Andrzej Betiuk  
Lords and castles  
Software Design Document

Contents

[1. INTRODUCTION 3](#_Toc148447582)

[1.1 Purpose 3](#_Toc148447583)

[1.2 Scope 3](#_Toc148447584)

[1.3 Overview 3](#_Toc148447585)

[1.4 Reference Material 3](#_Toc148447586)

[1.5 Definitions and Acronyms 3](#_Toc148447587)

[2. SYSTEM ARCHITECTURE 4](#_Toc148447588)

[2.1 Architectural Design 4](#_Toc148447589)

[2.2 Decomposition Description 6](#_Toc148447590)

[2.3 Design Rationale 6](#_Toc148447591)

[3. DATA DESIGN 7](#_Toc148447592)

[3.1 Data Description 7](#_Toc148447593)

[3.2 Data Dictionary 7](#_Toc148447594)

[4. HUMAN INTERFACE DESIGN 7](#_Toc148447595)

[4.1 Overview of User Interface 7](#_Toc148447596)

[5. MoSCow list 7](#_Toc148447597)

[6. APPENDICES 10](#_Toc148447598)

# 1. INTRODUCTION

## 1.1 Purpose

The purpose of this document is to visualize the game and its features. The intended audience of this game is rts[[1]](#footnote-1) players who also like city sim [[2]](#footnote-2)games. The game will be build upon sdl2 and libraries based on it (sdl2\_image for example).

The game-loop will consist of 2 parts.

Building loop

* building resource buildings.
* resource buildings collecting resources.
* Building specialized building to get more specific resources.
* Building defences or other military buildings.
* Repairing or rebuilding buildings if they are damaged.

Combat loop

* Recruiting soldiers.
* Defend or attack with the soldiers.
* Try to kill the enemy king with your soldiers.

This project will also be programmed hierarchically object-oriented programming style where it is comprised of objects that are children or parents of each other.

## 1.2 Scope

See [6. MoSCow list](#_6._MoSCow_list)

## 1.3 Overview

Provide a high-level overview of this document and elucidate its organizational structure.

## 1.4 Reference Material

* Link to [SDL2](https://www.libsdl.org/)
* Link to [hierarchical pathfinding](https://web.archive.org/web/20190411040123/http:/aigamedev.com/open/article/clearance-based-pathfinding/)

## 1.5 Definitions and Acronyms

|  |  |
| --- | --- |
| word | Meaning |
| rts | Realtime strategy – a strategy game that is in real world seconds. |
| npc | non playable character – a character that is unable to be controlled directly. |
| pathfinding | A way for npcs to find their way around the map. |
| City sim | A city-building game, or town-building game, is a genre of simulation video game where players act as the overall planner and leader of a city or town, looking down on it from above, and being responsible for its growth and management strategy. |
| hierarchical pathfinding | hierarchical pathfinding is a type of pathfinding that splits the map into sections and calculates common paths so the npc uses less memory for pathfinding. [link](https://web.archive.org/web/20190411040123/http:/aigamedev.com/open/article/clearance-based-pathfinding/) |
|  |  |

# 2. SYSTEM ARCHITECTURE

## 2.1 Architectural Design

Flowchart of the player gameplay loop

A diagram of a company

Description automatically generated

Flowchart of workers (farmers, miners, etc)

A diagram of a process flow

Description automatically generated

Soldier flowchart

A diagram of a soldier

Description automatically generated

Global relational view

A diagram of a company's structure

Description automatically generated

## 2.2 Decomposition Description

Provide a decomposition of subsystems within the architectural design. Supplement this with relevant text as needed. Options include providing either a functional description or an object-oriented description. For a functional description, present top-level data flow diagrams (DFD) and structural decomposition diagrams. Conversely, for an object-oriented description, include subsystem models, object diagrams, generalization hierarchy diagrams (if applicable), aggregation hierarchy diagrams (if applicable), interface specifications, and sequence diagrams.

## 2.3 Design Rationale

Delve into the rationale behind selecting the architecture detailed in section 3.1. Discuss critical considerations and trade-offs that factored into the decision-making process. It is permissible to mention other architectures that were contemplated, provided an explanation accompanies why they were not chosen.

# 3. DATA DESIGN

## 3.1 Data Description

Elaborate on how the information domain of the system is translated into data structures. Clarify how major data entities or system entities are stored, processed, and organized. List any relevant databases or data storage components.

