Regional ICT Skills Championship 2017

Application Development

SAMPLE TASK

Regional ICT Skills Championship 2017

1 Instructions

1.1 Introduction

This document describes the tasks to be performed for the Regional ICT Skills Championship 2016 in the category Application Development. We advise you to read this document carefully and to have an overview of the provided media like attached images and text. In order to complete the tasks, you are free to use whichever of the provided tools you like.

You could find yourself in the situation that there is not enough time to complete all remaining tasks. You are allowed to skip the implementation of certain tasks in favour for others. However, it is expected that you deliver a smoothly running application at the end of the competition to facilitate the assessment of your work.

Unless it was specified otherwise by your supervisor, you should find the required data and documents on the **Desktop** of your virtual machine.

1.2 Rules

Please comply with the following rules throughout the competition:

- Accessing the internet is not allowed
- The use of any communication device is prohibited (mobile, tablet, etc.)
- The use of your own storage devices is prohibited (USB sticks / hard disks, etc.)
- Communication between other candidates is prohibited
- You are not allowed to bring books, notes, etc. to the competition

The non-compliance with these rules may result in the disqualification from the competition.

1.3 Evaluation

Each task is rated by testing the implementation of the required functionality. In addition, the following criteria applies to all tasks and has to match your solution:

- All written code must be properly formatted, indented and commented;
- SQL queries must be protected from any possible injections;
- No error should occur / proper exception handling is implemented;

1.4 Deliverables

All deliverables must be submitted in a zip file named "AppDev_Name_FirstName.zip" and it must be placed at the folder location indicated by the supervisor. The deliverables are **executable files**, **source code** and the **database** (db-files, backup or SQL-export). Please also fill out the requested information in the file **registrationForm.docx** and include it in the zip file.

At the end of the competition you must also submit all documents and notes to the supervisor.

Only the contents of the zip file will be considered for the evaluation of the work. Only electronically delivered notes (*readme.txt or readme.rtf*) will be considered for evaluation. Notes on this document will not be used for the evaluation.

1.5 Tools

A System with the following software configuration will be provided to you to develop your application:

- JDK 8
- NetBeans 8.1
- Eclipse Neon
    - WindowBuilder Pro
    - e(fx)clipse
- .NET Framework 4.6
- Microsoft Visual Studio Community 2015
- Microsoft SQL Server 2014 Express
    - JDBC Connector
    - Management Studio
- MySQL 5.7
    - JDBC / .NET Connector
    - Workbench
- LibreOffice 5.1
- Adobe Reader DC
- Notepad++
- Windows 10

1.6 Documentation

To perform the tasks of this project the following manuals are available:

- T-SQL
- MySQL
- .NET (Visual Studio)
- JDK
- JavaFX

2 Project Introduction

Your task is to implement a two-player dice game called "Yatzy". Apart from the core gameplay mechanics, you also have to implement a database backed highscore list.

2.1 Gameplay

Players take turns rolling five dice. After each roll, the player chooses which dice to keep, and which to reroll. A player may reroll some or all of the dice up to two times on a turn. The player must put a score into a score box each turn. The game ends when all score boxes are used. The player with the highest total score wins the game.

2.2 Scoring

The game board is divided in two sections (left / right). The following tables show the points that can be earned in these sections.

2.2.1 Left Section

Category	Description	Score	Example
Aces	Any combination	The sum of dice with the number 1	scores 3
Twos	Any combination	The sum of dice with the number 2	scores 6
Threes	Any combination	The sum of dice with the number 3	scores 12
Fours	Any combination	The sum of dice with the number 4	scores 8
Fives	Any combination	The sum of dice with the number 5	scores 0
Sixes	Any combination	The sum of dice with the number 6	scores 18
Bonus	If the left subtotal is over 62 points	35	

2.2.2   Right Section

Category	Description	Score	Example
Three-Of-A-Kind	At least three dice the same	Sum of all dice (0 if less than three dice the same)	scores 17
Four-Of-A-Kind	At least four dice the same	Sum of all dice (0 if less than four dice the same)	scores 24
Full House	Three of one number and two of another	25 (0 if no Full House)	scores 25
Small Straight	Four sequential dice (1-2-3-4, 2-3-4-5, or 3-4-5-6)	30 (0 if no Small Straight)	scores 30
Large Straight	Five sequential dice (1-2-3-4-5 or 2-3-4-5-6)	40 (0 if no Large Straight)	scores 40
Yahtzee	All five dice the same	50 (0 if no Yahtzee)	scores 50
Chance	Any combination	Sum of all dice	scores 13

3   Storyboard

The Application consists of four different windows (main menu, highscore, name input dialog and game screen) as showed in the following pictures. The arrows show the navigation flow.


