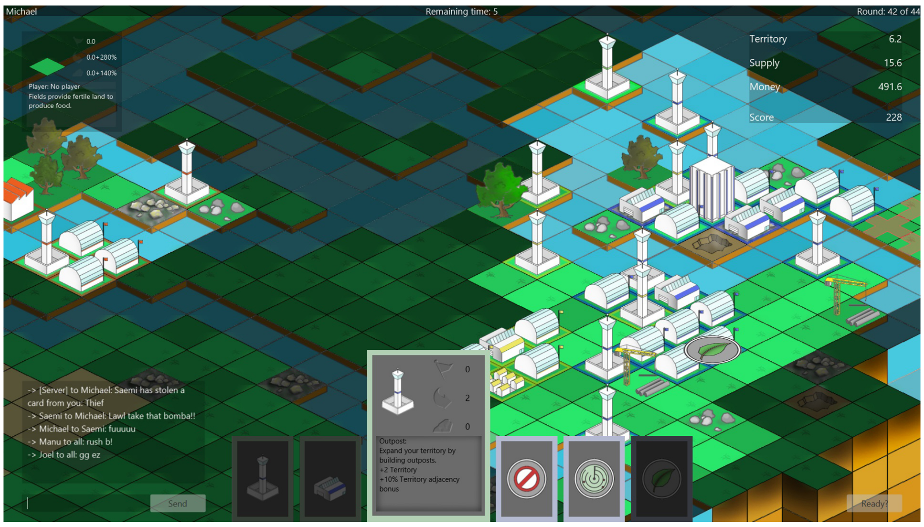
Nameless Lands
Manual

Developed by: Michael Plüss, Samuel Emde, Joel Gschwind and Manuel Rickli

May 21, 2017

Contents



Territory	6.2
Supply	15.6
Money	491.6
Score	228

0.0
0.0+280%
0.0+140%
Player: No player
Fields provide fertile land to produce food.

-> [Server] to Michael: Saemi has stolen a card from you: Thief
-> Saemi to Michael: Lawl take that bomba!!
-> Michael to Saemi: fuuuuu
-> Manu to all: rush b!
-> Joel to all: gg ez

0
2
0
Outpost:
Expand your territory by building outposts.
+2 Territory
+10% Territory adjacency bonus

1 Colonizing a new land

The earth is overcrowded and its natural resources have almost been farmed to the last crumble. Luckily, scientists have found a new planet which is suitable for human life.
Three nations dared the costly and difficult undertaking of colonizing the new world with its new, pure and yet unnamed lands with the goal of claiming as much of the planet as possible: Russia, China and the United States. All of them suspicious and eager to sabotage the others at any possible instance.
You are in command of one of the colonies. It is your mission to spread your influence as quickly as possible! But be aware: others are near you. Try to halt their progress, or they will be able to claim the land as theirs!

2 Using the Jar

2.1 Server

To get a server up and running, at least one argument has to be given by starting the jar. The arguments can have the following form:

- server
- server [port-number]

If no port-number is specified, the number 1337 will be used by default.

2.2 Client

Similar to the server, arguments have to be given. They can have the following form:

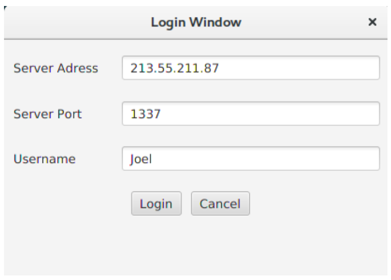
- client
- client [ip-address]:[port-number]
- client [ip-address]:[port-number] [username]

The arguments are then inserted in the textfields of the login window. If only the argument 'client' is given, the local host address is chosen along with the default port-number and the system-name. Without the argument for the username, the system-name will be used for the playername prompt.

3 Getting started

3.1 Login

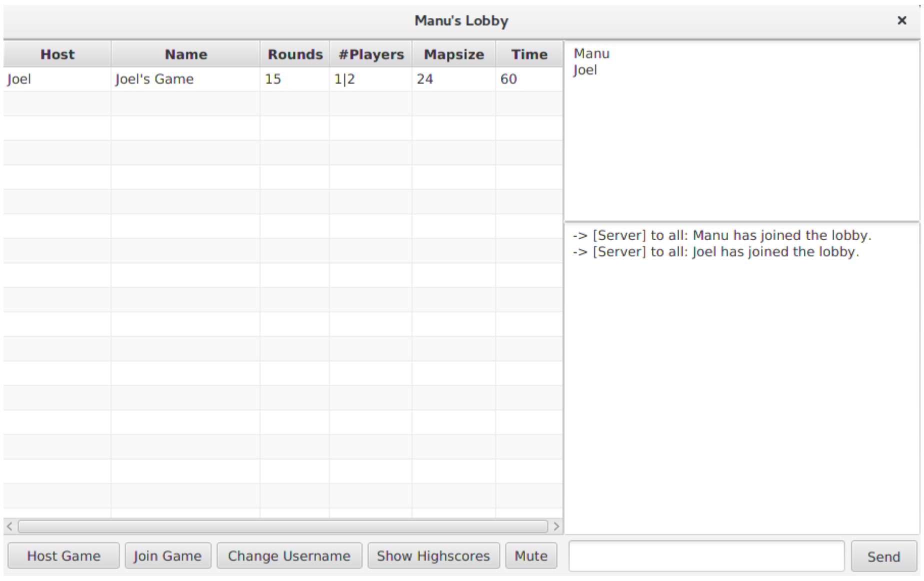
Entering the Lobby requires a connection to the server. To establish this connection, the correct ip-address and server port have to be typed in along with a unique playername. If all goes well, by pressing enter, a connection to a running sever is established and the lobby window opens.


