

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6 public class gameStatKeeper : MonoBehaviour
7 {
8     /// <summary>
9     /// keeps info about each passport and records player mistakes that
10     may be used when calculating player ranks
11     /// the classes are kinda uselessly complicated but it works
12     /// </summary>
13
14     [SerializeField]
15     private List<statEntry> stats;
16     [SerializeField]
17     private otherStats otherStat;
18     [SerializeField]
19     private List<otherStats> otherStats;
20     [SerializeField]
21     public allTheStats Allstats;
22     [SerializeField]
23     private statEntry entry;
24     //for comparison of the mistake
25     private int allPicMistakes;
26     private int allDateMistakes;
27     private int allNameMistakes;
28     private int allColorMistakes;
29     private int allPersonMistakes;
30     private int allEmbMistakes;
31     private bool doOnce=true;
32     // lang 0 eng 1 fin 2 swe 3 rus
33     public string[] emblemTip;
34     public string[] photoTip;
35     public string[] personTip;
36     public string[] nameTip;
37     public string[] dateTip;
38     public string[] colorTip;
39     public List<mistakeCalcHelper> calcHelpers;
40     public Text tipText;
41     public Text scoreText;
42     private newGameManager newGameManager;
43     private xmlDatabaseManager xml;
44     // Start is called before the first frame update
45     void Start()
46     {
47         newGameManager = GameObject.FindObjectOfType<newGameManager>();
48         xml = GameObject.FindObjectOfType<xmlDatabaseManager>();
49     }
```

```

49
50     // Update is called once per frame
51     void Update()
52     {
53         //hard coded score texts for different languages
54         if (xml.db.language == 0)
55         {
56             scoreText.text = "Score: " + newGameManager.wins;
57         }else
58         if (xml.db.language == 1)
59         {
60             scoreText.text = "Pisteet: " + newGameManager.wins;
61         }
62         else
63         if (xml.db.language == 2)
64         {
65             scoreText.text = "Poäng: " + newGameManager.wins;
66         }
67         else
68         if (xml.db.language == 3)
69         {
70             scoreText.text = "Счет: " + newGameManager.wins;
71         }
72         //update the stats
73         Allstats = calculateStuff(Allstats);
74         if (GameObject.FindObjectOfType<newGameManager>().gameOver != true)
75         {
76             //reset everything to zero so nothing breaks is inefficent
77             doOnce = false;
78             calcHelpers[0].theNumber = 0;
79             calcHelpers[1].theNumber = 0;
80             calcHelpers[2].theNumber = 0;
81             calcHelpers[3].theNumber = 0;
82             calcHelpers[4].theNumber = 0;
83             calcHelpers[5].theNumber = 0;
84
85             //could be more dynamic calculates what was the worst stat was
86             used for giving tips on what the player should improve on
87             for (int i = 0; i < Allstats.otherStats.Count - 1; i++)
88             {
89                 if (Allstats.otherStats[i].wasItCorrect == false &&
90                     Allstats.statEntries[i].colorWasChanged ==
91                     true&&Allstats.otherStats[i].colorWasSelected!=true)
92                 {
93                     calcHelpers[0].theNumber++;
94                 }
95                 if (Allstats.otherStats[i].wasItCorrect == false &&
96                     Allstats.statEntries[i].personWasChanged == true &&

```

```
Allstats.otherStats[i].personWasSelected != true)
93     {
94         calcHelpers[1].theNumber++;
95     }
96     if (Allstats.otherStats[i].wasItCorrect == false &&
        Allstats.statEntries[i].photoWasChanged == true &&
        Allstats.otherStats[i].photoWasSelected != true)
97     {
98         calcHelpers[2].theNumber++;
99     }
100    if (Allstats.otherStats[i].wasItCorrect == false &&
        Allstats.statEntries[i].emblemWasChanged == true &&
        Allstats.otherStats[i].emblemWasSelected != true)
101    {
102        calcHelpers[3].theNumber++;
103    }
104    if (Allstats.otherStats[i].wasItCorrect == false &&
        Allstats.statEntries[i].dateWasChanged == true &&
        Allstats.otherStats[i].dateWasSelected != true)
105    {
106        calcHelpers[4].theNumber++;
107    }
108    if (Allstats.otherStats[i].wasItCorrect == false &&
        Allstats.statEntries[i].nameWasChanged == true &&
        Allstats.otherStats[i].nameWasSelected != true)
109    {
110        calcHelpers[5].theNumber++;
111    }
112
113    }
114 }
115 //run for a while so it is sorted for sure
116 for(int i = 0; i < 20; i++)
117 {
118     calcHelpers.Sort(sorter);
119 }
120
121
122
123
124 calcHelpers.Sort(sorter);
125 //gives the player a tip on what to improve on when the game is
    over
126 if (GameObject.FindObjectOfType<newGameManager>().gameOver ==
    true)
127 {
128
129     if (calcHelpers[0].theNameOfTheStat == "color")
130     {
```

