



Bilkent University
Department of Computer Engineering

Object-Oriented Software Engineering Project

CS319 Project: River Adventure

Final Project Report

Nazlı Özge Uçan, Meder Kutbidin Uulu, Aras Heper, Hande Teke

Course Instructor: Uğur Doğrusöz

TA: İstemi Bahçeci

1. Implementation Process

At the beginning of the implementation we divided the project into four parts that every member of the group could implement his/her own part. We have three main parts of the project: User Interface, Controllers and Models. After distributing the packages to each of us, we created all classes with their methods that are needed in the game development according to the project design.

The Models were the first in our to-do list, as we needed them in all other parts of the game. They are partly implemented and tested before starting to UI part. However it was difficult to write and test the code without having UI part. So it was time to start UI part of the project. Before this point, **MetaModel** and **RealModel** of the **GameModel** was partly finished.

While designing UI part of the game, we strictly followed the design that were shown during the mockup, and it was really successful. Although we have various obstacles, boosts and coins in the game, we had only single classes for each of them. Therefore, we came to a conclusion to create new classes for each single type of the objects to draw them easily in UI, at the same time not violating Object-Oriented Programming patterns, because we used inheritance and polymorphism.

Additionally, during the development of UI, we got unrelated to code problems a lot: xml and IDE problems, without fixing those bugs, the main code did not work. Although they seem irrelevant to the project, they took quite a long time, and finally somehow we managed to fix it.

Of course, in a meantime, Controllers were also implemented, because we strictly followed MVC pattern of OOP to connect Model and UI of the game.

2. Changes in the Design

After the design phase, we started implementation. While doing the implementation phase, we had to make minor changes in some of the methods and attributes of some classes. But the major change would be the child classes of each RiverObject. For each child class of RiverObject (Obstacle, Trap, Collectible) and Boost (ExclusiveBoost, CharacterSpecifiedBoost, RareBoost) we created their child classes. For ExclusiveBoost, Deceleration, HealthPack, Invincibility, Shield and MinimisationPower extends ExclusiveBoost. For RareBoost, Bonus and RandomBox extends RareBoost. For CharacterSpecifiedBoost, BirdsOfCroc, BF and Berserk extends CharacterSpecifiedBoost.

For Collectible, Coin and CoinBag extends Collectible. For Trap, Maximise and Accelerate extends Trap. For Obstacle, Log, Stone, ChemicalHazard and NuclearBomb extends Trap. Those changes can be seen in the class diagram below. Beside the changes mentioned above, the design pattern and other issues remain the same as in the design document.

Figure 1. UML Class Diagram

3. User Guide and Installing the Software

Installation: Having both the supplied .jar file and the latest version of the JVM is sufficient. To run the game, run the .jar file with JVM.

Aim of the Game: The aim is to play as long as you can while collecting coins and trying not to hit any obstacle. The game finishes when the user has 0 health.

Menu Guidance: In the Main Menu of River Adventure game, there are following options below.

Play Game: This button in the Main Menu leads user to the game. In the game, there is a “Coins” label, which shows how many coins the player had collected. “Time” label, which shows how long the player is alive. Also, in the game there is a “Pause” button, which pauses the game and shows the menu to return either to Main Menu, to Continue the game or to Settings, however this optionality can be accessed with space button on the keyboard. The high scores of the player are saved on the memory and when the player run out of health, the best scores are shown.

Settings: In this option, the player can change the music sound level or mute completely and go back to Main Menu.

Store: In the Store, the player can purchase different characters according to the coins he/she had collected.

Help: The user guide of the game is accessed in this option. If some parts of the game is unclear to the player, he/she can read the guidance in this option.

4. Status of the Implementation

The RiverAdventure game is not fully implemented according to the project design we had before, but it is functional, main parts are done. Based on the current implementation, following functionalities or classes are not implemented completely:

- Extend the RareBoost class
- Sound settings
- Accelerate and Decelerate classes
- CharacterSpecifiedBoost classes (and its child classes)
- Cannot fully store constant data
- Cannot show highscores
- Store features are not completely finished