Login Window

Server Adress	213.55.211.87
Server Port	1337
Username	Joel

3.2 Lobby

Arriving at the lobby, more functions await the player!



Host	Name	Rounds	#Players	Mapsize	Time
Joel	Joel's Game	15	1|2	24	60

Manu
Joel

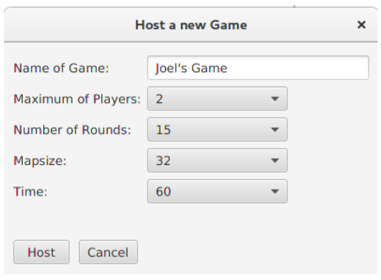
```
-> [Server] to all: Manu has joined the lobby.
-> [Server] to all: Joel has joined the lobby.
```

3.2.1 Hosting a game

If a new game is desired, this can be achieved by pressing the 'Host Game' button. This action displays a new window, in which the different settings of the new game can be set:

- Name of Game: The name of the game. Cannot be altered afterwards.
- Maximum of Players: Sets the number of players allowed in the game
- Number of Rounds: Sets the number of rounds and by this, determines the length of the game
- Mapsize: Sets the size of the map. For example 32 results in a map in a 32x32 grid.
- Time: Sets the time in seconds for the rounds. All actions have to be done within this timelimit.

By pressing 'Host', the PreGame window will be displayed (see PreGame).



Name of Game: Joel's Game
Maximum of Players: 2
Number of Rounds: 15
Mapsize: 32
Time: 60

3.2.2 Joining a game

If a game is already being hosted, a player can join it. This is only possible, if joining the game does not exceed the allowed number of players. To join a game, the button 'Join Game' has to be pressed, while the desired game is selected in the table above.
This action will also display the PreGame window (see below).

3.2.3 Chatting

To communicate with other people in the lobby, the chat can be used. By default, everything is sent to each player in the lobby. To send a message to a specific player, the Tab key should be used. By pressing Tab, the current players are displayed one after another in the field. When the desired player is displayed, the message to be sent can simply be entered aside.

3.2.4 Highscores

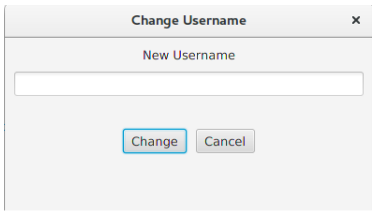
By pressing the 'Show Highscores' button, a window with the top ten scores is displayed.

3.2.5 Muting the sound

By pressing 'Mute' the sound is muted.

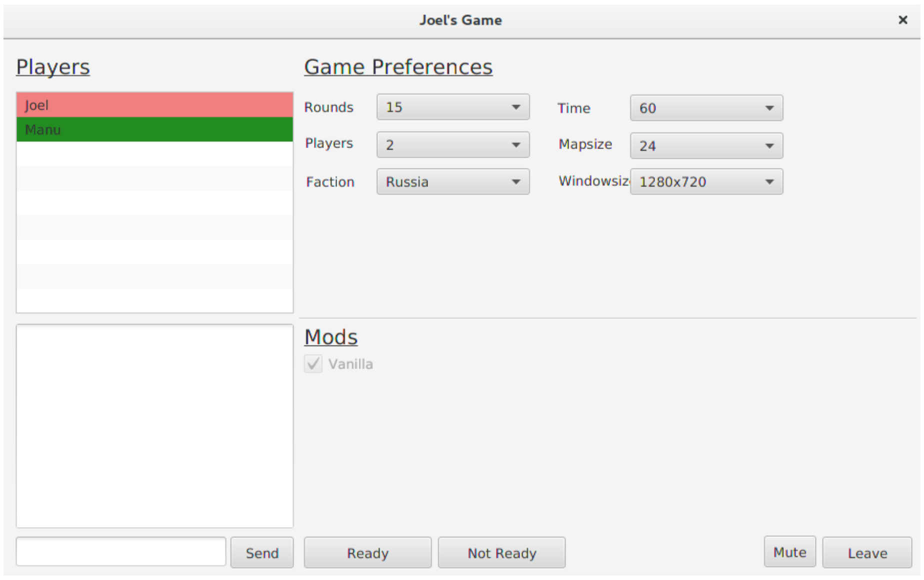
3.2.6 Changing the name

There is the possibility to change the playername after loggin into the lobby. This can be achieved by pressing 'Change Username' and entering the desired name in the displayed window. Like before, the name has to be different from the name of the players currently logged in.



