CS 319 - Object-Oriented Software Engineering

System Design Report

Rush Hour

Gurup Şurup

Muhammet Said Demir

Ata Coşkun

Zeynep Nur Öztürk

Asuman Aydın

Tarık Emin Kaplan

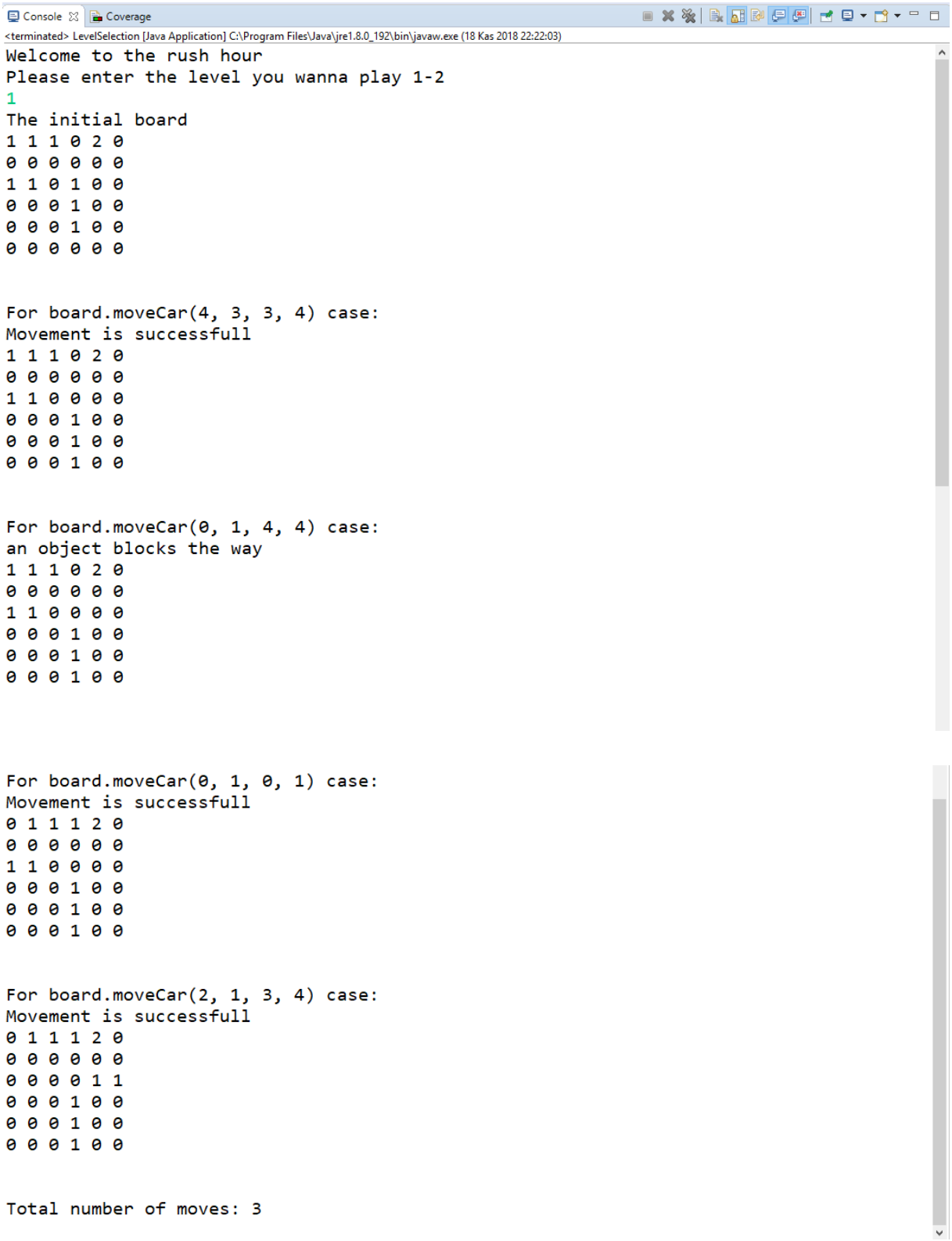
Supervisor: Eray Tüzün

TA: **Muhammed Çavuşoğlu**

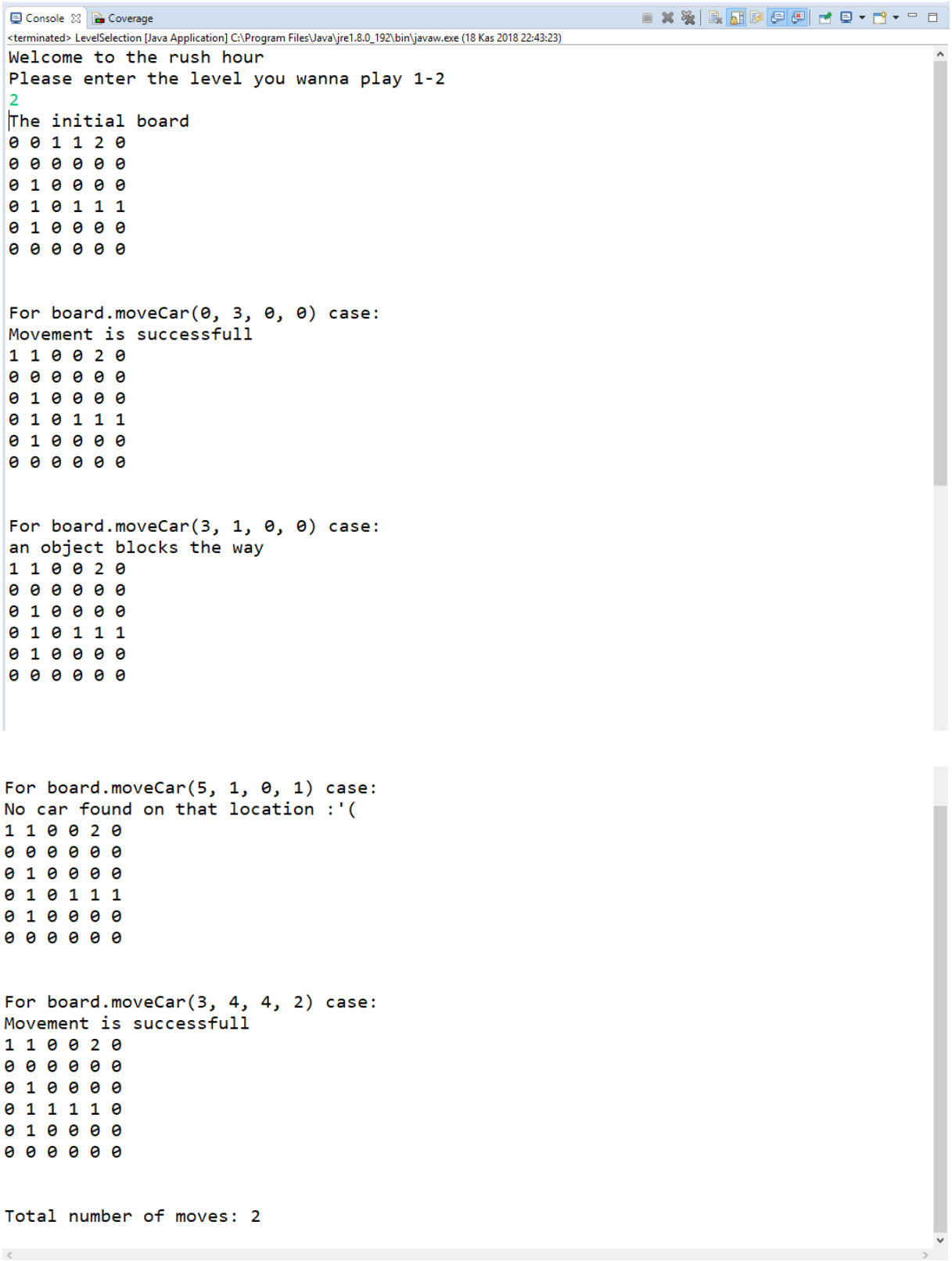
1. Introduction

Basic foundations of the project had been implemented, which are the key elements. LevelSelection, GameEngine, Board, Car, Obstacle classes had been implemented (not fully). At the moment, we're able to design a level, put cars&obstacles to the board of the level and move the cars. Also, when the user wants to move a vertical car both in vertical and horizontal directions, it only moves vertically and ignores the horizontal part. Same thing goes for horizontal case too. We haven't designed a fully developed interaction with the user tho, movements are hard coded movements. And we haven't fully developed a user interface yet, player cannot see the GUI of the objects. But, on the board the free spaces are represented with 0, places with cars are represented with 1 and for the obstacles, it is 2. We'll handle more advanced interactions (Mouse listeners) and better designed GUI until our demo day (22) hopefully. Our code is uploaded to GitHub so you can look into it for further knowledge. Without further ado, here are some test pictures.

For the first map:



For the second map:



2. Design Changes

There is a major design change in our project. Which is, we've swapped to java instead of C#. We've found that on almost every tutorial, the features of unity are being used, which we're strictly banned, and we were all familiar with java instead of C#. So, we've decided to change the portal. Now we're writing the code on Java with using Eclipse IDE. We were also familiar with Eclipse and it increases the readability, writability and helps us to write code faster with auto completing some keywords.

