

# A Gamified Quiz!

Vilket ord av följande är ett verb?

- a) Ringde
- B) Leksak
- c) Boll

**Correct Answer!**

## The Quiz - A Gamified Quiz!

***Program manual and code documentation.***

*Developed by:*

*Ali Nazar & Michel Nicklön*

*Department of Computer and System Sciences- Stockholm University*

The Quiz is a gamified question and answer program. The Quiz is designed with gamification elements and was built as a prototype for a bachelor thesis project. The goal with the project was to check which gamification elements had the most effect on users in increasing their motivation for learning.

This document is for those who want to know how to use the program or to get an overview of the classes in the project and how the program is designed and implemented. The project is coded using HTML, PHP, Javascript and JQuery. The program is designed with the functionality to upload your own created questions as XML files and use them in program.

Please have in thought that this program is although a prototype that still needs alot of work and amendment in terms of functionality and user experience.

***This program is open source and is free for anyone to use in any way that they may desire.***

### *How to use the program*

The program can either be used with the inbuilt questions(in swedish) or you also have the option to create your own questions in the form of an xml file and upload it to the program.

- 1) **We recommend** that you test the program on a webserver. Once the program folder is on the server, you enter the url to start.html file. if you want to run the program on localhost jump to point 2.If you choose to run the program on a webserver and with the inbuilt questions you can jump to point 3.  
If you choose to run the program on a webserver with your own generated xml files containing your questions please jump to point 4.
- 2) In order to make the program work on localhost you will need to either install XAMPP or WAMPP and start the apache function in either of the two mentioned programs. I Also recommend to install Aptana Studio as it is a great IDE for web development projects. Once you have installed the programs, put the quiz project in the htdocs folder of your xampp or wampp folder located generally in the c drive. once you have done this use the import feature of aptana studio to import the project into aptana studio from the htdocs folder.Now in the browser you will either type localhost/Quiz/start.html if you want to use the already built in questions that are part of the program. If you choose to open the start.html then in the "frågemapp" field, please type:  
"points" for the points instance  
"leaderboard" for the leaderboard instance  
"badges" for the badges instance  
"levels" for the levels instance

3) URL to the start.html file and then in the frågemapp field type:

- “points” for the points instance
- “leaderboard” for the leaderboard instance
- “badges” for the badges instance
- “levels” for the levels instance

once you are completed with an instance, you can simply press the back button of the browser and start any other instance.

4) if you chose to try out your own created questions then URL to “start\_upload.html” in the browser. Here you just select the instance of your choice and upload the XML file containing your questions in the upload section.

## How to create your own questions file?

If you want to create your own questions xml file and use it them in your program then open the “QuestionBank” folder that is placed in the Quiz folder and read the pdf inside of it.

You are now ready to try out the quiz!

### *Overview of the classes*

- Start.html
- Start\_upload.html
- redirecter.php
- instancestyle.css
- frontpagesheet.css
- Folder: instance1 & instance2 & instance3 & instance4 **contains:**
- achievementbox.js
- achievementbox.css
- badgebox.js
- CountPosition.php
- MainQuiz.php
- SaveScore.php
- localhostTestingRedirer.php
- xpbars.css
- xpbarsprogress.

## **Description of the files and their containing functions.**

### **Start.html**

This class will be the first class to be called in order to start the quiz. This class contains plain HTML and call to the redirecter.php script in a html form. This page is designed so that we can choose what game element we want to test.

### **Redirecter.php**

This PHP class will detect what option the user selected and will redirect the user to the appropriate instance of his choice.

### **InstanceStyle.CSS**

The css class that designs the MainQuiz.php page and its containing html elements.

### **Frontpagesheet.CSS**

This CSS class also designs the MainQuiz.php page and mainly the border around the questions.

Now the following folders:

- instance 1
- instance 2
- instance 3
- instance 4

These folders contains repetetive code files and you might be feeling that why are there repetetive classes in each instance? and the main reason for that is a sudden change of requirements while building the program and not having enough time to amend this.

### **Achievementbox.CSS**

This CSS stylesheet designs the achievementboxes that are generated from the achievementbox.js script. This CSS stylesheet also designs the “wrong” and “correct answer!” popups and other popups such as if a player answers all of the questions right then the program will generate a special popup for him and this stylesheet designs that popup as well.

## Achievementbox.JS

This script contains a function called `Box()` that takes an argument called “achieved”. This function is primarily for construction and generating various achievement boxes. This script is mainly used in the badges instances. The `achievementbox.CSS` stylesheet styles the generated achievementbox. The function `box` uses JQuery to create a `fadeIn` and `fadeOut` effect of the achievementbox that is generated. This function also generates the “correct answer” box once the user answer a question rightly.

## Badgebox.JS

This script contains a function called `badgebox()` that takes an argument of name “medal” and this function, depending upon what type of argument value it gets, will generate a badgebox similar to an achievement box. The following should be passed in to the function in order for it to generate a badge:

- 1) “brons” for a brons badge
- 2) “silver” for a silver badge
- 3) “gold” for a gold badge
- 4) “platina” for a platina badge

This script also contains a function called `increasebadgesunit()` which performs the task of increasing the users total amount of badges statistics in the leaderboard.

## CountPosition.PHP

This php class is used to calculate the current position the user is on by comparing the scores from other contestants. This class is used in the leaderboard instance. This class is called from the `showexperienceandposition()` function inside the `xpbarprogress.js` script inside the “instance 2” folder.

## MainQuiz.PHP

This is the main HTML page for the Quiz, that is the main page that we see when running the program. Why it is in .PHP form is that we are using php functionality inside of it to generate new question after a question has been answered as well as for other purposes.

