

C:\Users\Error\Desktop\JavaC#Difference\C#\GameManager.txt

```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4
5 public class GameManager
6 {
7
8     //Registriert den player (packt ihn in die players list)
9     public void RegisterPlayer(bool isHuman, String name)
10     {
11         Player player = new Player(isHuman, name);
12         players.Add(player);
13     }
14
15     //muss die runde wiederholt werden?
16     public int GetZeichenBenutzt()
17     {
18         bool[] zeichenMakierer = new bool[Enum.GetNames(typeof(Zeichen)).Length];
19         List<Zeichen> zeichenBenutzt = new List<Zeichen>();
20         int VerschiedeneZeichenBenutzt = 0;
21
22         //makiert im bool array welche enum Zeichen benutzt worden sind.
23         for (int j = 0; j < players.Count; j++)
24         {
25             if (players[j].m_stillPlaying)
26             {
27                 zeichenMakierer[(byte)players[j].getZeichen()] = true;
28             }
29         }
30
31         //loop durch das array um zu gucken ob irgend ein zeichen nicht gewählt worden ist
32         for (int j = 0; j < zeichenMakierer.Length; j++)
33         {
34             if (zeichenMakierer[j] == true)
35             { VerschiedeneZeichenBenutzt++; }
36         }
37         return VerschiedeneZeichenBenutzt;
38     }
39
40     //Wer hat das stärkere zeichen gewählt? Wer ist diese runde ausgeschieden?
41     public void Evaluierung()
42     {
43         List<Zeichen> zeichenBenutzt = new List<Zeichen>();
44         bool[] zeichenBenutztBool = new bool[Enum.GetNames(typeof(Zeichen)).Length];
45
46         //makiert im array welche enum Zeichen benutzt worden sind.
47         for (int j = 0; j < players.Count; j++)
48         {
49             if (players[j].m_stillPlaying)
50             {
51                 zeichenBenutztBool[(byte)players[j].getZeichen()] = true;
52             }
53         }
54     }
55 }
```

C:\Users\Error\AppData\Local\Temp__25F6.tmp

```
1 import java.util.ArrayList;
2 import java.util.List;
3 import java.util.Scanner;
4
5 public class GameManager
6 {
7     //Player input string to number converter
8     public static int tryParse(String value, int defaultVal)
9     {
10         try
11         {
12             return Integer.parseInt(value);
13         }
14         catch (NumberFormatException e)
15         {
16             return defaultVal;
17         }
18     }
19
20     //Registriert den player (packt ihn in die players list)
21     public void RegisterPlayer(boolean isHuman, String name)
22     {
23         Player player = new Player(isHuman, name);
24         players.add(player);
25     }
26
27     //muss die runde wiederholt werden?
28     public int GetZeichenBenutzt()
29     {
30         boolean[] zeichenMakierer = new boolean[Zeichen.values().length];
31         List<Zeichen> zeichenBenutzt = new ArrayList<Zeichen>();
32         int VerschiedeneZeichenBenutzt = 0;
33
34         //makiert im bool array welche enum Zeichen benutzt worden sind.
35         for (int j = 0; j < players.size(); j++)
36         {
37             if (players.get(j).m_stillPlaying)
38             {
39                 zeichenMakierer[players.get(j).getZeichen().ordinal()] = true;
40             }
41         }
42
43         //loop durch das array um zu gucken ob irgend ein zeichen nicht gewählt worden ist
44         for (int j = 0; j < zeichenMakierer.length; j++)
45         {
46             if (zeichenMakierer[j] == true)
47                 {VerschiedeneZeichenBenutzt++;}
48         }
49         return VerschiedeneZeichenBenutzt;
50     }
51
52
53     //Wer hat das stärkere zeichen gewählt? Wer ist diese runde ausgeschieden?
54     public void Evaluierung()
55     {
56         List<Zeichen> zeichenBenutzt = new ArrayList<Zeichen>();
57         boolean[] zeichenBenutztBool = new boolean[Zeichen.values().length];
58
59         //makiert im array welche enum Zeichen benutzt worden sind.
60         for (int j = 0; j < players.size(); j++)
61         {
62             if (players.get(j).m_stillPlaying)
63             {
64                 zeichenBenutztBool[players.get(j).getZeichen().ordinal()] = true;
65             }
66         }
67     }
```