```

...er\boder-gods-valmis\Assets\scripts\gameStatKeeper.cs 4
131         if (GameObject.FindObjectOfType<xmlDatabaseManager> 7
132             ().db.language == 0)
133         {
134             tipText.text = colorTip[0];
135         }
136         if (GameObject.FindObjectOfType<xmlDatabaseManager> 7
137             ().db.language == 1)
138         {
139             tipText.text = colorTip[1];
140         }
141         if (GameObject.FindObjectOfType<xmlDatabaseManager> 7
142             ().db.language == 2)
143         {
144             tipText.text = colorTip[2];
145         }
146         if (GameObject.FindObjectOfType<xmlDatabaseManager> 7
147             ().db.language == 3)
148         {
149             tipText.text = colorTip[3];
150         }
151     }
152     if (calcHelpers[0].theNameOfTheStat == "person")
153     {
154         if (GameObject.FindObjectOfType<xmlDatabaseManager> 7
155             ().db.language == 0)
156         {
157             tipText.text = personTip[0];
158         }
159         if (GameObject.FindObjectOfType<xmlDatabaseManager> 7
160             ().db.language == 1)
161         {
162             tipText.text = personTip[1];
163         }
164         if (GameObject.FindObjectOfType<xmlDatabaseManager> 7
165             ().db.language == 2)
166         {
167             tipText.text = personTip[2];
168         }
169         if (GameObject.FindObjectOfType<xmlDatabaseManager> 7
170             ().db.language == 3)
171         {
172             tipText.text = personTip[3];
173         }
174     }
175     if (calcHelpers[0].theNameOfTheStat == "photo")
176     {
177         if (GameObject.FindObjectOfType<xmlDatabaseManager> 7
178             ().db.language == 0)
179         {
180             tipText.text = photoTip[0];
181         }
182         if (GameObject.FindObjectOfType<xmlDatabaseManager> 7
183             ().db.language == 1)
184         {
185             tipText.text = photoTip[1];
186         }
187         if (GameObject.FindObjectOfType<xmlDatabaseManager> 7
188             ().db.language == 2)
189         {
190             tipText.text = photoTip[2];
191         }
192         if (GameObject.FindObjectOfType<xmlDatabaseManager> 7
193             ().db.language == 3)
194         {
195             tipText.text = photoTip[3];
196         }
197     }
198 }

```

```
171         tipText.text = photoTip[0];
172     }
173     if (GameObject.FindObjectOfType<xmlDatabaseManager>
174         ().db.language == 1)
175     {
176         tipText.text = photoTip[1];
177     }
178     if (GameObject.FindObjectOfType<xmlDatabaseManager>
179         ().db.language == 2)
180     {
181         tipText.text = photoTip[2];
182     }
183     if (GameObject.FindObjectOfType<xmlDatabaseManager>
184         ().db.language == 3)
185     {
186         tipText.text = photoTip[3];
187     }
188     if (calcHelpers[0].theNameOfTheStat == "emblem")
189     {
190         if (GameObject.FindObjectOfType<xmlDatabaseManager>
191             ().db.language == 0)
192         {
193             tipText.text = emblemTip[0];
194         }
195         if (GameObject.FindObjectOfType<xmlDatabaseManager>
196             ().db.language == 1)
197         {
198             tipText.text = emblemTip[1];
199         }
200         if (GameObject.FindObjectOfType<xmlDatabaseManager>
201             ().db.language == 2)
202         {
203             tipText.text = emblemTip[2];
204         }
205         if (GameObject.FindObjectOfType<xmlDatabaseManager>
206             ().db.language == 3)
207         {
208             tipText.text = emblemTip[3];
209         }
210     }
211     if (calcHelpers[0].theNameOfTheStat == "date")
212     {
213         if (GameObject.FindObjectOfType<xmlDatabaseManager>
214             ().db.language == 0)
215         {
216             tipText.text = dateTip[0];
217         }
218         if (GameObject.FindObjectOfType<xmlDatabaseManager>
```

```
        ().db.language == 1)
212     {
213         tipText.text = dateTip[1];
214     }
215     if (GameObject.FindObjectOfType<xmlDatabaseManager>
        ().db.language == 2)
216     {
217         tipText.text = dateTip[2];
218     }
219     if (GameObject.FindObjectOfType<xmlDatabaseManager>
        ().db.language == 3)
220     {
221         tipText.text = dateTip[3];
222     }
223 }
224 if (calcHelpers[0].theNameOfTheStat == "name")
225 {
226     if (GameObject.FindObjectOfType<xmlDatabaseManager>
        ().db.language == 0)
227     {
228         tipText.text = nameTip[0];
229     }
230     if (GameObject.FindObjectOfType<xmlDatabaseManager>
        ().db.language == 1)
231     {
232         tipText.text = nameTip[1];
233     }
234     if (GameObject.FindObjectOfType<xmlDatabaseManager>
        ().db.language == 2)
235     {
236         tipText.text = nameTip[2];
237     }
238     if (GameObject.FindObjectOfType<xmlDatabaseManager>
        ().db.language == 3)
239     {
240         tipText.text = nameTip[3];
241     }
242 }
243 }
244
245
246 }
247 //sorts the list thing
248 public static int sorter(mistakeCalcHelper _theOneThatWillBeAdded,
        mistakeCalcHelper _theOnebeingComparedTo)
249 {
250
251     if (_theOneThatWillBeAdded.theNumber >
        _theOnebeingComparedTo.theNumber)
```

