```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class AI_Blok_FindSpot : MonoBehaviour
6  {
7
8      //Checked Rows
9      private bool downed = false;
10     private bool checkedRow1 = false;
11     private bool checkedRow2 = false;
12     private bool checkedRow3 = false;
13     private bool checkedRow4 = false;
14     private bool checkedRow5 = false;
15     private bool checkedRow6 = false;
16     private bool checkedRow7 = false;
17     private bool checkedRow8 = false;
18     private bool checkedRow9 = false;
19     private bool checkedRow10 = false;
20     private bool checkedRow11 = false;
21     private bool checkedRow12 = false;
22     private bool checkedRow13 = false;
23     private bool checkedRow14 = false;
24     private bool checkedRow15 = false;
25     private bool checkedRow16 = false;
26     private bool checkedRow17 = false;
27     private bool checkedRow18 = false;
28     private bool checkedRow19 = false;
29     private bool checkedRow20 = false;
30
31     private int myBottomBlocks = 2;
32     private int mySpot = 0;
33     private int spot;
34
35     private float timerTime;
36
37     private Vector3 cubeQuart = new Vector3(0.25f, 0.25f, 0.25f);
38
39     //Rows
40
41     private List<Vector3> row1 = new List<Vector3>();
42     private List<Vector3> row2 = new List<Vector3>();
43     private List<Vector3> row3 = new List<Vector3>();
44     private List<Vector3> row4 = new List<Vector3>();
45     private List<Vector3> row5 = new List<Vector3>();
46     private List<Vector3> row6 = new List<Vector3>();
47     private List<Vector3> row7 = new List<Vector3>();
48     private List<Vector3> row8 = new List<Vector3>();
49     private List<Vector3> row9 = new List<Vector3>();
```

```
50      private List<Vector3> row10 = new List<Vector3>();
51      private List<Vector3> row11 = new List<Vector3>();
52      private List<Vector3> row12 = new List<Vector3>();
53      private List<Vector3> row13 = new List<Vector3>();
54      private List<Vector3> row14 = new List<Vector3>();
55      private List<Vector3> row15 = new List<Vector3>();
56      private List<Vector3> row16 = new List<Vector3>();
57      private List<Vector3> row17 = new List<Vector3>();
58      private List<Vector3> row18 = new List<Vector3>();
59      private List<Vector3> row19 = new List<Vector3>();
60      private List<Vector3> row20 = new List<Vector3>();
61
62      //Row Vacancies
63      private List<Vector3> row1Vacancies = new List<Vector3>();
64      private List<Vector3> row2Vacancies = new List<Vector3>();
65      private List<Vector3> row3Vacancies = new List<Vector3>();
66      private List<Vector3> row4Vacancies = new List<Vector3>();
67      private List<Vector3> row5Vacancies = new List<Vector3>();
68      private List<Vector3> row6Vacancies = new List<Vector3>();
69      private List<Vector3> row7Vacancies = new List<Vector3>();
70      private List<Vector3> row8Vacancies = new List<Vector3>();
71      private List<Vector3> row9Vacancies = new List<Vector3>();
72      private List<Vector3> row10Vacancies = new List<Vector3>();
73      private List<Vector3> row11Vacancies = new List<Vector3>();
74      private List<Vector3> row12Vacancies = new List<Vector3>();
75      private List<Vector3> row13Vacancies = new List<Vector3>();
76      private List<Vector3> row14Vacancies = new List<Vector3>();
77      private List<Vector3> row15Vacancies = new List<Vector3>();
78      private List<Vector3> row16Vacancies = new List<Vector3>();
79      private List<Vector3> row17Vacancies = new List<Vector3>();
80      private List<Vector3> row18Vacancies = new List<Vector3>();
81      private List<Vector3> row19Vacancies = new List<Vector3>();
82      private List<Vector3> row20Vacancies = new List<Vector3>();
83      private List<int> row1VacancyInts = new List<int>();
84      private List<int> row2VacancyInts = new List<int>();
85      private List<int> row3VacancyInts = new List<int>();
86      private List<int> row4VacancyInts = new List<int>();
87      private List<int> row5VacancyInts = new List<int>();
88      private List<int> row6VacancyInts = new List<int>();
89      private List<int> row7VacancyInts = new List<int>();
90      private List<int> row8VacancyInts = new List<int>();
91      private List<int> row9VacancyInts = new List<int>();
92      private List<int> row10VacancyInts = new List<int>();
93      private List<int> row11VacancyInts = new List<int>();
94      private List<int> row12VacancyInts = new List<int>();
95      private List<int> row13VacancyInts = new List<int>();
96      private List<int> row14VacancyInts = new List<int>();
97      private List<int> row15VacancyInts = new List<int>();
98      private List<int> row16VacancyInts = new List<int>();
```

```
 99      private List<int> row17VacancyInts = new List<int>();
100      private List<int> row18VacancyInts = new List<int>();
101      private List<int> row19VacancyInts = new List<int>();
102      private List<int> row20VacancyInts = new List<int>();
103      private List<int> openSpots = new List<int>();
104      private List<int> preferredSpots = new List<int>();
105
106      //Spaces
107
108      private Vector3 boxA1 = new Vector3(0.5f, 0.3f, -3.6f);
109      private Vector3 boxB1 = new Vector3(0.5f, 0.3f, -2.6f);
110      private Vector3 boxC1 = new Vector3(0.5f, 0.3f, -1.6f);
111      private Vector3 boxD1 = new Vector3(0.5f, 0.3f, -0.6f);
112      private Vector3 boxE1 = new Vector3(0.5f, 0.3f, 0.3f);
113      private Vector3 boxF1 = new Vector3(0.5f, 0.3f, 1.3f);
114      private Vector3 boxG1 = new Vector3(0.5f, 0.3f, 2.3f);
115      private Vector3 boxH1 = new Vector3(0.5f, 0.3f, 3.3f);
116      private Vector3 boxI1 = new Vector3(0.5f, 0.3f, 4.3f);
117      private Vector3 boxJ1 = new Vector3(0.5f, 0.3f, 5.3f);
118
119      private Vector3 boxA2 = new Vector3(0.5f, 1.3f, -3.6f);
120      private Vector3 boxB2 = new Vector3(0.5f, 1.3f, -2.6f);
121      private Vector3 boxC2 = new Vector3(0.5f, 1.3f, -1.6f);
122      private Vector3 boxD2 = new Vector3(0.5f, 1.3f, -0.6f);
123      private Vector3 boxE2 = new Vector3(0.5f, 1.3f, 0.3f);
124      private Vector3 boxF2 = new Vector3(0.5f, 1.3f, 1.3f);
125      private Vector3 boxG2 = new Vector3(0.5f, 1.3f, 2.3f);
126      private Vector3 boxH2 = new Vector3(0.5f, 1.3f, 3.3f);
127      private Vector3 boxI2 = new Vector3(0.5f, 1.3f, 4.3f);
128      private Vector3 boxJ2 = new Vector3(0.5f, 1.3f, 5.3f);
129
130      private Vector3 boxA3 = new Vector3(0.5f, 2.3f, -3.6f);
131      private Vector3 boxB3 = new Vector3(0.5f, 2.3f, -2.6f);
132      private Vector3 boxC3 = new Vector3(0.5f, 2.3f, -1.6f);
133      private Vector3 boxD3 = new Vector3(0.5f, 2.3f, -0.6f);
134      private Vector3 boxE3 = new Vector3(0.5f, 2.3f, 0.3f);
135      private Vector3 boxF3 = new Vector3(0.5f, 2.3f, 1.3f);
136      private Vector3 boxG3 = new Vector3(0.5f, 2.3f, 2.3f);
137      private Vector3 boxH3 = new Vector3(0.5f, 2.3f, 3.3f);
138      private Vector3 boxI3 = new Vector3(0.5f, 2.3f, 4.3f);
139      private Vector3 boxJ3 = new Vector3(0.5f, 2.3f, 5.3f);
140
141      private Vector3 boxA4 = new Vector3(0.5f, 3.3f, -3.6f);
142      private Vector3 boxB4 = new Vector3(0.5f, 3.3f, -2.6f);
143      private Vector3 boxC4 = new Vector3(0.5f, 3.3f, -1.6f);
144      private Vector3 boxD4 = new Vector3(0.5f, 3.3f, -0.6f);
145      private Vector3 boxE4 = new Vector3(0.5f, 3.3f, 0.3f);
146      private Vector3 boxF4 = new Vector3(0.5f, 3.3f, 1.3f);
147      private Vector3 boxG4 = new Vector3(0.5f, 3.3f, 2.3f);
```

```
148     private Vector3 boxH4 = new Vector3(0.5f, 3.3f, 3.3f);
149     private Vector3 boxI4 = new Vector3(0.5f, 3.3f, 4.3f);
150     private Vector3 boxJ4 = new Vector3(0.5f, 3.3f, 5.3f);
151
152     private Vector3 boxA5 = new Vector3(0.5f, 4.3f, -3.6f);
153     private Vector3 boxB5 = new Vector3(0.5f, 4.3f, -2.6f);
154     private Vector3 boxC5 = new Vector3(0.5f, 4.3f, -1.6f);
155     private Vector3 boxD5 = new Vector3(0.5f, 4.3f, -0.6f);
156     private Vector3 boxE5 = new Vector3(0.5f, 4.3f, 0.3f);
157     private Vector3 boxF5 = new Vector3(0.5f, 4.3f, 1.3f);
158     private Vector3 boxG5 = new Vector3(0.5f, 4.3f, 2.3f);
159     private Vector3 boxH5 = new Vector3(0.5f, 4.3f, 3.3f);
160     private Vector3 boxI5 = new Vector3(0.5f, 4.3f, 4.3f);
161     private Vector3 boxJ5 = new Vector3(0.5f, 4.3f, 5.3f);
162
163     private Vector3 boxA6 = new Vector3(0.5f, 5.3f, -3.6f);
164     private Vector3 boxB6 = new Vector3(0.5f, 5.3f, -2.6f);
165     private Vector3 boxC6 = new Vector3(0.5f, 5.3f, -1.6f);
166     private Vector3 boxD6 = new Vector3(0.5f, 5.3f, -0.6f);
167     private Vector3 boxE6 = new Vector3(0.5f, 5.3f, 0.3f);
168     private Vector3 boxF6 = new Vector3(0.5f, 5.3f, 1.3f);
169     private Vector3 boxG6 = new Vector3(0.5f, 5.3f, 2.3f);
170     private Vector3 boxH6 = new Vector3(0.5f, 5.3f, 3.3f);
171     private Vector3 boxI6 = new Vector3(0.5f, 5.3f, 4.3f);
172     private Vector3 boxJ6 = new Vector3(0.5f, 5.3f, 5.3f);
173
174     private Vector3 boxA7 = new Vector3(0.5f, 6.3f, -3.6f);
175     private Vector3 boxB7 = new Vector3(0.5f, 6.3f, -2.6f);
176     private Vector3 boxC7 = new Vector3(0.5f, 6.3f, -1.6f);
177     private Vector3 boxD7 = new Vector3(0.5f, 6.3f, -0.6f);
178     private Vector3 boxE7 = new Vector3(0.5f, 6.3f, 0.3f);
179     private Vector3 boxF7 = new Vector3(0.5f, 6.3f, 1.3f);
180     private Vector3 boxG7 = new Vector3(0.5f, 6.3f, 2.3f);
181     private Vector3 boxH7 = new Vector3(0.5f, 6.3f, 3.3f);
182     private Vector3 boxI7 = new Vector3(0.5f, 6.3f, 4.3f);
183     private Vector3 boxJ7 = new Vector3(0.5f, 6.3f, 5.3f);
184
185     private Vector3 boxA8 = new Vector3(0.5f, 7.3f, -3.6f);
186     private Vector3 boxB8 = new Vector3(0.5f, 7.3f, -2.6f);
