPTS**

### **3.1 Initialization**

Main script that handles gameplay and initialization is attached to the highest game object in the hierarchy and Unity Message method called Start handles initialization. It creates all necessary army, country, and province objects. For easier access country and province objects are added to dictionaries that allow access to them by certain name. This allows access to these even if their clan's name is changed. For example, Tokugawa clan was previously called Matsudaira and later changed name. Same goes for provinces. Armies are assigned to their respective countries.

After this countries' province lists are populated with currently controlled provinces.

Next is diplomacy initialization. Some countries were at war with each other or had ongoing battles and alliances.

Major clans' cores and claims are added. Other clans get their cores based on whether their owned provinces initial consolidation is full and claims if these provinces are not fully consolidated.

Modifiers are assigned to countries based on their historical advantages and strong points. Or maluses if they had some disadvantage.

The last part is UI. After everything initialized UI components are updated with the initialized values including first player goal and map icons.

### **3.2 Turn Processing**

One turn is equal to one year.

At the start of each turn the first thing that is evaluated is battles. They are evaluated in a random order, to make it more unpredictable and simulate a possibility of battle happening at any time of the year.

After this some minor conditional adjustments are made. For example, countries that were destroyed lose their vassals. Capital controller gains control of the shogunate, if it is not Oda, Toyotomi or Akechi.

Goals and Events are handled through their respective methods.

If victory conditions are satisfied game activates a win screen and allows to restart or exit. If player loses all their provinces a lose screen is activated and player can restart or exit as well.

Next, specific country changes are handled. This includes changes in army and economy. Non-player countries also evaluate their attack possibilities and attack provinces or, otherwise, send reinforcements to allies.

At the end UI changes are handled, such as current goal and date.

### 3.3 Interface Scripts

To handle changes to the UI elements such as lists a separate set of scripts exists. They are generally meant to replicate enough template GameObjects and modify them. However, province UI is more complicated.

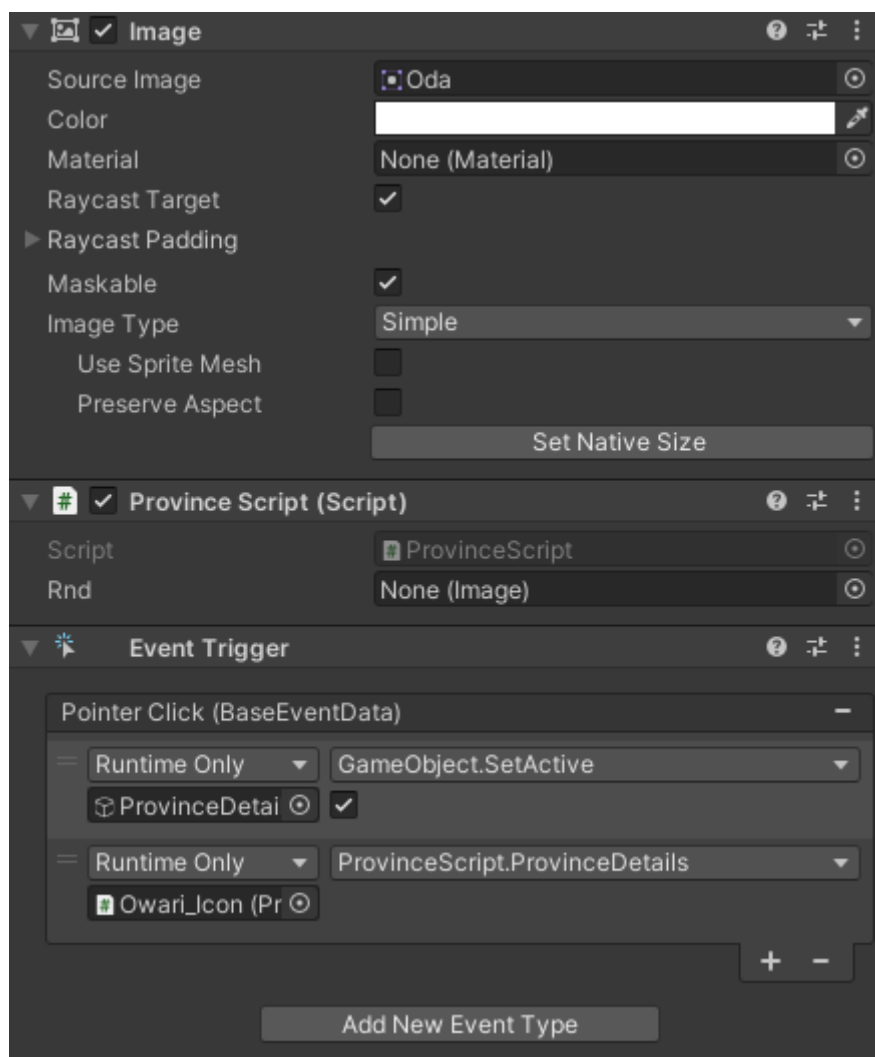


Figure 25. Icon Component example

### 3.4 Province UI

To handle changes to the province UI, a separate script handles everything that changes province or province details window appearance. In this script changes to province shapes' color and icons are made.

