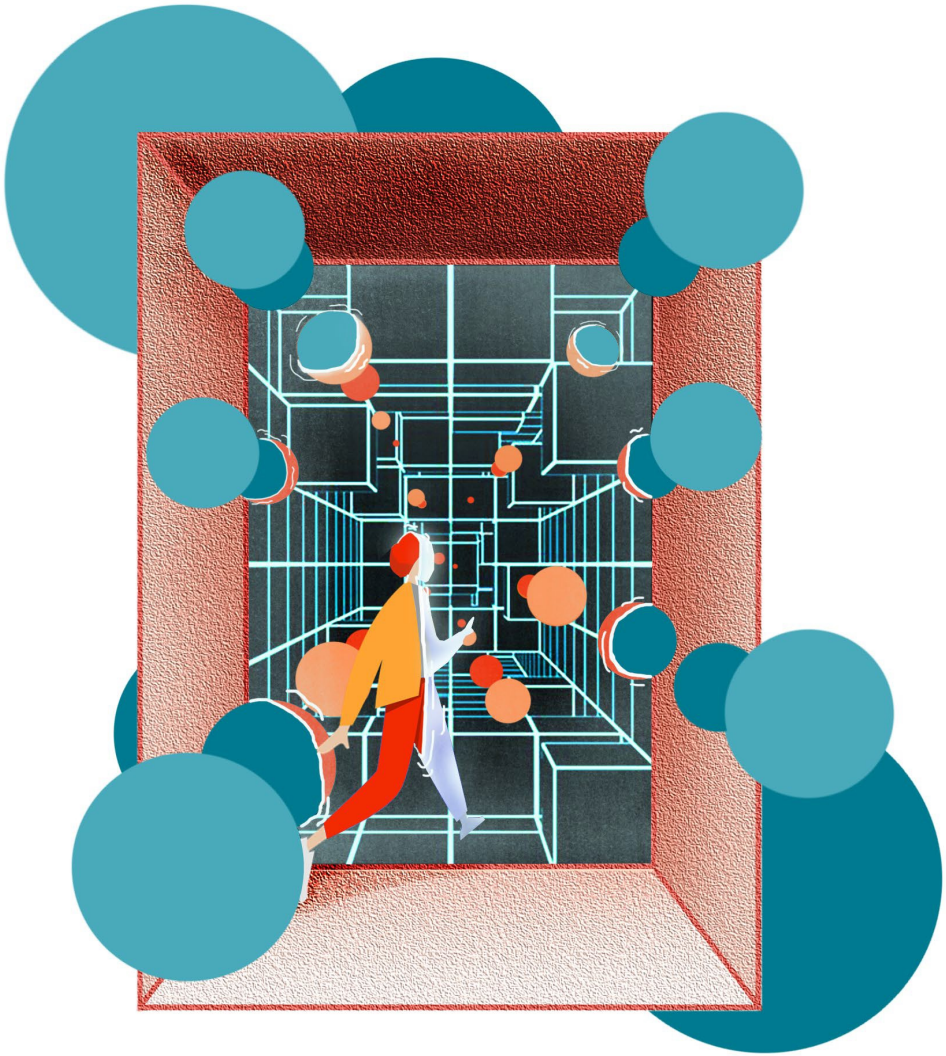
INTERACTIVE ‘KEEPLAND’

