Mock Exam ADO Week 11, November 2019

Time: 90 minutes.

Important notes:

- You are allowed to use everything on paper (books, notes, etc.) and on your laptop, but only what you bring in: you are not allowed to borrow something from someone else.
- During the exam it is not allowed to use the network. You should make the exam yourself: so no communication with MSDN or google for help and no communication with other students, like using Facebook, e-mail, Skype, Dropbox, mobile phone or whatever.

+-

Let's play The Witch Hunt!

Introduction

The Witch Hunt is a game modelling a conflict between two groups: the evil characters (witches) and the good characters (king, knights, wizards). Each player is assigned a role affiliated with one of these groups. Because of a magical spell, players do not know the roles of the other players. The goal as a player (good or evil) is to try kill the rival group.

Your assignment for this exam is to create an app for this game. The app enables users to add new players, view players, kill (remove) other players and check game statistics.

To add a new player, the app requires a player's name and role. In the startup project, the role of a player can be selected from a combobox. There are four roles: KING, KNIGHT, WITCH, WIZARD. The combobox is already populated with these roles.

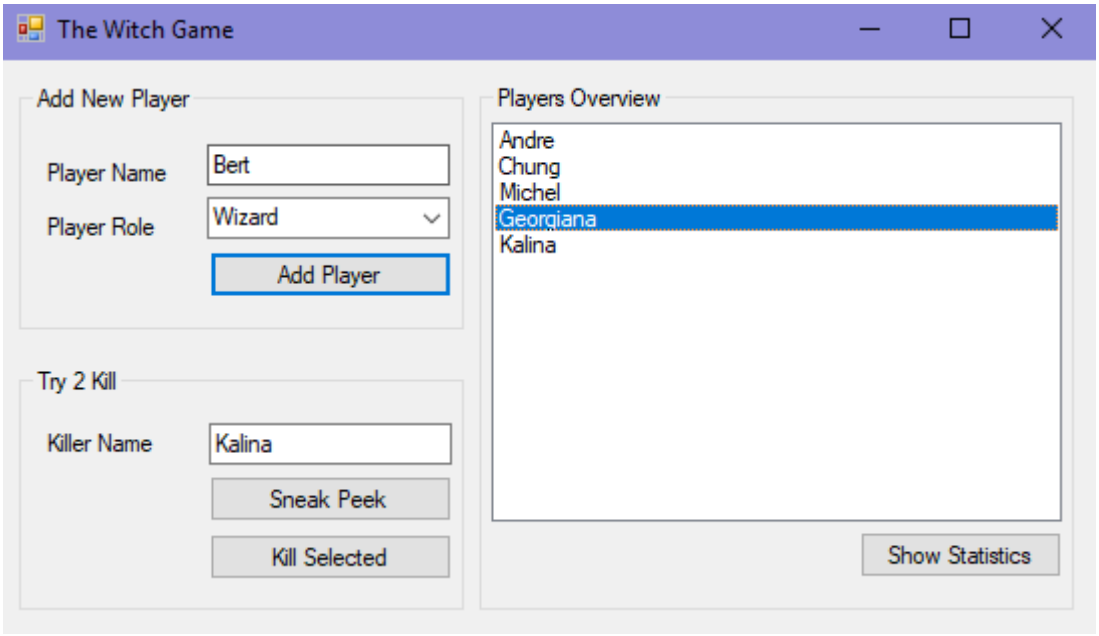
The app keeps a collection of the players' names and corresponding roles, but shows only their names and does not show their roles. The players' names are always showed in a listbox.

The app also has the possibility to show statistics of the number of each kind of players which are still in the game. For instance: KING: 0 KNIGHTS: 2 WIZARDS: 2 WITCHES: 6.

The user interface for this app has already been made (see Screenshot 1), and you can find it in the startup project.

You will need to make the app work. More specific instructions will be provided in the assignments. In case you need to add more functionality, feel free to do so.