187     private Vector3 boxC8 = new Vector3(0.5f, 7.3f, -1.6f);
188     private Vector3 boxD8 = new Vector3(0.5f, 7.3f, -0.6f);
189     private Vector3 boxE8 = new Vector3(0.5f, 7.3f, 0.3f);
190     private Vector3 boxF8 = new Vector3(0.5f, 7.3f, 1.3f);
191     private Vector3 boxG8 = new Vector3(0.5f, 7.3f, 2.3f);
192     private Vector3 boxH8 = new Vector3(0.5f, 7.3f, 3.3f);
193     private Vector3 boxI8 = new Vector3(0.5f, 7.3f, 4.3f);
194     private Vector3 boxJ8 = new Vector3(0.5f, 7.3f, 5.3f);
195
196     private Vector3 boxA9 = new Vector3(0.5f, 8.3f, -3.6f);
```

```
197     private Vector3 boxB9 = new Vector3(0.5f, 8.3f, -2.6f);
198     private Vector3 boxC9 = new Vector3(0.5f, 8.3f, -1.6f);
199     private Vector3 boxD9 = new Vector3(0.5f, 8.3f, -0.6f);
200     private Vector3 boxE9 = new Vector3(0.5f, 8.3f, 0.3f);
201     private Vector3 boxF9 = new Vector3(0.5f, 8.3f, 1.3f);
202     private Vector3 boxG9 = new Vector3(0.5f, 8.3f, 2.3f);
203     private Vector3 boxH9 = new Vector3(0.5f, 8.3f, 3.3f);
204     private Vector3 boxI9 = new Vector3(0.5f, 8.3f, 4.3f);
205     private Vector3 boxJ9 = new Vector3(0.5f, 8.3f, 5.3f);
206
207     private Vector3 boxA10 = new Vector3(0.5f, 9.3f, -3.6f);
208     private Vector3 boxB10 = new Vector3(0.5f, 9.3f, -2.6f);
209     private Vector3 boxC10 = new Vector3(0.5f, 9.3f, -1.6f);
210     private Vector3 boxD10 = new Vector3(0.5f, 9.3f, -0.6f);
211     private Vector3 boxE10 = new Vector3(0.5f, 9.3f, 0.3f);
212     private Vector3 boxF10 = new Vector3(0.5f, 9.3f, 1.3f);
213     private Vector3 boxG10 = new Vector3(0.5f, 9.3f, 2.3f);
214     private Vector3 boxH10 = new Vector3(0.5f, 9.3f, 3.3f);
215     private Vector3 boxI10 = new Vector3(0.5f, 9.3f, 4.3f);
216     private Vector3 boxJ10 = new Vector3(0.5f, 9.3f, 5.3f);
217
218     private Vector3 boxA11 = new Vector3(0.5f, 10.3f, -3.6f);
219     private Vector3 boxB11 = new Vector3(0.5f, 10.3f, -2.6f);
220     private Vector3 boxC11 = new Vector3(0.5f, 10.3f, -1.6f);
221     private Vector3 boxD11 = new Vector3(0.5f, 10.3f, -0.6f);
222     private Vector3 boxE11 = new Vector3(0.5f, 10.3f, 0.3f);
223     private Vector3 boxF11 = new Vector3(0.5f, 10.3f, 1.3f);
224     private Vector3 boxG11 = new Vector3(0.5f, 10.3f, 2.3f);
225     private Vector3 boxH11 = new Vector3(0.5f, 10.3f, 3.3f);
226     private Vector3 boxI11 = new Vector3(0.5f, 10.3f, 4.3f);
227     private Vector3 boxJ11 = new Vector3(0.5f, 10.3f, 5.3f);
228
229     private Vector3 boxA12 = new Vector3(0.5f, 11.3f, -3.6f);
230     private Vector3 boxB12 = new Vector3(0.5f, 11.3f, -2.6f);
231     private Vector3 boxC12 = new Vector3(0.5f, 11.3f, -1.6f);
232     private Vector3 boxD12 = new Vector3(0.5f, 11.3f, -0.6f);
233     private Vector3 boxE12 = new Vector3(0.5f, 11.3f, 0.3f);
234     private Vector3 boxF12 = new Vector3(0.5f, 11.3f, 1.3f);
235     private Vector3 boxG12 = new Vector3(0.5f, 11.3f, 2.3f);
236     private Vector3 boxH12 = new Vector3(0.5f, 11.3f, 3.3f);
237     private Vector3 boxI12 = new Vector3(0.5f, 11.3f, 4.3f);
238     private Vector3 boxJ12 = new Vector3(0.5f, 11.3f, 5.3f);
239
240     private Vector3 boxA13 = new Vector3(0.5f, 12.3f, -3.6f);
241     private Vector3 boxB13 = new Vector3(0.5f, 12.3f, -2.6f);
242     private Vector3 boxC13 = new Vector3(0.5f, 12.3f, -1.6f);
243     private Vector3 boxD13 = new Vector3(0.5f, 12.3f, -0.6f);
244     private Vector3 boxE13 = new Vector3(0.5f, 12.3f, 0.3f);
245     private Vector3 boxF13 = new Vector3(0.5f, 12.3f, 1.3f);
```

```
246     private Vector3 boxG13 = new Vector3(0.5f, 12.3f, 2.3f);
247     private Vector3 boxH13 = new Vector3(0.5f, 12.3f, 3.3f);
248     private Vector3 boxI13 = new Vector3(0.5f, 12.3f, 4.3f);
249     private Vector3 boxJ13 = new Vector3(0.5f, 12.3f, 5.3f);
250
251     private Vector3 boxA14 = new Vector3(0.5f, 13.3f, -3.6f);
252     private Vector3 boxB14 = new Vector3(0.5f, 13.3f, -2.6f);
253     private Vector3 boxC14 = new Vector3(0.5f, 13.3f, -1.6f);
254     private Vector3 boxD14 = new Vector3(0.5f, 13.3f, -0.6f);
255     private Vector3 boxE14 = new Vector3(0.5f, 13.3f, 0.3f);
256     private Vector3 boxF14 = new Vector3(0.5f, 13.3f, 1.3f);
257     private Vector3 boxG14 = new Vector3(0.5f, 13.3f, 2.3f);
258     private Vector3 boxH14 = new Vector3(0.5f, 13.3f, 3.3f);
259     private Vector3 boxI14 = new Vector3(0.5f, 13.3f, 4.3f);
260     private Vector3 boxJ14 = new Vector3(0.5f, 13.3f, 5.3f);
261
262     private Vector3 boxA15 = new Vector3(0.5f, 14.3f, -3.6f);
263     private Vector3 boxB15 = new Vector3(0.5f, 14.3f, -2.6f);
264     private Vector3 boxC15 = new Vector3(0.5f, 14.3f, -1.6f);
265     private Vector3 boxD15 = new Vector3(0.5f, 14.3f, -0.6f);
266     private Vector3 boxE15 = new Vector3(0.5f, 14.3f, 0.3f);
267     private Vector3 boxF15 = new Vector3(0.5f, 14.3f, 1.3f);
268     private Vector3 boxG15 = new Vector3(0.5f, 14.3f, 2.3f);
269     private Vector3 boxH15 = new Vector3(0.5f, 14.3f, 3.3f);
270     private Vector3 boxI15 = new Vector3(0.5f, 14.3f, 4.3f);
271     private Vector3 boxJ15 = new Vector3(0.5f, 14.3f, 5.3f);
272
273     private Vector3 boxA16 = new Vector3(0.5f, 15.3f, -3.6f);
274     private Vector3 boxB16 = new Vector3(0.5f, 15.3f, -2.6f);
275     private Vector3 boxC16 = new Vector3(0.5f, 15.3f, -1.6f);
276     private Vector3 boxD16 = new Vector3(0.5f, 15.3f, -0.6f);
277     private Vector3 boxE16 = new Vector3(0.5f, 15.3f, 0.3f);
278     private Vector3 boxF16 = new Vector3(0.5f, 15.3f, 1.3f);
279     private Vector3 boxG16 = new Vector3(0.5f, 15.3f, 2.3f);
280     private Vector3 boxH16 = new Vector3(0.5f, 15.3f, 3.3f);
281     private Vector3 boxI16 = new Vector3(0.5f, 15.3f, 4.3f);
282     private Vector3 boxJ16 = new Vector3(0.5f, 15.3f, 5.3f);
283
284     private Vector3 boxA17 = new Vector3(0.5f, 16.3f, -3.6f);
285     private Vector3 boxB17 = new Vector3(0.5f, 16.3f, -2.6f);
286     private Vector3 boxC17 = new Vector3(0.5f, 16.3f, -1.6f);
287     private Vector3 boxD17 = new Vector3(0.5f, 16.3f, -0.6f);
288     private Vector3 boxE17 = new Vector3(0.5f, 16.3f, 0.3f);
289     private Vector3 boxF17 = new Vector3(0.5f, 16.3f, 1.3f);
290     private Vector3 boxG17 = new Vector3(0.5f, 16.3f, 2.3f);
291     private Vector3 boxH17 = new Vector3(0.5f, 16.3f, 3.3f);
292     private Vector3 boxI17 = new Vector3(0.5f, 16.3f, 4.3f);
293     private Vector3 boxJ17 = new Vector3(0.5f, 16.3f, 5.3f);
294
```

```
295     private Vector3 boxA18 = new Vector3(0.5f, 17.3f, -3.6f);
296     private Vector3 boxB18 = new Vector3(0.5f, 17.3f, -2.6f);
297     private Vector3 boxC18 = new Vector3(0.5f, 17.3f, -1.6f);
298     private Vector3 boxD18 = new Vector3(0.5f, 17.3f, -0.6f);
299     private Vector3 boxE18 = new Vector3(0.5f, 17.3f, 0.3f);
300     private Vector3 boxF18 = new Vector3(0.5f, 17.3f, 1.3f);
301     private Vector3 boxG18 = new Vector3(0.5f, 17.3f, 2.3f);
302     private Vector3 boxH18 = new Vector3(0.5f, 17.3f, 3.3f);
303     private Vector3 boxI18 = new Vector3(0.5f, 17.3f, 4.3f);
304     private Vector3 boxJ18 = new Vector3(0.5f, 17.3f, 5.3f);
305
306     private Vector3 boxA19 = new Vector3(0.5f, 18.3f, -3.6f);
307     private Vector3 boxB19 = new Vector3(0.5f, 18.3f, -2.6f);
308     private Vector3 boxC19 = new Vector3(0.5f, 18.3f, -1.6f);
309     private Vector3 boxD19 = new Vector3(0.5f, 18.3f, -0.6f);
310     private Vector3 boxE19 = new Vector3(0.5f, 18.3f, 0.3f);
311     private Vector3 boxF19 = new Vector3(0.5f, 18.3f, 1.3f);
312     private Vector3 boxG19 = new Vector3(0.5f, 18.3f, 2.3f);
313     private Vector3 boxH19 = new Vector3(0.5f, 18.3f, 3.3f);
314     private Vector3 boxI19 = new Vector3(0.5f, 18.3f, 4.3f);
315     private Vector3 boxJ19 = new Vector3(0.5f, 18.3f, 5.3f);
316
317     private Vector3 boxA20 = new Vector3(0.5f, 19.3f, -3.6f);
318     private Vector3 boxB20 = new Vector3(0.5f, 19.3f, -2.6f);
319     private Vector3 boxC20 = new Vector3(0.5f, 19.3f, -1.6f);
320     private Vector3 boxD20 = new Vector3(0.5f, 19.3f, -0.6f);
321     private Vector3 boxE20 = new Vector3(0.5f, 19.3f, 0.3f);
322     private Vector3 boxF20 = new Vector3(0.5f, 19.3f, 1.3f);
323     private Vector3 boxG20 = new Vector3(0.5f, 19.3f, 2.3f);
324     private Vector3 boxH20 = new Vector3(0.5f, 19.3f, 3.3f);
325     private Vector3 boxI20 = new Vector3(0.5f, 19.3f, 4.3f);
326     private Vector3 boxJ20 = new Vector3(0.5f, 19.3f, 5.3f);
327
328     //VacancyInts
329     //-4, -3, -2, -1, 0, 1, 2, 3, 4, 5
330
331     // Start is called before the first frame update
332     void Start()
333     {
334         timerTime = Random.Range(0.2f, 0.6f);
335         AddPositions();
336         CheckRows();
337         ChooseSpot();
338     }
339
340     void Update()
341     {
342         timerTime -= Time.deltaTime;
343         if (timerTime <= 0.0f)
```

```
344         {
345             GoToSpot();
346         }
347     }
348     //Add the positions to the Lists
349     void AddPositions()
350     {
351         row1.Add(boxA1);
352         row1.Add(boxB1);
353         row1.Add(boxC1);
354         row1.Add(boxD1);
355         row1.Add(boxE1);
356         row1.Add(boxF1);
357         row1.Add(boxG1);
358         row1.Add(boxH1);
359         row1.Add(boxI1);
360         row1.Add(boxJ1);
361 
362         row2.Add(boxA2);
363         row2.Add(boxB2);
364         row2.Add(boxC2);
365         row2.Add(boxD2);
366         row2.Add(boxE2);
367         row2.Add(boxF2);
368         row2.Add(boxG2);
369         row2.Add(boxH2);
370         row2.Add(boxI2);
371         row2.Add(boxJ2);
372 
373         row3.Add(boxA3);
374         row3.Add(boxB3);
375         row3.Add(boxC3);
376         row3.Add(boxD3);
377         row3.Add(boxE3);
378         row3.Add(boxF3);
379         row3.Add(boxG3);
380         row3.Add(boxH3);
381         row3.Add(boxI3);