Furthermore, we've made a field trip to play the actual board game at Bilkent Station and we had a chance to read the rule book. Some features of Card class had been changed. Like there were different types of cards. And some features of Board class had been changed, only the left and right parts could move only. Also, we've played the single mode of the same game too, which gave us ideas like the map size or some levels. We've took some pictures of the levels and solutions of them. Also, the levels of multiplayer too, so that we'll base our game maps on the real game maps.

Some minor changes also had been made, isValid method on the board class had been removed due to its simplicity.

3. Lessons learnt

We'll to be honest, we've seen that the calculation at home did not match with the bazaar. In our world, using C# and Unity seemed clever and we could learn things about them. But when it comes to the implementation part, we've seen that we needed lots of time to learn about Unity and its aspects. So, we've head back to our old, dear, wise friend; which is Java. Also, when we were writing codes, we've seen that our design could use a little bit of improvements. Also, we've faced the cold hard truth that we've almost forgot how to Java. Another lesson is that, we've seen that writing fundamental codes is easy but as you go deeper and deeper the code gets harder and harder to implement. But that's a lesson that we'll learn more and more as the time comes.

4. User’s Guide

Well, all you need is Java and preferably Eclipse to run it. The source codes are in our GitHub account, but if you cannot reach it, you can always mail us and we can send the code in a zip file. See you at the demo!