



COLLECT JOE

Projektarbeit LB2 Modul 404

Exposee

Collect Joe ist ein Spiel, bei dem es darum geht, durch abschiessen der aufleuchtenden Boxen, so viele Punkte wie möglich zu sammeln.

mam

marianne.marti@pitteam.ch

Inhaltsverzeichnis

Vorwort	2
Projektübersicht	2
Spielfeld	2
Einstellungen	3
Rangliste	4
Score Eintragen	4
Umsetzungsanleitung	5
Zustandsdiagramm	5
Formulare	5
Spielfeld Form «frmField»	5
Optionen Form «frmOptions»	6
Ranglisten Form «frmScoreList»	7
Spielstand speichern «frmEditScore»	8
Programmierung	9
Optionen Form «frmOptions»	9
Ranglisten Form «frmScoreList»	11
Spielstand speichern «frmEditScore»	12
Spielfeld Form «frmField»	13
Anhang	18
Benennungsrichtlinien	18

Vorwort

Im nachfolgenden Text stehen alle «Attribut»- und «Methoden»-Namen zwischen Anführungszeichen z.B. «frmField».

Datentypen werden in eckigen Klammern geführt z.B. [string].

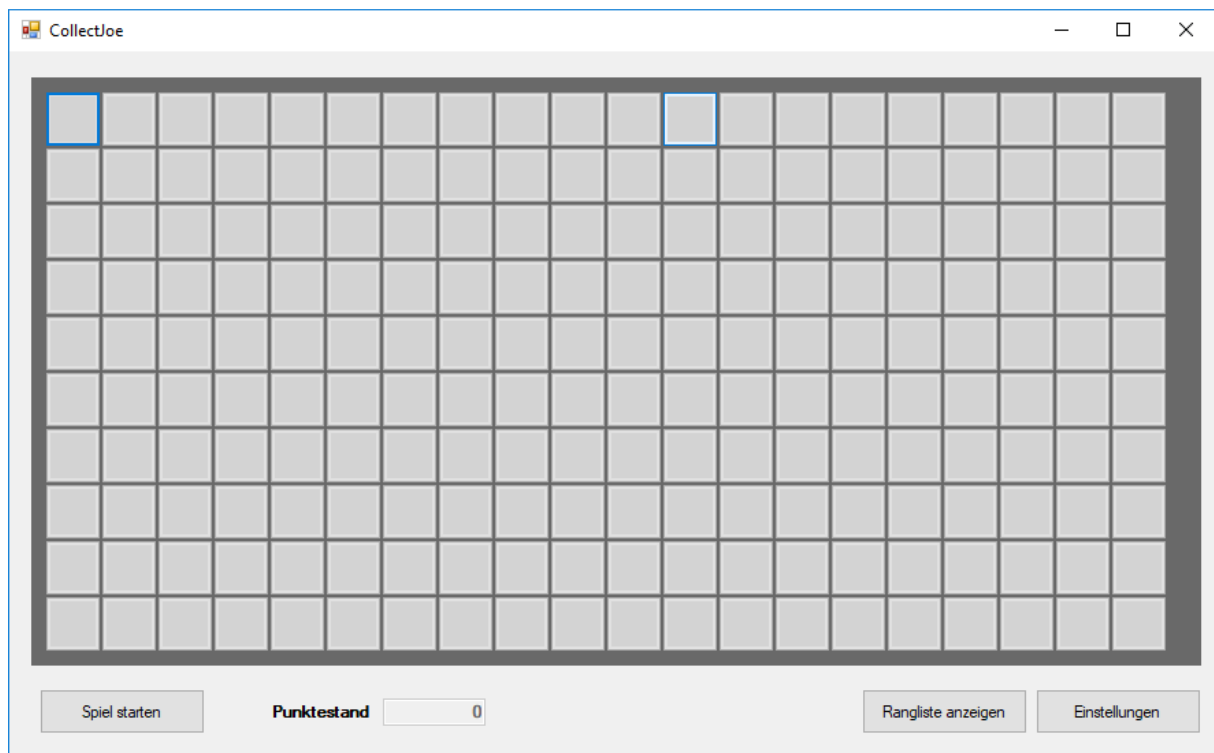
Sichtbarkeitsoperatoren stehen in geschweiften Klammern z.B. {public}.

Parameter und Rückgabetypen stehen in runden Klammern. Zuerst der Parameter und anschliessend nach einem Pfeil der Rückgabotyp z.B. (string)→(int)

Projektübersicht

Spielfeld

«Collect Joe» ist ein Spiel, bei dem es darum geht, durch abschiessen der aufleuchtenden Boxen, so viele Punkte wie möglich zu sammeln.



In der Standardeinstellung geben die blauen Boxen 1en Punkt, die grünen 10 Punkte und die roten - 50 Punkte. Immer nach 100 Millisekunden leuchtet eine beliebige Box auf dem Spielfeld in einer beliebigen Farbe auf. In dieser Zeit kann die Box angeklickt werden und man erhält die jeweiligen Punkte für deren Abschuss.

Einstellungen

Mit dem Button «Einstellungen» wird die Form «frmOptions» geöffnet. Das Fenster erscheint immer im Vordergrund und muss geschlossen werden, bevor Du mit dem Spiel beginnen kann.

