

COSC 3P91 – Assignment 2 – 6920201 & 6973523

BRETT TERPSTRA & MICHAEL BOULOS, Brock University, Canada

1 Info

The attached project is a strategy game played on the system console, which acts as the interface between a player and the game world. The game runs on a runnable `gameEngine` object which facilitates game events, keeps track of game events, and defines the top-level behaviour between game objects. Appropriately, execution of the game begins within the main function, which instantiates the game engine and calls its `run` function.

Within the `run` function is the game loop for this strategy game. The game loop involves six main user-triggered main events, and one main underlying update event:

User-Triggered Event	Event Description
Build	When this event is triggered, the engine builds the building specified in the command line arguments associated with the command, adding it to the user map contained in the engine members.
Train	When this event is triggered, the engine trains/produces the inhabitants specified in the command line arguments associated with the command, adding them to the list of inhabitants in the associated user map.
Upgrade	When the upgrade event is triggered, the engine upgrades the unit specified by the index argument and type argument associated with the command, the engine will upgrade the building accordingly to the next allowed stage given the index is valid within the list of units in the map and when the village resource requirements are met.
Explore	When the explore event is triggered, the engine will display the next possible map to attack, generated based on find a suitable village relative to the player village's stats.
Attack Explored	When the attack event is triggered, the engine will utilize the stats of both the player village and the last non-player village explored to generate an attack result.
Background Event	Event Description
Update	The update event calls every game loop; it utilizes an in-map timer object, specified in its own utility timer class to facilitate the timing on when certain resources will be mined/produced and other time sensitive game events.

Note: When classes and methods like `Tile` and `move()` are unimplemented or unused, it is due to them being supplemented as design elements that will later come into use once the engine is required to simulate certain events at a more fine-grain level and when the game GUI is entirely implemented.

2 Usage

The following describe the appropriate usage of the game and how to interact with game events:

2.1 Player Menu

When starting the game, the following menu will appear:

```
+-----{ Current Village State }-----+
| Resource Type | Max | Amount |
+-----+-----+-----+
| Wood | 5000 | 50 |
| Iron | 2500 | 0 |
| Gold | 1000 | 0 |
+-----+-----+-----+
+-----{ Village Buildings }-----+
| Name | Level | Health |
+-----+-----+-----+
| CasaDeNarino | 1 | 100 |
+-----+-----+-----+
+-----{ Village Inhabitants }-----+
| Name | Level |
+-----+-----+
| Worker | 1 |
| Worker | 1 |
| Collector | 1 |
+-----+-----+

~ Player Options:
1. Build {command: '1 <building name>'}
2. Train inhabitants {command: '2 <unit name>'}
3. Upgrade {command: '3 i<index>' / {command: '3 b<index>'}
4. Explore
5. Print Village Stats
6. Quit
7. Attack last explored
```

Fig. 1. The option menu and village state when starting the game

Each command, under ‘~ Player Options:’, is illustrated next to a digit. To run a command, enter the digit associated with it by typing it in as console input.

Any command that has a **command template** on its right, i.e. `Build {command: ‘1 <building name>’}`, requires additional arguments to be input when writing the command which are outlined by the specified template. Any command that does not have a template on its right can simply be ran by entering just the digit associated with it. Below are the usages of commands that require a special argument.

2.2 Building

To build a building within the player village, enter the command `1` followed by the name – non case-sensitive -- of the building you would like to build; for example, to build a Lumber Mine building, one would enter:

```

~ Player Options:
1. Build {command: '1 <building name>'}
2. Train inhabitants {command: '2 <unit name>'}
3. Upgrade {command: '3 i<index>' / {command: '3 b<index>'}
4. Explore
5. Print Village Stats
6. Quit
7. Attack last explored

1 lushanline

```

Fig. 2. An example of the usage of the Build command, input in green

2.3 Train

To train/produce an inhabitant within the player village, enter the command 2 followed by the name – non case-sensitive -- of the inhabitant you would like to train; for example, to train a worker, one would enter:

```

~ Player Options:
1. Build {command: '1 <building name>'}
2. Train inhabitants {command: '2 <unit name>'}
3. Upgrade Building {command: '3 i<index>' / {command: '3 b<index>'}
4. Explore
5. Print Village Stats
6. Quit
7. Attack last explored

2 worker

```

Fig. 2. An example of the usage of the Train command, input in green

2.4 Upgrade

To upgrade an inhabitant within the player village, enter the command 3 followed by a unit letter – i for inhabitant or b for building – and the index of the unit within the village state list, the list that is printed at the beginning of the game or when command 5 is called, of the inhabitant or building you would like to upgrade; for example, to train the second worker on the inhabitants list, one would enter:

-Note: the list is zero-indexed, so enter '2' for the third unit, '1' for the second, '0' for the first, etc.

+-----{ Current Village State }-----+			
Resource Type	Max	Amount	
Wood	5000	50	
Iron	2500	0	
Gold	1000	0	
+-----{ Village Buildings }-----+			
Name	Level	Health	
CasaDeMarino	1	100	
+-----{ Village Inhabitants }-----+			
Name	Level		
Worker	1		
Worker	1		
Collector	1		

```

~ Player Options:
1. Build {command: '1 <building name>'}
2. Train inhabitants {command: '2 <unit name>'}
3. Upgrade {command: '3 i<index>' / {command: '3 b<index>'}
4. Explore
5. Print Village Stats
6. Quit
7. Attack last explored

3 i1

```

Fig. 2. An example of the usage of the Upgrade command on the second worker unit in the 'Village Inhabitants' list, input in green

2.5 Printing the state of the Player village

Command 5 prints out the village state, the list as seen in the beginning of the game, this lists the current state of resources, buildings, and inhabitants of the Village taken at the moment the command is called. Below is what gets printed when '5' is typed as a command:

```
+-----{ Home Village }-----+
| Resource Type | Max | Amount |
+-----+-----+-----+
| Wood | 5000 | 0 |
| Iron | 2500 | 0 |
| Gold | 1000 | 0 |
+-----{ Village Buildings }-----+
| Name | Level | Health |
+-----+-----+-----+
| CasaDeNarino | 1 | 100 |
| LumberMine | 1 | 100 |
+----{ Village Inhabitants }-----+
| Name | Level |
+-----+-----+
| Worker | 1 |
| Worker | 1 |
| Collector | 1 |
+-----+-----+
```

DOCUMENT END