This script also handles display of player and province details stats as well as button availability and battle status.

### 3.5 Province Details UI

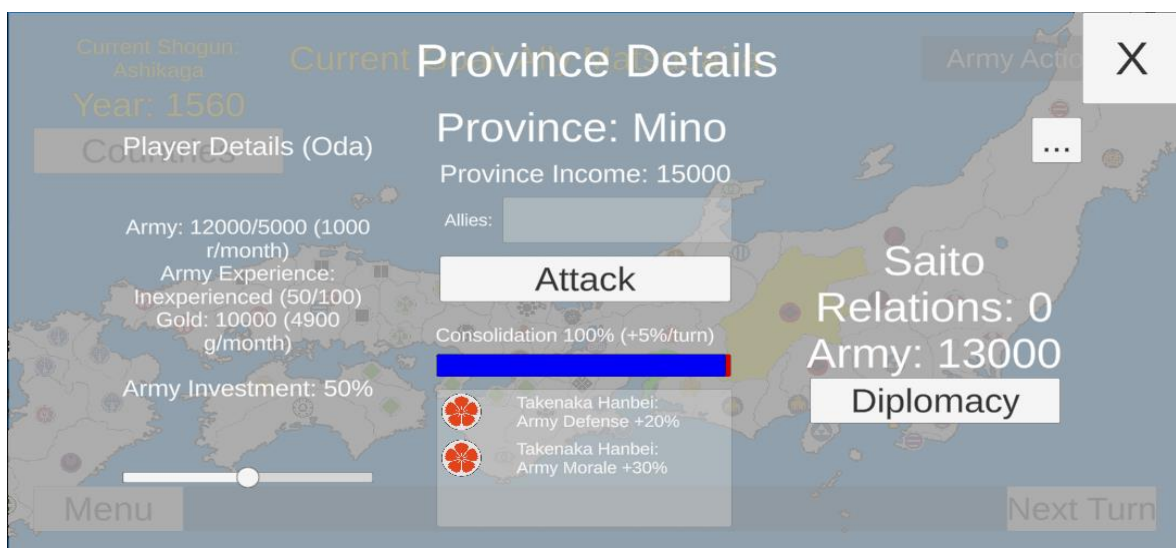


Figure 26. Province Details Window

This window opens when a player clicks on the province icon. It has 3 different views: current general view, diplomatic view, and battle view. Players can access diplomatic view by clicking diplomacy button and battle view by clicking button with three dots. Players can also attack province and change army investment with a slider. In diplomatic view player can enter treaty to increase relations, enter alliance, demand vassalage or annexation. In battle view a player can send reinforcements.

To open this window each icon has an event trigger that activates these windows and runs a method in UI script that assigns stats to UI components.

### 3.6 Province and Clan Interactions

Some interactions are restricted by certain condition, for example player cannot attack owned provinces or demand vassalage from themselves. So, all of these interactions

are disabled if a player opens their own province details. These conditions and button availability are also handled in Province UI script.

Player cannot attack if province already has battle, owner is ally, vassal or overlord or player army has no actions left.

Player cannot enter treaties to increase relations if player has no gold, or at war with the target.

Player cannot demand vassalage if player is a vassal themselves, if target is already vassal, if target is a shogun or at war with the target.

Players cannot demand annexation if at war with the target.

Players cannot enter alliance if relations are less than cordial. Alliance button changes to end alliance if player is already allied to target. In this case the player cannot end alliance if target is player's vassal or vice versa.

The last condition checks if the player can send reinforcements to the battle. Players can only send reinforcements to battle if either side is allied to player but cannot if both or neither are allied, or target owner is a vassal.

## **4 POST-DEVELOPMENT ADDITIONS**

### **4.1 Release on Google Play**

After the initial version of the game with one faction was completed, I released it on Google Play. Since release game managed to accumulate more than a thousand downloads.

