Quiz game (for Knowledge Island)

Content

1. Introduction ........................................................................................................................... 1
2. The idea .................................................................................................................................... 2
   2.1. Current situation ............................................................................................................. 2
   2.2. Goals ............................................................................................................................... 3
3. Available tools for development ............................................................................................ 3
4. Concept of the Quiz game ...................................................................................................... 4
   4.1. Functional requirements ............................................................................................... 4
   4.2. Data model ..................................................................................................................... 4
   4.3. Design Issues ................................................................................................................ 5
      4.3.1. How does the quiz game choose the questions for a player? ................................ 5
      4.3.1.1. Example for calculating the UserImportance ..................................................... 6
5. References ............................................................................................................................ 6

1. Introduction
This paper gives a summary of my work for the project „Knowledge Island“ of group 15 in the course “Serious Games”. “Knowledge Island” is a role-playing game consisting of a gaming part, which takes place on an island, and a quiz part to increase the knowledge of the player. The goals of the project are to create a game, which is fun to play and the player can gain knowledge.

We decided to create our game with the game engine “Unity 3D”. Nobody of our group had any experience in creating games with this game engine, except of the individually created moodspace projects, which we had to do the week before starting the group project. The team decided to split the work of the group project, and therefore I was responsible to create the quiz part for the game.

At first I tried with some tutorials from the internet to create my own quiz, but I finally realized, that it is not possible to create a whole quiz game within a couple of weeks, because unity 3D is a completely new development environment for me and consists of a huge framework. Therefore I searched for already existing assets for a quiz game on the internet and found some assets within the unity developer network forum (see [1]): This quizmaker consists of a Quiz Maker Editor and Quiz Manager:

*Quiz Maker Editor is Unity 3D editor extension. It allows you to set up main properties of your quiz, e.g. quiz name, number of questions, number of levels, basic background images etc. But more important, Quiz Editor is flexible tool for entering questions. There are four types of questions for now: Select correct answer(s) (see image below), choose correct image(s), matching (see images below) and image matching.*[1]
According to the description this quizmaker was exactly the part of software I was looking for. But during some basic tests I found out, that it is not working very conveniently. However, the whole development of the game “Knowledge Island” with all its details is not feasible in the available time, even though 4 people are working on that project, because we are all newbies in the development with Unity 3D. Nevertheless the goal of the project is to write down the ideas and to create a concept of the game in a way that it could be implemented by developers at a later time.

Therefore I came up with another idea, which can be realized by myself in a feasible amount of time. So I decided to create a quiz engine by myself using the programming platform of Microsoft .Net with the programming language C#, because I am familiar with this development environment and I am able to create this part within the available time. I know, that this is not (completely) integrable in the group project “Knowledge Island”, but I am sure that the basic logic can be reused.

2. The idea
The idea is to create a quiz game in combination with a physical game (e.g. basketball, dart,...). The quiz game consists of different questions specific to the context of the player, for instance all players are working in the same office with the same tools/technology and therefore the questions are based on that tools/technologies. The players of the game always stay the same and the game is played regularly. Each player is able to create questions for the quiz game and assign a difficulty and an importance level to the created question.

In the first step of the game, each player has for instance 5 tries to shoot some baskets. In the second step the players, who will have no hit or who will have the most hits will take part in the quiz round. The players, who had at least one hit, but did not win are not taking part in the quiz round. If the winner is able to answer the question correctly, he/she will get additional points. If the players with no hit are able to answer the questions in the quiz correctly, they will get zero points, otherwise they will get -1 point. The game keeps track of the points.

2.1. Current situation
I am working in a software development company with 10 developers. The most of our projects are developed with the same tools and the same technology (Microsoft .Net Framework) and therefore every developer has an extended knowledge in this technologies and uses more or less the same tools. Until now we are shooting some baskets during the breaks and keep track of the points on a sheet of paper. We are playing at least 2 times a day.

The idea is to extend the basketball game with a useful quiz part in a way that it is fun. As mentioned above the players/developers are motivated to create questions by themselves for the quiz part, which will lead to discussions and communications and therefore increases the share of knowlegde within the company.
2.2. Goals

There are 4 big goals of the quiz game:

- **Motivate the users to create new questions:** It is very important that there are always new interesting questions in the quiz game, otherwise the quiz will get boring and nobody would like to play it.
  - Therefore a ranking with the amount of created questions for each user is displayed on the start screen.
  - The player with the most created questions will get a reward from the management once a year.

- **Motivate users to learn:** the users should be motivated to gain knowledge
  - This is done via a competition between the users, therefore another ranking with the actual score of each user depending on the correctly answered questions is displayed on the start screen.

- **Creating new questions should be very fast and easy:** When the developer for example is reading an article about a new technology and has a good question in his/her mind, then it should be very fast and easy to create this question in the game. If it is a pain, the user would not create the question.
  - Fast startup: There is no login required for playing the quiz. A login is only required for creating questions, and therefore a fast login-method is used (e.g. windows logon).

- **The game should be fun:** The users should like to play the game, because it is fun:
  - There is a foto and the nickname of each user displayed on the screen, so it is more personal and the users like it more than a usual game.
  - Players can create funny citations and insider jokes depending on the day or the user (e.g. “Hey Herbi! Tell me the answer, thoroughbred software developer!”). This citations and jokes are presented from the virtual guy “Bill” in the quiz, who is asking the questions as well.

3. Available tools for development

I searched on the internet for a possibility to create the software together with the group. Therefore I found a great online-portal of Microsoft, which is currently available for free, to organize the team development within the Team Foundation Server 2012 (TFS 2012) [2]. The TFS is a software for Application Lifecycle Management (ALM) and contains:

- source control: source code repository, which can be shared with a group of developers
- work item tracking: it is possible to organize the development task, which have to be done, within TFS
- build management: check-in’s to the repository can trigger a build of the software
- reporting: you can organize sprints and TFS returns reports regarding available time, work hours, etc.