3.3 PreGame

The PreGame window serves as a separate lobby, only for the players enlisted in the game. The chat in encapsulated from the other chats, meaning only players in this game see the messages.


Joel's Game

Players

Joel

Manu

Game Preferences

Rounds 15

Time 60

Players 2

Mapsize 24

Faction Russia

Windowsiz 1280x720

Mods

✓ Vanilla

3.3.1 Altering the settings

The host has the ability to alter the settings of the game before it is started. These changes are synchronized with every other PreGame window or table entry in the lobby of the other players, so they are always up to date.
Players who have joined the game can choose their faction.

3.3.2 Ready?

The background of the playername in the list shows if the player is ready or not. By pressing 'Ready', the background of the player who pressed the button turns green. If every player is ready, a timer counts down from three and the game starts! This happens only, when more than one player is in the game, but also, if the number of maximum players is not yet reached.

3.3.3 Leaving

By pressing 'Leave', the player gets back to the lobby and is no longer a member of the game. If the player leaving is also host of the game, the game is deleted and every member will be removed and put back into the lobby.

4 Playing the game

4.1 Goal

Competing against other colonizing settlers, the goal is to be the most successful in doing so. How well this is done by a player is measured in three resources (see Resources) and determines eventually who succeeded the most and will be able to claim the nameless land for himself.

4.2 Factions

4.2.1 Russia

As one of the biggest countries, they do not want things to change in the new world. Their specialization lies on the territory in which they see the real power of a nation.

4.2.2 China

To own vast amounts of land is of no use if you do not have the supply to give your civilizaition a good heads up. Following this rule, china has specialized some of their bulidings to produce an extra amount of supply.

	Territory	Supply	Money
Tile bonuses:	-	+5%	-

4.2.3 USA

Valuing the earths treasures and wealth itself, the USA thrives for an economical advantage. Not only have they mastered the art of gathering rare materials, but also the trading with goods.

4.3 Resources

To play a card, resources are needed. These are provided and used by buildings and their production can be enhanced by the placing of the building and by effects.

4.3.1 Territory

Each building either gives or needs territory. To be able to place a building that needs territory, free space has to be available. If no free space is available, only buildings that add space are allowed to be built.

4.3.2 Supply

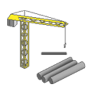
The supply resets each round, which means the player cannot store it. The amount of supply a player has at hand in the current round is determined by the supply-producing buildings he owns. Various effects and the placement of the buildings can make a significant difference in the production!

4.3.3 Money

Since capitalism prevailed throughout the history of men, money is still used and therefore needed for almost everything. Unlike supply, money can be stored.

4.4 Buildings

The main way of receiving resources are buildings. Their placement can have an effect on the production of its resource (see Field types) and should be considered.

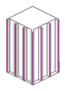


4.4.1 Special buildings

Each nation believes in a different resource to be the key to success. Therefore they specialized some of their structures for the purpose of delivering more of their resource of choice. These structures are only accessible for their nation and can be used to gain a certain advantage.

Russia

Pioneer probe

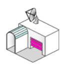


This unique probe was designed to be the first building to be erected in the new world and grants an additional bonus for territory.

	Territory	Supply	Money
Resources per turn:	3	2	1

Visibility range:	3 tiles

Secret service



Why change things that have proved to be working? By spying on their own people, the russians know about everything that is going on and are therefore able to control the expansion of their territory better.

	Territory	Supply	Money
Cost:	2	2	7
Resources per turn:	4	-1	-1
Adjacency bonuses:	+20%	-	-

Visibility range:	3 tiles

China

Colony ship

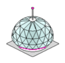
This is the first structure of the chinese colonialists and is designed to produce great amounts of supply right away.

	Territory	Supply	Money
Resources per turn:	2	3	1

Visibility range:	2 tiles

Fusion reactor

Advanced techonlogy led to this technical marvel which produces more energy than anyone could wish for.



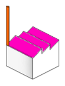
	Territory	Supply	Money
Cost:	1	5	5
Resources per turn:	-1	3	1
Adjacency bonuses:	-	+20%	-

Visibility range:	1 tile

USA

Extraction drone

In a world with no existing market, the first thing to do is gathering materials which can be used right away to gain economically advantage over other nations. Their start-building is designed for this very purpose.

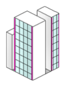


	Territory	Supply	Money
Resources per turn:	2	2	2

Visibility range:	2 tiles

Stock exchange

The interest in money peaks in this building. Like in the old world, it is the center of all tradings with stocks.