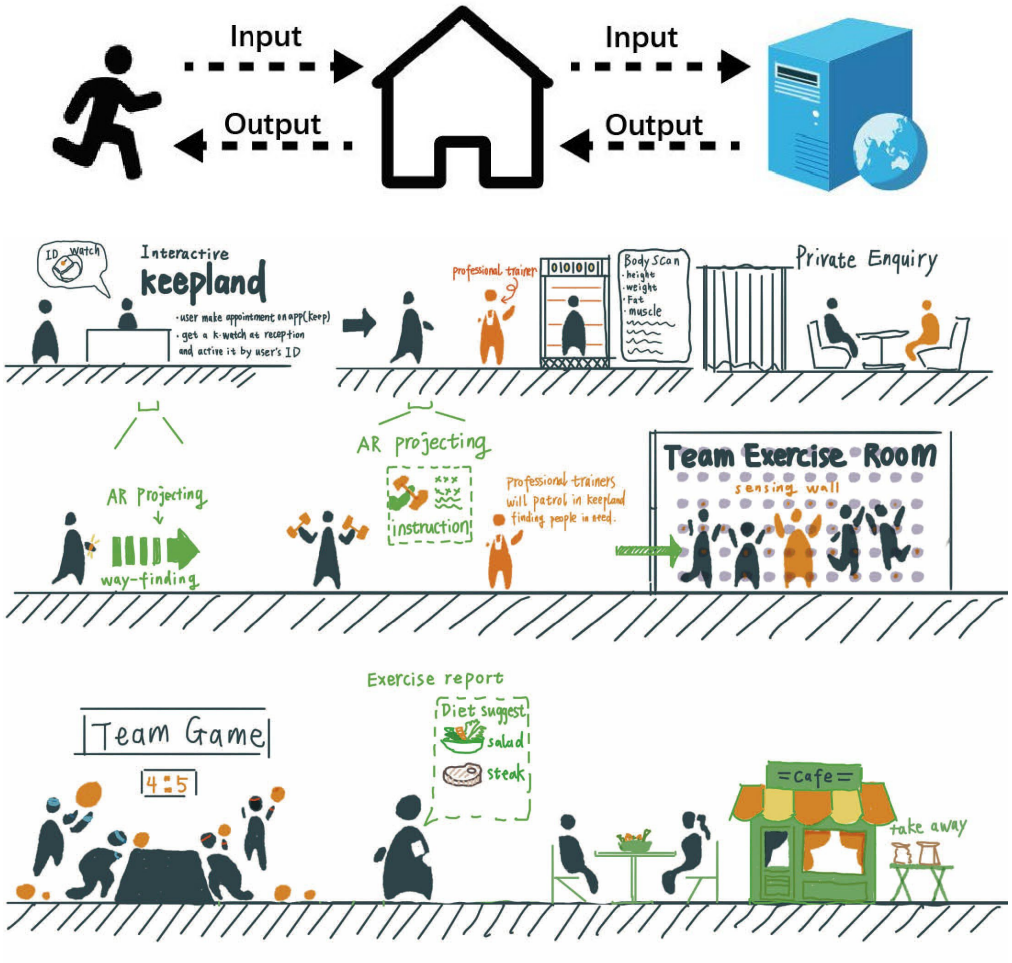
INTERACTIVE KEEPLAND provides a physical training space for the online fitness app named ‘Keep’. The design encourages the ebb and flow of boundaries between real and virtual spaces. It challenges how interior spaces can interact with visitors, both digitally and physically.

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 surounded by residential high rise building.