I created a new project for the quiz game to manage within the TFS in the cloud: [https://herbisoft.tfspreview.com/](https://herbisoft.tfspreview.com/). So the developers can connect to this project and create the game together while using the same code repository and coordinating the different tasks.
4. Concept of the Quiz game
This chapter includes the specification of the software.

4.1. Functional requirements

<table>
<thead>
<tr>
<th>Functional requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create questions</td>
<td>Users can create questions (multiple choice questions or questions with plain text answers). Questions have an importance level, a difficulty level and a category assigned.</td>
</tr>
<tr>
<td>Score of Physical Game</td>
<td>One or more users can take part in the physical game and the winner and the looser (0 hits) take part in the quiz game. The scores for each user in the physical game are entered and can be increased by the quiz game.</td>
</tr>
<tr>
<td>Display statistic</td>
<td>Different statistics for each user is displayed (created questions, correct/wrong answered question, hits per game, sum of played games)</td>
</tr>
<tr>
<td>Rate questions</td>
<td>Users can rate questions, which influences the importance level of the question</td>
</tr>
<tr>
<td>Citations</td>
<td>The user is able to create citations and jokes within the game and relate them to players</td>
</tr>
</tbody>
</table>

4.2. Data model
This section gives a datamodel and the description of the datamodel.
Description of the datamodel:

- Every question can be either a multiple choice question or a question consisting of plain text. Each question has different answers assigned, which can be correct (IsCorrect=true) or can be incorrect (IsCorrect=false, used for wrong multiple choice answers).
- A user can participate in a game (UserGame) and therefore the score and the information, whether the user won or lost, is saved. The user also have different categories assigned and therefore he/she will only be asked the questions, which are within the assigned categories.
- For every user and every question the system keeps track, whether a question is more or less important for a user to be asked. This is done via the UserImportanceLevel.
- The user is also able to rate questions, which has an influence of the importance level of the question itself. The higher the rate, the more important the question.

4.3. Design Issues
In this chapter some important questions for creating the quiz-game are discussed:

4.3.1. How does the quiz game choose the questions for a player?
The quiz game chooses the questions on a random basis for each player, but there are some rules the algorithm implements:

- Each player has a difficulty level and one or more categories assigned and therefore he/she will only get questions, which are equal or below his/her difficulty level and are within the assigned categories.
- During one game the same question must not be chosen more than once (even for different users taking part in the game)
- Every question has an importance level. The lower the importance level the more important is the question. The importance level of the question is the source for calculating the UserImportance level.
- If the user answers the question the UserImportance level for that user is incremented, which means that this question looses on importance for that user. In particular, if the user answers the question correctly, the UserImportance level is incremented with 5, if the user answers the question incorrectly, the UserImportance level is incremented only with 2 (which leads to a higher chance the user will be asked the same question again in the future).
- Also the UserImportanceLevel of the other users participating in the game is incremented with 1, because they are present and probably can watch the answers of the asked question.
4.3.1.1. Example for calculating the UserImportance

We have for example 2 players (Alice and Bob) in our game.

- **Action:** 2 new questions (questionX, questionY) with different importance-levels (1, 4) are created:

<table>
<thead>
<tr>
<th>Game-Nr</th>
<th>Player-Nr</th>
<th>Question-Nr</th>
<th>Question Importance-Level</th>
<th>User-Importance-Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Alice</td>
<td>X</td>
<td>1</td>
<td>1</td>
<td>The importance level of the question is initial assigned to the User-Importance-level (1 means “very important”, 4 means “not so important”)</td>
</tr>
<tr>
<td>0</td>
<td>Alice</td>
<td>Y</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Bob</td>
<td>X</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Bob</td>
<td>Y</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

The probability to get questionX is 4 times higher than to get questionY (for both players).

- **Action in Game1:** Alice gets questionX and answers it correctly.

<table>
<thead>
<tr>
<th>Game-Nr</th>
<th>Player-Nr</th>
<th>Question-Nr</th>
<th>Question Importance-Level</th>
<th>User-Importance-Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alice</td>
<td>X</td>
<td>1</td>
<td>6</td>
<td>5 is added to the user importance level (correct answer)</td>
</tr>
<tr>
<td>1</td>
<td>Alice</td>
<td>Y</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Bob</td>
<td>X</td>
<td>1</td>
<td>2</td>
<td>1 is added to the user importance level (because Bob was present, when Alice answered the question)</td>
</tr>
<tr>
<td>1</td>
<td>Bob</td>
<td>Y</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

The probability that Alice will get questionX again is now much lower. Also the probability for Bob to get questionX was decreased.

- **Action in Game1:** Bob gets questionY and answers it wrong.

<table>
<thead>
<tr>
<th>Game-Nr</th>
<th>Player-Nr</th>
<th>Question-Nr</th>
<th>Question Importance-Level</th>
<th>User-Importance-Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alice</td>
<td>X</td>
<td>1</td>
<td>6</td>
<td>1 is added to the user importance level (because Alice was present, when Bob answered the question)</td>
</tr>
<tr>
<td>1</td>
<td>Alice</td>
<td>Y</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Bob</td>
<td>X</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Bob</td>
<td>Y</td>
<td>4</td>
<td>6</td>
<td>2 is added to the user importance level (wrong answer)</td>
</tr>
</tbody>
</table>

The probability that Bob will get questionY again is now a little bit lower. Also the probability for Alice to get questionY was decreased.

5. References