	Territory	Supply	Money
Cost:	3	6	2
Resources per turn:	-2	0	3
Adjacency bonuses:	-	-	+20%

Visibility range:	1 tile

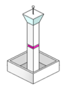
4.4.2 Mall

A place for trading goods and selling products. Results in a thriving economy which produces money for its settlers.

	Territory	Supply	Money
Cost:	1	1	1
Resources per turn:	-1	0	1
Adjacency bonuses:	-	-	+10%
Adjacency bonuses (USA):	+10%	+10%	+10%

Visibility range:	1 tile

4.4.3 Outpost

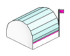


To produce territory, outposts are the way to go. Their vision is remarkably large and allow the player to expand quickly.

	Territory	Supply	Money
Cost:	0	2	0
Resources per turn:	2	0	0
Adjacency bonuses:	+10%	-	-

Visibility range:	2 tiles

4.4.4 Farm



A growing civilization needs supply to live and expand. By building farms, the amount of supply avaible for each round rises.

	Territory	Supply	Money
Cost:	1	1	0
Resources per turn:	-1	1	0

Visibility range:	1 tile

4.5 Actions

Action cards are crucial for winning the game. They either help the player by enhancing the production of the buildings or by interfering with the enemy. But caution is adviced, for all actions affect everyone in its radius.

4.5.1 Fertilizer

Unleash the full potential of the supply-production!

The fertilizer enhances the production of supply by one in a 3x3 area for two rounds.

Radius	Duration	Effect
1 tile	2 turns	+1 Supply production

	Territory	Supply	Money
Cost:	1	0	2

4.5.2 Block

Not being able to build anything for two rounds is something one wishes for his enemies. The solution lies in this card!

This effect prevents cards to be played in a 3x3 radius for two rounds.

Radius	Duration	Effect
1 tile	2 turns	Prevents playing of cards

	Territory	Supply	Money
Cost:	1	3	2

4.5.3 Radar

If the enemy's activities should be more than guesses, vision is needed!

The radar is used to gain vision of the map by revealing a 3x3 radius. It's duration is two rounds.

Radius	Duration	Effect
2 tiles	2 turns	Reveals map

	Territory	Supply	Money
Cost:	2	0	1

4.5.4 Infiltrate

Better than blocking the production, is stealing it!

This effect gives all the production of a building on the tile, the card is played on, to the player who used it.

Radius	Duration	Effect
-	1 turn	Converts building

	Territory	Supply	Money
Cost:	2	2	2

4.5.5 Airstrike

The Airstrike gets rid of everything in its radius for good.

Destroys the building on the tile the effect takes place.

Radius	Duration	Effect
-	-	Destroys building

	Territory	Supply	Money
Cost:	3	0	10

4.5.6 Thief

An involutary donation, consisting of an enemy's card to the players deck, may not please the donor, but kindness has no place in this cruel wrold!

Steals a card from the player who owns the tile the card is played on.

Radius	Duration	Effect
-	-	Steals card

	Territory	Supply	Money
Cost:	0	2	4

4.6 Field types

4.6.1 Gras

Idyllic fields were always associated with fertility and prosperity. This proved to be true in the new world as well.

	Territory	Supply	Money
Tile bonuses:	-	+10%	-

4.6.2 Desert

A harsh place that offers nothing but space.

	Territory	Supply	Money
Tile bonuses:	+10%	-	-

4.6.3 Ores

Rocks containing shimmering pieces of metal were always highly valued throughout the history.

	Territory	Supply	Money
Resources per turn:	0	0	1

4.6.4 Water

Being one of the major aspects when placing buildings, water grants an improvement to produciton.

	Territory	Supply	Money
Adjacency bonuses:	-	+10%	+10%

4.6.5 Forest

Although almost entirely covered with trees, sometimes the forest floor has a free space. This particular spot can be valuable.

	Territory	Supply	Money
Tile bonuses:	-	-	+10%

4.6.6 Crater

Craters form when a patch of land was being bombed. Only build here when you do not care about the wellbeing of your offspring.

	Territory	Supply	Money
Tile bonuses:	-10%	-10%	-10%

4.6.7 Boulders

Massive boulders make it impossible to build anything on these tiles. Only explosive power might help here.

4.6.8 Trees

The buildings are not meant to be built around trees! Therefore they act in the same manner as the boulders do.

4.7 Ingame functions

4.7.1 The menu

By pressing Escape, an overlay with options will appear. This does by no means pause the game.

Mute
This toggle button shows if the sound is muted or not.

Exit Game
If for some reason a player wants to yield, this button will close the running game and display the lobby. This button should also be used if the game has ended and the players wants to return to the lobby instead of closing the game.

