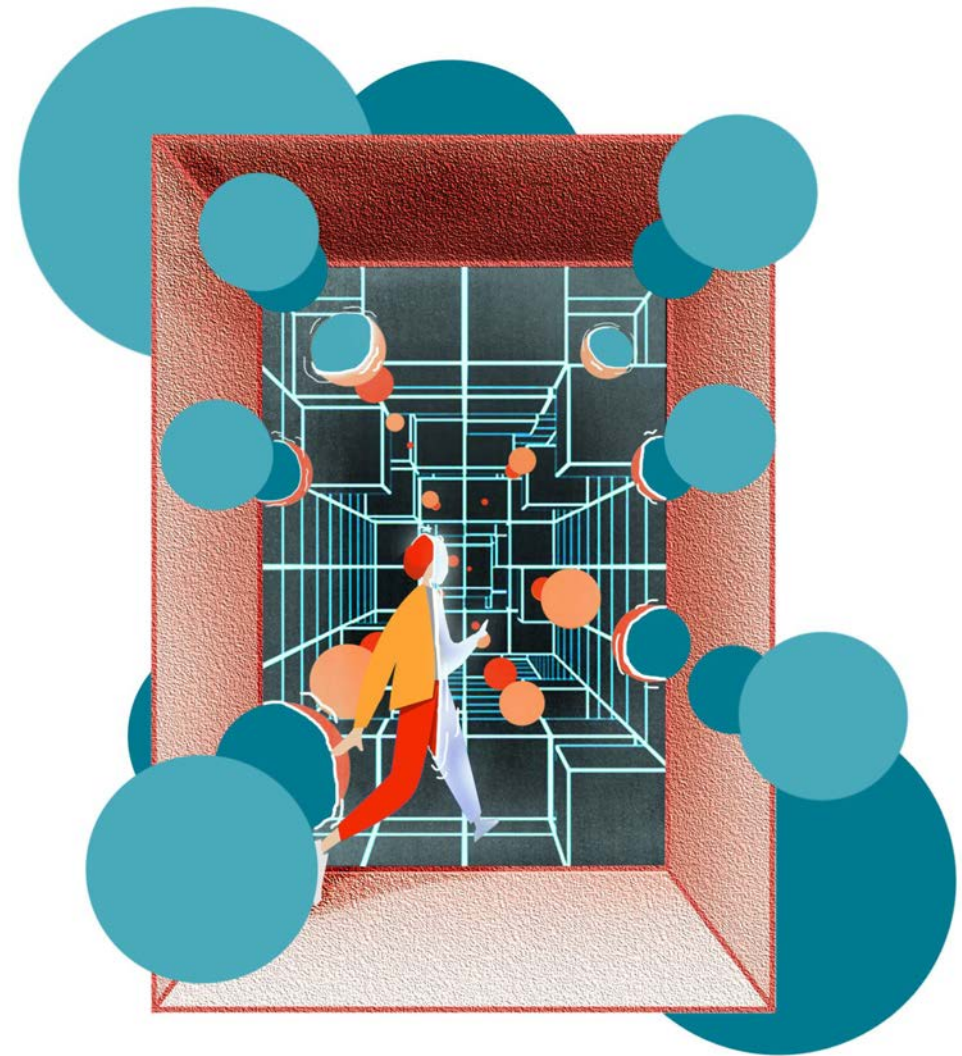


INTERACTIVE 'KEEPLAND'

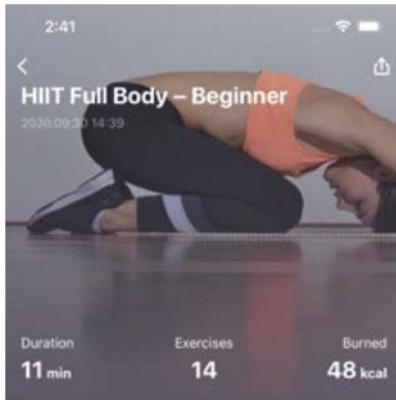
INTERACTIVE KEEPLAND provides a physical training space for the online fitness app named 'Keep'. The design encourages the ebb and flow of boundaries between real and virtual spaces. It challenges how interior spaces can interact with visitors, both digitally and physically.

Based on a sensing system and augmented reality, Interactive Keepland offers a hybrid fitness service that combines online services and physical gym facilities.