(after game is finished)
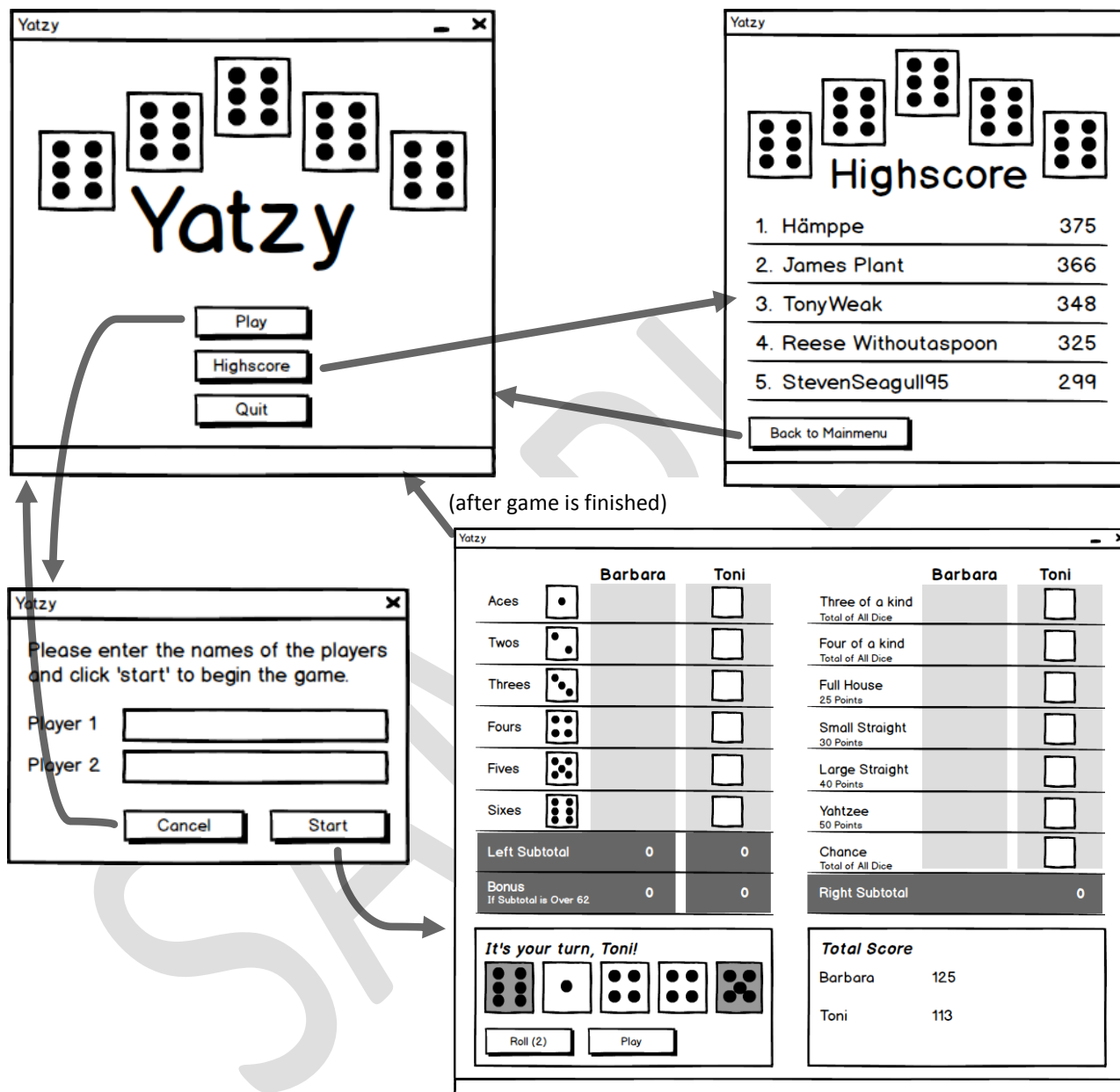
4 Implementation

4.1 Main menu

The main menu window is displayed after the user starts the application. Please create the window like shown in the mockup below.