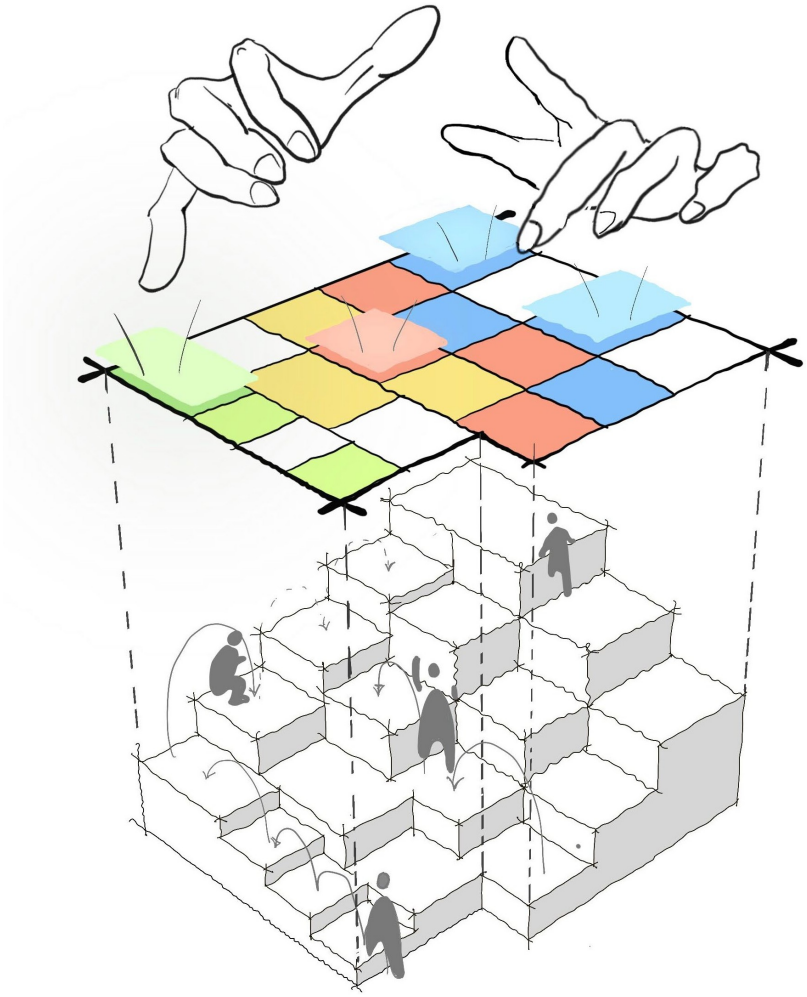
CONCEPT

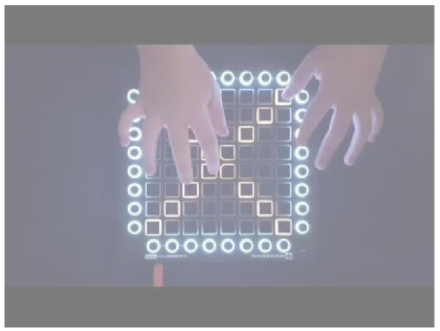


Based on the sensing system and AR , the Interactive Keepland offer a hybrid fitness service that combines online services and gym services.

When users get into the interactive keepland they get a smart watch for computer to recognize their ID and monitoring their moves and heart rates. Then facilities such as body scanning machine help to build a data model of users and updates it. Project mapping can do things like way-finding and intercative fitness games. The salad bar could suggest food to users by analysis how much calories they burn.

DJ PAD JUMPING BOX

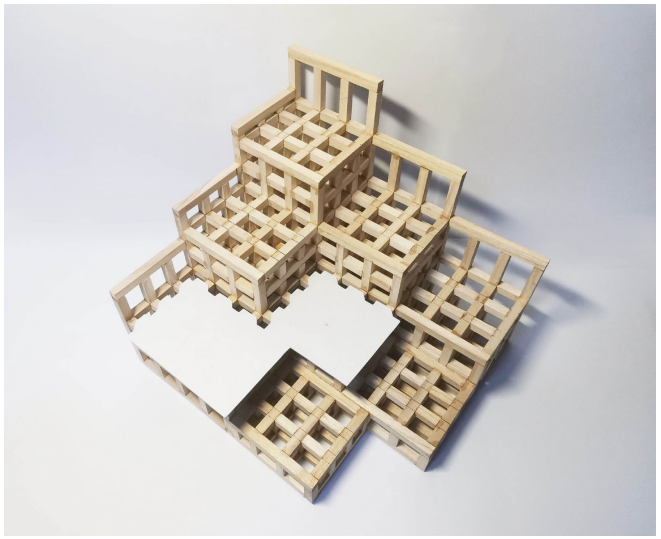


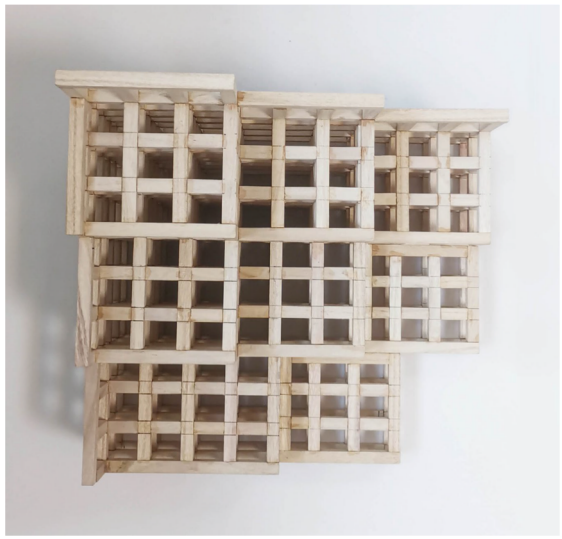


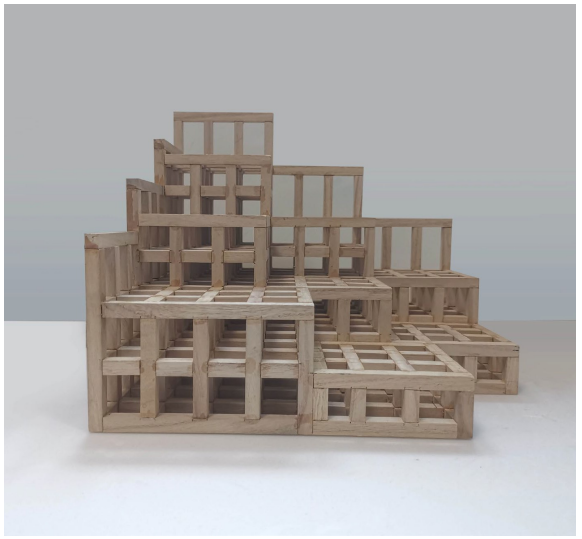
DJ PAD*JUMPING BOX is an interactive fitness installtion. The user can follow the instructions of the projection on it to jump on the jumping box, and at the same time, when stepping on the jumping box, the sensor will be triggered to play musical instrument sounds, such as drums and electric guitar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