The screenshot shows a Windows-style dialog box titled 'Optionen' with a close button (X) in the top right corner. The dialog is divided into three main sections: 'Spielfeld', 'Wertungen', and 'Farben'.
- The 'Spielfeld' section contains two input fields: 'Boxen horizontal' with the value '20' and 'Boxen vertikal' with the value '10'.
- The 'Wertungen' section contains three input fields labeled '1. Boxtyp', '2. Boxtyp', and '3. Boxtyp' with values '1', '10', and '-50' respectively.
- The 'Farben' section contains three color selection boxes: 'Boxen' (light gray), 'Spielfeld' (dark gray), and three 'Boxtyp' entries (1: blue, 2: green, 3: red).
At the bottom left of the dialog is a button labeled 'Übernehmen'.

Der Dialog «Optionen» beinhaltet drei «Gruppenboxen» die die einzelnen zu wählenden Bereiche abgrenzt.

Im Bereich «Spielfeld» kannst Du die Anzahl Boxen in horizontaler und vertikaler Richtung beeinflussen. Hier sollen horizontal maximal 20 Boxen, vertikal maximal 10 Boxen angezeigt werden können.

Im Bereich «Wertungen» bestimmst Du, für welche Feldfarbe Du wie viele Punkte addieren möchtest.

Im Bereich «Farben» kannst Du die Farbe der Boxen, des Spielfeldhintergrundes und der drei aufleuchtenden «Boxtypen» bestimmen.

Mit dem «Übernehmen» Button werden die Änderungen aktiv.

Rangliste

Wenn Du auf der Form «frmField» den Button «Rangliste anzeigen» anklickst, dann öffnet sich das Form Rangliste «frmScoreList».

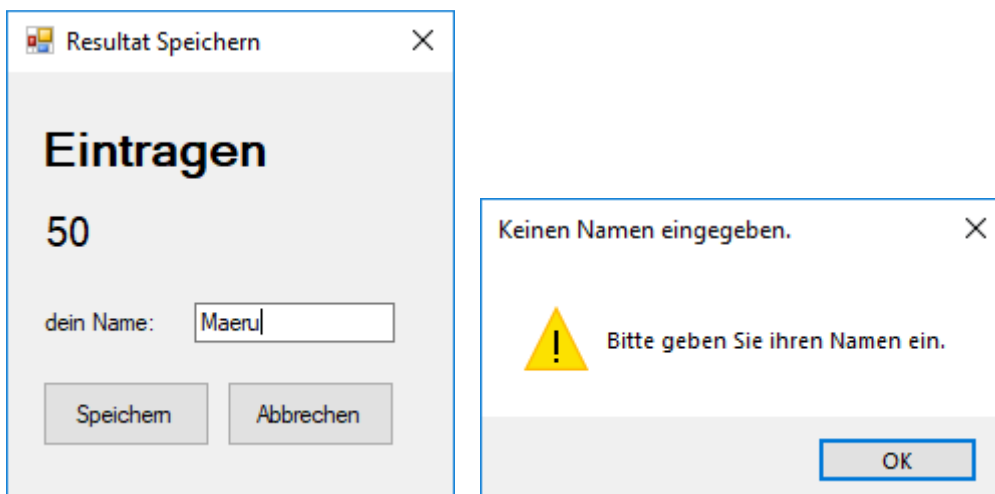


The screenshot shows a Windows-style dialog box titled 'Rangliste' with a close button (X) in the top right corner. The main area of the dialog contains a list of player names and their scores, displayed in a monospaced font: 'Looser;0', 'Maeru;80', 'Spideman;11', 'Semi Looser;2', and 'Samy;0'. To the right of the list is a vertical scrollbar. At the bottom of the dialog is a button labeled 'Schliessen'.

Auf diesem Formular siehst Du die Rangliste der bisher gespielten Spiele. Mit «Schliessen» kannst Du das Form wieder schliessen.

Score Eintragen

Wenn Du das Spiel fertig gespielt hast, dann erscheint das Dialogfeld «frmEditScore».



The screenshot shows two overlapping dialog boxes. The larger one on the left is titled 'Resultat Speichern' and has a close button (X) in the top right corner. It contains the title 'Eintragen' and a large text field displaying the number '50'. Below this, there is a label 'dein Name:' followed by a text input field containing the name 'Maeru'. At the bottom are two buttons: 'Speichern' and 'Abbrechen'. The smaller dialog box on the right is titled 'Keinen Namen eingegeben.' and has a close button (X) in the top right corner. It features a yellow warning triangle icon with an exclamation mark and the text 'Bitte geben Sie ihren Namen ein.' At the bottom is an 'OK' button.

Auf dieser Form siehst Du den erreichten Score. Du kannst deinen Namen eintragen und Speichern. Wenn kein Name angegeben wird, erscheint die [MessageBox] aus der rechten Abbildung.

Umsetzungsanleitung

Bitte halte dich bei der Umsetzung an die Vorgaben. Sind die Namen vergeben, musst Du diese zwingend so benennen. Die wichtigen Namen und Objekttypen stehen zwischen «Anführungszeichen».

