

UNIVERSITY OF STAVANGER, NORWAY

Picture Game - Dat240

(Group DaPaSaHa)

Daniel Cruz López

Håvard Furdal Seim

Pablo González Pérez

Samuel Helland

November 19, 2022

Github link: <https://github.com/dat240-2022/DaPaSaHa>

Abstract

Development of a one or two player game on image recognition, which can be used to collect data on how humans think when viewing images and identifying objects. In this case, the player will be the one who tries to guess the image and there will be an oracle that will select random segments of the image for the guesser to try to guess the image. The game has been created using C#, CSHTML and the use of GUEST/POSTs for connecting to a database.

Contents

1. Introduction	3
2. Design	4
a. Class Diagram	4
b. Sequence Diagram	4
3. Backend	5
a. Game Domain	5
b. User Domain	6
c. Player Domain	6
4. Frontend	7
5. Summary and Contributions	9
a. Summary of the project	9
b. Contributions	10

1 Introduction

The project consisted of creating a game where a player had to guess the image using a number of segments displayed on the screen and another player was in charge of selecting segments so that the guesser guessed the word correctly. The guessing player has up to three attempts to guess the title of the image. If the player failed three times or all the segments were already displayed on the screen, then the game is over and the players score no points. Otherwise, the fewer segments the player requires to guess the word and the fewer times they miss the word, the lower the penalty will be on their score. In the leaderboard the player with the lowest score will be the first (obviously without taking into account the score 0 which is for those who have lost).

Another important point is the identity of the proposer. This can be an AI if the game is in "single player" mode. In case the mode is "multiplayer" the proposer will be a human. Regardless of the player's identity, their role will be the same: select segments of an image to help the guesser guess the word as quickly as possible.

For the development of the game we decided to meet for the first time in the laboratory class and show ideas for development. After having created the closest thing to a class diagram on paper, we decided to get to work with the code.

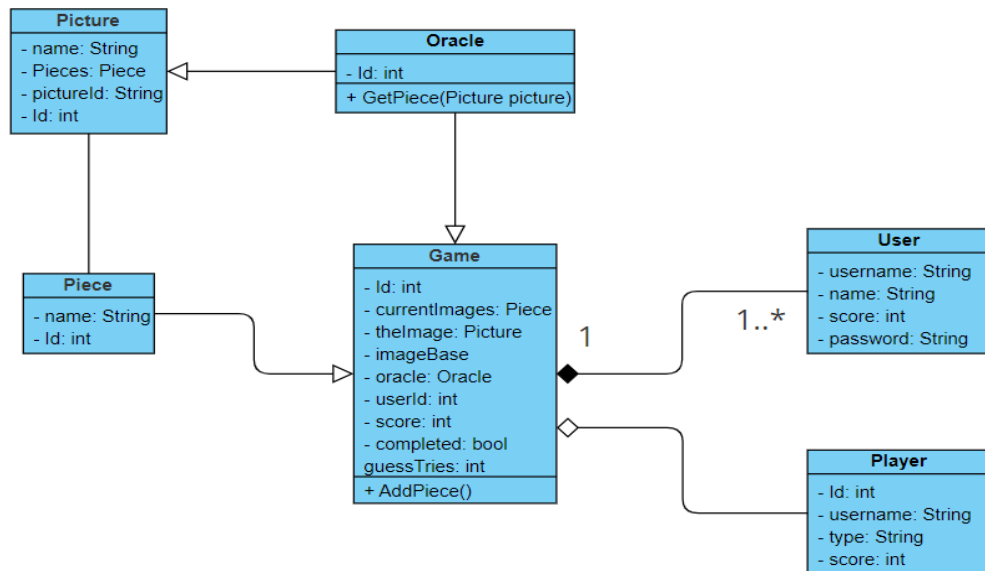
Taking the second laboratory of the course as a reference, we try to make the skeleton of the code and carry out the implementation of the log in and the database. Once that was done, we decided to adapt the code to the third laboratory of the course and carry out the rest of the processes.

It should be noted that not all ideas proposed on the first day. We haven't had much time to develop the project and the lack of one more member is noticeable.