- «Play» starts a new game (see chapter 4.2)
- «Highscore» opens the highscore window (see chapter 4.4)
- «Quit» terminates the application
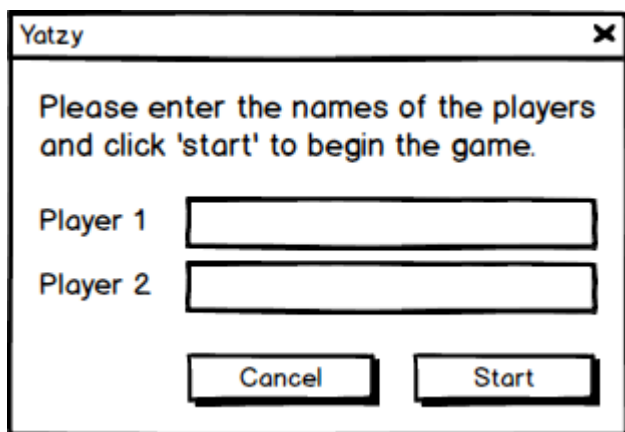- Always hide the main menu window if the user opens another one
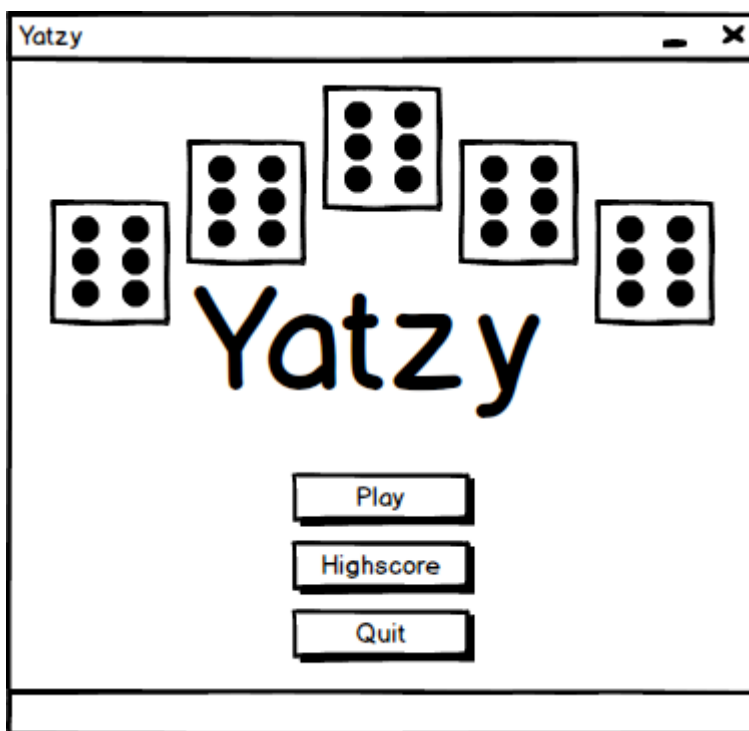
Mockup 1: Main menu window

4.2 New Game

If the user chooses to start a new game, a dialog should be displayed to allow the players to enter their names.

- Create a dialog like shown below
- «Cancel» closes the dialog and returns to the main menu
- Validate the user input. If the user clicks on «Start» and a textbox is still empty or just filled with whitespace:
  - Prevent the start of a new game
  - Display an error message
- «Start» closes the dialog (if no validation error has occurred) and opens the game screen (see chapter 4.3)

Mockup 2: Name entry dialog

4.3 Game Screen

- Below are two different mockups to give you an idea of how the game screen should look like. The first one shows the game screen after a new game has been started, the second shows the game screen during a game. Implement the window layout according to the mockups and implement the gameplay logic (see chapter ⬚ JDK 8
- NetBeans 8.1
- Eclipse Neon
    - WindowBuilder Pro
    - e(fx)clipse
- .NET Framework 4.6
- Microsoft Visual Studio Community 2015
- Microsoft SQL Server 2014 Express
    - JDBC Connector
    - Management Studio
- MySQL 5.7
    - JDBC / .NET Connector
    - Workbench
- LibreOffice 5.1
- Adobe Reader DC
- Notepad++
- Windows 10

4.4 Documentation

To perform the tasks of this project the following manuals are available:

- T-SQL
- MySQL
- .NET (Visual Studio)
- JDK
- JavaFX

Project Introduction and the requirements below). Use the provided images for the dices.