Zustandsdiagramm

Zeichne ein Zustandsdiagramm für die Applikation «CollectJoe». Wie viele Zustände hast Du? Wie sind die Übergänge von einem Zustand zum nächsten?

Achtung: Du musst dein Zustandsdiagramm am Ende der ersten Doppellektion als Datei, mit deinem Namen versehen, abgeben.

Beispiel: MustermannMax.gif oder MustermannMax.vsd

Wenn Du das Diagramm von Hand zeichnest bitte fotografieren und speichern.

Formulare

Das Programm besteht aus vier Formularen. Als erstes wollen wir die Formulare aufbauen. Benenne die Formulare genau so, wie sie im Titel angegeben werden. Wie Du ein neues Formular in dein Projekt einbindest, findest Du in den Kursunterlagen beschrieben. Benenne **jede** Komponente auf den Formularen mit einem passenden Namen. Halte dich dabei an die Richtlinien aus dem Anhang dieses Dokuments.

Spielfeld Form «frmField»

Erstelle ein neues C# Windows Forms Projekt und speichere es unter dem Namen «CollectJoe». Baue das Spielfeld auf wie in Abbildung 1.

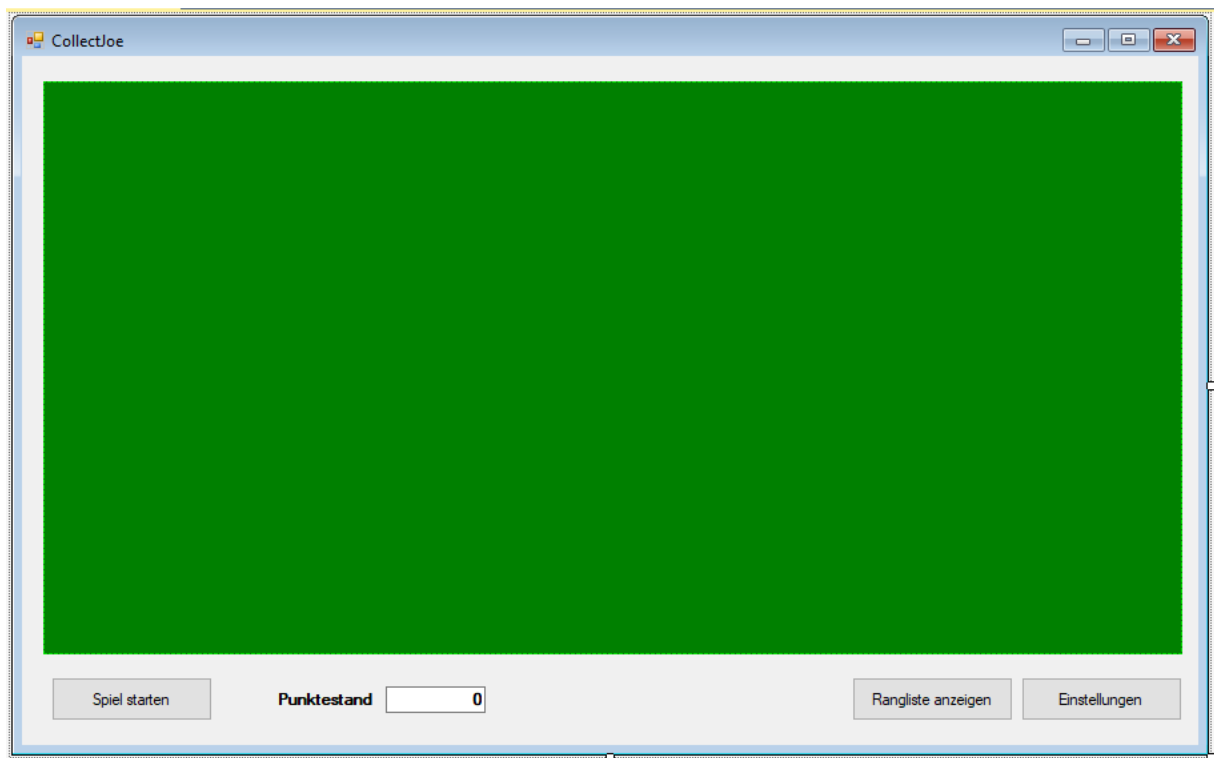


Abb. 1

Benenne **jede** Komponente auf deinem Spielfeldform mit einem passenden Namen. Halte dich dabei an die Richtlinien aus dem Anhang dieses Dokuments.

Verwende für die grüne Fläche ein [Panel]. Der Punktestand ist ein [Label]. Für den Punktezähler kannst Du entweder ein [Label] oder eine [TextBox] mit (Enabled = false) verwenden. Die restlichen Elemente sind vom Typ [Button].

Ziehe zusätzlich einen [Timer] auf dein Spielfeld. Diesen brauchst Du um die einzelnen Boxen aufzudecken und um das Spiel zu terminieren.

Optionen Form «frmOptions»

Füge zur Spielfeld Form ein neues Formular hinzu und benenne es um in «frmOptions». Baue das Optionen-Form auf wie in Abbildung 2.

Abb. 2


Das Form selbst hat die Attribute (FormBorderStyle = FixedSingle), (MaximizeBox = false) und (MinimizeBox = false), damit man das Form in seiner Grösse nicht mehr verändern kann.