4.7.2 Chat

The chat is not only used for sending messages between players, but also for some game informations to certain players. For example when the thief effect is successful, the victim and the thief both receive a message via chat to inform them, which card was stolen.

4.7.3 Ready

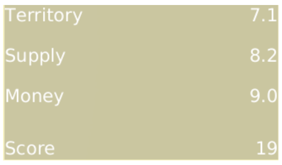
To be ready, the player can either press the 'Ready?' button at the bottom left or simply press the Space Bar.

4.7.4 Drag and Zoom

To get a better picture of the map, it is possible to zoom in and out with the mouse wheel and move it by holding down the right mouse button and dragging it.

4.8 Resources window

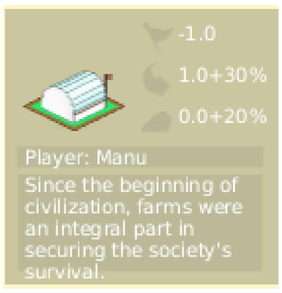
In the top right corner is a window which displays the currently available resources and the score.



Territory	7.1
Supply	8.2
Money	9.0
Score	19

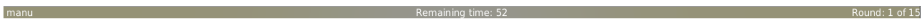
4.9 Information window

In the top left corner is a window which displays information about the object the mouse is hovering over. Pressing on the window toggles a new window with information about the owner of the tile and a description. For example if it is over a building, the production of it is shown along with the playername of the owner and a description of the building.



-1.0
1.0+30%
0.0+20%
Player: Manu
Since the beginning of civilization, farms were an integral part in securing the society's survival.

4.10 Top bar



manu Remaining time: 52 Round: 1 of 15

4.10.1 Time

In the middle of this bar is the timer, which shows how many seconds are left of this round.

4.10.2 Rounds

To the right of the time is the number of the current round.

4.10.3 Name

The player's name is to the left of the time.

4.11 Rules

4.11.1 First round

Arriving at the nameless lands, the first thing to do is establish some space by building the starter-building. This building can be placed on gras, deserts, forest floor or ores. The placing should be thought through, for the upfollowing building has to be adjacent to the starter building.

4.11.2 Finishing the round

If the building is placed, the player either set himself as ready or wait until the time for this round runs out. If every player pressed ready, the next round is initiated. Between two rounds, the buildings are constructed and produce resources right away.

4.11.3 Two buildings on the same field

In case two players placed their building on the same field, the one who did first is favoured. If this is the case in the first round, the second player does not get any additional cards in the second, but has to choose a different location for his starter-building. If this scenario happens after the starter-building is placed, the card of the building is simply put back into the deck.

4.11.4 Last player in game

If only one player remains in the game, he automatically wins it!

4.11.5 Ending of the game

When the last round is played or only one player is still in the ongoing game, the score of the winner will be displayed. To return to the lobby, the 'Exit Game' button in the menu must be used.

5 Modding

The possibility to change and add to the content of the game was a major concern during the production of "Nameless Lands". This is why we made it possible to modify many aspects of the game. You may change the graphics of the game, the cards, the buildings and the effects as well as the conditions for all of them. When modding, you have to be aware that differing game data on clients and server may lead to errors and crashes. Make sure that all the players play the same version of the game! When in doubt, the game data on the server counts.

5.1 File system

All modifyable game data may be found in the "mod" folder. Each mod is in its own folder. The original game data is stored in the "vanilla" folder. Feel free to add new mod folders. Each mod folder requires two sub-folders: "graphics" and "definitions". Unsurprisingly the "graphics" folder stores the graphics of the mod and the "definitions" folder stores the game logic. In the "vanilla" folder you will also find a folder named "stylesheets" and in it a file with the name "gameGUI.css". This file is not crucial to the working of the game but offers a possibility to slightly alter the look of the In-Game-Gui. Feel free to take an example on the vanilla file structure if anything is unclear.

mod1
mod2
mods.txt
vanilla

Figure 1: An example how this might look.

5.2 Adding a mod

To add your mod to the game, you need to enter it into the "mods.txt" file which may be found in the "mod" folder. Simply write the name of the folder of your mod on a new line. Any mods specified in this file will be loaded. The order you write the mods in also defines their loading order: top first, bottom last.

5.3 Writing content

Now let's discuss writitng new content!

5.3.1 Adding new content

To add new content you can simply create a new file in the "definitions" folder. Any file in this folder will be read out and parsed. The name of the files is not important and of no concern to the parser. It only helps for keeping the overview over the game data.

5.3.2 Modifying existing content

Alternatively you may modify existing content. You may either modify existing files or add new files which overwrite existing content. To do the latter, just define game data with the same textID as an existing definition and make sure that your new definition comes later in the loading order. (This means putting it below the original game data in the "mods.txt" file.)

5.4 Kinds of game data

There are many different kinds of game data that you may define. Most of it is correlated to a so-called textID. The textID is a way to identify a game object and is used to communicate game data between server and client.