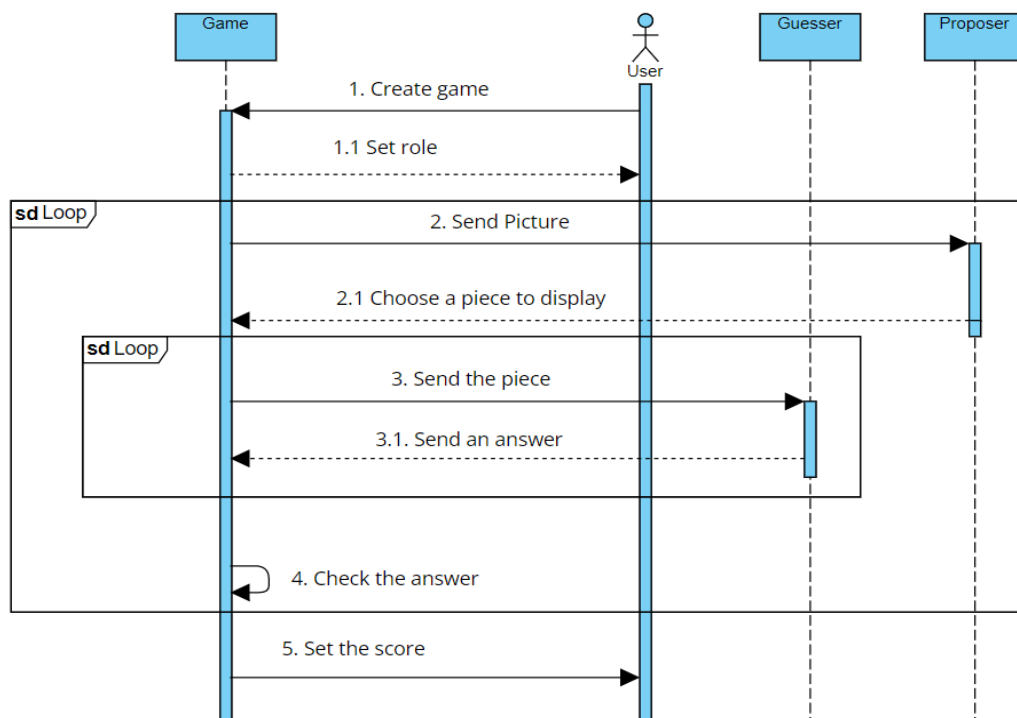
We have met a large number of times to discuss the distribution of tasks, move forward with the project, show new ideas, show different process developments to see how it helps the project, etc. All this added to the fact that we have tended to have similar ideas at all times about the development of the project, makes its progress more fluid.

2 Design

2.1 Class Diagram



2.2 Sequence Diagram



3. Back End

3.1 Game Domain

Game Class

The Game class contains the game logic. In this class the player is added, the id of the random image that has been chosen is stored, as well as the number of pieces that are being selected, the score when the game ends, its status, which is true if the game is finished and false if not. It also stores the user's id, so we can set it for when the player wants to resume the game in case they have logged out without finishing the game. Finally, it stores the number of attempts that the player has taken.

It has a function, add pieces. As the name of the function itself says, it adds a piece to the game list.

In addition to the pipeline to make a connection to the database, either to create a game or get an existing one, it has a service for when we want to add a random piece.

When a new game is created, the Oracle, Piece and Picture classes are called, which we will explain next.

Oracle Class

The Oracle class uses the IProposer interface to get the piece of the picture. The function looks for a random piece of the picture and returns its index.

Picture Class

The Picture class only has the name, a parts list, and the ID.

Piece Class

The piece class is very simple. It only has a name and ID.

3.2 User Domain

The user domain contains the user class and the pipeline to connect to the database, having different get (id, user, etc), a file to create users and store them in the database, and a file to check that the user exists.

Regarding the user class, the object contains the following attributes: username, name, score (which is initialized to 0), password and iD. We use the validator to create name, username and password. With this, we achieve that these attributes are not written incorrectly or that they are not written at all.

3.3 Player Domain

Finally we have the player domain. First we have the pipeline to connect to the database. We use the get to avoid having to generate a new player every time the user creates a game if they already have one.

Regarding the player class we have two constructors. If no attribute is passed, it is understood that the session was not started and the type of player will be a guest. Otherwise, passing the player's id and username will create the player with those attributes. This will store the score at the end of the game and will be the one taken as a reference for the leaderboard.