## 3.2 Data Dictionary

Alphabetically catalog system entities or major data, complete with types and comprehensive descriptions. If a functional description was provided in Section 3.2, enumerate all functions and function parameters. In the case of an object-oriented description, list objects, along with their respective attributes, methods, and method parameters.

# 4. HUMAN INTERFACE DESIGN

## 4.1 Overview of User Interface

This part explains the general UI of the game.

### 4.1.1 Player interface in the menu

### 4.1.2 Player interface in gameplay scene

* The Left Panel with Icons: This panel serves as the category selector for the buildings to be displayed in the middle panel. For instance, if the player intends to construct a stone quarry, they would first choose the icon corresponding to the industry category of buildings.
* The Middle Panel: The middle panel is designed for selecting the specific building the player wishes to place on the map. For example, if the player wants to build a stone quarry, they would select the stone quarry icon from this panel, allowing them to place it on the map.
* The 3rd and Last Panel: This panel is reserved for displaying essential statistics such as the player's current funds and population.

A screenshot of a video game

Description automatically generated

A group of buildings with text

Description automatically generated

player interface for viewing the settings of the game.

# 5. MoSCow list

|  |
| --- |
| Must |
| Name: must have a collection system. |
| Requirement: there must be a way to collect resources. |
| Exception: when there are no resources to collect, or the player isn’t in the gameplay scene. |
| Definition of done: when the player can collect resources. |

|  |
| --- |
| Must |
| Name: pathfinding. |
| Requirement: the npcs should be able to find their way across the map. |
| Exception: when out of gameplay scene or when there isn’t need for it, for example there is npc. |
| Definition of done: when a npc can go from point A to B whilst avoiding obstacles. |

|  |
| --- |
| Must |
| Name: stockpile. |
| Requirement: there should be a place to store the collected resource. |
| Exception: when there isn’t a stockpile, or the player is outside the gameplay scene. |
| Definition of done: when there is a stockpile where npcs can “store” the resources gathered. |

|  |
| --- |
| Must |
| Name: player UI |
| Requirement: the players must have a interface to play the game. |
| Exception: none |
| Definition of done: when the player has a UI to play the game. |

|  |
| --- |
| Should |
| Name: combat. |
| Requirement: there is combat in the game in the form of soldiers that fight as units in a traditional rts game. |
| Exception: when the player has won, has no soldiers to command or the player is outside the gameplay scene. |
| Definition of done: when the soldiers are able to engage in combat. |

|  |
| --- |
| Should |
| Name: basic settings. |
| Requirement: the player should be able to change things as input and resolution. |
| Exception: when the changes could cause problems while playing the game. |
| Definition of done: the player is able to change settings of his game. |

|  |
| --- |
| Should |
| Name: basic save |
| Requirement: the player should be able to save basic things as settings or customization options. |
| Exception: when saving may create problems for the game. |
| Definition of done: when the player can save basic things such as settings. |

|  |
| --- |
| Should |
| Name: tutorial |
| Requirement: there should be an option to play the tutorial to explain the game. |
| Exception: when in the gameplay scene. |
| Definition of done: when the player can play the tutorial in other to learn how the game works. |

|  |
| --- |
| Could |
| Name: multiple soldier types. |
| Requirement: the games has multiple soldier types that have different stats and attributes. |
| Exception: when the player is outside of the gameplay scene. |
| Definition of done: when there is more than 1 soldier type. |

|  |
| --- |
| Could |
| Name: hierarchical pathfinding. |
| Requirement: the game uses hierarchical pathfinding for its pathfinding, adding optimization. |
| Exception: when the player is outside of the gameplay scene. |
| Definition of done: when the npcs use hierarchical pathfinding. |

|  |
| --- |
| Wont |
| Name: make it multiplayer. |
| Requirement: there won’t be any multiplayer function or option. |

|  |
| --- |
| Wont |
| Name: campaign. |
| Requirement: the game won’t have a campaign. |

# 6. APPENDICES

(Optional) Appendices may be included, either directly or by reference, to provide supplementary details conducive to the comprehension of the Software Design Document.

It is advisable to note that this document template serves as a foundation for delineating the architecture, design, and functioning of the software. It is crucial to tailor the content to the specific project's needs while adhering to best practices in software design and documentation.

1. Rts stands for real-time strategy game [↑](#footnote-ref-1)
2. A city-building game, or town-building game, is a genre of simulation video game where players act as the overall planner and leader of a city or town, looking down on it from above, and being responsible for its growth and management strategy. [↑](#footnote-ref-2)