Benenne **jede** Komponente auf deiner Optionsform mit einem passenden Namen. Halte dich dabei an die Richtlinien aus dem Anhang dieses Dokuments.

Verwende für die einzelnen umrandeten Bereiche eine [GroupBox] (eine für «Spielfeld», eine für «Wertungen» und eine für «Farben»). Die Eingabefelder sind [Textboxes] Du kannst aber auch ein [NumericUpDown] verwenden, die Beschriftungen [Labels]. Die Farbauswahlfelder auf der rechten Seite sind alles [Buttons] mit den Attributen (Text = leer) und (BackColor) jeweils einer unterschiedlichen Farbe. «Übernehmen» ist ebenfalls ein [Button], der gleich gross ist wie die Buttons auf der Spielfeld-Form. Setze zusätzlich einen [ColorDialog] auf dein Formular. Diesen brauchst Du bei der Programmierung für die einzelnen Farb-Buttons.

Ranglisten Form «frmScoreList»

Füge zur Spielfeld Form ein neues Formular hinzu und benenne es um in «frmScoreList». Baue das Ranglisten-Form auf wie in Abbildung 3.



The screenshot shows a Windows form titled 'Rangliste'. The form has a title bar with a close button (X). The main content area has a large label 'Rangliste' at the top. Below the label is a text box containing the following text: 'Looser;0', 'Maeru;80', 'Spiderman;11', 'Semi Looser;2', and 'Samy;0'. The text box has a vertical scrollbar on the right side. At the bottom of the form is a button labeled 'Schliessen'.

Abb. 3

Das Form selbst hat die Attribute (FormBorderStyle = FixedSingle), (MaximizeBox = false) und (MinimizeBox = false), damit man das Form in seiner Grösse nicht mehr verändern kann.

Benenne **jede** Komponente auf deinem Ranglistenform mit einem passenden Namen. Halte dich dabei an die Richtlinien aus dem Anhang dieses Dokuments.

Die Überschrift ist ein [Label]. Die Rangliste selbst ist eine [TextBox] mit den Attributen (Multiline = true) und (ScrollBars = vertical), Du kannst aber auch eine passendere Komponente verwenden, wenn Du ein findest. «Schliessen» ist ein [Button].

Spielstand speichern «frmEditScore»

Füge zur Spielfeld Form ein neues Formular hinzu und benenne es um in «frmEditScore». Baue das Spielstand-Form auf wie in Abbildung 4.

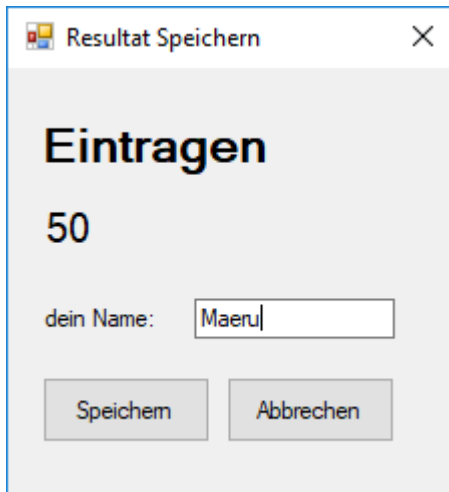


Abb. 4

Das Form selbst hat die Attribute (FormBorderStyle = FixedSingle), (MaximizeBox = false) und (MinimizeBox = false), damit man das Form in seiner Grösse nicht mehr verändern kann.

Benenne **jede** Komponente auf deinem Spielstand-eintragen-form mit einem passenden Namen. Halte dich dabei an die Richtlinien aus dem Anhang dieses Dokuments.

Die Überschrift, der Punktestand darunter und «dein Name:» sind [Label]. Das Eingabefeld für den Namen ist eine [TextBox]. «Speichern» und «Abbrechen» sind [Buttons].

Programmierung

Hier findest Du die Anleitungen für die Programmierung der Funktionalität.

Alle grau ausgewiesenen Funktionsnamen kannst Du selber wählen, resp. ergeben sich durch deine Namenswahl bei den Komponenten. Alle anderen im Titel ausgewiesenen Methodennamen bitte so verwenden.

Optionen Form «frmOptions»

Methode «GetHorizontal» {public} () → (int)

Erstelle eine neue Methode «GetHorizontal». Diese Methode soll den Wert aus der [TextBox] für «Boxen horizontal» als [int] zurückgeben. Programmiere diese Methode auch, wenn Du ein [NumericUpDown] verwendet hast. Dabei sind folgende Regeln zu beachten:

Bedingung	Return Wert
Wert keine Zahl oder Leer	20
Wert > 20	20
Wert < 1	10
Sonst	Wert zurück geben

Methode «GetVertical» {public} () → (return int)

Erstelle eine neue Methode «GetVertical». Diese Methode soll den Wert aus der [TextBox] für «Boxen vertikal» als [int] zurückgeben. Programmiere diese Methode auch, wenn Du ein [NumericUpDown] verwendet hast. Dabei sind folgende Regeln zu beachten:

Bedingung	Return Wert
Wert keine Zahl oder Leer	10
Wert > 10	10
Wert < 1	10
Sonst	Wert zurück geben

Methode «GetValue» {public} (string) → (int)

Erstelle eine neue Methode «GetValue». Diese Methode soll den Wert aus den [TextBoxen] für «1. Boxtyp» - «3.Boxtyp» als [int] zurückgeben. Welcher «Boxtyp» ausgelesen wir bestimmst Du, in dem Du den Namen der [TextBox] als String-Parameter der Methode mitgibst.