382         row3.Add(boxJ3);
383 
384         row4.Add(boxA4);
385         row4.Add(boxB4);
386         row4.Add(boxC4);
387         row4.Add(boxD4);
388         row4.Add(boxE4);
389         row4.Add(boxF4);
390         row4.Add(boxG4);
391         row4.Add(boxH4);
392         row4.Add(boxI4);
```

```
393         row4.Add(boxJ4);
394
395         row5.Add(boxA5);
396         row5.Add(boxB5);
397         row5.Add(boxC5);
398         row5.Add(boxD5);
399         row5.Add(boxE5);
400         row5.Add(boxF5);
401         row5.Add(boxG5);
402         row5.Add(boxH5);
403         row5.Add(boxI5);
404         row5.Add(boxJ5);
405
406         row6.Add(boxA6);
407         row6.Add(boxB6);
408         row6.Add(boxC6);
409         row6.Add(boxD6);
410         row6.Add(boxE6);
411         row6.Add(boxF6);
412         row6.Add(boxG6);
413         row6.Add(boxH6);
414         row6.Add(boxI6);
415         row6.Add(boxJ6);
416
417         row7.Add(boxA7);
418         row7.Add(boxB7);
419         row7.Add(boxC7);
420         row7.Add(boxD7);
421         row7.Add(boxE7);
422         row7.Add(boxF7);
423         row7.Add(boxG7);
424         row7.Add(boxH7);
425         row7.Add(boxI7);
426         row7.Add(boxJ7);
427
428         row8.Add(boxA8);
429         row8.Add(boxB8);
430         row8.Add(boxC8);
431         row8.Add(boxD8);
432         row8.Add(boxE8);
433         row8.Add(boxF8);
434         row8.Add(boxG8);
435         row8.Add(boxH8);
436         row8.Add(boxI8);
437         row8.Add(boxJ8);
438
439         row9.Add(boxA9);
440         row9.Add(boxB9);
441         row9.Add(boxC9);
```

```
442         row9.Add(boxD9);
443         row9.Add(boxE9);
444         row9.Add(boxF9);
445         row9.Add(boxG9);
446         row9.Add(boxH9);
447         row9.Add(boxI9);
448         row9.Add(boxJ9);
449
450         row10.Add(boxA10);
451         row10.Add(boxB10);
452         row10.Add(boxC10);
453         row10.Add(boxD10);
454         row10.Add(boxE10);
455         row10.Add(boxF10);
456         row10.Add(boxG10);
457         row10.Add(boxH10);
458         row10.Add(boxI10);
459         row10.Add(boxJ10);
460
461         row11.Add(boxA11);
462         row11.Add(boxB11);
463         row11.Add(boxC11);
464         row11.Add(boxD11);
465         row11.Add(boxE11);
466         row11.Add(boxF11);
467         row11.Add(boxG11);
468         row11.Add(boxH11);
469         row11.Add(boxI11);
470         row11.Add(boxJ11);
471
472         row12.Add(boxA12);
473         row12.Add(boxB12);
474         row12.Add(boxC12);
475         row12.Add(boxD12);
476         row12.Add(boxE12);
477         row12.Add(boxF12);
478         row12.Add(boxG12);
479         row12.Add(boxH12);
480         row12.Add(boxI12);
481         row12.Add(boxJ12);
482
483         row13.Add(boxA13);
484         row13.Add(boxB13);
485         row13.Add(boxC13);
486         row13.Add(boxD13);
487         row13.Add(boxE13);
488         row13.Add(boxF13);
489         row13.Add(boxG13);
490         row13.Add(boxH13);
```

```
491         row13.Add(boxI13);
492         row13.Add(boxJ13);
493
494         row14.Add(boxA14);
495         row14.Add(boxB14);
496         row14.Add(boxC14);
497         row14.Add(boxD14);
498         row14.Add(boxE14);
499         row14.Add(boxF14);
500         row14.Add(boxG14);
501         row14.Add(boxH14);
502         row14.Add(boxI14);
503         row14.Add(boxJ14);
504
505         row15.Add(boxA15);
506         row15.Add(boxB15);
507         row15.Add(boxC15);
508         row15.Add(boxD15);
509         row15.Add(boxE15);
510         row15.Add(boxF15);
511         row15.Add(boxG15);
512         row15.Add(boxH15);
513         row15.Add(boxI15);
514         row15.Add(boxJ15);
515
516         row16.Add(boxA16);
517         row16.Add(boxB16);
518         row16.Add(boxC16);
519         row16.Add(boxD16);
520         row16.Add(boxE16);
521         row16.Add(boxF16);
522         row16.Add(boxG16);
523         row16.Add(boxH16);
524         row16.Add(boxI16);
525         row16.Add(boxJ16);
526
527         row17.Add(boxA17);
528         row17.Add(boxB17);
529         row17.Add(boxC17);
530         row17.Add(boxD17);
531         row17.Add(boxE17);
532         row17.Add(boxF17);
533         row17.Add(boxG17);
534         row17.Add(boxH17);
535         row17.Add(boxI17);
536         row17.Add(boxJ17);
537
538         row18.Add(boxA18);
539         row18.Add(boxB18);
```

```
540         row18.Add(boxC18);
541         row18.Add(boxD18);
542         row18.Add(boxE18);
543         row18.Add(boxF18);
544         row18.Add(boxG18);
545         row18.Add(boxH18);
546         row18.Add(boxI18);
547         row18.Add(boxJ18);
548
549         row19.Add(boxA19);
550         row19.Add(boxB19);
551         row19.Add(boxC19);
552         row19.Add(boxD19);
553         row19.Add(boxE19);
554         row19.Add(boxF19);
555         row19.Add(boxG19);
556         row19.Add(boxH19);
557         row19.Add(boxI19);
558         row19.Add(boxJ19);
559
560         row20.Add(boxA20);
561         row20.Add(boxB20);
562         row20.Add(boxC20);
563         row20.Add(boxD20);
564         row20.Add(boxE20);
565         row20.Add(boxF20);
566         row20.Add(boxG20);
567         row20.Add(boxH20);
568         row20.Add(boxI20);
569         row20.Add(boxJ20);
570     }
571     //Row by row, checks which rows have been checked.
572     void CheckRows()
573     {
574         if (openSpots.Count == 0)
575         {
576             if (!checkedRow1)
577             {
578                 checkedRow1 = true;
579                 CheckRow1();
580             }
581             else if (!checkedRow2)
582             {
583                 checkedRow2 = true;
584                 CheckRow2();
585             }
586             else if (!checkedRow3)
587             {
588                 checkedRow3 = true;
```

```
589                 CheckRow3();
590             }
591             else if(!checkedRow4)
592             {
593                 checkedRow4 = true;
594                 CheckRow4();
595             }
596             else if (!checkedRow5)
597             {
598                 checkedRow5 = true;
599                 CheckRow5();
600             }
601             else if (!checkedRow6)
602             {
603                 checkedRow6 = true;
604                 CheckRow6();
605             }
606             else if (!checkedRow7)
607             {
608                 checkedRow7 = true;
609                 CheckRow7();
610             }
611             else if (!checkedRow8)
612             {
613                 checkedRow8 = true;
614                 CheckRow8();
615             }
616             else if (!checkedRow9)
617             {
618                 checkedRow9 = true;
619                 CheckRow9();
620             }
621             else if (!checkedRow10)
622             {
623                 checkedRow10 = true;
624                 CheckRow10();
625             }
626             else if (!checkedRow11)
627             {
628                 checkedRow11 = true;
629                 CheckRow11();
630             }
631             else if (!checkedRow12)
632             {
633                 checkedRow12 = true;
634                 CheckRow12();
635             }
636             else if (!checkedRow13)
637             {
```

```
638                 checkedRow13 = true;
639                 CheckRow13();
640             }
641             else if (!checkedRow14)
642             {
643                 checkedRow14 = true;
644                 CheckRow14();
645             }
646             else if (!checkedRow15)
647             {
648                 checkedRow15 = true;
649                 CheckRow15();
650             }
651             else if (!checkedRow16)
652             {
653                 checkedRow16 = true;
654                 CheckRow16();
655             }
656             else if (!checkedRow17)
657             {
658                 checkedRow17 = true;
659                 CheckRow17();
660             }
661             else if (!checkedRow18)
662             {
663                 checkedRow18 = true;
664                 CheckRow18();
665             }
666             else if (!checkedRow19)
667             {
668                 checkedRow19 = true;
669                 CheckRow19();
670             }
671             else if (!checkedRow20)
672             {
673                 checkedRow20 = true;
674                 CheckRow20();
675             }
676         }
677     }
678     void CheckRow1()
679     {
680         //Check all spaces to see if there's anything there
681         foreach (Vector3 vec in row1)
682         {
683             //If something isn't there, check the 6 spaces above that
                  spot. If nothing's there,
684             //add the position to the row's Vacancy list and that index
                  to a the row's list of indeces.
```

```
685             if (!Physics.CheckBox(vec, cubeQuart))
686             {
687                 Vector3 vecPlus1 = new Vector3(vec.x, vec.y + 1, vec.z);
688                 Vector3 vecPlus2 = new Vector3(vec.x, vec.y + 2, vec.z);
689                 Vector3 vecPlus3 = new Vector3(vec.x, vec.y + 3, vec.z);
690                 Vector3 vecPlus4 = new Vector3(vec.x, vec.y + 4, vec.z);
691                 Vector3 vecPlus5 = new Vector3(vec.x, vec.y + 5, vec.z);
692                 Vector3 vecPlus6 = new Vector3(vec.x, vec.y + 6, vec.z);
693                 Vector3 vecPlus7 = new Vector3(vec.x, vec.y + 7, vec.z);
694                 Vector3 vecPlus8 = new Vector3(vec.x, vec.y + 8, vec.z);
695                 Vector3 vecPlus9 = new Vector3(vec.x, vec.y + 9, vec.z);
696                 Vector3 vecPlus10 = new Vector3(vec.x, vec.y + 10,
                      vec.z);
697
698                 if (!Physics.CheckBox(vecPlus1, cubeQuart) && !