5.4.1 Cards

Cards are possibly one of the first things which you might want to add or modify. Cards are the way to interact with the world in the game and as such have a very direct impact on the game experience.

5.4.2 Buildings

Buildings are also very important to the game experience. Many cards are used to construct buildings. However some buildings are not created by the players but are generated with the map, such as trees or boulders.

5.4.3 Tiles

Tiles are what the map is built out of. They include things like fields, rivers and ore occurrences. In the vanilla game, there is no way to change the tiles on the map, but nonetheless they have a big impact on the game by giving boni on the production of resources. Tiles are also the way that the behaviour of the map generation is defined.

5.4.4 Civilizations

Civilizations (indirectly) define which cards and starting buildings a player recieves. Indirectly, because this is not defined in the civilization object but in the card object, where the civilization is referenced.

5.4.5 Sprites

Sprites are a more technical kind of game data. Most of the sprite definitions of the original game are found in the "System_Graphics.txt" file where you will find a reminder to be careful when changing the Sprites. This is because the textIDs of the sprites are hardcoded into the draw method of the code. However don't be shy to change the pictures of the Sprites. This is no problem at all.

5.4.6 Effects

Effects are special in that they are not defined by them selves like the other game objects but are always nested into the definition of another game object. Effects define the actual scripts that are executed when for example a card is played. A large portion of this modding documentation will also focus on the writing of scripts. More on that in the section about scripting.

5.4.7 Conditions

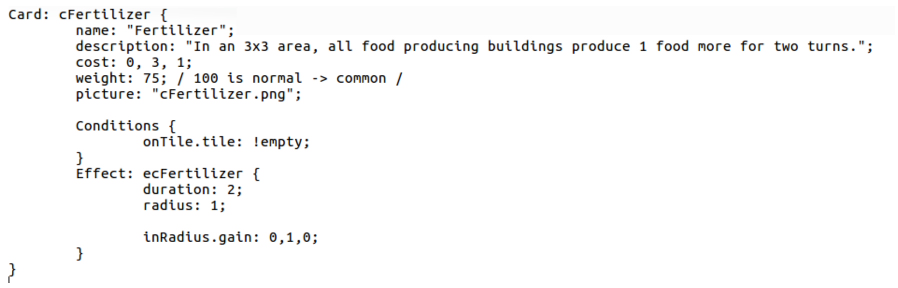
Conditions are very similar to effects as they are also always nested into another object. Differently than effects they do not need a textID. They are also written very similar to effects. More detail is to be found in the section about scripting.

5.4.8 Pictures

Pictures, like Conditions are nested into other definitions and do not have a textID. Many objects can hold pictures. These objects are: Tiles, Buildings, Cards and Sprites. You may attach multiple pictures to a single object. The object will automatically chose one of the pictures that fulfills its conditions, if there are some.

5.5 Definitions

Every definition starts with a keyword. Normally, this keyword is followed up by a ":" and a textID. Afterwards, the values of the object are defined inside curly brackets. The closing of the curly brackets signifies the end of the definition of this game object.



```
Card: cFertilizer {
        name: "Fertilizer";
        description: "In an 3x3 area, all food producing buildings produce 1 food more for two turns.";
        cost: 0, 3, 1;
        weight: 75; / 100 is normal -> common /
        picture: "cFertilizer.png";

        Conditions {
                onTile.tile: !empty;
        }
        Effect: ecFertilizer {
                duration: 2;
                radius: 1;

                inRadius.gain: 0,1,0;
        }
}
```

Figure 2: The definition of the Fertilizer card as an example.

Values are defined after a consistent scheme. First you write the value that you want to define, then you put a colon (":"). Then come the actual values, called parameters, seperated by commas. End the definition of the value with a semicolon (";"). Values that only contain letters and numbers do not need apostrophes, but if there is any character that is neither letter nor cipher, you need to put it inbetween apostrophes as seen above in the figure. This also goes for negative numbers! Hyphens, used to mark negative numbers, have to be put into apostrophes.

5.6 List of definition values

This table gives an overview over the definition values that are available and how to use them.

5.6.1 General definition values:

- name: The name of this object.
  - name: Name; (String)
- description: The description of this object. Is only displayed on cards.
  - description: Description; (String)
- picture: Adds a picture or animation to the graphical representation of this object. Objects may hold multiple pictures. In that case you have to define a variation value for each picture. The object will automatically chose one of the pictures to display. Pictures are special because they can also be nested into the definition of objects as an object themselves. This gives the opportunity to add Conditions to pictures.
  - picture: Path [,Frames] [,Variation]; (String, int, int)
    Path is the path of the picture, relative to the graphics folder. Frames is an optional value that defines how many frames are to be extracted from the image file. This automatically animates the object. Variation is a value that you only have to define when an object has multiple pictures. May be any integer value.
- radius: Defines the radius in which this object acts. This value may be used in two ways: Either inside an effect definiton, where it defines the area in which the effect applies or inside other object definitions where it is used by Conditions to check things inside the area around the object. The standard value is 0, where the effect only applies to the tile whereon it lies. A value of 1 will ultimately include an area of the size 3x3.
  - radius: Radius; (int)