THE SITE is a 2-floor building located at Tongjiang road, Nehe, China. It is facing a school and surrounded by residential high rise building.

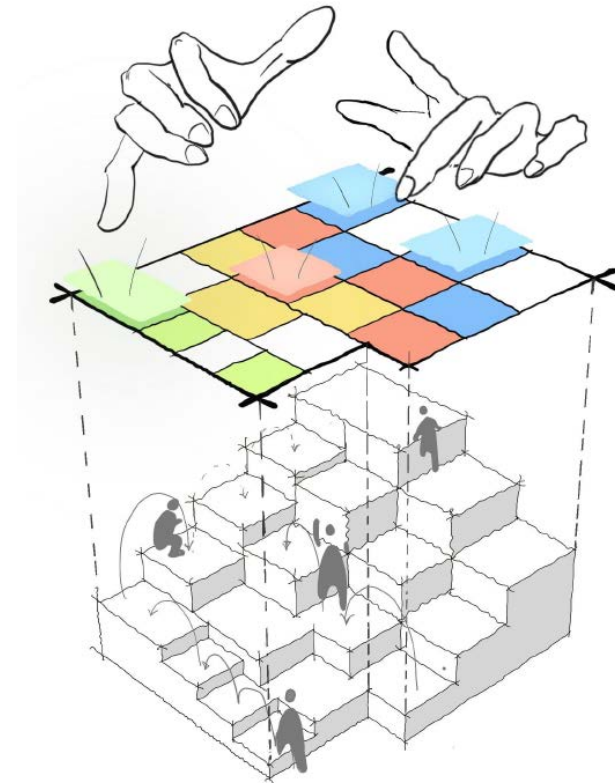


CLIENTS ANALYSIS



Screenshot from Keep

Training Detail	
Arm circles	00:20
Forward elbow circles	00:20
Lateral hops	00:20
Plank	00:30
Good mornings 10x	00:44
Body Saw Plank	00:30
Box squat 12x	00:41
Body Saw Plank	00:30
Jump Squat on Toes 13x	00:23
Hand-Release Knee Push Up 10x	00:32
Forward elbow circles	00:20
Rolling Like a Ball	00:20
Calf Stretch Against Wall (Left)	00:30
Calf Stretch Against Wall (Right)	00:30



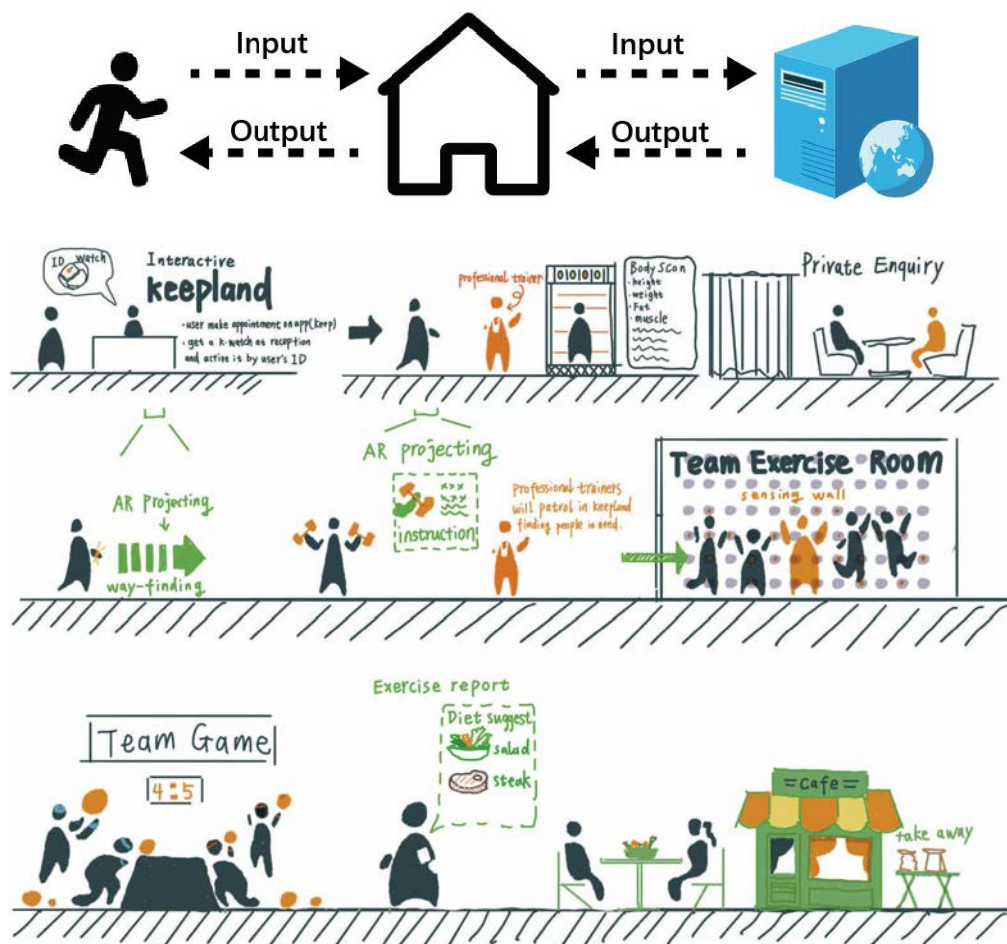
This project is a physical training space for Keep. Keep is an online fitness app. It comprises an e-store for fitness-related products (including exercise equipment) and online fitness courses. Users are provided with training schemes and tutorial videos after filling in an information request form and stating their goals.