                      Physics.CheckBox(vecPlus2, cubeQuart) && !
                      Physics.CheckBox(vecPlus3, cubeQuart)
699                       && !Physics.CheckBox(vecPlus4, cubeQuart) && !
                      Physics.CheckBox(vecPlus5, cubeQuart) && !
                      Physics.CheckBox(vecPlus6, cubeQuart)
700                       && !Physics.CheckBox(vecPlus7, cubeQuart) && !
                      Physics.CheckBox(vecPlus8, cubeQuart) && !
                      Physics.CheckBox(vecPlus9, cubeQuart)
701                           && !Physics.CheckBox(vecPlus10, cubeQuart))
702                 {
703                     row1Vacancies.Add(vec);
704                 }
705             }
706         }
707         foreach (Vector3 vec in row1Vacancies)
708         {
709             row1VacancyInts.Add(Mathf.RoundToInt(vec.z));
710         }
711         row1VacancyInts.Sort();
712         //Checks each of the verified empty spaces except the last one on
              the right
713         //To see if there's a block to the right of that block.
714         //If there is, add it to list openSpots. If no open spots, go to
              next row.
715         foreach (int i in row1VacancyInts)
716         {
717             if (i <= 4)
718             {
719                 if (row1VacancyInts.Contains(i + 1))
720                 {
721                     openSpots.Add(i);
722                 }
723             }
724         }
```

```
725         if (openSpots.Count == 0)
726         {
727             CheckRows();
728         }
729     }
730     void CheckRow2()
731     {
732         //Check all spaces to see if there's anything there.
733         foreach (Vector3 vec in row2)
734         {
735             //If something isn't there, check the 6 spaces above that spot. If nothing's there,
736             //add the position to the row's Vacancy list and that index to a the row's list of indeces.
737             if (!Physics.CheckBox(vec, cubeQuart))
738             {
739                 Vector3 vecPlus1 = new Vector3(vec.x, vec.y + 1, vec.z);
740                 Vector3 vecPlus2 = new Vector3(vec.x, vec.y + 2, vec.z);
741                 Vector3 vecPlus3 = new Vector3(vec.x, vec.y + 3, vec.z);
742                 Vector3 vecPlus4 = new Vector3(vec.x, vec.y + 4, vec.z);
743                 Vector3 vecPlus5 = new Vector3(vec.x, vec.y + 5, vec.z);
744                 Vector3 vecPlus6 = new Vector3(vec.x, vec.y + 6, vec.z);
745                 Vector3 vecPlus7 = new Vector3(vec.x, vec.y + 7, vec.z);
746                 Vector3 vecPlus8 = new Vector3(vec.x, vec.y + 8, vec.z);
747                 Vector3 vecPlus9 = new Vector3(vec.x, vec.y + 9, vec.z);
748                 Vector3 vecPlus10 = new Vector3(vec.x, vec.y + 10, vec.z);
749
750                 if (!Physics.CheckBox(vecPlus1, cubeQuart) && !Physics.CheckBox(vecPlus2, cubeQuart) && !Physics.CheckBox(vecPlus3, cubeQuart)
751                    && !Physics.CheckBox(vecPlus4, cubeQuart) && !Physics.CheckBox(vecPlus5, cubeQuart) && !Physics.CheckBox(vecPlus6, cubeQuart)
752                     && !Physics.CheckBox(vecPlus7, cubeQuart) && !Physics.CheckBox(vecPlus8, cubeQuart) && !Physics.CheckBox(vecPlus9, cubeQuart)
753                         && !Physics.CheckBox(vecPlus10, cubeQuart))
754                 {
755                     row2Vacancies.Add(vec);
756                 }
757             }
758         }
759         foreach (Vector3 vec in row2Vacancies)
760         {
761             row2VacancyInts.Add(Mathf.RoundToInt(vec.z));
762         }
763         row2VacancyInts.Sort();
764         //Checks eacb of the verified empty spaces except the last one on
```

```
            the right
765         //To see if there's a block to the right of that block.
766         //If there is, add it to list openSpots.
767         foreach (int i in row2VacancyInts)
768         {
769             if (i <= 4)
770             {
771                 if (row2VacancyInts.Contains(i + 1))
772                 {
773                     openSpots.Add(i);
774                 }
775             }
776         }
777         if (openSpots.Count == 0)
778         {
779             CheckRows();
780         }
781     }
782     void CheckRow3()
783     {
784         //Check all spaces to see if there's anything there.
785         foreach (Vector3 vec in row3)
786             //If something isn't there, check the 6 spaces above that
                  spot. If nothing's there,
787             //add the position to the row's Vacancy list and that index
                  to a the row's list of indeces.
788             if (!Physics.CheckBox(vec, cubeQuart))
789             {
790                 Vector3 vecPlus1 = new Vector3(vec.x, vec.y + 1, vec.z);
791                 Vector3 vecPlus2 = new Vector3(vec.x, vec.y + 2, vec.z);
792                 Vector3 vecPlus3 = new Vector3(vec.x, vec.y + 3, vec.z);
793                 Vector3 vecPlus4 = new Vector3(vec.x, vec.y + 4, vec.z);
794                 Vector3 vecPlus5 = new Vector3(vec.x, vec.y + 5, vec.z);
795                 Vector3 vecPlus6 = new Vector3(vec.x, vec.y + 6, vec.z);
796                 Vector3 vecPlus7 = new Vector3(vec.x, vec.y + 7, vec.z);
797                 Vector3 vecPlus8 = new Vector3(vec.x, vec.y + 8, vec.z);
798                 Vector3 vecPlus9 = new Vector3(vec.x, vec.y + 9, vec.z);
799                 Vector3 vecPlus10 = new Vector3(vec.x, vec.y + 10,
                      vec.z);
800
801                 if (!Physics.CheckBox(vecPlus1, cubeQuart) && !
                      Physics.CheckBox(vecPlus2, cubeQuart) && !
                      Physics.CheckBox(vecPlus3, cubeQuart)
802                     && !Physics.CheckBox(vecPlus4, cubeQuart) && !
                      Physics.CheckBox(vecPlus5, cubeQuart) && !
                      Physics.CheckBox(vecPlus6, cubeQuart)
803                     && !Physics.CheckBox(vecPlus7, cubeQuart) && !
                      Physics.CheckBox(vecPlus8, cubeQuart) && !
                      Physics.CheckBox(vecPlus9, cubeQuart)
```

```
804                             && !Physics.CheckBox(vecPlus10, cubeQuart))
805                 {
806                     row3Vacancies.Add(vec);
807                 }
808             }
809         foreach (Vector3 vec in row3Vacancies)
810         {
811             row3VacancyInts.Add(Mathf.RoundToInt(vec.z));
812         }
813         row3VacancyInts.Sort();
814         //Checks eacb of the verified empty spaces except the last one on the right
815         //To see if there's a block to the right of that block.
816         //If there is, add it to list openSpots.
817         foreach (int i in row3VacancyInts)
818         {
819             if (i <= 4)
820             {
821                 if (row3VacancyInts.Contains(i + 1))
822                 {
823                     openSpots.Add(i);
824                 }
825             }
826         }
827         if (openSpots.Count == 0)
828         {
829             CheckRows();
830         }
831     }
832     void CheckRow4()
833     {
834         //Check all spaces to see if there's anything there.
835         foreach (Vector3 vec in row4)
836             //If something isn't there, check the 6 spaces above that spot. If nothing's there,
837             //add the position to the row's Vacancy list and that index to a the row's list of indeces.
838             if (!Physics.CheckBox(vec, cubeQuart))
839             {
840                 Vector3 vecPlus1 = new Vector3(vec.x, vec.y + 1, vec.z);
841                 Vector3 vecPlus2 = new Vector3(vec.x, vec.y + 2, vec.z);
842                 Vector3 vecPlus3 = new Vector3(vec.x, vec.y + 3, vec.z);
843                 Vector3 vecPlus4 = new Vector3(vec.x, vec.y + 4, vec.z);
844                 Vector3 vecPlus5 = new Vector3(vec.x, vec.y + 5, vec.z);
845                 Vector3 vecPlus6 = new Vector3(vec.x, vec.y + 6, vec.z);
846                 Vector3 vecPlus7 = new Vector3(vec.x, vec.y + 7, vec.z);
847                 Vector3 vecPlus8 = new Vector3(vec.x, vec.y + 8, vec.z);
848                 Vector3 vecPlus9 = new Vector3(vec.x, vec.y + 9, vec.z);
849                 Vector3 vecPlus10 = new Vector3(vec.x, vec.y + 10,
```

```
                    vec.z);
850
851                 if (!Physics.CheckBox(vecPlus1, cubeQuart) && !
                      Physics.CheckBox(vecPlus2, cubeQuart) && !
                      Physics.CheckBox(vecPlus3, cubeQuart)
852                      && !Physics.CheckBox(vecPlus4, cubeQuart) && !
                     Physics.CheckBox(vecPlus5, cubeQuart) && !
                     Physics.CheckBox(vecPlus6, cubeQuart)
853                      && !Physics.CheckBox(vecPlus7, cubeQuart) && !
                     Physics.CheckBox(vecPlus8, cubeQuart) && !
                     Physics.CheckBox(vecPlus9, cubeQuart)
854                      && !Physics.CheckBox(vecPlus10, cubeQuart))
855                 {
856                     row4Vacancies.Add(vec);
857                 }
858             }
859         foreach (Vector3 vec in row4Vacancies)
860         {
861             row4VacancyInts.Add(Mathf.RoundToInt(vec.z));
862         }
863         row4VacancyInts.Sort();
864         //Checks eacb of the verified empty spaces except the last one on
              the right
865         //To see if there's a block to the right of that block.
866         //If there is, add it to list openSpots.
867         foreach (int i in row4VacancyInts)
868         {
869             if (i <= 4)
870             {
871                 if (row4VacancyInts.Contains(i + 1))
872                 {
873                     openSpots.Add(i);
874                 }
875             }
876         }
877         if (openSpots.Count == 0)
878         {
879             CheckRows();
880         }
881     }
882     void CheckRow5()
883     {
884         //Check all spaces to see if there's anything there.
885         foreach (Vector3 vec in row5)
886             //If something isn't there, check the 6 spaces above that
                  spot. If nothing's there,
887             //add the position to the row's Vacancy list and that index
                  to a the row's list of indeces.
888             if (!Physics.CheckBox(vec, cubeQuart))
```

```
889             {
890                 Vector3 vecPlus1 = new Vector3(vec.x, vec.y + 1, vec.z);
891                 Vector3 vecPlus2 = new Vector3(vec.x, vec.y + 2, vec.z);
892                 Vector3 vecPlus3 = new Vector3(vec.x, vec.y + 3, vec.z);
893                 Vector3 vecPlus4 = new Vector3(vec.x, vec.y + 4, vec.z);
894                 Vector3 vecPlus5 = new Vector3(vec.x, vec.y + 5, vec.z);
895                 Vector3 vecPlus6 = new Vector3(vec.x, vec.y + 6, vec.z);
896                 Vector3 vecPlus7 = new Vector3(vec.x, vec.y + 7, vec.z);
897                 Vector3 vecPlus8 = new Vector3(vec.x, vec.y + 8, vec.z);
898                 Vector3 vecPlus9 = new Vector3(vec.x, vec.y + 9, vec.z);
899                 Vector3 vecPlus10 = new Vector3(vec.x, vec.y + 10,
                      vec.z);
900 
901                 if (!Physics.CheckBox(vecPlus1, cubeQuart) && !