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

PHYSICAL MODEL

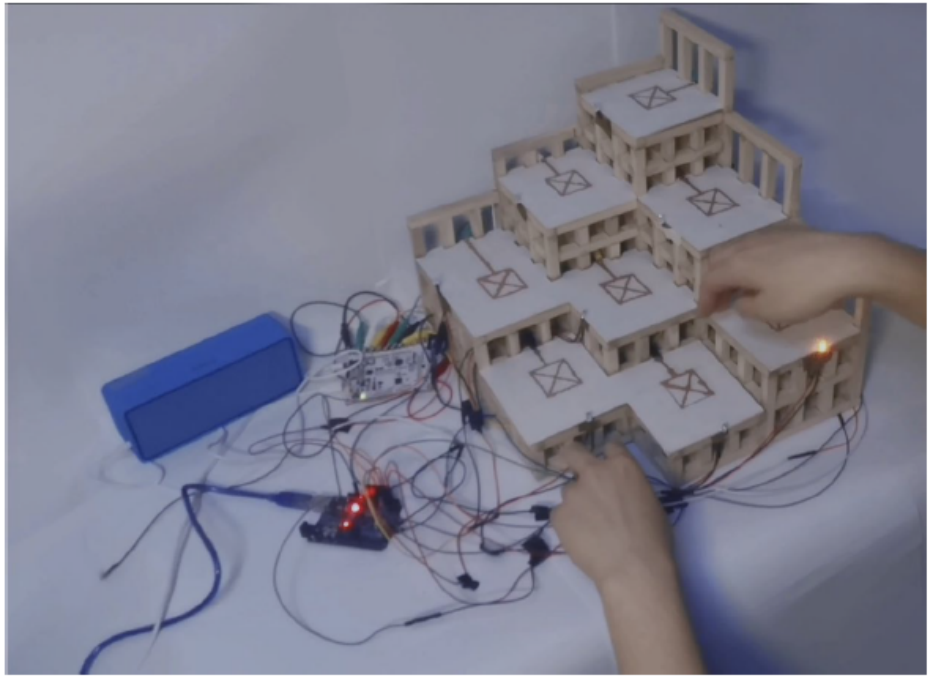






INTERACTIVE JUMPING BOX PROTOTYPING

Click the vedio link to see how it works: https://vimeo.com/538778001



Material: Ardunio UNO board; Duinopeak 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 Ardunio 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.

```
void setup() {
pinMode(2,OUTPUT);//LED Light on Drum kit No.1
pinMode(3,OUTPUT);//LED Light on Drum kit No.2
pinMode(4,OUTPUT);//LED Light on Synthesizer sound C
pinMode(5,OUTPUT);//LED Light on Synthesizer sound E
pinMode(6,OUTPUT);//LED Light on Synthesizer sound A
pinMode(7,OUTPUT);//LED Light on Synthesizer sound G
pinMode(8,OUTPUT);//LED Light on Synthesizer sound B
pinMode(9,OUTPUT);//LED Light on Synthesizer sound A-
digitalWrite(2,HIGH);//LED Light on Drum kit No.1
delay(1000);
digitalWrite(2,LOW);
delay(1000);
digitalWrite(3,HIGH);//LED Light on Drum kit No.2
delay(1000);
digitalWrite(3,LOW);
delay(1000);
digitalWrite(2,HIGH);//LED Light on Drum kit No.1
delay(1000);
digitalWrite(2,LOW);
delay(1000);
digitalWrite(3,HIGH);//LED Light on Drum kit No.2
delay(1000);
digitalWrite(3,LOW);
delay(1000);
digitalWrite(2,HIGH);//LED Light on Drum kit No.1
delay(1000);
digitalWrite(2,LOW);
delay(1000);
digitalWrite(3,HIGH);//LED Light on Drum kit No.2
delay(1000);
digitalWrite(3,LOW);
delay(1000);
digitalWrite(2,HIGH);//LED Light on Drum kit No.1
delay(1000);
digitalWrite(2,LOW);
delay(1000);
digitalWrite(3,HIGH);//LED Light on Drum kit No.2
delay(1000);
digitalWrite(3,LOW);
delay(1000);
}

void loop() {
digitalWrite(2,HIGH);//TURN THE LED Light ON FOR THE Drum kit No.1
digitalWrite(4,HIGH);//TURN THE LED Light on FOR THE Synthesizer sound C
delay(1000);
digitalWrite(2,LOW);//TURN THE LED OFF ON THE DRUM KIT NO.1
digitalWrite(4,LOW);//TURN THE LED OFF on THE Synthesizer sound C
delay(1000);
digitalWrite(3,HIGH);//LED Light on Drum kit No.2
delay(1000);
digitalWrite(3,LOW);//TURN THE LED FOR DRUM KIT NO.2 OFF
delay(1000);
digitalWrite(2,HIGH);//TURN THE LED Light ON FOR THE Drum kit No.1
digitalWrite(4,HIGH);//TURN THE LED Light on FOR THE Synthesizer sound C
delay(1000);