Game is available on Google Play via following link:

<https://play.google.com/store/apps/details?id=com.KonstantinAleksandrov.Sengoku>

### **4.2 Additional factions**

After release a lot of additional playable factions were added to the game. Most of them playable from the start, but some of them are formed at certain points during the game. Not all factions have content at the moment, but it will be added in the future.

### **4.3 Advertisements**

At some point during development after some playable factions were added along with an ability to restart game, I decided to add interstitial ads. They play when player restarts campaign and provide some a small income without harming the gameplay.

### **4.4 In-app purchases**

Some time after advertisements were added I decided to add an ability to turn off ads for a small price. At the moment it is about 1 euro. It may differ in other regions due to taxes and laws. Game's first in-app purchase happened in January 2024.

## CONCLUSION

As a result of my work, I created a game that is available on Google Play and has more than a thousand downloads, summarized a crucial period of Japanese history in a game form using various mechanics, thus providing not only entertainment, but a historical background as well. I tested various mechanics and reworked some of them during the development.

When making this game, I tried to focus on playability and to have a game with at least one playable country with as few bugs as possible and good performance. This approach was successful, so now the game can be played on both PC in editor and on handheld devices. Of course, there is still a lot of room for improvement and advancement, but the main goals of the project have been met.

During development I had to rework some mechanics and add new ones. For example, I changed goals from sequentially fired methods returning Boolean completion status to static object-based goals that can be added several at the same time and have to be evaluated only once.

Event mechanic is also a result of similar rework. Initially events were handled along with goals as triggered effects. But to make them more interactive, I decided to make them their own mechanic and add UI for them. This allowed me to add more of them and add possibility of choosing different paths for the player in multiple choice events. Some other “quality of life” improvements were added as well, like pause menu and main menu where player can choose which clan to play as.

Eventually, after the first clan content was done, I decided to publish this game on Google Play and later implemented ads and monetization.

In the future I plan to keep adding content for remaining major clans and add more mechanics. For example, I am considering adding technology development in some form and by requests from players music, different languages, and multiplayer. I am also considering adding some sort of content progression, when player finishes one clan’s questline it unlocks next one and so on. This will also allow pay to skip or watch full ad to unlock possibilities.

## REFERENCES

- [1] Sansom, George (1961). A History of Japan, 1334–1615. Stanford University Press. page 217. ISBN 0804705259.
- [2] Ackroyd, Joyce. (1982) Lessons from History: The Tokushi Yoron, page 331.
- [3] Unity Documentation. (2018). EventSystem.  
<https://docs.unity3d.com/2018.2/Documentation/ScriptReference/EventSystems.EventSystem.html> (15.04.2024).
- [4] Unity Documentation. (2018). StandaloneInputModule.  
<https://docs.unity3d.com/2018.2/Documentation/ScriptReference/EventSystems.StandaloneInputModule.html> (15.04.2024).
- [5] Takeuchi, Rizō (1985). Nihonshi shōjiten, page 233.
- [6] Turnbull, Stephen (1987). Battles of the Samurai. Arms and Armour Press. page 37. ISBN 978-0-85368-826-6.
- [7] Weston, Mark. “Oda Nobunaga: The Warrior Who United Half of Japan”. Giants of Japan: The Lives of Japan's Greatest Men and Women. New York: Kodansha International, 2002. pages 140–145.
- [8] Sato, Hiroaki (1995). Legends of the Samurai. Overlook Duckworth. pages 234–37. ISBN 978-1-59020-730-7.
- [9] Mizutani, Tetsuya. “Chapter 5: Management Strategies Learned from the Three Local Great Masters - Nobunaga Shows His Vision of “Tenka Fubu””. Fujitsu Family Association Tokai Branch. Fujitsu.
- [10] Matsunaga, Yoshinobu (6 November 2022). “What is Nobunaga's ambition in “Gifu”?”. Asahi Shimbun. Tokyo.
- [11] Ishikawa, Takuji (2 September 2019). “Nobunaga Observations: The Realities of the Ruler of Japan - Chapter 9: Ashikaga Yoshiaki”. GOETHE. Gentosha.
- [12] Tonomura, Miki (15 July 2022). “The chain of innovations that led Oda Nobunaga to unify the country”. Nikkei BizGate. The Nikkei.

[13] Suzuki, Hiroki (7 June 2022). "Why did the 150-year-long Sengoku period end... The real reason why Oda Nobunaga, a small Sengoku Daimyo in Owari, was able to become a "Tenkabito"?. Nikkei Business. Nikkei Business Publications.

[14] Owada, Tetsuo (24 May 2020). "Oda Nobunaga (Part 1): A Genius Unbound by Old Ways, Creative Policies and Personnel Evaluation". Otonanswer. Media Vague.