Beispiel für den Aufruf: `_deineVariable.GetValue("txtValueType1");`

Für den Return-Wert sind folgende Regeln zu beachten:

Bedingung	Return Wert
Keine TextBox (Komponente nicht gefunden)	1
Wert keine Zahl oder Leer	1
Sonst	Wert zurück geben

TIP 1: Die Komponente zum im Parameter übertragenen Namen findest Du mit:

```
Control[] controls = this.Controls.Find(deinParameter, true);
```

TIP 2: Die gefundene Komponente in eine «Textbox» umwandeln machst Du mit.

```
activeTextBox = (TextBox)controls[0];
```

Methode «GetColor» {public} (string) → (Color)

Erstelle eine neue Methode «GetColor». Diese Methode soll das Attribut (BackColor) aus den Buttons der Farben (im Optionsform auf der rechten Seite) als [Color] zurückgeben. Welcher Button ausgelesen wir bestimmst Du, indem Du den Namen des Buttons als String-Parameter der Methode mitgibst.

Beachte die Tipps unter Methode «GetValue» und adaptiere sie für Button-Komponenten.

EventHandler «BtnColor_Click»

Wenn Du auf einen der Farbbuttons auf der rechten Seite klickst, dann soll der [ColorDialog] geöffnet werden. Wenn der [ColorDialog] mit dem [Button] «OK» geschlossen wird, schreibst Du die gewählte Farbe in das Attribut (BackColor) des aufrufenden Buttons.

TIP 1: Erstelle ein [click-Event] mit Doppelklick auf einen Farbbutton. Passe den Namen des Events an und verbinde das Event neu mit dem Button.

TIP 2: Für die Verwendung des [ColorDialogs] google nach «ColorDialog» + C#.

EventHandler «BtnUse_Click»

Dies ist das Klick-Event für den «Übernehmen»-[Button]. Überprüfe, ob alle ausgewählten Farben eindeutig sind (es darf nicht mehrmals z.B. die Farbe «Rot» ausgewählt werden). Wenn die Farben unterschiedlich sind, wir das Formular versteckt (this.Hide()). Wenn nicht, erscheint eine [MessageBox] die dich auf die fehlerhafte Eingabe hinweist.

EventHandler «FrmOptions_FormClosing»

Wir das X auf der Options-Form geklickt, darf das Formular nicht geschlossen werden. Es muss versteckt werden. Dazu musst Du dieselben Prüfungen wie beim «BtnUse_Click» vornehmen. Setze zusätzlich nach dem (this.Hide()) noch ein (e.Cancel = true), damit die Standard-Methode (Close()) unterbrochen wird.

Lagere den Code aus dem «BtnUse_Click» Handler aus in eine separate Methode, damit Du diesen Code nicht zweimal schreiben musst.

Ranglisten Form «frmScoreList»

Attribute

Erstelle ein privates Attribut vom Datentyp [string] zur Übertragung des Speicherpfades der «HighScore»-Datei. Beachte dabei die Benennungsrichtlinien im Anhang.

Konstruktor «frmScoreList» (string) ➔ ()

Passe den Konstruktor so an, dass er einen [string] Parameter mit dem «ScorePfad» entgegennimmt. Setze nach der «InitializeComponent»-Methode das «HighScore»-Speicherpfad Attribut auf den übergebenen «ScorePfad».

Methode «RefreshScore» {public} () ➔ ()

Erstelle eine Methode «RefreshScore» mit der Du das «HighScore»-File auslesen und in die «Rangliste» [Textbox] schreiben kannst. Wenn das Attribut mit dem Speicherpfad der Rangliste leer ist schreibst Du «Rangliste nicht verfügbar.» in die [Textbox].

TIP: Schau dir die Methode «ReadAllText» aus der Klasse «File» an.

EventHandler «BtnClose_Click»

Dies ist das Klick-Event für den «Schliessen»-[Button]. Rufe nur die Methode (this.Hide()) für diese Form auf.

EventHandler «BtnScoreList_Closing»

Dieser Handler wird aufgerufen, wenn die Form über das X geschlossen wird. Rufe die Methode (this.Hide()) für diese Form auf und stoppe die Verarbeitung anschliessend mit (e.Cancel = true).

Spielstand speichern «frmEditScore»

Attribute

Erstelle ein privates Attribut vom Datentyp [string] zur Übertragung des Speicherpfades der «HighScore»-Datei. Beachte dabei die Benennungsrichtlinien im Anhang.

Konstruktor «frmScoreList» {public} (string) → ()

Passe den Konstruktor so an, dass er einen [string] Parameter mit dem «ScorePfad» entgegennimmt. Setze nach der «InitializeComponent»-Methode das «HighScore»-Speicherpfad Attribut auf den übergebenen «ScorePfad».

