Trivial\_i1a Documentation: Second Deliverable

##### 

### Mario Germánico Pascual Tomás

Daniel Gonzalez Gómez

Edgar Herrero Uría

Adrián Prada Gil

Ainhoa Longo Pérez

### DEVELOPMENT OF THE DESKTOP TRIVIAL GAME

Proposal of the problem

The NoGame Company has decided to develop a desktop version of the trivial Game. For that, the development group has to reuse the application in charge of extracting the question, which will be shown in a graphical user interface where the Trivial Game will be played.

The board representation can be simplified transforming the traditional circumference in a track that follows a straight line with an end square.

On the other hand, the number of competitors depends on the available number of colours. Besides every player will have a login and a password that will have assigned information related to the punctuation.

The gameplay is based on the traditional way of playing the Trivial. To begin with players will login and from there, the computer will generate dice rolls for each player. After that, the player will choose the square he wants to move. Chosen the square a question according to the colour of the square will appear, if the player matches the question of the given category, he will be assigned the wedge of the that colour and he goes to another square. But if he fails the question the turn changes to the next player. Once a player has all the wedges he goes to the end square in which a question is done that if it is known, ensures the victory.

Finally the company is interested in having a differentiation of roles between players with privileges when obtaining information about other users and normal users. The company is interested in using the app in other kind of platforms too so the game graphical interface and the logic should be separated.

Identification of the stakeholders

#### Trivial Development team:

They are the ones in charge of creating the trivial game and develop the architecture in order to facilitate the creation of future trivia games.

Some of their objectives are:

- Develop the Trivial game in an efficient way, controlling the costs and following the standards of usability thinking in the final user.

- Create a reusable architecture that can be implemented in other games based on question/answer mechanisms.

#### People in charge of NoGame:

These are the directors of the corporation, in charge of the budget, from which they allocate funds for the project and administrating the different games developed.

Some of their objectives are:

- The duration of the project should be short and the costs as minimum as possible.

- Get the maximum Profit.

- Get an application whose architecture can be used in other variants of the game.

#### Development team of NoGame:

They are the ones in charge of developing future trivia game that will use the architecture developed by Trivial Development team as a help in the process.

Some of their objectives are:

- Use that architecture to facilitate the development of new versions of the game.

- Finding the best technological alternatives to implement the game, and communicating them to the Trivial Development team.

#### Players of the Game:

The final users of the product, they want to play the game in an easy and intuitive way.

Some of their objectives are:

- Playing the game in an easy way without having to spend too much time trying to understand how the application works.

Identification of the quality attributes

#### Availability

* The system must have available all the functionality 24-7(every day at any time).

#### Modifiability

* Scalability of the system, we probably may do changes in the program.
* Facility to change the parser system, the questions and answers files format.
* Facility to change the connection with the database if in the future we want to change the database administrator program.

#### Performance

* We have to make sure that our systems responds to the user in a reasonable amount of time.
* Latency should be as low as possible in order to provide the user a good experience within the application.
* Throughput (number of events that take place within a given amount of time) should be as high as possible.

#### Security

* We have to ensure the security and strength of the system.
* We have to provide a good architecture in order to avoid non-ethical attack against our system that try to break into the server or that try to mislead the system.
* The system should be able to avoid DDoS attacks (very important in an application like this one).

#### Usability

* The system must be the most intuitive as possible, in order to help the user to understand its functioning.
* A manual must be available for any user that doesn't understand how to use the system.

#### Adaptability

* The system must work for any operating system.

#### Time to market

* Short development cycle

#### Cost-Benefit

* Null development cost

First approach to the solution

#### First approach

Once we have analyzed the requirements of the system, we have arrived to a possible solution based on a Batch system. That is a way of reducing the interation with the user avoiding an extra cost in specialized operators of the system focused on managing the translation and the databe. This solution will be formed of an application that will process text files with different formats extracting its content and storing it in a database.

#### Risks related to the solution

1. Guarantee of the database integrity in case of erroneous data introduce.
2. Protection against the introduction of not valid data both the parser and the database.

To avoid the problems caused by the risks mentioned, the following solutions are proposed:

1. Creating a backup of the database.
2. Creating a series of preconditions and exceptions that controls the flow of data that enters the system.

Quality attributes and stakeholders

|  |  |  |  |
| --- | --- | --- | --- |
| Stakeholders  vs  Attributes | ST-01 | ST-02 | ST-03 |
| AT001 |  | X |  |
| AT002 | X |  | X |
| AT003 | X |  | X |
| AT004 |  |  | X |
| AT005 |  | X | X |
| AT006 |  | X | X |
| AT007 |  | X | X |
| AT008 | X | X | X |
| AT009 | X | X | X |
| AT010 | X | X | X |
| AT011 |  | X | X |
| AT012 |  |  | X |
| AT013 |  |  | X |
| AT014 | X | X |  |
| AT015 | X | X |  |

Business description of the solution

In our application we will have 3 differentiated systems

#### Graphical user interface:

This section of the system will be the one in charge of showing the game board and the questions, making possible to interact with the Trivial Game. To do this, the Swing library of Java programming language is used.

#### Logic:

This part of the application performs all the operations related to the functionality of the game, it mainly get the questions, manage the players, the position of the player along the track…

#### Database:

Composed of text documents: one for the questions and another one for the users and all the information related to the login and the punctuation.

Scheme representation of the problem:



Quality scenarios



Component Diagram

# ASW Trivial

User´s system manual

General functionality

This version of the application have a more visual interaction to the user and the program will be divided into several phases:

* At first a logging window will appear in which the user will use his username and password to differenciate between administrator or common user, this username and password will be compare in a txt document.
* Then the game begins, a squared board will appear in which the game will be played, a dice will appear and the available boxes will appear.
* Player chooses a square and a dialog with the question of that category is shown, in which you have to choose one answer, and the game will be elapsed in this way till you reach the center of the board.
* Finally when you reach the center one question of each theme, if you answer correctly the game finishes and you win.

The load of the questions file is done in a automatically way.

GIFT format

|  |  |
| --- | --- |
| **// text** | Comment until end of line (optional) |
| **::title::** | Question title (optional) |
| **text** | Question text (becomes title if no title specified) |
| **!theme!** | Question Theme(optional) |
| **{ ... =right ... }** | Correct answer for multiple choice, (multiple answer? -- see page comments) or fill-in-the-blank |
| **{ ... ~wrong ... }** | Incorrect answer for multiple choice or multiple answer |