C:\Users\Error\Desktop\JavaC#Difference\C#\GameManager.txt

```
56 //put alle benutzten zeichen ins zeichenBenutzt (zeichen) list
57 for (int i = 0; i < Enum.GetNames(typeof(Zeichen)).Length; i++)
58 {
59     if (zeichenBenutztBool[i])
60     {
61         zeichenBenutzt.Add((Zeichen)i);
62     }
63 }
64
65 Zeichen unbenutztesZeichen = Zeichen.Stein; //dummy da Java rumheult. Mann kann es
66
67 for (int i = 0; i < Enum.GetNames(typeof(Zeichen)).Length; i++)
68 {
69     if (!zeichenBenutztBool[i])
70     {
71         unbenutztesZeichen = (Zeichen)i;
72     }
73 }
74
75 Zeichen winningZeichen;
76
77 switch (unbenutztesZeichen)
78 {
79     case Zeichen.Schere:
80         winningZeichen = Zeichen.Papier;
81         break;
82     case Zeichen.Stein:
83         winningZeichen = Zeichen.Schere;
84         break;
85     case Zeichen.Papier:
86         winningZeichen = Zeichen.Stein;
87         break;
88     default:
89         winningZeichen = Zeichen.Stein; //Java heult rum das variable may n
90         break;
91 }
92
93 //kick leute die verloren haben aus dieser runde raus.
94 for (int i = 0; i < players.Count; i++)
95 {
96     if (players[i].getZeichen() != winningZeichen)
97     {
98         players[i].m_stillPlaying = false;
99         Console.WriteLine("Player " + players[i].getName() + " hat leider v
100     }
101 }
102
103
104 public void ResetRound()
105 {
106     for (int i = 0; i < players.Count; i++)
107     {
108         players[i].m_stillPlaying = true;
109     }
110     Console.WriteLine("Nähste Runde!");
111 }
112
113 public void Reset()
114 {
115     for (int i = 0; i < players.Count; i++)
116     {
117         players[i].m_stillPlaying = true;
118         players[i].SetPoints(0);
119     }
120 }
121
122 public int GetNumberOfStillPlayingPlayers()
```

C:\Users\Error\AppData\Local\Temp__25F6.tmp

```
68         //put alle benutzten zeichen ins zeichenBenutzt (zeichen) list
69         for (int i = 0; i < Zeichen.values().length; i++)
70         {
71             if (zeichenBenutztBool[i])
72             {
73                 zeichenBenutzt.add(Zeichen.values()[i]);
74             }
75         }
76
77         Zeichen unbenutztesZeichen = Zeichen.Stein; //dummy da Java rumheult. Mann kann es
78
79         for (int i = 0; i < Zeichen.values().length; i++)
80         {
81             if (!zeichenBenutztBool[i])
82             {
83                 unbenutztesZeichen = Zeichen.values()[i];
84             }
85         }
86
87         Zeichen winningZeichen;
88
89         switch (unbenutztesZeichen)
90         {
91             case Schere:
92                 winningZeichen = Zeichen.Papier;
93                 break;
94             case Stein:
95                 winningZeichen = Zeichen.Schere;
96                 break;
97             case Papier:
98                 winningZeichen = Zeichen.Stein;
99                 break;
100             default:
101                 winningZeichen = Zeichen.Stein; //Java heult rum das variable may not be in
102                 break;
103         }
104
105         //kick leute die verloren haben aus dieser runde raus.
106         for (int i = 0; i < players.size(); i++)
107         {
108             if(players.get(i).getZeichen() != winningZeichen)
109             {
110                 players.get(i).m_stillPlaying = false;
111                 System.out.println("Player " + players.get(i).getName() + " hat lei
112             }
113         }
114     }
115
116     public void ResetRound()
117     {
118         for (int i = 0; i < players.size(); i++)
119         {
120             players.get(i).m_stillPlaying = true;
121         }
122         System.out.println("Nächste Runde!");
123     }
124
125     public void Reset()
126     {
127         for (int i = 0; i < players.size(); i++)
128         {
129             players.get(i).m_stillPlaying = true;
130             players.get(i).SetPoints(0);
131         }
132     }
133
134     public int GetNumberOfStillPlayingPlayers()
```