5.6.2 Definition values specific to Cards:

- cost: The cost of the card.
  - cost: Territory, Food, Money; (int, int, int)
  - cost: Resource, Cost; (String, int)
    In the place of "Resource" you write "territory", "food" or "money".
- weight: The probability of this card to randomly spawn in a deck of cards. Cards with higher weights are more likely to spawn.
  - weight: Weight; (int)
- amount: Defines the exact amount how many times this card should appear in a deck of cards.
  - amount: Amount; (int)

- civilization: Restricts this card to only be in decks of players that have the specified civilization.
  - civilization: CivilizationID; (String)
    The textID of the civilization.
- startercard: Defines this card as a starting card for the given civilization. In the beginning of the game, the player will randomly be given exactly one of the cards that is flagged as a startercard for his/her civilization.
  - startercard: Civilization; (String)
    The textID of the civilization, that this card is a starting card for.

5.6.3 Definition values specific to Tiles:

- spawn: Defines the spawning behaviour of this tile during the map generation. There are three modes: "default" where the tile is used as a standard map tile (see fields), "cluster" where patches of tiles of this type spawn on the map (see forests) and "line" where lines of this type of tile are layed on the map (see rivers).
  - spawn: Spawn-type, Weight; (String, int)
    The spawn type is either "default", "cluster" or "line" and the weight defines the probability of this tile compared to other tiles of the same spawn type.
- spawnbuilding: If this value is defined, the tile will sometimes spawn a building of the given type on itself during map generation.
  - spawnbuilding: Building, Probability; (String, double)
    Building refers to the textID of the wished building. The probability is a double value between 0 and 1. Attention: Do not forget to put double values into apostrophes, or the comma will not be parsed correctly!
- produce: The resources that this tile will naturally produce per turn.
  - produce: Territory, Food, Money; (int, int, int)
  - produce: Resource, Production; (String, int)
    In the place of "Resource" you write "territory", "food" or "money".
- waterfall: Tiles with this flag will cause the edge of the map to display a waterfall instead of the standard dirt when on the border of the map.
  - waterfall: Waterfall; (boolean)
    The boolean value has to be "true" or "false", written in lowercase.

5.6.4 Definition values specific to Buildings:

- produce: The resources that this building will produce per turn.
  - produce: Territory, Food, Money; (int, int, int)
  - produce: Resource, Production; (String, int)
    In the place of "Resource" you write "territory", "food" or "money".

5.6.5 Definition values specific to Effects:

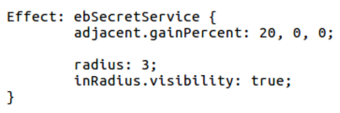
- duration: Defines how many turns this effect will last. An effect with the duration -1 will last forever.
  - duration: Duration; (int)

5.7 Scripting

Knowing how to script is very important for modding, since most ways to influence the game is are by script. But don't panic! It's very easy.

5.7.1 Where to put a script

Scripts are single lines of text inside an effect. Effects may be added to any other object, but are not always executed. For example, effects added to sprites or pictures are not applied.



```
Effect: ebSecretService {
        adjacent.gainPercent: 20, 0, 0;

        radius: 3;
        inRadius.visibility: true;
}
```

Figure 3: The effect of the Secret Service as an example.

You may add multiple effects to an object, that is no problem. All the effects will be applied in that case.

5.7.2 Structure of a script

Scripts consist of three parts: the location, the command and the values. Some scripts do not require the values and thus end after the command.

Syntax location . command : value1 , value2 , value3 ;

Location To understand the importance of locations, first you need to know that effects are always played onto tiles on the map. The effect is then applied on tiles in certain positions relative to the tile the effect is played on. That's what the location part of the script is for. Effects can influence the tile they are played on itself, surrounding neighbour tiles, tiles in a specific direction or even all the tiles within a certain radius around the effect's tile.

- onTile. Points to the tile that the effect is played on.
- adjacent. Points to the four direct neighbours of the effect's tile.
- inRadius. Points to each tile inside the radius. For example a radius of 2 covers an area of the size 5x5. The radius has to be defined (standard is zero). This includes the tile the effect is played on.
- topTile. The tile atop the effect's tile. Visually top-right.
- bottomTile. The tile ot the bottom of the effect's tile. Visually lower left.

- leftTile. The tile to the left of the effect's tile. Visually top-left.
- rightTile. The tile to the right of the effect's tile. Visually lower right.
- topLeftTile. The tile to the top left of the effect's tile. Visually to the top.
- topRightTile. The tile to the top right of the effect's tile. Visually to the right.
- bottomLeftTile. The tile to the bottom left of the effect's tile. Visually to the left.
- bottomRightTile. The tile to the bottom right of the effect's tile. Visually to the bottom.