9

ICT-Berufsbildung Schweiz
Aarbergergasse 30, CH-3011 Bern
Telefon +41 58 360 55 50
info@ict-berufsbildung.ch
www.ict-berufsbildung.ch

Regional ICT Skills Championship 2017

Mockup 3: Game Screen (start)

All the scores are zero and the score boxes are displayed empty. The dices are not shown yet, because no player has rolled the dices at this time.

Regional ICT Skills Championship 2017

Mockup 4: Game Screen (in game).

In this example, Toni is playing the 8th round. He already rolled the dices once. He selected the two dices with number four to keep. The other ones would be rerolled if Toni chooses to do so. Additionally, he selected the score box of category „Fours". Therefore, the score box displays the points that Toni would earn if he finishes the round by clicking „Play".

4.4.1   Requirements
General

- Use the player names from the new game dialog
- It should be randomly decided which one of the players starts
- The game ends, if all score boxes are used for both players
- If the game has ended, save the scores of the two players in the database, so that you are able to generate a highscore list (see chapter 4.5). Additionally, display a message box to congratulate the winner:
  - Ask the user, if he wants to play again (same usernames)
  - If the user does not want to play again, take him back to the main menu

Rolling Dices

- Always display the name of the player who's turn it is
- Use a random number generator to simulate the rolling of dices
- The "Roll" button should show, how many times the player can reroll the dices
- After each roll, the player can select the dices he wants to keep
- Only the dices which aren't selected should be rerolled

Scoring

- If a player clicks on a score box, display the points he would earn if he finishes the round using this category. Refer to chapter 2.2 for the scoring details
- Display an error message if the player clicks on "Play" and hasn't rolled a dice or selected a score category yet.
- The round must be finished by clicking "Play"
- The score boxes should only displayed in the column of the user whose turn it is. For the other user, display only the scores without the boxes.
- Left Subtotal: calculate the sum of points of the left section (without bonus)
- Bonus: 35 points, if the left subtotal is greater than 62, otherwise zero
- Right Subtotal: calculate the sum of points of the right section
- Total: Calculate the grand total (left subtotal + bonus + right subtotal)

4.5 Highscore

Create a window that displays the five players with the highest scores (see Mockup 5). The scores have to be loaded from the database. The application should create the necessary structures if they do not exist yet, please add the following logic to setup the required database tables to the application start-up process:

- Connect to the database
- Check if the table(s) exist
  - If they do not exist, create them
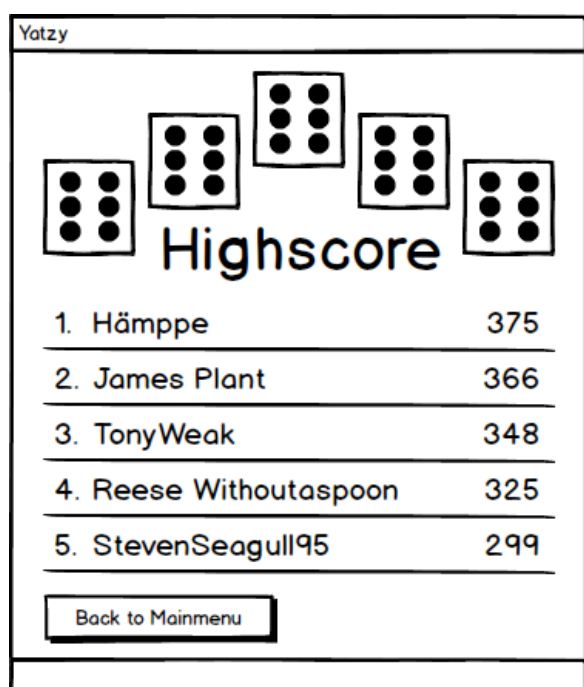
Mockup 5: Highscore window

5 Delivery

Please hand in your deliverables (see chapter 1.4) on the USB flash drive provided, or where the supervising person instructs you.