                      Physics.CheckBox(vecPlus2, cubeQuart) && !
                      Physics.CheckBox(vecPlus3, cubeQuart)
902                     && !Physics.CheckBox(vecPlus4, cubeQuart) && !
                     Physics.CheckBox(vecPlus5, cubeQuart) && !
                     Physics.CheckBox(vecPlus6, cubeQuart)
903                         && !Physics.CheckBox(vecPlus7, cubeQuart) && !
                     Physics.CheckBox(vecPlus8, cubeQuart) && !
                     Physics.CheckBox(vecPlus9, cubeQuart)
904                             && !Physics.CheckBox(vecPlus10, cubeQuart))
905                 {
906                     row5Vacancies.Add(vec);
907                 }
908             }
909         foreach (Vector3 vec in row5Vacancies)
910         {
911             row5VacancyInts.Add(Mathf.RoundToInt(vec.z));
912         }
913         row5VacancyInts.Sort();
914         //Checks eacb of the verified empty spaces except the last one on
              the right
915         //To see if there's a block to the right of that block.
916         //If there is, add it to list openSpots.
917         foreach (int i in row5VacancyInts)
918         {
919             if (i <= 4)
920             {
921                 if (row5VacancyInts.Contains(i + 1))
922                 {
923                     openSpots.Add(i);
924                 }
925             }
926         }
927         if (openSpots.Count == 0)
928         {
929             CheckRows();
```

```
930         }
931     }
932     void CheckRow6()
933     {
934         //Check all spaces to see if there's anything there.
935         foreach (Vector3 vec in row6)
936             //If something isn't there, check the 6 spaces above that spot. If nothing's there,
937             //add the position to the row's Vacancy list and that index to a the row's list of indeces.
938             if (!Physics.CheckBox(vec, cubeQuart))
939             {
940                 Vector3 vecPlus1 = new Vector3(vec.x, vec.y + 1, vec.z);
941                 Vector3 vecPlus2 = new Vector3(vec.x, vec.y + 2, vec.z);
942                 Vector3 vecPlus3 = new Vector3(vec.x, vec.y + 3, vec.z);
943                 Vector3 vecPlus4 = new Vector3(vec.x, vec.y + 4, vec.z);
944                 Vector3 vecPlus5 = new Vector3(vec.x, vec.y + 5, vec.z);
945                 Vector3 vecPlus6 = new Vector3(vec.x, vec.y + 6, vec.z);
946                 Vector3 vecPlus7 = new Vector3(vec.x, vec.y + 7, vec.z);
947                 Vector3 vecPlus8 = new Vector3(vec.x, vec.y + 8, vec.z);
948                 Vector3 vecPlus9 = new Vector3(vec.x, vec.y + 9, vec.z);
949                 Vector3 vecPlus10 = new Vector3(vec.x, vec.y + 10, vec.z);
950 
951                 if (!Physics.CheckBox(vecPlus1, cubeQuart) && !Physics.CheckBox(vecPlus2, cubeQuart) && !Physics.CheckBox(vecPlus3, cubeQuart)
952                    && !Physics.CheckBox(vecPlus4, cubeQuart) && !Physics.CheckBox(vecPlus5, cubeQuart) && !Physics.CheckBox(vecPlus6, cubeQuart)
953                    && !Physics.CheckBox(vecPlus7, cubeQuart) && !Physics.CheckBox(vecPlus8, cubeQuart) && !Physics.CheckBox(vecPlus9, cubeQuart)
954                         && !Physics.CheckBox(vecPlus10, cubeQuart))
955                 {
956                     row6Vacancies.Add(vec);
957                 }
958             }
959         foreach (Vector3 vec in row6Vacancies)
960         {
961             row6VacancyInts.Add(Mathf.RoundToInt(vec.z));
962         }
963         row6VacancyInts.Sort();
964         //Checks eacb of the verified empty spaces except the last one on the right
965         //To see if there's a block to the right of that block.
966         //If there is, add it to list openSpots.
967         foreach (int i in row6VacancyInts)
968         {
```

```
969             if (i <= 4)
970             {
971                 if (row6VacancyInts.Contains(i + 1))
972                 {
973                     openSpots.Add(i);
974                 }
975             }
976         }
977         if (openSpots.Count == 0)
978         {
979             CheckRows();
980         }
981     }
982     void CheckRow7()
983     {
984         //Check all spaces to see if there's anything there.
985         foreach (Vector3 vec in row7)
986             //If something isn't there, check the 6 spaces above that spot. If nothing's there,
987             //add the position to the row's Vacancy list and that index to a the row's list of indeces.
988             if (!Physics.CheckBox(vec, cubeQuart))
989             {
990                 Vector3 vecPlus1 = new Vector3(vec.x, vec.y + 1, vec.z);
991                 Vector3 vecPlus2 = new Vector3(vec.x, vec.y + 2, vec.z);
992                 Vector3 vecPlus3 = new Vector3(vec.x, vec.y + 3, vec.z);
993                 Vector3 vecPlus4 = new Vector3(vec.x, vec.y + 4, vec.z);
994                 Vector3 vecPlus5 = new Vector3(vec.x, vec.y + 5, vec.z);
995                 Vector3 vecPlus6 = new Vector3(vec.x, vec.y + 6, vec.z);
996                 Vector3 vecPlus7 = new Vector3(vec.x, vec.y + 7, vec.z);
997                 Vector3 vecPlus8 = new Vector3(vec.x, vec.y + 8, vec.z);
998                 Vector3 vecPlus9 = new Vector3(vec.x, vec.y + 9, vec.z);
999                 Vector3 vecPlus10 = new Vector3(vec.x, vec.y + 10, vec.z);
1000
1001                if (!Physics.CheckBox(vecPlus1, cubeQuart) && !Physics.CheckBox(vecPlus2, cubeQuart) && !Physics.CheckBox(vecPlus3, cubeQuart)
1002                   && !Physics.CheckBox(vecPlus4, cubeQuart) && !Physics.CheckBox(vecPlus5, cubeQuart) && !Physics.CheckBox(vecPlus6, cubeQuart)
1003                    && !Physics.CheckBox(vecPlus7, cubeQuart) && !Physics.CheckBox(vecPlus8, cubeQuart) && !Physics.CheckBox(vecPlus9, cubeQuart)
1004                        && !Physics.CheckBox(vecPlus10, cubeQuart))
1005                {
1006                    row7Vacancies.Add(vec);
1007                }
1008            }
```

```
1009          foreach (Vector3 vec in row7Vacancies)
1010          {
1011              row7VacancyInts.Add(Mathf.RoundToInt(vec.z));
1012          }
1013          row7VacancyInts.Sort();
1014          //Checks eacb of the verified empty spaces except the last one on the right
1015          //To see if there's a block to the right of that block.
1016          //If there is, add it to list openSpots.
1017          foreach (int i in row7VacancyInts)
1018          {
1019              if (i <= 4)
1020              {
1021                  if (row7VacancyInts.Contains(i + 1))
1022                  {
1023                      openSpots.Add(i);
1024                  }
1025              }
1026          }
1027          if (openSpots.Count == 0)
1028          {
1029              CheckRows();
1030          }
1031      }
1032      void CheckRow8()
1033      {
1034          //Check all spaces to see if there's anything there.
1035          foreach (Vector3 vec in row8)
1036              //If something isn't there, check the 6 spaces above that spot. If nothing's there,
1037              //add the position to the row's Vacancy list and that index to a the row's list of indeces.
1038              if (!Physics.CheckBox(vec, cubeQuart))
1039              {
1040                  Vector3 vecPlus1 = new Vector3(vec.x, vec.y + 1, vec.z);
1041                  Vector3 vecPlus2 = new Vector3(vec.x, vec.y + 2, vec.z);
1042                  Vector3 vecPlus3 = new Vector3(vec.x, vec.y + 3, vec.z);
1043                  Vector3 vecPlus4 = new Vector3(vec.x, vec.y + 4, vec.z);
1044                  Vector3 vecPlus5 = new Vector3(vec.x, vec.y + 5, vec.z);
1045                  Vector3 vecPlus6 = new Vector3(vec.x, vec.y + 6, vec.z);
1046                  Vector3 vecPlus7 = new Vector3(vec.x, vec.y + 7, vec.z);
1047                  Vector3 vecPlus8 = new Vector3(vec.x, vec.y + 8, vec.z);
1048                  Vector3 vecPlus9 = new Vector3(vec.x, vec.y + 9, vec.z);
1049                  Vector3 vecPlus10 = new Vector3(vec.x, vec.y + 10, vec.z);
1050
1051                  if (!Physics.CheckBox(vecPlus1, cubeQuart) && !
                        Physics.CheckBox(vecPlus2, cubeQuart) && !
                        Physics.CheckBox(vecPlus3, cubeQuart)
```

```
1052                        && !Physics.CheckBox(vecPlus4, cubeQuart) && !
                        Physics.CheckBox(vecPlus5, cubeQuart) && !
                        Physics.CheckBox(vecPlus6, cubeQuart)
1053                        && !Physics.CheckBox(vecPlus7, cubeQuart) && !
                        Physics.CheckBox(vecPlus8, cubeQuart) && !
                        Physics.CheckBox(vecPlus9, cubeQuart)
1054                         && !Physics.CheckBox(vecPlus10, cubeQuart))
1055                {
1056                    row8Vacancies.Add(vec);
1057                }
1058            }
1059        foreach (Vector3 vec in row8Vacancies)
1060        {
1061            row8VacancyInts.Add(Mathf.RoundToInt(vec.z));
1062        }
1063        row8VacancyInts.Sort();
1064        //Checks eacb of the verified empty spaces except the last one on
            the right
1065        //To see if there's a block to the right of that block.
1066        //If there is, add it to list openSpots.
1067        foreach (int i in row8VacancyInts)
1068        {
1069            if (i <= 4)
1070            {
1071                if (row8VacancyInts.Contains(i + 1))
1072                {
1073                    openSpots.Add(i);
1074                }
1075            }
1076        }
1077        if (openSpots.Count == 0)
1078        {
1079            CheckRows();
1080        }
1081    }
1082    void CheckRow9()
1083    {
1084        //Check all spaces to see if there's anything there.
1085        foreach (Vector3 vec in row9)
1086            //If something isn't there, check the 6 spaces above that
              spot. If nothing's there,
1087            //add the position to the row's Vacancy list and that index
              to a the row's list of indeces.
1088            if (!Physics.CheckBox(vec, cubeQuart))
1089            {
1090                Vector3 vecPlus1 = new Vector3(vec.x, vec.y + 1, vec.z);
1091                Vector3 vecPlus2 = new Vector3(vec.x, vec.y + 2, vec.z);
1092                Vector3 vecPlus3 = new Vector3(vec.x, vec.y + 3, vec.z);
1093                Vector3 vecPlus4 = new Vector3(vec.x, vec.y + 4, vec.z);
```

```
1094                Vector3 vecPlus5 = new Vector3(vec.x, vec.y + 5, vec.z);
1095                Vector3 vecPlus6 = new Vector3(vec.x, vec.y + 6, vec.z);
1096                Vector3 vecPlus7 = new Vector3(vec.x, vec.y + 7, vec.z);
1097                Vector3 vecPlus8 = new Vector3(vec.x, vec.y + 8, vec.z);
1098                Vector3 vecPlus9 = new Vector3(vec.x, vec.y + 9, vec.z);
1099                Vector3 vecPlus10 = new Vector3(vec.x, vec.y + 10,
                      vec.z);
1100
1101                if (!Physics.CheckBox(vecPlus1, cubeQuart) && !
                      Physics.CheckBox(vecPlus2, cubeQuart) && !
                      Physics.CheckBox(vecPlus3, cubeQuart)
1102                    && !Physics.CheckBox(vecPlus4, cubeQuart) && !