Methode «SetScore» {public} (string) → ()

Erstelle eine Methode «SetScore» die den Parameter Score als [string] übernimmt. In der Methode speicherst Du den übergebenen Score in das dazugehörige [Label].

Wird verwendet vor dem öffnen des Forms in «frmField».

Methode «ReSetNameAndScore» {public} () → ()

Erstelle eine Methode «ReSetNameAndScore» die dir die [Textbox] und das «Score»-[Label] mit einem leeren String initialisiert.

Wird verwendet beim Starten des Spiels in «frmField».

EventHandler «btnSave_Click»

Dies ist das Klick-Event für den «Speichern-Button». Überprüfe, ob der Name eingegeben wurde. Wenn nicht öffne eine [MessageBox] mit dem Text «Bitte geben Sie ihren Namen ein.», dem Titel «Keinen Namen eingegeben.» einem «OK» Button für die Bestätigung und dem «MessageIcon» (Exclamation). Wurde der Name eingegeben, Speichere Namen und Score mit «;» getrennt und einem (NewLine) am Ende des Eintrags, in die «HighScore» Datei. Verstecke die Form mit der Methode (this.Hide()).

TIP: Schau dir die Methode «AppendAllText» aus der «File» Klasse an. Eine neue Zeile fügst Du mit «Environment.NewLine» ein.

EventHandler «btnCancel_Click»

Dies ist das Klick-Event für den «Abbrechen»-[Button]. Rufe nur die Methode (this.Hide()) für diese Form auf.

EventHandler «frmEditScore_FormClosing»

Dieser Handler wird aufgerufen, wenn die Form über das X geschlossen wird. Rufe die Methode (this.Hide()) für diese Form auf und stoppe die Verarbeitung anschliessend mit (e.Cancel = true).

Spielfeld Form «frmField»

Attribute

Deklariere in deiner Klasse «frmField» folgende Attribute mit den dazugehörigen Datentypen. Verwende als Attributnamen einen Namen, der das Attribut identifiziert. Alle Attribute sind explizit als privat zu deklarieren. Beachte die Namensgebungsrichtlinien aus dem Anhang.

Zum Beispiel das Attribut in der Zeile 1:

```
private frmOptions _fieldOptions = new frmOptions();
```

Attribut-ID	Beschreibung	Datentyp	Initialisierung
1	<i>Optionen Form</i>	frmOptions	new frmOptions()
2	<i>HighScore Eintragen Form</i>	frmEditScore	
3	<i>HighScore Anzeigen Form</i>	frmScoreList	
4	<i>Zufallszahlengenerator für zufällige Anzeigefabre</i>	Random	new Random()
5	<i>Dauer eines einzelnen Spieles</i>	int	39999
6	<i>Bis jetzt gespielte Zeit in diesem Spiel</i>	int	
7	<i>Pfad zur HighScore Datei</i>	string	
8	<i>Anzahl Boxen in der Horizontalen</i>	int	
9	<i>Anzahl Boxen in der Vertikalen</i>	int	
10	<i>Höhe einer Box</i>	int	
11	<i>Breite einer Box</i>	int	
12	<i>Farbe der Box</i>	Color	
13	<i>Liste mit allen Spielfeld-Boxen</i>	List<Button>	new List<Button>()
14	<i>Index des zuletzt aufgedeckten Buttons in der Liste</i>	int	0
15	<i>Dictionary mit den Farben und dem Wert der aufleuchtenden Boxen</i>	Dictionary<Color, int>	new Dictionary<Color, int>()
16	<i>Äusserer Abstand vom Spielfeldpanel bis zur ersten Box von oben/ unten, rechts/links.</i>	Int[]	new int[] {10,10}

Methode «SetOptions» {public} () ➔ ()

Die Methode «SetOptions» wird immer dann aufgerufen, wenn Du die Optionsparameter aus dem Form «Options» übernehmen möchtest. Es werden folgende Aufgaben in dieser Methode behandelt:

Farbe des Spielfelds setzen

Rufe die Methode «GetColor» aus der «frmOptions»-Klasse auf und speichere den Rückgabewert in das Attribut «BackColor» der Komponente für das Spielfeld-Panel.

Anzahl Buttons auf dem Spielfeld setzen (IDs 8 und 9)

Rufe die Methode «GetHorizontal» aus der «frmOptions»-Klasse auf und speichere den Rückgabewert in das Attribut «Anzahl Boxen in der Horizontalen» (ID=8).

Rufe die Methode «GetVertical» aus der «frmOptions»-Klasse auf und speichere den Rückgabewert in das Attribut «Anzahl Boxen in der Vertikalen» (ID=9).

Höhe und Breite der Buttons setzen (IDs 10 und 11)

Berechne die Höhe der einzelnen Boxen indem Du die Höhe des Spielfeldes minus «den äusseren Abstand» (ID=16) * 2 (Oben und Unten Position [0]) dividiert durch die «Anzahl Boxen in der Vertikalen» (ID=9). Speichere die Höhe ins Attribut «höhe einer Box» (ID=10).