4. Front End

In this section we describe everything related to the visualization of the web page for the user. For this, the CSHTML file type has been used due to the project format used. In total, we have 5 files that complement the HTML section. We have used the GET/POSTs to make our game work, relating the html section with the back end and the database. Here only the visual section of the game is shown, all the logical part was previously explained in the third section of the report. Next, we will make a brief description of each of the files that make up the front end.

INDEX.CSHTML

This file contains everything related to the user's login, where the username and password will be passed to then perform the GEST to the database. Likewise, the user can click on "Register" if they do not have a user. This option will send you to the registration section. Otherwise, if the user does not want to register, they can simply click on the "play as guest" option. This will send you to the menu as a guest.

REGISTER.CSHTML

To access this file, the actor must first have gone to the start of the web page and clicked on the "register" option. Regarding the page, it is as simple as a couple of text boxes that will ask for a name, username and password. This information will be stored in the database through the POST. Finally the actor will be sent back to the home page where he can log in.

MENU.CSHTML

If the user was able to log in successfully, they will appear on this new page. It is a very simple menu with three options: create a new game, see the leaderboard or log out.

In the first case, the user will create a game if it has not been previously created. If it is, the user will return to the game that hasn't finished yet, or create a new one if the user wants to delete the previous record.

In the second case, the user will go to the leaderboard section, where he will see the ranking of players.

In the last case, the user will return to the home page, unable to go back and access the menu without logging in again. It is a security measure implemented by the group.

LEADERBOARD.CSHTML

This page is very simple, it only shows the ranking of players by score from least to most. The lower your score, the higher your ranking. The user in question can only view the leaderboard or return to the menu. To obtain the list of players we perform a GET to the database in the table of players and we order them by score.

GAME.CSHTML

This page is the most complex of all. Create a new game or return an existing game if the user had an unfinished game (status completed = false). This page shows a text box where the player can write the word that he thinks is correct. You will have a maximum of 3 attempts to guess the word. In case of not wanting to risk it, the player can ask the proposer for a segment of the image. This will cause a request to be made to the proponent and it will randomly select a new segment (in case it is an AI). The guesser will be shown the new segment on the screen.

This process will be repeated in a loop until one of the following events occurs:

- The player spends his 3 guesses and does not guess the word. In that case the game will be over and the player will not get any points.
- The proponent does not have more segments to select. Exactly the same thing will happen as in the first case.
- The player guesses the word. It will be game over and score will be assigned based on the number of segments used and attempts remaining. After that, the player will return to the menu where he can create a new game or even view the updated leaderboard.
- The player closes the page. The game will be maintained and the user will be able to resume it later.

5. Summary and Contributions

5.1 Summary of the project

From the beginning of the project the group organized several meetings to establish the ideas for the development of the project. Unfortunately, the creation of the group was late, therefore, there was not much time for the development of the project. Also, we were 4 members, but we were able to manage to distribute the work so that the group was productive.

First we had meetings about the design of the project. We didn't want to start programming until we were sure that the ideas were clear and the project structure was the most appropriate. We had to deal with the load of work and exam preparation to organize the meetings. It was a bit difficult to arrange meetings, but we did our best to have as many as possible. Fortunately, we created a discord server where we were able to communicate and share resources.

The feeling from the group was that the back end was more complicated to develop than the front end. The front end overall was pretty easy to do. The most complex thing was the visualization of the image for the guesser as the segments appeared. On the other hand, the back end was complicated by the creation of the game. The log in part was the easiest, as well as adding the images and their management.

5.2 Contributions

Task	Developers
Block Diagram	
Class Diagram	Daniel
Sequence Diagram	Daniel
Game Domain	Samuel, Pablo
User Domain	Daniel, Pablo
Player Domain	Daniel, Pablo
Images	Samuel
Index.cshtml	Daniel, Pablo, Håvard
Menu.cshtml	Daniel, Pablo, Håvard
Register.cshtml	Pablo, Håvard
Game.cshtml	Samuel, Pablo
LeaderBoard.cshtml	Daniel
Logout.cshtml	Håvard
Session management	Håvard, Pablo

References

<https://github.com/dat240-2022/assignments> → Course assignments