                     Physics.CheckBox(vecPlus5, cubeQuart) && !
                     Physics.CheckBox(vecPlus6, cubeQuart)
1103                     && !Physics.CheckBox(vecPlus7, cubeQuart) && !
                     Physics.CheckBox(vecPlus8, cubeQuart) && !
                     Physics.CheckBox(vecPlus9, cubeQuart)
1104                          && !Physics.CheckBox(vecPlus10, cubeQuart))
1105                {
1106                    row9Vacancies.Add(vec);
1107                }
1108            }
1109        foreach (Vector3 vec in row9Vacancies)
1110        {
1111            row9VacancyInts.Add(Mathf.RoundToInt(vec.z));
1112        }
1113        row9VacancyInts.Sort();
1114        //Checks eacb of the verified empty spaces except the last one on
             the right
1115        //To see if there's a block to the right of that block.
1116        //If there is, add it to list openSpots.
1117        foreach (int i in row9VacancyInts)
1118        {
1119            if (i <= 4)
1120            {
1121                if (row9VacancyInts.Contains(i + 1))
1122                {
1123                    openSpots.Add(i);
1124                }
1125            }
1126        }
1127        if (openSpots.Count == 0)
1128        {
1129            CheckRows();
1130        }
1131    }
1132    void CheckRow10()
1133    {
1134        //Check all spaces to see if there's anything there.
```

```
1135            foreach (Vector3 vec in row10)
1136                //If something isn't there, check the 6 spaces above that spot. If nothing's there,
1137                //add the position to the row's Vacancy list and that index to a the row's list of indeces.
1138                if (!Physics.CheckBox(vec, cubeQuart))
1139                {
1140                    Vector3 vecPlus1 = new Vector3(vec.x, vec.y + 1, vec.z);
1141                    Vector3 vecPlus2 = new Vector3(vec.x, vec.y + 2, vec.z);
1142                    Vector3 vecPlus3 = new Vector3(vec.x, vec.y + 3, vec.z);
1143                    Vector3 vecPlus4 = new Vector3(vec.x, vec.y + 4, vec.z);
1144                    Vector3 vecPlus5 = new Vector3(vec.x, vec.y + 5, vec.z);
1145                    Vector3 vecPlus6 = new Vector3(vec.x, vec.y + 6, vec.z);
1146                    Vector3 vecPlus7 = new Vector3(vec.x, vec.y + 7, vec.z);
1147                    Vector3 vecPlus8 = new Vector3(vec.x, vec.y + 8, vec.z);
1148                    Vector3 vecPlus9 = new Vector3(vec.x, vec.y + 9, vec.z);
1149                    Vector3 vecPlus10 = new Vector3(vec.x, vec.y + 10, vec.z);
1150
1151                    if (!Physics.CheckBox(vecPlus1, cubeQuart) && !Physics.CheckBox(vecPlus2, cubeQuart) && !Physics.CheckBox(vecPlus3, cubeQuart)
1152                       && !Physics.CheckBox(vecPlus4, cubeQuart) && !Physics.CheckBox(vecPlus5, cubeQuart) && !Physics.CheckBox(vecPlus6, cubeQuart)
1153                        && !Physics.CheckBox(vecPlus7, cubeQuart) && !Physics.CheckBox(vecPlus8, cubeQuart) && !Physics.CheckBox(vecPlus9, cubeQuart)
1154                            && !Physics.CheckBox(vecPlus10, cubeQuart))
1155                    {
1156                        row10Vacancies.Add(vec);
1157                    }
1158                }
1159            foreach (Vector3 vec in row10Vacancies)
1160            {
1161                row10VacancyInts.Add(Mathf.RoundToInt(vec.z));
1162            }
1163            row10VacancyInts.Sort();
1164            //Checks eacb of the verified empty spaces except the last one on the right
1165            //To see if there's a block to the right of that block.
1166            //If there is, add it to list openSpots.
1167            foreach (int i in row10VacancyInts)
1168            {
1169                if (i <= 4)
1170                {
1171                    if (row10VacancyInts.Contains(i + 1))
1172                    {
1173                        openSpots.Add(i);
```

```
1174                }
1175            }
1176        }
1177        if (openSpots.Count == 0)
1178        {
1179            CheckRows();
1180        }
1181    }
1182    void CheckRow11()
1183    {
1184        //Check all spaces to see if there's anything there
1185        foreach (Vector3 vec in row11)
1186            //If something isn't there, check the 6 spaces above that spot. If nothing's there,
1187            //add the position to the row's Vacancy list and that index to a the row's list of indeces.
1188            if (!Physics.CheckBox(vec, cubeQuart))
1189            {
1190                Vector3 vecPlus1 = new Vector3(vec.x, vec.y + 1, vec.z);
1191                Vector3 vecPlus2 = new Vector3(vec.x, vec.y + 2, vec.z);
1192                Vector3 vecPlus3 = new Vector3(vec.x, vec.y + 3, vec.z);
1193                Vector3 vecPlus4 = new Vector3(vec.x, vec.y + 4, vec.z);
1194                Vector3 vecPlus5 = new Vector3(vec.x, vec.y + 5, vec.z);
1195                Vector3 vecPlus6 = new Vector3(vec.x, vec.y + 6, vec.z);
1196                Vector3 vecPlus7 = new Vector3(vec.x, vec.y + 7, vec.z);
1197                Vector3 vecPlus8 = new Vector3(vec.x, vec.y + 8, vec.z);
1198                Vector3 vecPlus9 = new Vector3(vec.x, vec.y + 9, vec.z);
1199                Vector3 vecPlus10 = new Vector3(vec.x, vec.y + 10, vec.z);
1200
1201                if (!Physics.CheckBox(vecPlus1, cubeQuart) && !Physics.CheckBox(vecPlus2, cubeQuart) && !Physics.CheckBox(vecPlus3, cubeQuart)
1202                   && !Physics.CheckBox(vecPlus4, cubeQuart) && !Physics.CheckBox(vecPlus5, cubeQuart) && !Physics.CheckBox(vecPlus6, cubeQuart)
1203                    && !Physics.CheckBox(vecPlus7, cubeQuart) && !Physics.CheckBox(vecPlus8, cubeQuart) && !Physics.CheckBox(vecPlus9, cubeQuart)
1204                        && !Physics.CheckBox(vecPlus10, cubeQuart))
1205                {
1206                    row11Vacancies.Add(vec);
1207                }
1208            }
1209        foreach (Vector3 vec in row11Vacancies)
1210        {
1211            row11VacancyInts.Add(Mathf.RoundToInt(vec.z));
1212        }
1213        row1VacancyInts.Sort();
```

```
1214          //Checks eacb of the verified empty spaces except the last one on
                 the right
1215          //To see if there's a block to the right of that block.
1216          //If there is, add it to list openSpots.
1217          foreach (int i in row11VacancyInts)
1218          {
1219              if (i <= 4)
1220              {
1221                  if (row11VacancyInts.Contains(i + 1))
1222                  {
1223                      openSpots.Add(i);
1224                  }
1225              }
1226          }
1227          if (openSpots.Count == 0)
1228          {
1229              CheckRows();
1230          }
1231      }
1232      void CheckRow12()
1233      {
1234          //Check all spaces to see if there's anything there.
1235          foreach (Vector3 vec in row12)
1236              //If something isn't there, check the 6 spaces above that
                    spot. If nothing's there,
1237              //add the position to the row's Vacancy list and that index
                    to a the row's list of indeces.
1238              if (!Physics.CheckBox(vec, cubeQuart))
1239              {
1240                  Vector3 vecPlus1 = new Vector3(vec.x, vec.y + 1, vec.z);
1241                  Vector3 vecPlus2 = new Vector3(vec.x, vec.y + 2, vec.z);
1242                  Vector3 vecPlus3 = new Vector3(vec.x, vec.y + 3, vec.z);
1243                  Vector3 vecPlus4 = new Vector3(vec.x, vec.y + 4, vec.z);
1244                  Vector3 vecPlus5 = new Vector3(vec.x, vec.y + 5, vec.z);
1245                  Vector3 vecPlus6 = new Vector3(vec.x, vec.y + 6, vec.z);
1246                  Vector3 vecPlus7 = new Vector3(vec.x, vec.y + 7, vec.z);
1247                  Vector3 vecPlus8 = new Vector3(vec.x, vec.y + 8, vec.z);
1248                  Vector3 vecPlus9 = new Vector3(vec.x, vec.y + 9, vec.z);
1249                  Vector3 vecPlus10 = new Vector3(vec.x, vec.y + 10,
                        vec.z);
1250
1251                  if (!Physics.CheckBox(vecPlus1, cubeQuart) && !
                        Physics.CheckBox(vecPlus2, cubeQuart) && !
                        Physics.CheckBox(vecPlus3, cubeQuart)
1252                         && !Physics.CheckBox(vecPlus4, cubeQuart) && !
                        Physics.CheckBox(vecPlus5, cubeQuart) && !
                        Physics.CheckBox(vecPlus6, cubeQuart)
1253                         && !Physics.CheckBox(vecPlus7, cubeQuart) && !
                        Physics.CheckBox(vecPlus8, cubeQuart) && !
```

```
                    Physics.CheckBox(vecPlus9, cubeQuart)
1254                     && !Physics.CheckBox(vecPlus10, cubeQuart))
1255                {
1256                    row12Vacancies.Add(vec);
1257                }
1258            }
1259        foreach (Vector3 vec in row12Vacancies)
1260        {
1261            row12VacancyInts.Add(Mathf.RoundToInt(vec.z));
1262        }
1263        row12VacancyInts.Sort();
1264        //Checks eacb of the verified empty spaces except the last one on the right
1265        //To see if there's a block to the right of that block.
1266        //If there is, add it to list openSpots.
1267        foreach (int i in row12VacancyInts)
1268        {
1269            if (i <= 4)
1270            {
1271                if (row12VacancyInts.Contains(i + 1))
1272                {
1273                    openSpots.Add(i);
1274                }
1275            }
1276        }
1277        if (openSpots.Count == 0)
1278        {
1279            CheckRows();
1280        }
1281    }
1282    void CheckRow13()
1283    {
1284        //Check all spaces to see if there's anything there.
1285        foreach (Vector3 vec in row13)
1286            //If something isn't there, check the 6 spaces above that spot. If nothing's there,
1287            //add the position to the row's Vacancy list and that index to a the row's list of indeces.
1288            if (!Physics.CheckBox(vec, cubeQuart))
1289            {
1290                Vector3 vecPlus1 = new Vector3(vec.x, vec.y + 1, vec.z);
1291                Vector3 vecPlus2 = new Vector3(vec.x, vec.y + 2, vec.z);
1292                Vector3 vecPlus3 = new Vector3(vec.x, vec.y + 3, vec.z);
1293                Vector3 vecPlus4 = new Vector3(vec.x, vec.y + 4, vec.z);
1294                Vector3 vecPlus5 = new Vector3(vec.x, vec.y + 5, vec.z);
1295                Vector3 vecPlus6 = new Vector3(vec.x, vec.y + 6, vec.z);
1296                Vector3 vecPlus7 = new Vector3(vec.x, vec.y + 7, vec.z);
1297                Vector3 vecPlus8 = new Vector3(vec.x, vec.y + 8, vec.z);
1298                Vector3 vecPlus9 = new Vector3(vec.x, vec.y + 9, vec.z);
```

```
1299                    Vector3 vecPlus10 = new Vector3(vec.x, vec.y + 10,
                          vec.z);
1300
1301                    if (!Physics.CheckBox(vecPlus1, cubeQuart) && !
                          Physics.CheckBox(vecPlus2, cubeQuart) && !
                          Physics.CheckBox(vecPlus3, cubeQuart)
1302                        && !Physics.CheckBox(vecPlus4, cubeQuart) && !