Berechne die Breite der einzelnen Boxen indem Du die Breite des Spielfeldes minus «den äusseren Abstand» (ID=16) * 2 (Links und Rechts Position [1]) dividiert durch die «Anzahl Boxen in der Horizontalen» (ID=8). Speichere die Breite ins Attribut «Breite einer Box» (ID=11).

Farbe der Box-Buttons setzen

Rufe die Methode «GetColor» aus der «frmOptions»-Klasse auf und speichere den Rückgabewert in das Attribut «Farbe der Box» (ID=12).

Farben und Wertigkeiten auf den aufgedeckten Box-Buttons setzen

Leere das Attribut «Dictionary mit den Farben und dem Wert der aufleuchtenden Boxen» (ID=15).
TIP: Clear-Methode.

Rufe die Methoden «GetColor» und «GetValue» aus der «frmOptions»-Klasse auf und speichere den Rückgabewert in das Attribut «Dictionary mit den Farben und dem Wert der aufleuchtenden Boxen» (ID=15).

Die Farbe ist dabei der «Key» des Dictionary und der «Wert» ist das «Value» des Dictionary.
Wiederhole dies für «1. Feldtyp» bis «3. Feldtyp».

Methode «StopGame» {public} () → ()

Diese Methode wird aufgerufen, wenn Du das Game beenden musst.

Stoppe als erstes den Timer, damit keine neuen Felder mehr aufgedeckt werden. Wenn der Wert in der [TextBox] «Punktestand» kleiner ist als 0, dann gib eine [MessageBox] mit dem Text «GAME OVER!» oder etwas Ähnlichem aus. Die [MessageBox] soll einen Button «OK» haben und das [MessageBoxIcon] (Hand) haben. Ist der «Punktestand» grösser oder gleich 0, soll der Punktestand auf das Form (ID=2) übergeben werden. Verwende dazu die zuvor erstellte Methode «SetScore». Öffne jetzt das Form. Verwende dazu die Methode (ShowDialog), damit es modal (immer im Vordergrund) erscheint.

EventHandler «BtnBox_Click»

Erstelle einen neuen Event-Handler für alle Box-Buttons.

Lies die (BackColor) des aktiven Buttons aus und schaue, ob der «Dictionary mit den Farben und dem Wert der aufleuchtenden Boxen» (ID=15) diese Farbe enthält. Wenn ja, zähle den Wert aus dem Dictionary zu deinem Wert in der [TextBox] «Punktestand» dazu.
Ist der Wert in der [TextBox] negativ, dann rufe die [Methode] «StopGame» auf.

Methode «BuildButtonField» {public} () → ()

Erstelle eine Methode, die dir dynamisch alle Buttons auf dem Spielfeld erstellt. Verwende dazu zwei verschachtelte for-Schleifen.

Lösche zuerst das «Spielfeld-Panel» und die «Liste mit allen Spielfeld-Boxen» (ID=13).

TIP: Verwende die Methode (Clear()).

Setze für jeden Button folgende Attribute:

Attribut	Wert
Name	«btn_»+[0-n]
Height	ID=10 aus der vorhergehenden Liste
Width	ID=11 aus der vorhergehenden Liste
Location	Position im Spielfeld
Text	« »
BackColor	ID=12 aus der vorhergehenden Liste
Click event	Bereits erstelltes Event «BtnBox_Click»

Füge den neuen Button an das Panel pnlField an. Füge den neuen Button zur Liste (ID=13) hinzu.

EventHandler «BtnHideOptions_Click»

Dieser EventHandler soll ausgeführt werden, wenn das «Options»-[Form] entweder aus- oder eingeblendet wird. Wir wollen nur auf das ausgeblendet reagieren (sender.Visible == false).

Rufe an dieser Stelle die Methode «SetOptions» auf, damit die Optionen in die aktuelle Form übernommen werden. Baue das Box-Button-Feld noch einmal neu auf, indem du die Methode «BuildButtonField» aufrufst.

Konstruktor «frmField»

Erweitere deinen Konstruktor nach der Methode «InitializeComponent» um die folgenden Aufgaben:

Initialisiere High-Score Pfad

Initialisiere das zuvor deklarierte Attribut «Pfad zur HighScore Datei» (ID=7) mit dem Pfad, an dem deine Applikation liegt + dem Filenamen «highscore.txt». Gehe dazu vom BaseDirectory aus. Siehe TIP 1. Wenn das File noch nicht existiert, dann erstelle es mit einem leeren String drin. Siehe dazu TIP 2.

TIP 1: Den Applikationspfad findest Du mit dem Stichwort «BaseDirectory». Wenn Du keine weiteren Einstellungen vornimmst und die Applikation im DEBUG-Modus kompilierst, findest du die Datei im bin/Debug Verzeichnis. Siehe auch HILFREICHE LINKS. Mit der Methode «Path.Combine» kannst Du Pfadstücke zusammensetzen, ohne dass Du dich um die / (Slashes) kümmern musst.

TIP 2: Suche nach der Klasse «File» (System.IO). Darunter findest Du eine Methode «Exists» und «WriteAllText».

HILFREICHE LINKS:

1. <http://www.anotherchris.net/csharp/6-ways-to-get-the-current-directory-in-csharp/>
2. Google nach «Path.Combine» und C#

Initialisiere High-Score Forms mit High-Score Pfad

Initialisiere die beiden Score Forms «frmEditScore» (ID=2) und «frmScoreList» (ID=3) indem Du deren Constructor mit dem Attribut «ScorePath» (ID=7) aufrufst.

