COMP504

Object-Oriented Programming and Design

Final Project User Manual

Geocaching Game

Peisheng Guo (pg23)

Yiqing Lu (yl128)

Abstract

This document is the user manual of 'Geocaching': a two-team, multi-player game. The documentation starts by giving an introduction to the project. Then, there will be game process description on the game.

The main user manual is in the second section. System architecture is in the last section.

1. Introduction

In this project the multi-player game we are required to design need to have two important features. One is that players in the game are supposed to make use of the chat-room platform in order to make advantage in the game. The second is that the game must involve a geographical map of the world and should be highly based on it.

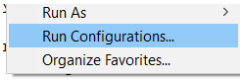
According to the requirement, our team designed a game called ‘Geocaching’.
The rest of this documentation is arranged as follows. Then, there will be game process description on the game. The main user manual could be seen in the second section. Finally, there is some thinking and analysis regarding the game design.

2. Game Process Description

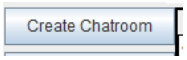
In this section, the full description of the game process is given. The focus would be operation on server side, rather than client side, since UI on client might differ.

2.1 How to start the program

Step 1 - Connect server & Create lobby:
On server side, open the terminal. Before opening the terminal, make sure run configuration is correctly set up.



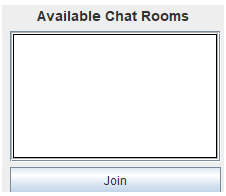
The configuration file is in “yl128_pg23_test.launch”, the user could export the configuration and run the program.
On server side, create a chat room as lobby.



On client side, open the terminal, connect to the server by entering server’s IP and press enter on keyboard.

Step 2 - Join Lobby :
After success connection, server could invite clients into game lobby. It could be done by inputting client’s IP address and press the button ‘invite’.

Another function of joining manually is also given. For the reason that lobby is essentially a chatroom, client can also join in lobby by choosing ‘chatroom’ in server’s list.

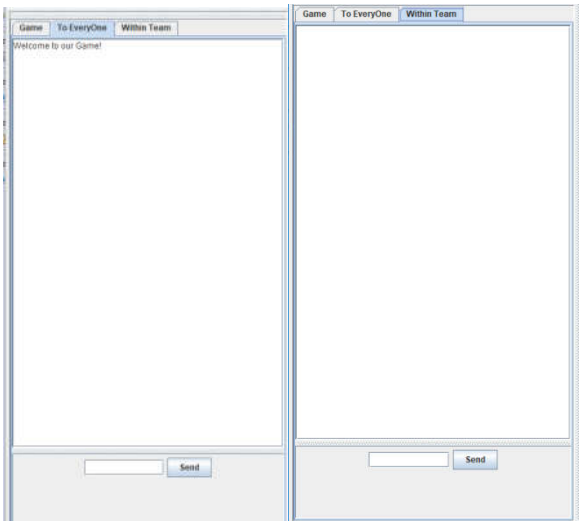


Step 3 - Start Game:
After joining in the lobby, players could exchange message in the lobby, as they wish. Then, server press ‘start’ in JFrame and game should pop up on both server’s and clients’ screen.

2.2 How to play

Step 1 - Join Team:
Join in team A or team B as wishes, suggestion would be join in the one with less people, due to equivalence.

Step 2 - Exchange Message:
Players could send message to everyone or within team by clicking the tag, enter message and press send. Message to everyone can actually be sent before joining team.



Step 3 - Place Treasure:
Each team could place treasure in the whatever spot on the map by **right click**. Each team has 10 treasure, which means they could put 10 treasure on the map at maximum. Press ‘finish’ button in the JFrame. The process of placing treasure ends once placing time (2 minutes) expires or pressing ‘finish’ button.

Step 4 - Looking for Treasure
Players would click on the map trying to find the competitor team's treasure. 'Find a treasure' is defined that a player **right click** on a point of within 10 degree of longitude and 10 degree of latitude away from a treasure of opposite side.

Step 5 - End game
If players in one team together find all the treasure settled by the competitor team, they would win. After one team win, there will be a window pop up at both side, stating the result (i.e. 'You lose' or 'You Win').

3. Game Design

3.1 Player Allocation

'Geocaching' is a game across two teams. Each team is supposed to have at least one player(s). There are 4 notes regarding to player allocation:
Note 1: The upper bound of players has not been tested yet. But it is supposed to support dozens of players. However, due to the limitation of network speed, there might be lag if there are too many players.
Note 2: For lower bound, although it is a game that encourages communication and cooperation among team members, the game supports single player in a team.
Note 3: The number of players in two teams could be different. But equal number of players is fairer, since the more players in the game, the more cooperation they might have and hence more chance to find all the treasure.
Note 4: Players could join in the game after game owner start the game and treasure hunting process. Team size is fixed after this process start.

3.2 Game Objective

The objective of the game is to hide the containers (treasure) of one's own team and seek the competitor team's containers as fast as possible (or as many as possible). The team that finds all the competitor team's containers faster will win the game. Suppose

neither team find all the competitor team's container, the team that find more treasure would win.

3.3 Collaborative Play

Users need collaborative play to have optimal strategy, both to cache their own treasure and find the competitor's. Members among a group could discuss a strategy to make the treasure elusive. In addition, while looking for the treasure, every member would need to share their information, such as spots one player has visited to avoid unnecessary revisit.

4. System Architecture

We use MVC architecture as infrastructure of our game. In addition, we use visitor pattern, command pattern to optimize the system.

For MVC, we divide the game into three sections: model, view and controller. The game view is the GUI of the game, where users could see and control on the interface. The model is to provide the functions to the interface. The controller is to control the model and the view and make sure they would work together. Specifically, through adapter, it tell the model when to use which functions and the view need to be told what is the result from the model functions. Adapter could be considered to be

For visitor pattern and command pattern, we set different data types for different data type commands/algorithms. These commands let the chat room know how and when to deal with these data types.
In addition, extended visitor pattern is used to set commands for those unknown data types. Game commands, since they are on server side, are unknown to clients. The commands would be sent in this way.