Bachelor Thesis Review

Faculty of Mathematics and Physics, Charles University

Thesis author	Šimon Tichý	
Thesis title	The Last Clan - RTS game in Unity	
Year submitted	2023	
Study program	Computer Science	
Specialization	Software and Data Engineering	
Review author	Jan Pacovský	Advisor
Department	Department of Distributed and Dependable Systems	

Overall

	good	OK	poor	insufficient
Assignment difficulty		X		
Assignment fulfilled		X		
Total size *. . . text and code, overall workload*		X		

The thesis focuses on developing a real-time strategy game using Unity's modern DOTS (Data-Oriented Technology Stack) ecosystem. This includes C# Job System that provides easy parallel task execution; and Entity Component System (ECS) that decouples the representation of game entities from their behavior, allowing cache-friendly data organization in memory.

The resulting game is a single-player game with up to three players in a player's team (clan) and up to three in the adversary one, all governed by the same simple AI. The game features multiple element types: three resources, two buildings, two land units, and a ship unit.

The objective is to be part of a clan with the last living land units. The game world consists of islands and offers limited resources.

A noteworthy aspect of the game is its unique mechanism where land units regularly consume food, and as the game progresses, fish depletion near docks forces ships to travel increasingly longer distances, which leads to a gradual decrease in food income speed. Players' strategies must therefore balance army sizes, food income speed, and stored resources to use them as efficiently as possible. To spread to a different island, a ship needs to construct a warehouse there which enables the production of additional land units.

Part of the solution is a map editor where players can easily create their own maps in addition to the default one.

The size of the non-generated source code, including comments, exceeds 350KB and spans over 8000 lines.

In the following texts, I will repeat some of the remarks that the student has already received but was unable to take into account, especially for the thesis text in the last three weeks when he was still finishing the program. As a result, both the text and program are in need of some polishing. Hoverwer, I think the thesis proves his capability of independent examination of a problem and applying the obtained knowledge.

Thesis Text

	good	OK	poor	insufficient
Form *. . . language, typography, references*	X			
Structure *. . . context, goals, analysis, design, evaluation, level of detail*			X	
Problem analysis			X	

Developer documentation		X		
User Documentation		X		

The text is well-written in proper English and is free from typographical errors. The structure of the thesis, according to the headings, is good. However, the content would benefit from even clearer distinctions. For example, the Programmer documentation contains parts that should be in the Analysis; and the Analysis covers only a part of the challenges that must have been addressed. For example description of the computer player AI is missing. Another issue is that it often does not analyze the other obvious options, and sometimes it misses a justification. For example, in the paragraph about Land pathfinding, it is stated that the A* must consider forces between units, which is not true since navigation comprises two distinct components: planning and execution. Nevertheless, the author mixes them together.

The User documentation provides a nice overview, including how to run the game from the provided executable but is missing information on how to build the game from source code. Developer documentation fills in the details on which packages need to be present.

Overall the text provides an introduction to the development of an RTS game that uses DOTS with ECS.

Thesis Code

	good	OK	poor	insufficient
Design *… architecture, algorithms, data structures, used technologies*		X		
Implementation *… naming conventions, formatting, comments, testing*		X		
Stability			X	

The architectural design is sensible, with the implementation incorporating classical game algorithms. The code is organized into distinct classes, following standard C# conventions, and includes comments and method documentation. Unit tests are not present.

The game crashed in roughly 20% of the games in a sample of approximately 30 games. Additionally, there are a couple of bugs that affect the gameplay; the ship units often get stuck when the A* plans the path too close to the border with the land (which is the typical A* behavior). A similar problem occurs rarely, also with the land units that are governed by the flow field in narrow passages. The units gathering resources sometimes fail to continue the gathering when the resource on a tile is depleted, but other nearby tiles still contain the resource. This occurs when the unit is on the way to the warehouse and is far enough from the now-depleted resource tile that it was originally gathering. Sending a ship to fishing is often challenging because the fish area to be clicked is too small in the default resolution.

Overall grade Good
Award level thesis No

Date 28.8.2023 Signature