According to the survey, only providing online fitness services cannot fully meet the needs of users. At the same time, Keep has also begun to develop physical fitness space in recent years. Further research please see my dissertation 'The Relationship Between Physical space and Online Services in the Field of Fitness'.

DJ PAD* JUMPING BOX is an interactive fitness installation. The user can follow the instructions of the projection on it to jump on the jumping box, and at the same time, when stepping on the jumping box, the sensor will be triggered to play musical instrument sounds, such as drums and electric guitar sounds.

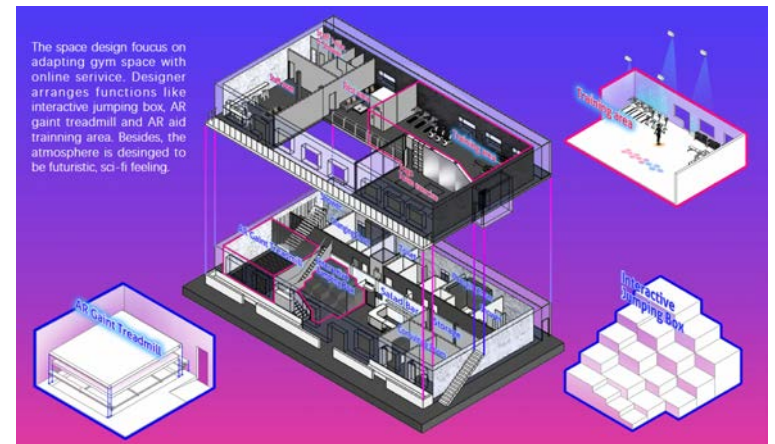
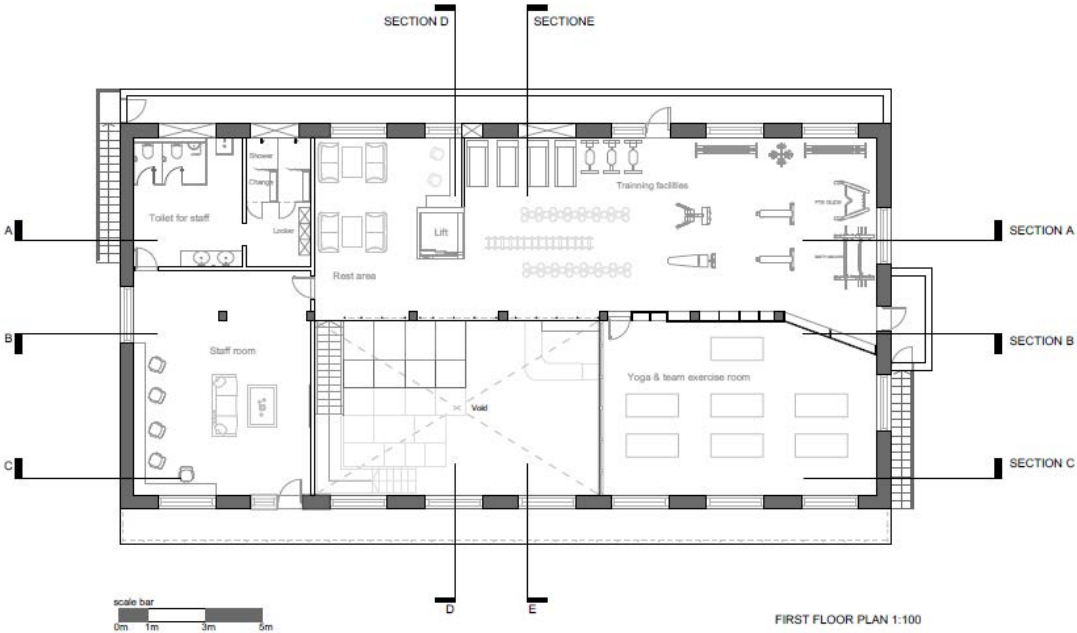
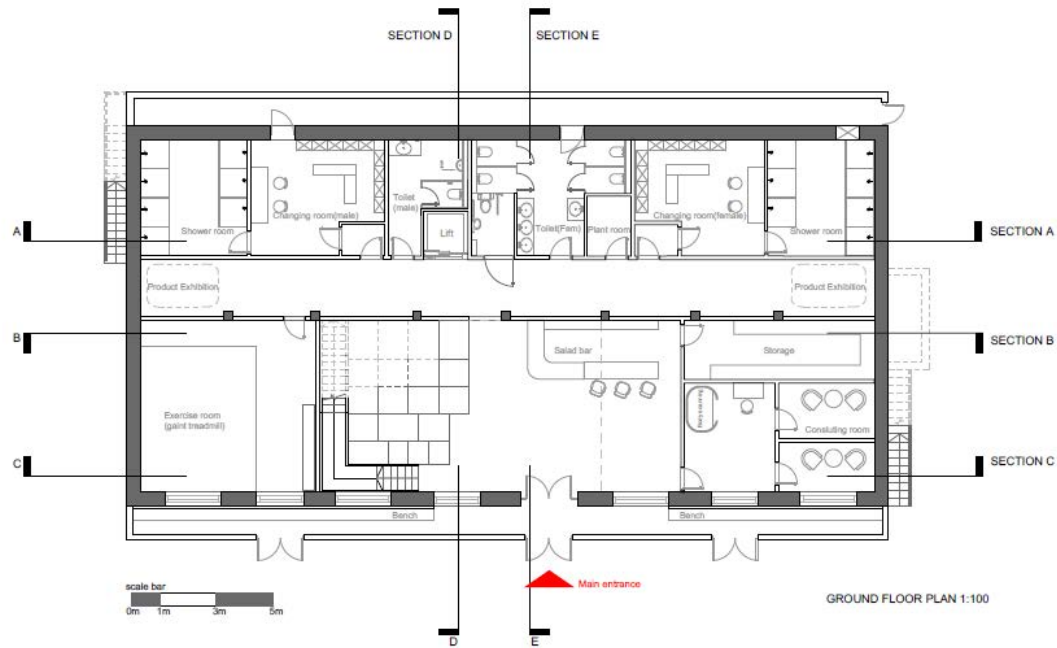
Users can also select different programs or tracks through the Keep app, and managers can design different programs and activities to link with the installation.

CONCEPT



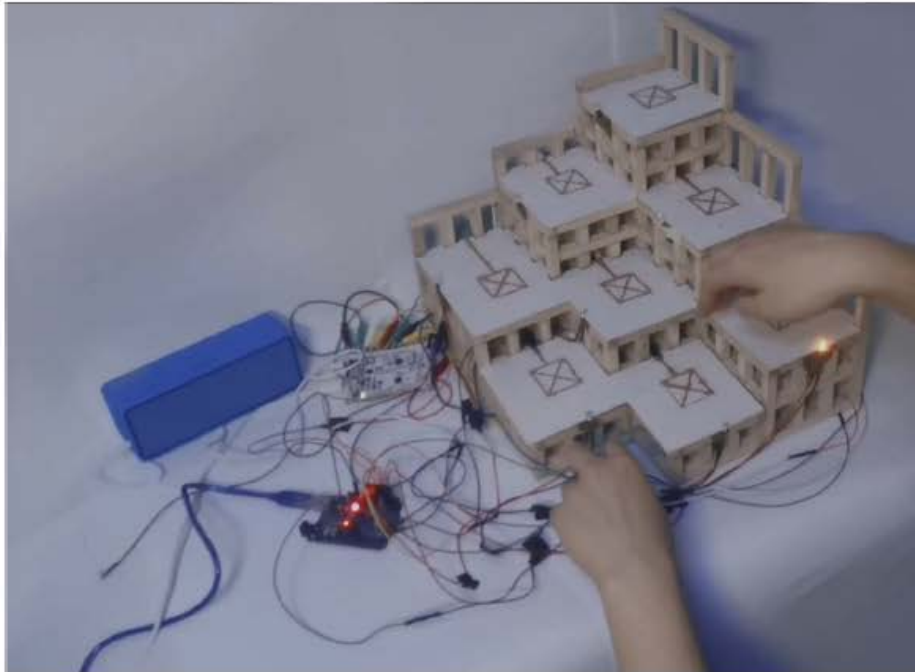
Based on the sensing system and AR, the Interactive Keepland offers a hybrid fitness service that combines online services and gym services.