## SaveScore.PHP

This php class is used in order to save the score of the contestant. This class is called from the `showexperienceandposition()` function inside the `xpbarprogress.js` script inside the “instance 2” folder. The class opens the folder that has the name of the contestant, and saves the points into the file and closes it.

## LocalhostTestingRedirecter.php

This php class is present in every instance for the developer so that he can test a particular instance and its functionality on localhost. If you want to use this redirecter on localhost then you simply just change the form action attribute in `start.html` and/or `start_upload.html` to “localhostTestingRedirecter.php”.

## Redirecter.php

This php class works as a redirecter to the different instances when the program is running on a webserver. you will have to manually change the links so it meets the requirements for your webserver in order to reach the specific instances.

## xpbar.CSS

This CSS stylesheet designs the experiencebar that is used in the levels instance.

## xpbarprogress.JS

This class has the main functionality to increase the experience bar further when the user has received more points. The main function that exists in every instance in this script file is `increasexp()` that takes an argument of the name “amount”.

This script also contains the following functions:

- **increaseLevel():** This function increases the level statistics in the leaderboard.
- **showexperience():** this function increases the experience of the user in the leaderboard.
- **wronganswer():** Takes an argument of the name “answer “ and generates a “wrong answer” box whenever the user answers a question wrongly.
- **fade\_out4()**
- **fadeout()**

in the instance 2 folder, we have some additional functions:

- **showexperienceandposition():** increases the experience points stats in the leaderboard and calls the `SaveScore.PHP` class to save the newly received experience

points into the the users score file. It also calls the CountPosition.PHP to calculate what position the user is at the moment compared to other people who have played the game. This is compared using the points available in each of the files inside the contestants folder.

- **correctAnswer():** generates a “correct answer!” box whenever the user has answered an answer correctly.
- **positiondialogue():** This function shows the user which position he is at the moment compared to the other players who have played the game. If the user has recieved more experiencepoints than other players he will have reached position 1 in the leaderboard.
- **fade\_out2()**
- **fade\_out5()**

in the instance 3 folder we have some additional functions:

- **fade\_out3()**

The instance 2 folder contains another php class called gencontestantfile.php and this file is used in this instance to generate a file for the user so the program can enter the experiencepoints the user is getting while playing the game. This file is the used to calculate his position compared to other players.

we also have some png documents that are pictures that we use in the program.

```

sequenceDiagram
    participant User
    participant StartForm
    participant InstructorObj
    participant InstanceID
    participant ViewQuizObj
    participant QuizForm
    participant selectAnswersObj
    participant selectAnswersObj2
    participant SelectScoreObj
    participant scoreSectionObj
    participant targetObj

    User->>StartForm: 1. Needs user name(s)
    StartForm->>InstructorObj: 2. Inputs/Validates
    StartForm->>InstructorObj: 3. Checks responses
    StartForm->>InstructorObj: 4. Publishes test history
    InstructorObj->>InstanceID: 5. Sends form ID
    InstanceID->>ViewQuizObj: 6. Redirects to correct
    ViewQuizObj->>QuizForm: 7. opens
    QuizForm->>ViewQuizObj: 7.1 Change Quiz Page
    ViewQuizObj->>selectAnswersObj: 7.1.1 Generate a new question form
    selectAnswersObj->>User: 7.2 show new question
    selectAnswersObj->>ViewQuizObj: 7.1.1.1 load new question
    ViewQuizObj->>selectAnswersObj: 7.1.1.2 answering
    selectAnswersObj->>selectAnswersObj2: 7.1.1.2.1 Row
    selectAnswersObj2->>selectAnswersObj: 7.1.1.2.2 showAnswer()
    selectAnswersObj->>selectAnswersObj: 7.1.1.2.2.1 selectAnswer()
    selectAnswersObj->>selectAnswersObj: 7.1.1.2.2.2 showAnswerAndPosition()
    selectAnswersObj->>SelectScoreObj: 7.1.1.2.4.1 show
    SelectScoreObj->>scoreSectionObj: 7.1.1.2.4.2.1 photoChange()
    scoreSectionObj->>targetObj: 7.1.1.2.4.2.2 target()
    targetObj->>scoreSectionObj: 7.1.1.2.4.2.2.1 showAnswerCount()
    scoreSectionObj->>ViewQuizObj: 7.1.1.2.6.3 user starts a new target
    ViewQuizObj->>selectAnswersObj: 7.1.1.2.6.3.1 show user correct answer Row
    selectAnswersObj->>User: 7.1.1.2.6.3.1.1 show user wrong answer
    selectAnswersObj->>ViewQuizObj: 7.1.1.2.6.3.1.2 wrong answer
    ViewQuizObj->>selectAnswersObj: 7.1.1.2.6.3.1.3 wrong answer
    selectAnswersObj->>User: 7.1.1.2.6.3.1.3.1 show user wrong answer
    
```

We are aware that the program has not been programmed that professionally with design in thought and there is still alot of amemdments in terms of code needed. Alot of repetitive code is also occuring. So anyone who takes up this project in the future is welcome to amend the code and share it to others.

The reason we weren't able to program the software better was mainly due to lack of time and that the submission of our thesis was nearing.



***All of the source code is available on the following git hub link:***

***<https://github.com/Alinator/QUIZ>***

***Ali Nazar***

***Michel Lemponen***

**Thesis project:**

**Gamification- ett framtida koncept inom IT stött lärande?**

**Department of computer and system sciences**

**Stockholm University**