Assignment

BATTLESHIP

Contents

1   Outcomes

After finishing this assignment, students can proficiently use:

- MARS MIPS simulator.

- Arithmetic & data transfer instructions.

- Conditional branch and unconditional jump instructions.

- Procedures.

2   Introduction

Battleship is a strategic board game between 2 players. It focuses on the process of planning while testing the problem solving skill of both players. The game rules are:

1. Each player sets up a fleet of battleships on their map (a 10x10 grid). A fleet must contain a predefined set of battleships with different sizes. For example, a fleet of ships can consist of 5 2x1 ships, 3 3x1 ships, 1 5x1 ship.
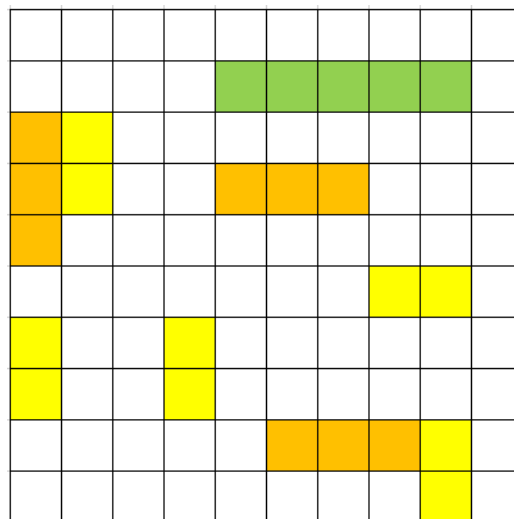

Figure 1: Example of battleships placed in a 10x10 grid

2. After the setup phase, both players will take turn targeting a box on the other player's map blindly. If it hits a part of a ship, an indicator

will be shown to mark that box. A ship is completely destroyed if all of the boxes it's occupying are targeted. A player will lose if all of their ships got destroyed first.

3 Requirements

In this project, we will work with a smaller grid size which is 7x7. Design and write MIPS assembly language for implementing a text-based Battleship game for two players as follows:

- Firstly, each player will input a series of 49 numbers for their maps. The numbers will be either 1 or 0. 1 means that box is occupied by a ship, otherwise it's 0. The players will need to setup the exact amount of ships with the same size in order to complete the setup phase. If there are violations such as different amount of ships, wrong size, etc., students need to be able to handle those cases.

- After the successful setup, the program should print out the amount of battleships and their sizes to the terminal to re-inform the players.

- Afterwards, each player will take turn to choose a box from the opponent's map to attack. If the attack hits a ship, that grid is marked and that part of the ship is destroyed.

- Finally, output the winner of the game to the terminal if all of the ships of one side are destroyed.

- Note: Bonus points are provided if you can write out the moves of each player into a separate text file for ease of review.

4 Submission

Students are requested to submit the MIPS program(s)/source code (.asm files) and the Assignment report to BK E-learning system (BKEL) no later than the last lab session of your group. Assignment must be done individually. Students have to demonstrate program(s) on MARS MIPS during the last lab session. Students who do not show up during the demonstration time will get 0 for the assignment. The report should not contain code. Instead, students should present the algorithms as well as the idea in your implementation.

5 Plagiarism

Similarity less than 30% in MIPS code is allowed. In other words, you will get 0 for assignment if your answers are similar to another student's more than 30%.

6 Rubric for evaluation

6.1 Friendly interface - 2 points

Students can design and implement an amicable user interface so that players can play easily without any confusion (2 points).

Students can design and implement a friendly user interface; however, players face some difficulties when playing the game (1.5 points).

Students can design and implement a user interface, but it is not friendly, or players need to do several steps for one move (1 point).

Student can design and implement a user interface, but it fails to allow playing (0.5 points).

6.2 Application implementation - 6 points

Students can implement an excellent application without any errors found (5.0 - 6 points).

Students can implement a good application with some minor errors, but players do not need to restart the application to continue (4.0 - 5.0 points).

Students can implement the application with some errors that prevent players from playing the game (2.0 - 4.0 point).

Students cannot implement the application so that players can play/run (0 - 2.0 points).

6.3 Report - 2 points

Students write such an excellent report that others can understand without any difficulty (2 points).

Students write a good report but it's quite simple or lack of some information to clarify the implementation (1.0 - 1.5 points).

Students write a report with a lot of code embedded without any explanation (0.5 - 1.0 points).

Students with no report submitted (0 point).

References