When users get into the interactive keepland, they receive a smartwatch and a computer to recognize their ID and monitor their moves and heart rates. Facilities such as body scanning machines help to build a data model of users and update it. Project mapping can do things like way-finding and interactive fitness games. The salad bar could suggest food to users by analyzing how many calories they burn.



INTERACTIVE JUMPING BOX PROTOTYPING

Click the video link to see how it works: <https://vimeo.com/538778001>



Material: Arduino UNO board; Duno Peak Touchtouch board; Speaker; LED with 220 ohm resistor*8; Conductive paint;

I made a simple version of the sensing and interactive system through learning programming and circuit design. In this prototype, I transform music rhythm information to signal of flashing of LED lights, and then the participants jumped to the corresponding jumping box according to light signal. The sensor circuit on the surface of the jumping box sends a signal to the data board after sensing the human contact, and the data board sends a command to the speaker to play the specified sound after receiving the signal.

The purpose of the following Arduino program is to make the lights flash according to the rhythm of the music score so as to guide the participants to follow the instructions of the lights to step on the corresponding box.

```
2 void setup() {
3   pinMode(2, OUTPUT); //LED Light on Drum Kit No.1
4   pinMode(3, OUTPUT); //LED Light on Drum Kit No.2
5   pinMode(4, OUTPUT); //LED Light on Synthesizer sound C
6   pinMode(5, OUTPUT); //LED Light on Synthesizer sound E
7   pinMode(6, OUTPUT); //LED Light on Synthesizer sound A
8   pinMode(7, OUTPUT); //LED Light on Synthesizer sound G
9   pinMode(8, OUTPUT); //LED Light on Synthesizer sound B
10  pinMode(9, OUTPUT); //LED Light on Synthesizer sound R-
11  digitalWrite(2, HIGH); //LED Light on Drum Kit No.1
12  delay(1000);
13  digitalWrite(2, LOW);
14  delay(1000);
15  digitalWrite(3, HIGH); //LED Light on Drum Kit No.2
16  delay(1000);
17  digitalWrite(3, LOW);
18  delay(1000);
19  digitalWrite(2, HIGH); //LED Light on Drum Kit No.1
20  delay(1000);
21  digitalWrite(2, LOW);
22  delay(1000);
23  digitalWrite(3, HIGH); //LED Light on Drum Kit No.2
24  delay(1000);
25  digitalWrite(3, LOW);
26  delay(1000);
27  digitalWrite(2, HIGH); //LED Light on Drum Kit No.1
28  delay(1000);
29  digitalWrite(2, LOW);
30  delay(1000);
31  digitalWrite(3, HIGH); //LED Light on Drum Kit No.2
32  delay(1000);
33  digitalWrite(3, LOW);
34  delay(1000);
35  digitalWrite(2, HIGH); //LED Light on Drum Kit No.1
36  delay(1000);
37  digitalWrite(2, LOW);
38  delay(1000);
39  digitalWrite(3, HIGH); //LED Light on Drum Kit No.2
40  delay(1000);
41  digitalWrite(3, LOW);
42  delay(1000);
43  }
44
45 void loop() {
46  digitalWrite(2, HIGH); //TURN THE LED Light ON FOR THE Drum Kit No.1
47  delay(1000);
48  digitalWrite(4, HIGH); //TURN THE LED Light on FOR THE Synthesizer sound C
49  digitalWrite(2, LOW); //TURN THE LED OFF ON THE DRUM KIT NO.1