C:\Users\Error\Desktop\JavaC#Difference\C#\GameManager.txt

```
123     {
124         int stillPlayingPlayers = 0;
125         for (int j = 0; j < players.Count; j++)
126         {
127             if (players[j].m_stillPlaying)
128             {
129                 stillPlayingPlayers++;
130             }
131         }
132         return stillPlayingPlayers;
133     }
134
135
136 public void PrintMitspielerZeichen()
137 {
138     for (int j = 0; j < players.Count; j++)
139     {
140         if (players[j].m_stillPlaying)
141         {
142             Console.WriteLine(players[j].getName() + " hat " + players[j].getZeichen());
143         }
144     }
145 }
146 public void PrintMitspielerPunkte()
147 {
148     for (int j = 0; j < players.Count; j++)
149     {
150         Console.WriteLine(players[j].getName() + " hat " + players[j].GetPoints() + " Punkte");
151     }
152 }
153
154
155 public void Run()
156 {
157     for (int i = 1; i < maxRunden + 1; i++)
158     {
159         while (GetNumberOfStillPlayingPlayers() > 1) //while there is more then 1 player
160         {
161             Console.WriteLine("Runde " + i + "!");
162             //Step 1, wähle zeichen
163             for (int j = 0; j < players.Count; j++)
164             {
165                 //Player (Human)
166                 if (players[j].m_stillPlaying)
167                 {
168                     if (players[j].m_isHuman == true)
169                     {
170                         Console.WriteLine(players[j].getName() + " wählt Zeichen");
171                         int PlayerInputInt;
172                         bool ZeichenExistiert = false;
173                         for (int k = 0; k < ZeichenLaenge; k++)
174                         {
175                             Console.WriteLine(k + " = " + (Zeichen[k]));
176                         }
177                         do
178                         {
179                             playerInput = Console.ReadLine();
180                             Int32.TryParse(playerInput, out PlayerInputInt);
181                             ZeichenExistiert = (PlayerInputInt < ZeichenLaenge);
182                         } while (!ZeichenExistiert);
183                         players[j].setZeichen((Zeichen)PlayerInputInt);
184                     }
185                     //Player (NPC)
186                     else
187                     {
188                         var rand = new Random();
189                         int randomIndex = rand.Next(0, ZeichenLaenge);
190                         players[j].setZeichen(Zeichen[randomIndex]);
191                     }
192                 }
193             }
194         }
195     }
196 }
```

C:\Users\Error\AppData\Local\Temp__25F6.tmp

```
135     {
136         int stillPlayingPlayers = 0;
137         for (int j = 0; j < players.size(); j++)
138         {
139             if (players.get(j).m_stillPlaying)
140             {
141                 stillPlayingPlayers++;
142             }
143         }
144         return stillPlayingPlayers;
145     }
146
147
148     public void PrintMitspielerZeichen()
149     {
150         for (int j = 0; j < players.size(); j++)
151         {
152             if (players.get(j).m_stillPlaying)
153             {
154                 System.out.println(players.get(j).getName() + " hat " + players.get(j).getZeichen());
155             }
156         }
157     }
158     public void PrintMitspielerPunkte()
159     {
160         for (int j = 0; j < players.size(); j++)
161         {
162             System.out.println(players.get(j).getName() + " hat " + players.get(j).GetPunkte());
163         }
164     }
165
166     public void Run()
167     {
168         for (int i = 1; i < maxRunden + 1; i++)
169         {
170             while (GetNumberOfStillPlayingPlayers() > 1) //while there is more then 1 player
171             {
172                 System.out.println("Runde " + i + "!");
173                 //Step 1, wähle zeichen
174                 for (int j = 0; j < players.size(); j++)
175                 {
176                     //Player (Human)
177                     if (players.get(j).m_stillPlaying)
178                     {
179                         if (players.get(j).m_isHuman == true)
180                         {
181                             System.out.println(players.get(j).getName() + " wählt " + players.get(j).getZeichen());
182                             int PlayerInputInt;
183                             boolean ZeichenExistiert = false;
184                             for (int k = 0; k < ZeichenLaenge; k++)
185                             {
186                                 System.out.println(k + " = " + Zeichen.get(k));
187                             }
188                             do
189                             {
190                                 playerInput = playerInputScanner.nextInt();
191                                 PlayerInputInt = tryParse(playerInput);
192                                 ZeichenExistiert = (PlayerInputInt < ZeichenLaenge);
193                             }while(!ZeichenExistiert);
194                             players.get(j).setZeichen(Zeichen.values()[PlayerInputInt]);
195                         }
196                     }
197                     //Player (NPC)
198                     else
199                     {
200                         System.out.println(players.get(j).getName() + " wählt " + players.get(j).getZeichen());
201                     }
202                 }
197             }
198         }
199     }
200 }
```

