Pirkanmaan Valloitus

External Documentation for Programming 3 Course Project

Hermanni Rytkölä & Joona Perasto

Faculty of Information Technology and Communication Sciences
Tampere University
Finland
1.12.2019

Contents

1 Game overview

Pirkanmaan Valloitus is a turn-based strategy game where you must expand your empire and build the Victory Monument building to win the game. The game can be played alone but multiplayer matches of up to 4 players can also be played.

1.1 Resources

The game has five resources: gold, food, wood, stone and ore. Each player starts in a random area on the map with 200 gold, 200 food, 200 wood and 100 stone. The resources can be spent on buildings or workers to increase your resource production.

Gold is used to build nearly every building and worker. Gold is also spent as an upkeep cost for most buildings and workers. A player with a Marketplace building can also use their gold to buy other resources.

Food is used to build some buildings and train workers. It is the main limiting factor of expanding your empire; colonies require some food to build and consume a large amount of food per turn. generated by Farms and most tiles.

Wood is used in the construction of most buildings. It can be acquired by building lumber camps in forested tiles. Wood is also spent by the Factory building to generate a large amount of gold per turn.

Stone is a building material used in mid-late game buildings. Stone can be acquired by building any type of mine. Maintaining a steady supply of stone from the start of the game is often a good idea.

Ore is a late-game material used to build advanced buildings and workers. It can be gained by mining or bought for a premium at the Marketplace building.

1.2 Rules and instructions

The main goal of the game is simple: you must expand your empire until you are able to build the Victory Monument. Up to 4 players can play on the same map, competing against each other for limited resources. The first player to build the Victory Monument building wins. Players can only build buildings within their claimed tiles. At the start of the game, every tile in a 2 tile radius around every player's city will be claimed to them. Additional tiles can be claimed by building the Colony building.

The Colony building is moderately expensive and consumes a large amount of food per turn when built. To continue expanding your empire, you must be able to feed your populace. This can be accomplished with the Farm building, which can be built on grassland.

Citizens can be placed on tiles to increase their resource production. Basic citizens can be trained at City from the very start of the game. More efficient Educated Citizens can be trained by building the expensive University building.

The game has 8 different types of tiles. Each tile generates a different amount of resources and only certain buildings may be built on each tile. There are a total of 14 buildings, 3 of which are direct upgrades to other buildings. All tiles support one building and one worker. Buildings can be demolished to replace the current building with another.

2 Features

The game implements many additional features that are not necessarily listed in the project grading description. This section will go through the marketplace system, 2.5D tile graphics, world generation and screen-game coordinate mapping.

2.1 Marketplace system

By building a marketplace within their empire, players can sell their resources for gold or buy resources in return for gold. Trading resources will effect the price of the traded resource. The marketplace prices are global and shared for every player.

2.2 2.5D tile system

2.2.1 Graphics

The game assets were 3D modeled in Blender and rendered into a texture atlas, from which the tiles could be drawn simply by mapping game object types to their respective X and Y coordinates in the texture atlas. The user is able to pan and zoom the game view, which allows the user to focus on their own empire or have an overview of the entire map.

The game also features a tile highlighting system, where all the claimed tiles by current player are highlighted slightly so the user knows where they can

place their buildings. The highlighting is updated in realtime when claimed tiles change, which occurs after building a new Colony.

2.2.2 Natural world generation based on random noise

The game world is generated using randomized noise using a seed given by the player. Several random values, such as height, forest level and stone level are assigned to each tile. These values are then averaged and compared to several preset constants in order to create a landscape that looks more natural than pure random noise. Because of the nature of our world generator, there was no reasonable way to inherit from the premade WorldGenerator singleton.

2.2.3 Screen-game coordinate mapping

To make this tile system possible, translation from screen coordinates to game coordinates and vice versa was required for the system to be playable. For every Y coordinate, the tile is moved down by half of the sprite's height of its top surface portion of the sprite. Tile height wasn't taken into account, but it works very well without it.

2.3 Tiles and buildings

The game has 8 different types of tiles. Each tile generates a different amount of resources and only certain buildings may be built on each tile. There are a total of 14 buildings, 3 of which are direct upgrades to other buildings. Only one building may be present on a tile.

2.4 Workers

There are two types of workers in the game: basic Citizens and Educated Citizens. Basic Citizens can be trained from the City building, which is spawned at the start of the game for every player. Educated Citizens are far more efficient than basic Citizens in terms of resource production, but they can only be trained from the expensive University building.

2.5 Sound effects

The game uses the QSound library to play context dependent sounds. There are four sounds implemented: a sound is played when ending the turn, selecting a tile, pressing a button on the UI and when trading resources on the market.

3 Program structure

The main gameplay logic is found within the classes `MapWindow` and `GameEventHandler`. Nearly all gameplay actions involve some interplay between the two. `MapWindow` utilizes the `MapGenerator` class to initialize the game area. All gameplay related objects are stored and accessed through the `ObjectManager` class. Player resources and the in-game marketplace are managed by the `GameEventHandler` class.

The class `GameView` is the UI-component for viewing the game board, which handles all graphics implementation, such as drawing `Sprite`-classes by adding them in the `GameScene` class. The `GameScene` class inherits `QGraphicsView` and keeps track of `Sprite`-objects by mapping them to coordinates and gameobjects. The `ElevatedTile` class, which inherits `TileBase`, is used to help drawing `Sprite`-classes at different height offsets.

4 Division of labour

At the start of the project, work was roughly divided between graphics implementation and gameplay mechanics. **Joona** worked primarily on graphics implementation, GUI design, 3D modelling tiles and buildings. **Hermanni** worked primarily on gameplay mechanics, tiles, buildings, workers, and UI button implementation.

	Hermanni	Joona
Game/Game/core	Everything	MapWindow GameEventHandler MapGenerator
Game/Game/tiles	Everything	ElevatedTileBase
Game/Game/graphics		Everything
Game/Game/workers	Everything	
Game/Game/buildings	Everything	
UnitTests	ObjectManager GameEventHandler	Debugging
Other	SetupDialog EndDialog Sound effects	General optimization

5 Grading

The project fulfills all minimum, intermediate and top-grade requirements. It has unit tests implemented for ObjectManager and GameEventHandler. It also has multiple different unique Tile-classes, multiple unique Building-classes and two unique Worker-classes. The project boasts multiple features which are discussed below. Extra work was also put into the project, which will count towards bonus points. This will also be discussed below.

5.1 Additional features

The game has a marketplace system, as detailed in section 2.1. The game is presented in beautiful 2.5d graphics, as detailed in section 2.2. The game has basic sound implementation for numerous features, detailed in section 2.5. Also, the game has natural noise-based world generation, which is detailed in section 2.2.2.

5.2 Bonus points for extra work

5.2.1 Graphical User Interface

The game has a very intuitive user interface. There's a side menu that contextually changes depending on what tile is selected, which prevents possible user errors. The UI gives immediate feedback on the users' actions through audio and visual feedback. There's a status bar at the top, which tells the user how many resources they have and the current status of the game.

5.2.2 Graphics

The game has great handcrafted graphics modelled in 3D and rendered to a texture atlas. Extra work was also put into translating screen coordinates to game coordinates and vice versa. Tile highlighting also took quite a bit of work.

5.2.3 Additional tiles and buildings

The game has a large amount of variation in tiles, buildings and workers, creating a visually interesting environment and allowing tactical gameplay.