                         Physics.CheckBox(vecPlus5, cubeQuart) && !
                         Physics.CheckBox(vecPlus6, cubeQuart)
1303                         && !Physics.CheckBox(vecPlus7, cubeQuart) && !
                         Physics.CheckBox(vecPlus8, cubeQuart) && !
                         Physics.CheckBox(vecPlus9, cubeQuart)
1304                             && !Physics.CheckBox(vecPlus10, cubeQuart))
1305                    {
1306                        row13Vacancies.Add(vec);
1307                    }
1308                }
1309            foreach (Vector3 vec in row13Vacancies)
1310            {
1311                row13VacancyInts.Add(Mathf.RoundToInt(vec.z));
1312            }
1313            row13VacancyInts.Sort();
1314            //Checks eacb of the verified empty spaces except the last one on
                  the right
1315            //To see if there's a block to the right of that block.
1316            //If there is, add it to list openSpots.
1317            foreach (int i in row13VacancyInts)
1318            {
1319                if (i <= 4)
1320                {
1321                    if (row13VacancyInts.Contains(i + 1))
1322                    {
1323                        openSpots.Add(i);
1324                    }
1325                }
1326            }
1327            if (openSpots.Count == 0)
1328            {
1329                CheckRows();
1330            }
1331        }
1332        void CheckRow14()
1333        {
1334            //Check all spaces to see if there's anything there.
1335            foreach (Vector3 vec in row14)
1336                //If something isn't there, check the 6 spaces above that
                      spot. If nothing's there,
1337                //add the position to the row's Vacancy list and that index
                      to a the row's list of indeces.
```

```
1338            if (!Physics.CheckBox(vec, cubeQuart))
1339            {
1340                Vector3 vecPlus1 = new Vector3(vec.x, vec.y + 1, vec.z);
1341                Vector3 vecPlus2 = new Vector3(vec.x, vec.y + 2, vec.z);
1342                Vector3 vecPlus3 = new Vector3(vec.x, vec.y + 3, vec.z);
1343                Vector3 vecPlus4 = new Vector3(vec.x, vec.y + 4, vec.z);
1344                Vector3 vecPlus5 = new Vector3(vec.x, vec.y + 5, vec.z);
1345                Vector3 vecPlus6 = new Vector3(vec.x, vec.y + 6, vec.z);
1346                Vector3 vecPlus7 = new Vector3(vec.x, vec.y + 7, vec.z);
1347                Vector3 vecPlus8 = new Vector3(vec.x, vec.y + 8, vec.z);
1348                Vector3 vecPlus9 = new Vector3(vec.x, vec.y + 9, vec.z);
1349                Vector3 vecPlus10 = new Vector3(vec.x, vec.y + 10,
                      vec.z);
1350
1351                if (!Physics.CheckBox(vecPlus1, cubeQuart) && !
                      Physics.CheckBox(vecPlus2, cubeQuart) && !
                      Physics.CheckBox(vecPlus3, cubeQuart)
1352                    && !Physics.CheckBox(vecPlus4, cubeQuart) && !
                     Physics.CheckBox(vecPlus5, cubeQuart) && !
                     Physics.CheckBox(vecPlus6, cubeQuart)
1353                     && !Physics.CheckBox(vecPlus7, cubeQuart) && !
                     Physics.CheckBox(vecPlus8, cubeQuart) && !
                     Physics.CheckBox(vecPlus9, cubeQuart)
1354                         && !Physics.CheckBox(vecPlus10, cubeQuart))
1355                {
1356                    row14Vacancies.Add(vec);
1357                }
1358            }
1359        foreach (Vector3 vec in row14Vacancies)
1360        {
1361            row14VacancyInts.Add(Mathf.RoundToInt(vec.z));
1362        }
1363        row14VacancyInts.Sort();
1364        //Checks eacb of the verified empty spaces except the last one on
             the right
1365        //To see if there's a block to the right of that block.
1366        //If there is, add it to list openSpots.
1367        foreach (int i in row14VacancyInts)
1368        {
1369            if (i <= 4)
1370            {
1371                if (row14VacancyInts.Contains(i + 1))
1372                {
1373                    openSpots.Add(i);
1374                }
1375            }
1376        }
1377        if (openSpots.Count == 0)
1378        {
```

```
1379                CheckRows();
1380            }
1381        }
1382        void CheckRow15()
1383        {
1384            //Check all spaces to see if there's anything there.
1385            foreach (Vector3 vec in row15)
1386                //If something isn't there, check the 6 spaces above that spot. If nothing's there,
1387                //add the position to the row's Vacancy list and that index to a the row's list of indeces.
1388                if (!Physics.CheckBox(vec, cubeQuart))
1389                {
1390                    Vector3 vecPlus1 = new Vector3(vec.x, vec.y + 1, vec.z);
1391                    Vector3 vecPlus2 = new Vector3(vec.x, vec.y + 2, vec.z);
1392                    Vector3 vecPlus3 = new Vector3(vec.x, vec.y + 3, vec.z);
1393                    Vector3 vecPlus4 = new Vector3(vec.x, vec.y + 4, vec.z);
1394                    Vector3 vecPlus5 = new Vector3(vec.x, vec.y + 5, vec.z);
1395                    Vector3 vecPlus6 = new Vector3(vec.x, vec.y + 6, vec.z);
1396                    Vector3 vecPlus7 = new Vector3(vec.x, vec.y + 7, vec.z);
1397                    Vector3 vecPlus8 = new Vector3(vec.x, vec.y + 8, vec.z);
1398                    Vector3 vecPlus9 = new Vector3(vec.x, vec.y + 9, vec.z);
1399                    Vector3 vecPlus10 = new Vector3(vec.x, vec.y + 10, vec.z);
1400
1401                    if (!Physics.CheckBox(vecPlus1, cubeQuart) && !Physics.CheckBox(vecPlus2, cubeQuart) && !Physics.CheckBox(vecPlus3, cubeQuart)
1402                       && !Physics.CheckBox(vecPlus4, cubeQuart) && !Physics.CheckBox(vecPlus5, cubeQuart) && !Physics.CheckBox(vecPlus6, cubeQuart)
1403                          && !Physics.CheckBox(vecPlus7, cubeQuart) && !Physics.CheckBox(vecPlus8, cubeQuart) && !Physics.CheckBox(vecPlus9, cubeQuart)
1404                               && !Physics.CheckBox(vecPlus10, cubeQuart))
1405                    {
1406                        row15Vacancies.Add(vec);
1407                    }
1408                }
1409            foreach (Vector3 vec in row15Vacancies)
1410            {
1411                row15VacancyInts.Add(Mathf.RoundToInt(vec.z));
1412            }
1413            row15VacancyInts.Sort();
1414            //Checks eacb of the verified empty spaces except the last one on the right
1415            //To see if there's a block to the right of that block.
1416            //If there is, add it to list openSpots.
1417            foreach (int i in row15VacancyInts)
```

```
1418            {
1419                if (i <= 4)
1420                {
1421                    if (row15VacancyInts.Contains(i + 1))
1422                    {
1423                        openSpots.Add(i);
1424                    }
1425                }
1426            }
1427            if (openSpots.Count == 0)
1428            {
1429                CheckRows();
1430            }
1431        }
1432        void CheckRow16()
1433        {
1434            //Check all spaces to see if there's anything there.
1435            foreach (Vector3 vec in row16)
1436                //If something isn't there, check the 6 spaces above that
                      spot. If nothing's there,
1437                //add the position to the row's Vacancy list and that index
                      to a the row's list of indeces.
1438                if (!Physics.CheckBox(vec, cubeQuart))
1439                {
1440                    Vector3 vecPlus1 = new Vector3(vec.x, vec.y + 1, vec.z);
1441                    Vector3 vecPlus2 = new Vector3(vec.x, vec.y + 2, vec.z);
1442                    Vector3 vecPlus3 = new Vector3(vec.x, vec.y + 3, vec.z);
1443                    Vector3 vecPlus4 = new Vector3(vec.x, vec.y + 4, vec.z);
1444                    Vector3 vecPlus5 = new Vector3(vec.x, vec.y + 5, vec.z);
1445                    Vector3 vecPlus6 = new Vector3(vec.x, vec.y + 6, vec.z);
1446                    Vector3 vecPlus7 = new Vector3(vec.x, vec.y + 7, vec.z);
1447                    Vector3 vecPlus8 = new Vector3(vec.x, vec.y + 8, vec.z);
1448                    Vector3 vecPlus9 = new Vector3(vec.x, vec.y + 9, vec.z);
1449                    Vector3 vecPlus10 = new Vector3(vec.x, vec.y + 10,
                          vec.z);
1450
1451                    if (!Physics.CheckBox(vecPlus1, cubeQuart) && !
                          Physics.CheckBox(vecPlus2, cubeQuart) && !
                          Physics.CheckBox(vecPlus3, cubeQuart)
1452                           && !Physics.CheckBox(vecPlus4, cubeQuart) && !
                           Physics.CheckBox(vecPlus5, cubeQuart) && !
                           Physics.CheckBox(vecPlus6, cubeQuart)
1453                            && !Physics.CheckBox(vecPlus7, cubeQuart) && !
                           Physics.CheckBox(vecPlus8, cubeQuart) && !
                           Physics.CheckBox(vecPlus9, cubeQuart)
1454                                && !Physics.CheckBox(vecPlus10, cubeQuart))
1455                    {
1456                        row16Vacancies.Add(vec);
1457                    }
```

```
1458            }
1459        foreach (Vector3 vec in row16Vacancies)
1460        {
1461            row16VacancyInts.Add(Mathf.RoundToInt(vec.z));
1462        }
1463        row16VacancyInts.Sort();
1464        //Checks eacb of the verified empty spaces except the last one on the right
1465        //To see if there's a block to the right of that block.
1466        //If there is, add it to list openSpots.
1467        foreach (int i in row16VacancyInts)
1468        {
1469            if (i <= 4)
1470            {
1471                if (row16VacancyInts.Contains(i + 1))
1472                {
1473                    openSpots.Add(i);
1474                }
1475            }
1476        }
1477        if (openSpots.Count == 0)
1478        {
1479            CheckRows();
1480        }
1481    }
1482    void CheckRow17()
1483    {
1484        //Check all spaces to see if there's anything there.
1485        foreach (Vector3 vec in row17)
1486            //If something isn't there, check the 6 spaces above that spot. If nothing's there,
1487            //add the position to the row's Vacancy list and that index to a the row's list of indeces.
1488            if (!Physics.CheckBox(vec, cubeQuart))
1489            {
1490                Vector3 vecPlus1 = new Vector3(vec.x, vec.y + 1, vec.z);
1491                Vector3 vecPlus2 = new Vector3(vec.x, vec.y + 2, vec.z);
1492                Vector3 vecPlus3 = new Vector3(vec.x, vec.y + 3, vec.z);
1493                Vector3 vecPlus4 = new Vector3(vec.x, vec.y + 4, vec.z);
1494                Vector3 vecPlus5 = new Vector3(vec.x, vec.y + 5, vec.z);
1495                Vector3 vecPlus6 = new Vector3(vec.x, vec.y + 6, vec.z);
1496                Vector3 vecPlus7 = new Vector3(vec.x, vec.y + 7, vec.z);
1497                Vector3 vecPlus8 = new Vector3(vec.x, vec.y + 8, vec.z);
1498                Vector3 vecPlus9 = new Vector3(vec.x, vec.y + 9, vec.z);
1499                Vector3 vecPlus10 = new Vector3(vec.x, vec.y + 10, vec.z);
1500
1501                if (!Physics.CheckBox(vecPlus1, cubeQuart) && !
                    Physics.CheckBox(vecPlus2, cubeQuart) && !
```

```
                    Physics.CheckBox(vecPlus3, cubeQuart)
1502                   && !Physics.CheckBox(vecPlus4, cubeQuart) && !