C:\Users\Error\Desktop\JavaC#Difference\C#\GameManager.txt

```
190         int rngZeichen = rand.Next(0, Enum.GetNames<Zeichen>().Length);
191         players[j].setZeichen((Zeichen)rngZeichen);
192     }
193 }
194
195     }
196     PrintMitspielerZeichen();
197
198     int verschiedeneZeichenBenutzt = GetZeichenBenutzt();
199
200     //falls alle zeichen benutzt worden sind wird die runde wiederholt
201     if (verschiedeneZeichenBenutzt == Enum.GetNames<Zeichen>().Length)
202     {
203         Console.WriteLine("Alle zeichen wurden benutzt, runde wird
204             continue;
205     }
206
207     //falls nur 1 zeichen benutzt worden sind wird die runde wiederholt
208     if (verschiedeneZeichenBenutzt == 1)
209     {
210         Console.WriteLine("Das Gleiche zeichen wurde benutzt, runde
211             continue;
212     }
213
214     Evaluierung();
215 }
216 //Give the winning player a point.
217 for (int j = 0; j < players.Count; j++)
218 {
219     if (players[j].m_stillPlaying)
220     {
221         players[j].AddPoint();
222     }
223 }
224 PrintMitspielerPunkte();
225 ResetRound();
226 }
227 }
228
229 //Settings
230 private int maxRunden = 10;
231 int ZeichenLaenge = Enum.GetNames<Zeichen>().Length;
232 //Scanner playerInputScanner = new Scanner(System.in);
233 String playerInput;
234 List<Player> players = new List<Player>();
235 }
```

C:\Users\Error\AppData\Local\Temp__25F6.tmp

```
201                                     int rngZeichen = ((int)(Math.random() * 100
202                                     players.get(j).setZeichen(Zeichen.values()[
203                                     }
204                                     }
205                                     }
206                                     }
207                                     PrintMitspielerZeichen();
208
209                                     int verschiedeneZeichenBenutzt = GetZeichenBenutzt();
210
211                                     //falls alle zeichen benutzt worden sind wird die runde wiederholt
212                                     if (verschiedeneZeichenBenutzt == Zeichen.values().length)
213                                     {
214                                         System.out.println("Alle zeichen wurden benutzt, runde wird
215                                         continue;
216                                     }
217
218                                     //falls nur 1 zeichen benutzt worden sind wird die runde wiederholt
219                                     if (verschiedeneZeichenBenutzt == 1)
220                                     {
221                                         System.out.println("Das Gleiche zeichen wurde benutzt, rund
222                                         continue;
223                                     }
224
225                                     Evaluierung();
226                                     }
227                                     //Give the winning player a point.
228                                     for (int j = 0; j < players.size(); j++)
229                                     {
230                                         if (players.get(j).m_stillPlaying)
231                                         {
232                                             players.get(j).AddPoint();
233                                         }
234                                     }
235                                     PrintMitspielerPunkte();
236                                     ResetRound();
237                                     }
238                                     }
239
240                                     //Settings
241                                     private int maxRunden = 10;
242                                     int ZeichenLaenge = Zeichen.values().length;
243                                     Scanner playerInputScanner = new Scanner(System.in);
244                                     String playerInput;
245                                     List<Player> players = new ArrayList<Player>();
246                                     }
```