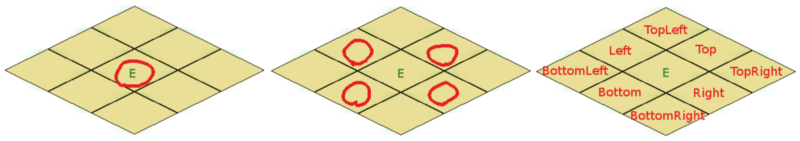


Figure 4: From left to right: onTile, adjacent and the various special ones.

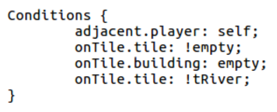
Command The command specifies the actual change that the effect causes on the map. This change is applied to each tile that is affected by the effect seperately.

- gain: The tile gains the specified amount of resources. If the tile belongs to a player, he or she will collect these resources in the next turn.
  - location.gain: Territory, Food, Money; (int, int, int)
  - location.gain: Resource, Amount; (String, int)
    In the place of "Resource" you write "territory", "food" or "money".
- gainPercent: The tile gains the specified amount of resources percentage modifiers.
  - location.gainPercent: TerritoryPercent, FoodPercent, MoneyPercent; (int, int, int)
  - location.gainPercent: Resource, Percentage; (String, int)
    In the place of "Resource" you write "territory", "food" or "money".
- build: Builds a building of the type of the given textID on the specified tile.
  - location.build: textID; (String)
- tile: Sets the affected tile to the type of the given textID.
  - location.tile: textID; (String)

- destroy: Destroys the building on the tile. Does not require any values.
  - location.destroy;
- block: Blocks the affected tiles, making it impossible for anyone to play cards onto them. Does not require any values.
  - location.block;
- visibility: Sets the visibility of a tile. A tile that is visible is not affected by the fog of war.
  - location.visibility: Visible; (boolean)
    In place of Visible you have to either write ”true” or ”false”.
- stealcard: Steals a card from the player on the effect's tile and puts it into the deck of the player that caused the effect. Nothing happens when the tile does not belong to another player. Does not require any values.
  - location.stealcard;
- stealeffects: This effect will cause the affected tile to convert to the player who played the effect. When the effect expires, it will go back to the original player. Nothing happens wehn the affected tile does not belong to another player. Does not require any values.
  - location.stealeffects;

5.7.3 Conditions

Conditions are quite similar to normal scripts, but used in a different context. They share the characteristic structure, but instead of commands, they feature targets. Also they use the values differently, as in that the target is checked against the values. Conditions are discerned from scripts by putting them inside a condition object. See the figure below as an example.



```
Conditions {
        adjacent.player: self;
        onTile.tile: !empty;
        onTile.building: empty;
        onTile.tile: !tRiver;
}
```

Figure 5: This condition may be found in the definition for the Farm card.

Syntax location . target : value1 , value2 , value3 ;

Use Conditions may be put into any other object but do not always take effect. Normally they apply in the context of cards or effects. Cards can only be played on tiles that fulfill the card's condition and effects are only applied on a tile when it fulfills the effect's condition. Also conditions may be added to pictures. When for example a building definition has a picture object defined in it and the picture has a condition, then the condition will be checked and only when it is fulfilled, the picture may be chosen to display the building. For an extreme example of this, look at the definition of the river tile, which is in it's

own file named "Tiles_River.txt".
Conditions do not need a textID.

Compliance For a card to be playable or an effect to be activated, all it's conditions have to be fulfilled in the fashion of an "and" logic condition. In the example of the figure above, all four conditions have to be fulfilled for a tile, before the player can play a Farm card on the tile.
There is a possibility to construct an "or" logic condition: A condition may be checked against multiple values by separating the value blocks with a "|". If any of the values fits the check, the condition will be fulfilled.

Negation Conditions may be checked against the negation of a statement, by using an exclamation mark. Put the exclamation mark in front of the value that you want to check against. It will function in the same way as negating a boolean in any major programming language.

Target

- tile: A check on the textID of the affected tiles. May also be checked against "empty" to know whether the tile exists at all.
  - location.tile: textID; (String)
- building: A check on the textID of the affected tiles buildings. May also be checked against "empty" to know whether the building exists at all.
  - location.building: textID; (String)
- produces: A check on the resources that the current tile produces.
  - location.produces: Resource; (String)
    For Resource you may put in "territory", "food", "money" or "empty".
- player: A check against the owner of the tile in comparison to the playing player.
  - location.player: Player; (String)
    For Player you may put in "self" or "empty". "self" references the player himself and "empty" means that the tile belongs to noone.
- civilization: A check on the textID of the civilization of the player on the tile. May also be checked against "empty" when the tile should not belong to any civilization.
  - location.civilization: textID; (String)