



4. Nach der Methode `buttonRegistrieren_Click(object, sender, EventArgs e)`
- Das Passwort muss 8 Zeichen lang sein, sonst gibt es eine Fehlermeldung.
  - Sollte die Registrierung erfolgreich sein, wird `ViewRegistrierBestaetigung()` aufgerufen.

```
15  
16 private void buttonRegistrieren_Click(object sender, EventArgs e)  
17 {  
18     errorProvider.Clear();  
19     if(textBoxPasswort.Text.Length > 7)  
20     {  
21         if(controller.userRegistrierung(textBoxBenutzername.Text, textBoxPasswort.Text))  
22         {  
23             Form registrierBestaetigung = new ViewRegistrierBestaetigung();  
24             registrierBestaetigung.Show();  
25         }  
26     }  
27     else  
28     {  
29         errorProvider.SetError(buttonRegistrieren, "Fehler beim erstellen des Benutzers!");  
30     }  
31 }  
32 else  
33 {  
34     errorProvider.SetError(textBoxPasswort, "Das Passwort muss mindestens 8 zeichen lang sein!");  
35 }  
36 }
```

5. Nach der Methode `buttonAnmelden_Click(object, sender, EventArgs e)`
- Sollten die Daten nicht mit denen in der Datenbank übereinstimmen, gibt es eine Fehlermeldung.

```
40  
41 private void buttonAnmelden_Click(object sender, EventArgs e)  
42 {  
43     errorProvider.Clear();  
44     if (textBoxPasswort.Text.Length < 7)  
45     {  
46         errorProvider.SetError(textBoxPasswort, "Das Passwort muss mindestens 8 zeichen lang sein!");  
47     }  
48     else  
49     {  
50         if (controller.userAnmeldung(textBoxBenutzername.Text, textBoxPasswort.Text))  
51         {  
52             gameOverview.Show();  
53         }  
54         else  
55         {  
56             errorProvider.SetError(buttonAnmelden, "Anmeldung ist fehlgeschlagen!");  
57         }  
58     }  
59 }
```

6. Nach der Methode `pressingEnter(object, sender, KeyEventArgs e)`
- Die Methode `buttonAnmelden_Click(object, sender, EventArgs e)` wird aufgerufen.

```
61  
62 private void pressingEnter(object sender, KeyEventArgs e)  
63 {  
64     if (e.KeyValue == (char)Keys.Enter)  
65     {  
66         buttonAnmelden_Click(sender, null);  
67     }  
68 }  
69 }
```

7. Nach der Methode `userRegistrierung(string benutzername, string password)`
- Sollte ein Benutzer noch nicht existieren, wird ein salt generiert, das Passwort gehashed und all diese Werte an die Datenbank übergeben.

```
19  
20 public bool userRegistrierung(string benutzername, string password)  
21 {  
22     if (!model.benutzerExistiert(benutzername))  
23     {  
24         string salt = model.generiereSalt();  
25         string hashedPassword = model.hashPassword(password, salt);  
26         return model.userInDatenbankEinfuegen(benutzername, hashedPassword, salt);  
27     }  
28     return false;  
29 }  
30  
31  
32 public bool userAnmelden(string benutzername, string password)
```

## 8. Nach der Methode `userAnmeldung(string benutzername, string passwort)`

- a. Sollte der User in der Datenbank existieren, wird dieser angemeldet und kann spielen.

```
33 public bool userAnmeldung(string benutzername, string passwort)
34 {
35     if (model.benutzerExistiert(benutzername))
36     {
37         User user = model.userAusDatenbankLesen(benutzername);
38         if (user != null)
39         {
40             if (model.hashPasswort(passwort, user.Salt) == user.HashPasswort)
41             {
42                 ControllerGame.userData = user;
43                 return true;
44             }
45         }
46         else
47         {
48             return false;
49         }
50
51         return true;
52     }
53     else
54     {
55         return false;
56     }
57 }
58
```