CS319 Object Oriented Software Engineering Project

Final Report

Iteration 1

IQ PUZZLER PRO

GROUP - 3G

Fatih Çelik

Cenk Er

Enes Yıldırım

Eren Yalçın

Burak Bayar

Ebru Kerem

1. Introduction

1.1 Progress Details

Halfway through the project, we have started to shape up our game environment on the implementation stage. Our aim on the first iteration was to develop a game which is playable but it is not fully functional. We created a basic menu and added the simple buttons which lets a user to choose an easy map and start to solve the puzzle. Although the game needs a lot more coding to achieve all of our goals, it will be easier with all of the UML diagrams and mockups that we created. It could be said that the hard part of the project is done.

1.2 Lifecycle Stages

Requirements Elicitation: Finding the requirements for making a puzzle game from scratch was not easy. Firstly, we needed to fully understand how the puzzle works so we bought it online to examine it. The puzzle had a very simple logic, so we had to come up with as many functionalities as possible. To do that, we came up to our friends and made them play to game and asked what could be done to make an interesting software version. After we identified the non-functional requirements, we created a use case diagram to show the possible activities a user can make and started to analysis stage.

Analysis: This stage was really challenging because as a group, we had to determine how the game was going to work in detail. In projects like this, every small detail should be discussed and determined which is pretty time consuming. Especially in analysis object model and creating a class diagram, there had been a lot of debate on implementation because we had to determine how the game physics will work and define

every attribute for our many objects. Fortunately, after this stage, we all had a picture for the game and this was very helpful for the next stages of the project.

System & Object Design: Firstly, we re-looked to our dynamic models and settled our design goals, agreed on which features we should focus and determined the tradeoffs. Than we further looked into our object design and defined our subsystems. After that, we created new class diagrams for our subsystem to complete the class design for our project which also included defining each of the classes and their functionalities in detail.

Implementation: This is the stage that our group is in but it should not be forgotten that testing is also ongoing throughout the implementation process. There is a lot to do on this stage but analysis and design stages created a clear path for our coding. Before our first demo, we are aiming to create a simpler version of the game that only includes the solo mode, with a user-friendly gameplay without any unnatural outputs. Main functionalities to be able to complete a puzzle such as dragging and dropping, rotating or flipping a puzzle piece should be provided.

1.3 What is Left

The project has many more functionalities to add. Furthermore, the graphical user interface needs a lot more work to be more pleasant in the eyes. Our first job on implementation is to provide the functionalities that we desire. This includes major functionalities such as options and online mode which includes making your own puzzles and competing with other players, and it also includes minor functionalities such as hints, timers, etc. After we make our system fully functional, we will focus on the GUI and create a better-looking interface with newly designed backgrounds, buttons and puzzle figures. Finally, we will add things such as visual effects sound effects and narrators to make our game as complete as possible.

2.Design Changes

- In the beginning, we decided to draw maps and puzzle pieces as a sprite class.However, after some research we found that drawing in the C# can be made only in the form class. We are still trying to find a way to make sprite a class but if we cannot find a way. We will write drawing methods in the form and try alternative methods of designing graphics.
- Moreover, we realize that we cannot give 3 hints for every level like we said in the design report because some levels are trivial to give 3 hints. Some of the maps has only 3 or 4 pieces which are not placed. Thus, we decided to give only one or two even zero hints for some levels.
- Also, we said that we will add hints as an image in the design report. However, we thought that hints are the puzzle pieces and we do not need another image or different construction for them. Therefore, we agreed to use puzzle piece classes for creating hint.

3.Lesson Learnt

For all of us, there has never been a project that required as much preparation, analysis, modelling, implementation and testing before. This showed us software engineering is not a linear process that only includes coding but it requires many more stages to create a well-organized project. We learned how to simplify a complex system into more clear subsystems by defining use-cases, states, sequences, activities through modelling. It also instructed us on how to work as a team and share tasks. Debating on very little details for many hours showed us that it

is not an easy progress to find optimal decisions for things such as requirements, system composition, object and their attributes, etc.

4. Users Guide

4.1 Intro

In IQ Puzzler, the player is tasked to complete a board to the point which there are no gaps on the grid left empty using the puzzle pieces that are available at the moment. The rules are rather simple.

4.2 Rules

- The player can only use the pieces that are available to them.
- In order to win the game, the board must be completed with no empty spaces.
- The player can not move the pieces that are part of the puzzle.
- The player can rotate or flip the pieces as they like.

4.3 System Requirements

- Windows 7 or better.
- 1,8 GHz or a faster cpu.
- 2 GB RAM or more.
- 4 MB disk space or more.
- At least 480p resolution with 24 fps screen.

4.4 Installation

The game is contained in a single exe file where the potantial player can download and play upon executing the executable file whenever he or she desires.

4.5 How To Use

4.5.1 Starting the game

As the game executes, the user faces the opening animation followed by the menu screen. From the menu, player can select the game mode he or she desires to play or can adjust the settings along with the option to create his or her own puzzle.

4.5.2 Playing Single Player

If the player wishes to choose the single player mode, first he or she needs to choose between the levels. Not all levels are unlocked initially and the player should expect to gain access to harder levels as the player goes further and further through the levels. The

reasoning behind this is preventing the player to try harder levels before learning the easier ones therefore, losing the desire to play a puzzle game that is too hard for a beginner.

After choosing the level, the game begins along with the timer. On the screen the player sees the board with already placed and immovable parts. The player is tasked to fill the gaps with the parts that are available for placement.

4.5.3 Playing Multiplayer

If the player wishes to choose the multiplayer mode, first he or she needs to enter the information needed for online playing. After this step is completed with success, the player needs to choose a puzzle to play from the browser. Exclusively for these puzzles, the player can see the leaderboards or scores and compare the scores with his or her scores. The gameplay is identical to single player for this section.

4.5.4 Using Puzzle Creator

This is a tool for the player to create or design his or her own puzzle to share with friends, share with the internet using multiplayer mode or just simply play by themselves. The interface for the Puzzle Creator is rather simple. The player has the option to auto-generate a filled puzzle and remove some selected pieces in order to create a puzzle or they can start from scratch and fill the puzzle all by themselves without any already placed parts yet for this option, the player has to finish the puzzle after it is completed in order to verify the validity of the puzzle and whether if the newly created puzzle has a solution or not.