digitalWrite(2,LOW);//TURN THE LED OFF ON THE DRUM KIT NO.1
digitalWrite(4,LOW);//TURN THE LED OFF on THE Synthesizer sound C
delay(1000);
digitalWrite(3,HIGH);//LED Light on Drum kit No.2
digitalWrite(5,HIGH);//TURN THE LED Light on FOR THE Synthesizer sound E
delay(1000);
digitalWrite(3,LOW);
digitalWrite(5,LOW);//TURN THE LED Light on FOR THE Synthesizer sound E
delay(1000);
digitalWrite(2,HIGH);//TURN THE LED Light ON FOR THE Drum kit No.1
digitalWrite(6,HIGH);//TURN THE LED Light on FOR THE Synthesizer sound A
delay(1000);
digitalWrite(2,LOW);//TURN THE LED OFF ON THE DRUM KIT NO.1
digitalWrite(6,LOW);//TURN THE LED OFF on THE Synthesizer sound C
delay(1000);
digitalWrite(3,HIGH);//LED Light on Drum kit No.2
delay(1000);
digitalWrite(3,LOW);
delay(1000);
digitalWrite(2,HIGH);//TURN THE LED Light ON FOR THE Drum kit No.1
digitalWrite(6,HIGH);//TURN THE LED Light on FOR THE Synthesizer sound A
delay(1000);
digitalWrite(2,LOW);//TURN THE LED OFF ON THE DRUM KIT NO.1
digitalWrite(6,LOW);//TURN THE LED OFF on THE Synthesizer sound C
delay(1000);
digitalWrite(3,HIGH);//LED Light on Drum kit No.2
digitalWrite(7,HIGH);//TURN THE LED Light on FOR THE Synthesizer sound G
delay(1000);
delay(1000);
digitalWrite(3,LOW);
digitalWrite(7,LOW);
delay(1000);
digitalWrite(2,HIGH);
digitalWrite(5,HIGH);
delay(1000);
digitalWrite(2,LOW);
digitalWrite(5,LOW);
delay(1000);
digitalWrite(3,HIGH);
delay(1000);
digitalWrite(3,LOW);
delay(1000);
digitalWrite(2,HIGH);
digitalWrite(5,HIGH);
delay(1000);
digitalWrite(2,LOW);
digitalWrite(5,LOW);
delay(1000);
digitalWrite(3,HIGH);
digitalWrite(7,HIGH);
delay(1000);
digitalWrite(3,LOW);
digitalWrite(7,LOW);
delay(1000);
digitalWrite(2,HIGH);
digitalWrite(8,HIGH);
delay(1000);
digitalWrite(2,LOW);
digitalWrite(8,LOW);
delay(1000);
digitalWrite(3,HIGH);
delay(1000);
digitalWrite(3,LOW);
delay(1000);
digitalWrite(2,HIGH);
digitalWrite(8,HIGH);
delay(1000);
digitalWrite(2,LOW);
digitalWrite(8,LOW);
delay(1000);
digitalWrite(3,HIGH);
digitalWrite(9,HIGH);
delay(1000);
digitalWrite(3,LOW);
digitalWrite(9,LOW);
delay(1000);
digitalWrite(2,HIGH);
delay(1000);
digitalWrite(2,LOW);
delay(1000);
digitalWrite(3,HIGH);
delay(1000);
digitalWrite(3,LOW);
delay(1000);
}
```