The user interface (Form1) looks like this:



Screenshot 1

- In the "Add New Player" groupbox, pressing button "Add Player" will add a new player with player name and role from the "Player Name" textbox and "Player Role" combobox, respectively.
- Groupbox "Try 2 Kill" is about the killing, detailed in assignment 2.
- In the "Players Overview" groupbox, corresponding listbox will always be populated with current player names, and button "Show Statistics" will show statistics of current players.

Now it's time to open the startup project; you will need to implement the adding (assignment 1), the killing (assignment 2) and the statistics (assignment 3).

Assignment 1: Adding New Players (10 + 10 + 25 = 45 points)

Assignment 1a:
Define an enum type to represent the player roles and use it anywhere possible in the code. The roles are:

- KING, KNIGHT, WITCH, WIZARD

Assignment 1b:
Add some collection(s) to keep track of all the players' names and corresponding roles.

Assignment 1c:
Implement button `btnAddNewPlayer`, that takes a given player name from `tbPlayerName` and a role from `cmbRole` and updates the collection(s) accordingly.

- Make sure that there cannot be two KINGS in a game. If a new player is added with role KING and there already exists a KING in the collection, show an appropriate message; for instance: `"There already exists a king"`. It is up to you where you show this information on the screen.
- Make sure that there cannot be two players with the same name, if the name already exists, show an appropriate message; for instance: `"There already exists a player with this name"`. It is up to you where you show this information on the screen.

Make sure you update the listbox accordingly; show only the names of the players.

Assignment 2: Killing Other Players (20 + 25 = 45 points)

In The Witch Hunt game, the goal is to kill your rival group. Any player from the current list of players can be the killer. The killer selects another player from the current players to be killed.
In the app, the killer is written in the `tbKiller` textbox, and the player to be killed is selected from the listbox (see Screenshot 1). Killing means that the player must be removed.

Assignment 2a:

To help the non-die-hards, there is a sneak peek option to show the roles of both the killer and the player being killed before actually killing.
In this assignment, you implement button `btnSneakPeek` that shows for both the killer and player to be killed their names and roles. For instance:

- If the killer name in `tbKiller` textbox is Kalina and the selected player in the listbox is Georgiana, show information about their names and roles, such as `"Kalina is a WIZARD, Georgiana is a WITCH"`.

It is up to you where you show this information on the screen.

Assignment 2b:

The way The Witch Hunt game works is that any player can kill any other player. However, if a player tries to kill another player which is a witch, he can do it only if he is a wizard. If a player is either a witch, king or knight and tries to kill another player which is a witch, he dies instead.
Implement button `btnKill` such that it also takes into account the role of the killer and makes sure that the killer can kill, or else be killed instead.
If the killer is killed instead, show also an appropriate message; for instance: `"Ops, you died!"`. It is up to you where you show this information on the screen.
For instance, in Screenshot 1:

- If player Kalina is a WIZARD and player Georgiana is a WITCH, Kalina kills Georgiana, meaning Georgiana is removed.
- If player Kalina is a KNIGHT and player Georgiana is a WITCH, Georgiana kills Kalina, meaning Kalina is removed.
- If player Kalina is a WITCH and player Georgiana is a WITCH, Kalina kills Georgiana, meaning Georgiana is removed.
- If player Kalina is either a KING, KNIGHT, WIZARD and player Georgiana is either a KING, KNIGHT, WIZARD, Kalina kills Georgiana, meaning Georgiana is removed.

For both assignments 2a, 2b, make sure that the killer name written in the `tbKiller` exists in the collection. If not, show an appropriate message, for instance: `"<<tbKiller.Text>> does not exist"`, where `<<tbKiller.Text>>` is of course the name written in the `tbKiller` textbox. It is up to you where you show this information on the screen.
You may also assume that the killer and the player being killed are not the same.
Make sure you update the listbox accordingly; show only the names of the players.

Assignment 3: Statistics (10 points)

Lastly, you want to keep track of which roles are still in the game. Implement button `btnStatistics` which show the number of players per role.
For instance: `"KING: 0 KNIGHTS: 2 WIZARDS: 2 WITCHES: 6"`.
Show this information on the screen. It is up to you where you show this information on the screen.

END of ADO mock exam.