```
252     {
253         return -1;
254     }
255     else
256     {
257         if (_theOneThatWillBeAdded.theNumber <
258             _theOnebeingComparedTo.theNumber)
259         {
260             return 1;
261         }
262         else return 0;
263     }
264 }
265 //passport stats for the calculation
266 public void saveStat(bool _date, bool _name, bool _emblem, bool
267     _photo, bool _person, bool _color, bool _paper)
268 {
269     entry = new statEntry();
270     entry.dateWasChanged = _date;
271     entry.nameWasChanged = _name;
272     entry.emblemWasChanged = _emblem;
273     entry.photoWasChanged = _photo;
274     entry.personWasChanged = _person;
275     entry.colorWasChanged = _color;
276     entry.paperWasChanged = _paper;
277     stats.Add(entry);
278 }
279 //player decision and combo + player input saving
280 public void saveOtherStats(int _combo, bool _wasItCorrect, bool
281     _photo, bool _color, bool _emblem, bool _name, bool _date, bool
282     _person, bool _paper)
283 {
284     otherStat = new otherStats();
285     otherStat.combo = _combo;
286     otherStat.wasItCorrect = _wasItCorrect;
287     if (_photo != _wasItCorrect)
288     {
289         otherStat.photoWasSelected = false;
290     }
291     if (_color != _wasItCorrect)
292     {
293         otherStat.colorWasSelected = false;
294     }
295     if (_emblem != _wasItCorrect)
296     {
297         otherStat.emblemWasSelected = false;
```

```
297     }
298     if (_name != _wasItCorrect)
299     {
300         otherStat.nameWasSelected = false;
301     }
302     if (_date != _wasItCorrect)
303     {
304         otherStat.dateWasSelected = false;
305     }
306     if (_person != _wasItCorrect)
307     {
308         otherStat.personWasSelected = false;
309     }
310     if (_paper != _wasItCorrect)
311     {
312         otherStat.paperWasSelected = false;
313     }
314
315     otherStats.Add(otherStat);
316     saveEveryStat();
317
318 }
319 // put both to the same list for easy access or pointless access
320 public void saveEveryStat()
321 {
322     Allstats.statEntries = stats;
323     Allstats.otherStats = otherStats;
324 }
325 //has all the info in one class a bit pointless
326 public allTheStats calculateStuff(allTheStats theStats)
327 {
328     theStats.failiureCount=0;
329     theStats.correctChoises = 0;
330     foreach(otherStats i in theStats.otherStats)
331     {
332         if (i.wasItCorrect == true)
333         {
334             theStats.correctChoises=theStats.correctChoises+1;
335         }
336     }
337     foreach (otherStats i in theStats.otherStats)
338     {
339         if (i.wasItCorrect == false)
340         {
341             theStats.failiureCount=theStats.failiureCount+1;
342         }
343     }
344     theStats.percentageOfCorrectChoises = (theStats.correctChoises /  
        theStats.otherStats.Count)*100;
```



```
345         theStats.passportsHandled = theStats.otherStats.Count;
346         return theStats;
347     }
348 }
349 [System.Serializable]
350 //passport stat class
351 public class statEntry
352 {
353     public bool photoWasChanged;
354     public bool colorWasChanged;
355     public bool emblemWasChanged;
356     public bool nameWasChanged;
357     public bool dateWasChanged;
358     public bool personWasChanged;
359     public bool paperWasChanged;
360 }
361 [System.Serializable]
362 //player info class
363 public class otherStats
364 {
365     public int combo;
366     public bool wasItCorrect;
367     public bool photoWasSelected;
368     public bool colorWasSelected;
369     public bool emblemWasSelected;
370     public bool nameWasSelected;
371     public bool dateWasSelected;
372     public bool personWasSelected;
373     public bool paperWasSelected;
374 }
375 [System.Serializable]
376 //combined stat class that has both player and passport stats from each attempt in a list
377 public class allTheStats
378 {
379     public float correctChoises=0;
380     public float percentageOfCorrectChoises=0;
381     public float failiureCount=0;
382     public float passportsHandled=0;
383     public List<statEntry> statEntries;
384     public List<otherStats> otherStats;
385 }
386 [System.Serializable]
387 public class mistakeCalcHelper
388 {
389     public int theNumber=0;
390     public string theNameOfTheStat;
```

