A game/tool for running/playing the TTRPG “Lancer”

Gwion Hughes[gwh18]

Supervisor: Bernie Tiddemen[bpt]
Degree scheme: G401 Computer Science with a year in industry
Module: CS39440
Document Creation Date: 05//02/2025
Version: 1.0
State: Complete

Project Description

My proposal for this project is to create a turn-based tactics game based on the TTRPG(Tabletop Roleplaying Game) Lancer produced by Massif Press[1]. This category of game revolves around opposing teams composed of(usually) 3-10 agents attempting to achieve an objective on a tile based map. This genre of game takes many influences from the TTRPG genre, specifically ones in which players managing a character have to cooperatively overcome a challenge set by a game master. I'll be producing this game in the Unity engine[2].

I'd like to be able to use this game as a tool to simulate combat scenes for a group of players playing Lancer. Therefore, the gameplay mechanics will be entirely based on the rules in Lancer. Other tools used by the Lancer community include COMP/CON[3], an app for tracking the condition of the player character and the mech, however it's main shortcoming is that it cannot be used to simulate combat scenes.

I will also make the project simpler by limiting the scope to simulate "low level" combat scenes. Player and enemy agents will not have accumulated a large amount of complexity through weapons, perks, and traits, making it significantly easier to model. I'll be making a tactics based game where the players control one team, and an opposing team(s) is controlled by a game master(who is also a player), to streamline playing in person games of Lancer.

Proposed Tasks

This is my first time trying game development. I'll be spending quite a bit of time familiarising myself with game design[4] and the engine[5].

Lancer Decomposition

Lancer's Tactical Turn Based combat rules need to be decomposed and included as requirements. Lancer is a game with a lot of nuanced game mechanics and a player agent is made powerful by one or more unique abilities and weapons that confer situational advantages.

Agents and their Map

The Map will be a hexagonal tile map[6]. There will be tiles that count as elevated terrain, difficult terrain, and impassable obstacles. Combat scenes will also contain an objective which may be to eliminate the enemy team or control a position on the map. I'll model how customizable player agents can be by importing a variety of weapons and components from the Lancer core rule book to allow making unique agents.

Each agent in the combat scene will need to be made aware of the moves available to it. It will also need to track its conditions such as its health. I plan to track all this information in the entity class. There's a cover and line of sight based system in Lancer. I will investigate methods of drawing a straight line as a path of tiles between two agents to determine if an agent is within another line of sight or in cover.

Frontend

I'll be using 2D assets for the sprites. This suits this genre which tends to sacrifice graphics to make their game more interpretable. I will focus on this aspect of the work once the work on the agents and the map have been finalised. I will use stock and freely available assets for sound and sprites.

A cornerstone of this genre is ensuring tactical information such as the conditions of a player's agents conveniently placed and interpretable. I'll take inspiration from other turn based tactics games. For example, I would like to take inspiration from Baldur's Gate 3's bottom bar displaying possible actions an agent can take, adopting its method of categorizing actions. I'll be taking inspiration from Advanced Wars and the Fire emblem series for communicating terrain types. I believe these games demonstrate that clear and vibrant coloured tiles are one of the most useful tools for displaying tactical map information in an intuitive non-intrusive way.

Tools useful for a Gamemaster

Being able to save the state of the game, an encounter and map editor, and customizable agent templates, are all features that are useful for running a digital TTRPG. From my experience as a game master, I would find generative AI integration and an auto-battle function useful tools to manage the gamemaster controlled agents. It could be possible to locally host a generative AI model using Ollama[7] and then integrate it through a chat screen the game master controls. Depending on the integration, it could give personalities to agents outside the player's control.

Project Deliverables

System Requirements Doc - A document of tests to ensure the rules Lancer Combat encounters are being upheld. Should be done by week 17th of February.
Combat Map - A map with all of the terrain rules included in Lancer. Should be accompanied by a demonstration of combat between two teams of very basic agents. The deadline is the mid project demonstration.
Customizable agents - Implement a system for customising agents, giving them different parts for different abilities and stats. Aim to complete by mid project demonstration.
System Requirement Testing - Done following the previous deliverable. By this point, the combat mechanics of the game should be considered complete and unlikely to change. This is testing the Lancer core rules have been accurately translated.
GUI - A user interface that should be considered "complete" by this stage of the project. I.e, not the development interface used for testing, but one that is expected to stay as is by the time the project is completed.
Gamemaster Tools - Integration with a ML model that could be used by the gamemaster to "Roleplay" the enemies. The ability to save the state of an encounter for future play. Includes Map and agent template editor.
Auto-Battler - A possible future goal. This is the option to have a team controlled by a computer adversary rather than the gamemaster.

Bibliography

[1]T. Parkinson-Morgan and M. Lopez, *Lancer TTRPG Core Rulebook*. Dark Horse Comics, 2024.

[2]Unity Technologies, “Unity,” *Unity*, 2019. https://unity.com/

[3]“COMP/CON,” *Compcon.app*, 2024. https://compcon.app/#/

[4]freeCodeCamp.org, “From Zero to Game Designer: how to start building video games even if you don’t have any experience,” *freeCodeCamp.org*, Jun. 10, 2018. https://www.freecodecamp.org/news/from-zero-to-game-designer-how-to-start-building-video-games-even-if-you-dont-have-any-experience-5e2f9f45f4bb/

[5]unity, “Unity Learn,” *Unity Learn*, 2020. https://learn.unity.com/

[6]Unity Technologies, “Unity - Manual: Hexagonal Tilemaps,” *Unity3d.com*, 2018. https://docs.unity3d.com/6000.0/Documentation/Manual/tilemaps/work-with-tilemaps/hexagonal-tilemaps.html (accessed Feb. 06, 2025).

[7]“ollama/ollama,” *GitHub*, Feb. 29, 2024. https://github.com/ollama/ollama