                    Physics.CheckBox(vecPlus5, cubeQuart) && !
                    Physics.CheckBox(vecPlus6, cubeQuart)
1503                    && !Physics.CheckBox(vecPlus7, cubeQuart) && !
                    Physics.CheckBox(vecPlus8, cubeQuart) && !
                    Physics.CheckBox(vecPlus9, cubeQuart)
1504                     && !Physics.CheckBox(vecPlus10, cubeQuart))
1505                {
1506                    row17Vacancies.Add(vec);
1507                }
1508            }
1509        foreach (Vector3 vec in row17Vacancies)
1510        {
1511            row17VacancyInts.Add(Mathf.RoundToInt(vec.z));
1512        }
1513        row17VacancyInts.Sort();
1514        //Checks eacb of the verified empty spaces except the last one on
               the right
1515        //To see if there's a block to the right of that block.
1516        //If there is, add it to list openSpots.
1517        foreach (int i in row17VacancyInts)
1518        {
1519            if (i <= 4)
1520            {
1521                if (row17VacancyInts.Contains(i + 1))
1522                {
1523                    openSpots.Add(i);
1524                }
1525            }
1526        }
1527        if (openSpots.Count == 0)
1528        {
1529            CheckRows();
1530        }
1531    }
1532    void CheckRow18()
1533    {
1534        //Check all spaces to see if there's anything there.
1535        foreach (Vector3 vec in row18)
1536            //If something isn't there, check the 6 spaces above that
                  spot. If nothing's there,
1537            //add the position to the row's Vacancy list and that index
                  to a the row's list of indeces.
1538            if (!Physics.CheckBox(vec, cubeQuart))
1539            {
1540                Vector3 vecPlus1 = new Vector3(vec.x, vec.y + 1, vec.z);
1541                Vector3 vecPlus2 = new Vector3(vec.x, vec.y + 2, vec.z);
1542                Vector3 vecPlus3 = new Vector3(vec.x, vec.y + 3, vec.z);
```

```
1543                Vector3 vecPlus4 = new Vector3(vec.x, vec.y + 4, vec.z);
1544                Vector3 vecPlus5 = new Vector3(vec.x, vec.y + 5, vec.z);
1545                Vector3 vecPlus6 = new Vector3(vec.x, vec.y + 6, vec.z);
1546                Vector3 vecPlus7 = new Vector3(vec.x, vec.y + 7, vec.z);
1547                Vector3 vecPlus8 = new Vector3(vec.x, vec.y + 8, vec.z);
1548                Vector3 vecPlus9 = new Vector3(vec.x, vec.y + 9, vec.z);
1549                Vector3 vecPlus10 = new Vector3(vec.x, vec.y + 10,
                      vec.z);
1550
1551                if (!Physics.CheckBox(vecPlus1, cubeQuart) && !
                      Physics.CheckBox(vecPlus2, cubeQuart) && !
                      Physics.CheckBox(vecPlus3, cubeQuart)
1552                       && !Physics.CheckBox(vecPlus4, cubeQuart) && !
                       Physics.CheckBox(vecPlus5, cubeQuart) && !
                       Physics.CheckBox(vecPlus6, cubeQuart)
1553                           && !Physics.CheckBox(vecPlus7, cubeQuart) && !
                       Physics.CheckBox(vecPlus8, cubeQuart) && !
                       Physics.CheckBox(vecPlus9, cubeQuart)
1554                            && !Physics.CheckBox(vecPlus10, cubeQuart))
1555                {
1556                    row18Vacancies.Add(vec);
1557                }
1558            }
1559        foreach (Vector3 vec in row18Vacancies)
1560        {
1561            row18VacancyInts.Add(Mathf.RoundToInt(vec.z));
1562        }
1563        row18VacancyInts.Sort();
1564        //Checks eacb of the verified empty spaces except the last one on
              the right
1565        //To see if there's a block to the right of that block.
1566        //If there is, add it to list openSpots.
1567        foreach (int i in row18VacancyInts)
1568        {
1569            if (i <= 4)
1570            {
1571                if (row18VacancyInts.Contains(i + 1))
1572                {
1573                    openSpots.Add(i);
1574                }
1575            }
1576        }
1577        if (openSpots.Count == 0)
1578        {
1579            CheckRows();
1580        }
1581    }
1582    void CheckRow19()
1583    {
```

```
1584          //Check all spaces to see if there's anything there.
1585          foreach (Vector3 vec in row19)
1586              //If something isn't there, check the 6 spaces above that spot. If nothing's there,
1587              //add the position to the row's Vacancy list and that index to a the row's list of indeces.
1588              if (!Physics.CheckBox(vec, cubeQuart))
1589              {
1590                  Vector3 vecPlus1 = new Vector3(vec.x, vec.y + 1, vec.z);
1591                  Vector3 vecPlus2 = new Vector3(vec.x, vec.y + 2, vec.z);
1592                  Vector3 vecPlus3 = new Vector3(vec.x, vec.y + 3, vec.z);
1593                  Vector3 vecPlus4 = new Vector3(vec.x, vec.y + 4, vec.z);
1594                  Vector3 vecPlus5 = new Vector3(vec.x, vec.y + 5, vec.z);
1595                  Vector3 vecPlus6 = new Vector3(vec.x, vec.y + 6, vec.z);
1596                  Vector3 vecPlus7 = new Vector3(vec.x, vec.y + 7, vec.z);
1597                  Vector3 vecPlus8 = new Vector3(vec.x, vec.y + 8, vec.z);
1598                  Vector3 vecPlus9 = new Vector3(vec.x, vec.y + 9, vec.z);
1599                  Vector3 vecPlus10 = new Vector3(vec.x, vec.y + 10, vec.z);
1600
1601                  if (!Physics.CheckBox(vecPlus1, cubeQuart) && !Physics.CheckBox(vecPlus2, cubeQuart) && !Physics.CheckBox(vecPlus3, cubeQuart)
1602                     && !Physics.CheckBox(vecPlus4, cubeQuart) && !Physics.CheckBox(vecPlus5, cubeQuart) && !Physics.CheckBox(vecPlus6, cubeQuart)
1603                        && !Physics.CheckBox(vecPlus7, cubeQuart) && !Physics.CheckBox(vecPlus8, cubeQuart) && !Physics.CheckBox(vecPlus9, cubeQuart)
1604                           && !Physics.CheckBox(vecPlus10, cubeQuart))
1605                  {
1606                      row19Vacancies.Add(vec);
1607                  }
1608              }
1609          foreach (Vector3 vec in row19Vacancies)
1610          {
1611              row19VacancyInts.Add(Mathf.RoundToInt(vec.z));
1612          }
1613          row19VacancyInts.Sort();
1614          //Checks eacb of the verified empty spaces except the last one on the right
1615          //To see if there's a block to the right of that block.
1616          //If there is, add it to list openSpots.
1617          foreach (int i in row19VacancyInts)
1618          {
1619              if (i <= 4)
1620              {
1621                  if (row19VacancyInts.Contains(i + 1))
1622                  {
```

```
1623                        openSpots.Add(i);
1624                    }
1625                }
1626            }
1627            if (openSpots.Count == 0)
1628            {
1629                CheckRows();
1630            }
1631        }
1632        void CheckRow20()
1633        {
1634            //Check all spaces to see if there's anything there.
1635            foreach (Vector3 vec in row20)
1636                //If something isn't there, check the 6 spaces above that spot. If nothing's there,
1637                //add the position to the row's Vacancy list and that index to a the row's list of indeces.
1638                if (!Physics.CheckBox(vec, cubeQuart))
1639                {
1640                    Vector3 vecPlus1 = new Vector3(vec.x, vec.y + 1, vec.z);
1641                    Vector3 vecPlus2 = new Vector3(vec.x, vec.y + 2, vec.z);
1642                    Vector3 vecPlus3 = new Vector3(vec.x, vec.y + 3, vec.z);
1643                    Vector3 vecPlus4 = new Vector3(vec.x, vec.y + 4, vec.z);
1644                    Vector3 vecPlus5 = new Vector3(vec.x, vec.y + 5, vec.z);
1645                    Vector3 vecPlus6 = new Vector3(vec.x, vec.y + 6, vec.z);
1646                    Vector3 vecPlus7 = new Vector3(vec.x, vec.y + 7, vec.z);
1647                    Vector3 vecPlus8 = new Vector3(vec.x, vec.y + 8, vec.z);
1648                    Vector3 vecPlus9 = new Vector3(vec.x, vec.y + 9, vec.z);
1649                    Vector3 vecPlus10 = new Vector3(vec.x, vec.y + 10, vec.z);
1650
1651                    if (!Physics.CheckBox(vecPlus1, cubeQuart) && !Physics.CheckBox(vecPlus2, cubeQuart) && !Physics.CheckBox(vecPlus3, cubeQuart)
1652                       && !Physics.CheckBox(vecPlus4, cubeQuart) && !Physics.CheckBox(vecPlus5, cubeQuart) && !Physics.CheckBox(vecPlus6, cubeQuart)
1653                        && !Physics.CheckBox(vecPlus7, cubeQuart) && !Physics.CheckBox(vecPlus8, cubeQuart) && !Physics.CheckBox(vecPlus9, cubeQuart)
1654                            && !Physics.CheckBox(vecPlus10, cubeQuart))
1655                    {
1656                        row20Vacancies.Add(vec);
1657                    }
1658                }
1659            foreach (Vector3 vec in row20Vacancies)
1660            {
1661                row20VacancyInts.Add(Mathf.RoundToInt(vec.z));
1662            }
```

```
1663            row20VacancyInts.Sort();
1664            //Checks eacb of the verified empty spaces except the last one on the right
1665            //To see if there's a block to the right of that block.
1666            //If there is, add it to list openSpots.
1667            foreach (int i in row20VacancyInts)
1668            {
1669                if (i <= 4)
1670                {
1671                    if (row20VacancyInts.Contains(i + 1))
1672                    {
1673                        openSpots.Add(i);
1674                    }
1675                }
1676            }
1677            if (openSpots.Count == 0)
1678            {
1679                CheckRows();
1680            }
1681        }
1682        //Chooses the spot to move to.
1683        void ChooseSpot()
1684        {
1685            foreach (int i in openSpots)
1686            {
1687                if (i % 2 == 0)
1688                {
1689                    preferredSpots.Add(i);
1690                }
1691            }
1692            if (preferredSpots.Count >= 1)
1693            {
1694                spot = preferredSpots[Random.Range(0, preferredSpots.Count)];
1695
1696            }
1697            else
1698            {
1699                spot = openSpots[Random.Range(0, openSpots.Count)];
1700            }
1701        }
1702        //Moves the gameobject to the spot chosen by comparing spot and mySpot.
1703        void GoToSpot()
1704        {
1705            if (mySpot > spot)
1706            {
1707                MoveLeft();
1708                mySpot--;
1709                timerTime = Random.Range(0.2f, 0.6f);
```

```
1710            }
1711            if (mySpot < spot)
1712            {
1713                MoveRight();
1714                mySpot++;
1715                timerTime = Random.Range(0.2f, 0.6f);
1716            }
1717            if (mySpot == spot)
1718            {
1719                if (downed == false)
1720                {
1721                    downed = true;
1722                    MoveDown();
1723                    Destroy(this);
1724                }
1725            }
1726        }
1727
1728        void MoveRight()
1729        {
1730            gameObject.GetComponent<Blok_Movement>().MoveRight();
1731        }
1732
1733        void MoveLeft()
1734        {
1735            gameObject.GetComponent<Blok_Movement>().MoveLeft();
1736        }
1737
1738        void MoveDown()
1739        {
1740            gameObject.GetComponent<Blok_Movement>().MoveDown();
1741        }
1742    }
1743
```