50  digitalWrite(4, LOW); //TURN THE LED OFF ON THE Synthesizer sound C
51  delay(1000);
52  digitalWrite(3, HIGH); //LED Light on Drum Kit No.2
53  delay(1000);
54  digitalWrite(3, LOW); //TURN THE LED FOR DRUM KIT NO.2 OFF
55  delay(1000);
56  digitalWrite(2, HIGH); //TURN THE LED Light ON FOR THE Drum Kit No.1
57  digitalWrite(4, HIGH); //TURN THE LED Light on FOR THE Synthesizer sound C
58  delay(1000);
59  digitalWrite(2, LOW); //TURN THE LED OFF ON THE DRUM KIT NO.1
60  digitalWrite(4, LOW); //TURN THE LED OFF ON THE Synthesizer sound C
61  delay(1000);
62  digitalWrite(3, HIGH); //LED Light on Drum Kit No.2
63  digitalWrite(5, HIGH); //TURN THE LED Light on FOR THE Synthesizer sound E
64  delay(1000);
65  digitalWrite(5, LOW); //TURN THE LED Light on FOR THE Synthesizer sound E
66  delay(1000);
67  digitalWrite(2, HIGH); //TURN THE LED Light ON FOR THE Drum Kit No.1
68  digitalWrite(6, HIGH); //TURN THE LED Light on FOR THE Synthesizer sound A
69  delay(1000);
70  digitalWrite(2, LOW); //TURN THE LED OFF ON THE DRUM KIT NO.1
71  digitalWrite(6, LOW); //TURN THE LED OFF ON THE Synthesizer sound C
72  delay(1000);
73  digitalWrite(3, HIGH); //LED Light on Drum Kit No.2
74  delay(1000);
75  digitalWrite(3, LOW);
76  delay(1000);
77  digitalWrite(2, HIGH); //TURN THE LED Light ON FOR THE Drum Kit No.1
78  digitalWrite(6, HIGH); //TURN THE LED Light on FOR THE Synthesizer sound A
79  delay(1000);
80  digitalWrite(2, LOW); //TURN THE LED OFF ON THE DRUM KIT NO.1
81  digitalWrite(6, LOW); //TURN THE LED OFF ON THE Synthesizer sound C
82  delay(1000);
83  digitalWrite(3, HIGH); //LED Light on Drum Kit No.2
84  digitalWrite(7, HIGH); //TURN THE LED Light on FOR THE Synthesizer sound G
85  delay(1000);
86  delay(1000);
87  digitalWrite(3, LOW);
88  digitalWrite(7, LOW);
89  delay(1000);
90  digitalWrite(2, HIGH);
91  digitalWrite(5, HIGH);
92  delay(1000);
93  digitalWrite(2, LOW);
94  digitalWrite(5, LOW);
95  delay(1000);
96  digitalWrite(3, HIGH);
97  delay(1000);
98  digitalWrite(3, LOW);
99  delay(1000);
100 digitalWrite(2, HIGH);
101 digitalWrite(5, HIGH);
102 delay(1000);
103 digitalWrite(2, LOW);
104 digitalWrite(5, LOW);
105 delay(1000);
106 digitalWrite(3, HIGH);
107 digitalWrite(7, HIGH);
108 delay(1000);
109 digitalWrite(3, LOW);
110 digitalWrite(7, LOW);
111 delay(1000);
112 digitalWrite(2, HIGH);
113 digitalWrite(8, HIGH);
114 delay(1000);
115 digitalWrite(2, LOW);
116 digitalWrite(8, LOW);
117 delay(1000);
118 digitalWrite(3, HIGH);
119 delay(1000);
120 digitalWrite(3, LOW);
121 delay(1000);
122 digitalWrite(2, HIGH);
123 digitalWrite(8, HIGH);
124 delay(1000);
125 digitalWrite(2, LOW);
126 digitalWrite(8, LOW);
127 delay(1000);
128 digitalWrite(3, HIGH);
129 digitalWrite(9, HIGH);
130 delay(1000);
131 digitalWrite(3, LOW);
132 digitalWrite(9, LOW);
133 delay(1000);
134 digitalWrite(2, HIGH);
135 digitalWrite(8, HIGH);
136 delay(1000);
137 digitalWrite(2, LOW);
138 delay(1000);
139 digitalWrite(3, HIGH);
140 digitalWrite(9, LOW);
141 delay(1000);
142 }
```