TIP: Der Konstruktor wird aufgerufen, wenn Du new der Klasse machst.

Spielfeld aufbauen

Rufe die Methode «SetOptions» auf um die Optionen aus der «frmOptions» zu übertragen.

Rufe die Methode «BuildButtonField» auf, um die Buttons auf das «Spielfeld»-[Panel] zu zeichnen.

Zusätzliche Events anhängen

Damit Du die Optionen direkt beim schliessen der Options-Form übernehmen kannst, musst du dem «Options» Form (ID=1) noch den zuvor erstellten [EventHandler] «BtnHideOptions_Click» anhängen.

EventHandler «BtnOptions_Click»

Öffne das «Optionen»-[Form] mit der Methode (ShowDialog), damit die Form modal dargestellt wird.

EventHandler «BtnStart_Click»

Dieser Handler startet das Spiel neu. Einträge wie zum Beispiel der «HightScore» die «bis jetzt gespielte Zeit in diesem Spiel» (ID=6), der «Index des zuletzt aufgedeckten Buttons in der Liste» (ID=14) und die [TextBox] «Spielstand» müssen auf ihren Ausgangspunkt zurückgestellt werden. Ebenfalls kannst Du gleich den Namen des Spielers im «HighScore Eintragen»-[Form] (ID=2) auf einen leeren String setzen, damit beim nächsten Eintrag wieder ein leeres Feld erscheint. Zuletzt startest Du den Timer.

EventHandler «TmrGame_Tick»

Erhöhe das Attribut «bis jetzt gespielte Zeit in diesem Spiel» (ID=6) um die seit dem letzten Tick vergangene Zeit. Decke den zuletzt aufgedeckten Button «Index des zuletzt aufgedeckten Buttons in der Liste» (ID=14) wieder zu (weise ihm eine neue (BackColor) zu).

Random Farbe bestimmen

Bestimme mit dem Zufallszahlengenerator einen neuen Wert zwischen 0 und dem höchsten Eintrag deines «Dictionary mit den Farben und dem Wert der aufleuchtenden Boxen» (ID=15).

Randmon Box_Button bestimmen

Bestimme mit dem Zufallszahlengenerator einen neuen Wert zwischen 0 und dem höchsten Eintrag deiner «Liste mit allen Spielfeld-Boxen» (ID=13).

Weise dieser Box die [BackColor] aus «Random Farbe bestimmen» zu.

GameOver für Timer

Prüfe mit Hilfe der «Dauer eines einzelnen Spieles» (ID=5) und «bis jetzt gespielte Zeit in diesem Spiel» (ID=6), ob die Spielzeit abgelaufen ist. Wenn ja, rufe die Methode «StopGame» auf.

EventHandler «BtnScore_Click»

Dieser Handler wird aufgerufen, wenn der Button «Rangliste Anzeigen» geklickt wurde. Rufe die Methode «RefreshScoreList» aus der «HightScore Anzeigen Form» (ID=3) auf. Öffne die Form nun mit der Methode (ShowDialog).

Zusatzpunkte

Du kannst Dir für weitere Funktionalitäten des Spiels Zusatzpunkte holen.

Einstellungen

Es dürfen nicht mehrmals dieselben Farben vergeben werden. Ist dies doch der Fall, erscheint eine [MessageBox] mit einer entsprechenden Warnung.

Die Spieldauer kann im «Optionen»-Form verändert werden. Sie wird auch an die aufrufende Form übergeben und die Prüfung im Timer-Event läuft gegen dieses Attribut.

Rangliste anzeigen

Die Rangliste wird sortiert, so dass der beste Spieler zuoberst erscheint.

Spiel gewonnen

Erreicht der Spieler mit seinem Spiel die «HighScore» Position, dann ertönt eine Fanfare. Diese soll direkt beim öffnen der «Rangliste eintragen»-Form ertönen.

Abschliessende Tipps

Am letzten Tag erscheinen und behaupten, deine GibbiX sei abgestürzt und du hättest nichts mehr, gilt nicht. Als Informatiker kennst Du Tools für Backup und Recovery, oder Versionsverwaltungen.

Sichere dir die jeweils letzte lauffähige Version. Sollten am letzten Tag alle Stricke reißen, kannst Du immer noch diese abgeben und eine anständige Note erhalten.

Anhang

Benennungsrichtlinien

Alle Attribute und Komponenten sind in «Camel Case» (erst Buchstabe klein, dann gross-klein im Wechsel), alle Methoden in «Pascal Case» (erst Buchstabe gross, dann klein-gross im Wechsel) zu erstellen. Dabei sind folgende Präfixe zu verwenden:

Komponenten	Präfixe
Forms	frm
Buttons	btn
Panel	pnl
TextBox	txt
Label	lbl
Timer	tmr
GroupBox	gpx
NumericUpDown	nud

Variablen und Konstanten	Präfixe
Attribute	<i>_ (Underscore)</i>
Konstanten	<i>DEINKONSTANTENNAME</i>
Variablen	<i>deinVariablenName</i>