

```
// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN

CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE
  uploading and positioning TOOLS; // mapping Teufelsberg
  // Steel Ring, ALDO VAN EYCK
  // Playtime, JAQUES TATI
}

(FIELD) STATION < problems - solution; // Protei Sailboat
  investigation - knowledge; // Drone Survival Guide
  // necessity of context

void setup() {
0 CONTEXT

CODE: start (one time only) FIELD STATION 0 as open hardware; // Open Source Ecology, MARICIN JAKUBOWSKI

INPUT = FIELD STATION 0 = COMPONENTS; // WikiHouse, ALISTAIR PARVIN
  // it's easier to ship recipes than cakes and biscuits

COMPONENTS: first suggestion of TOOLS; // Apollo Lunar Module
}

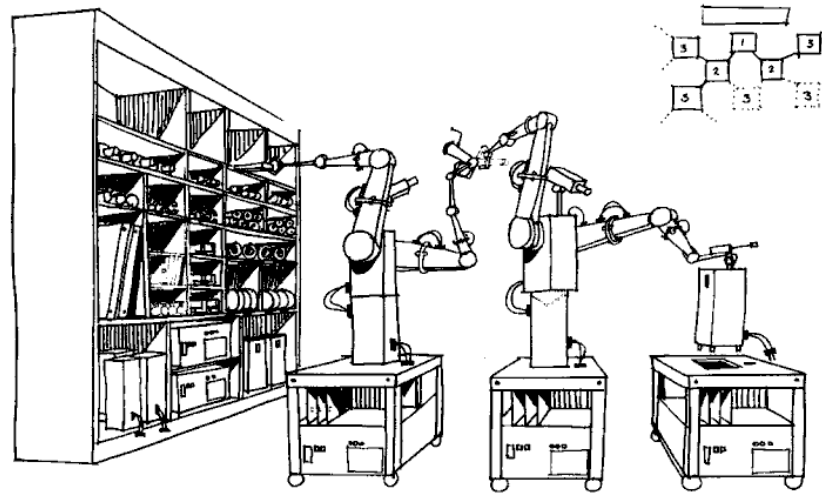
void loop() {
+ CONTEXT: FIELD STATION (n);

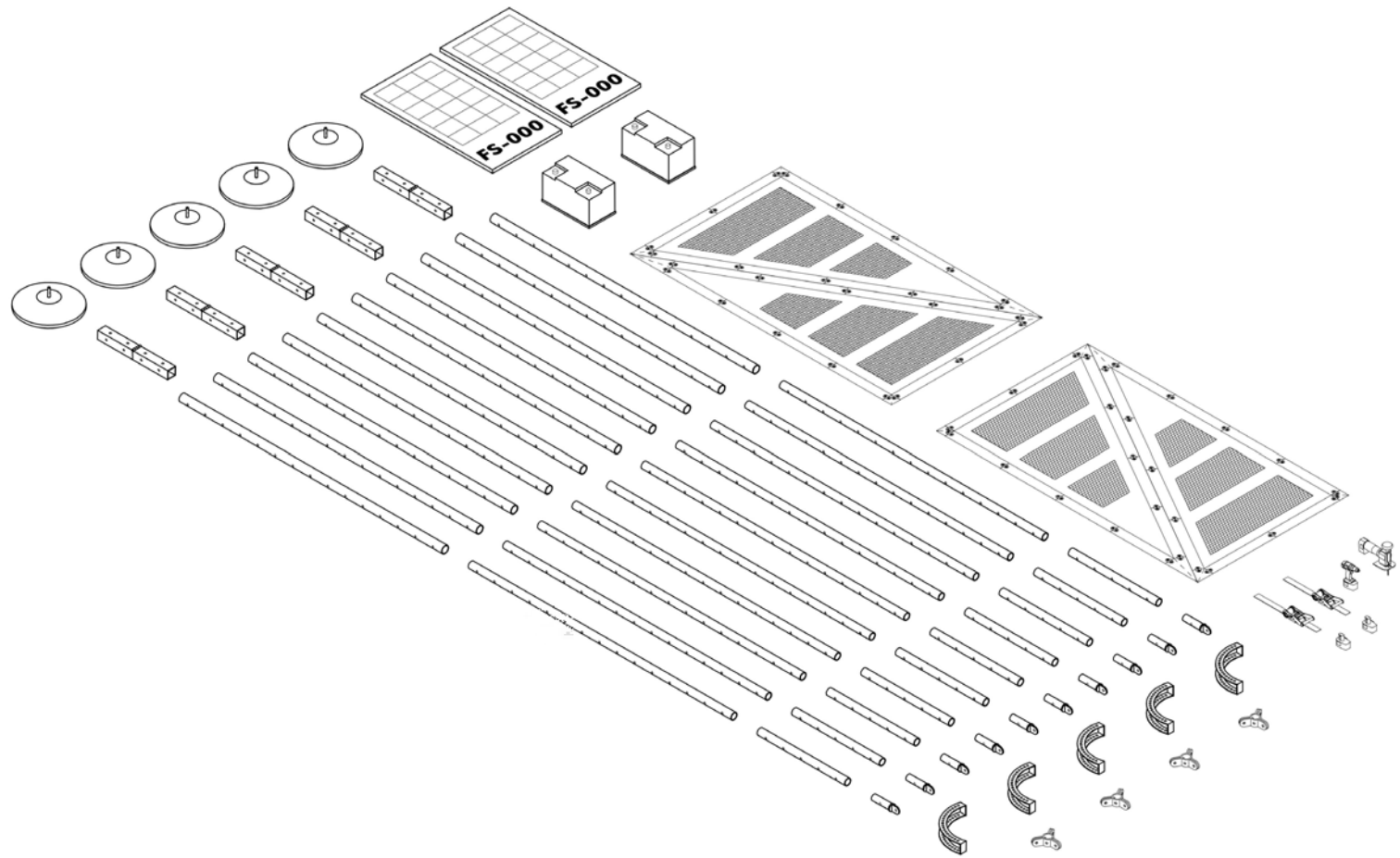
FIELD STATION (n+1): addition, mutation, implementation of FIELD STATION (n) // Open Structures

If END {
  activation of space?;
}

If notEND {
  FIELD STATION (n+1);
}
}
}
```

# CODE - (SERIES OF) FIELD STATION(S) (+)





// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN

CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

The spaces and times are pulled out from between people.

Authorship becomes universal.

Information is shared, rather than matter, and such it can be easily distributed and modified, taking on a history of its own.

To generate not one idea but the hundreds of thousands of design ideas.

? Role of the architect

Architects as programmer:  
the activation of space > creation of space  
event > form

? (COLLECTIVE) PHYSICAL SPACE (as counterpart of digital existence).

The architect has an opportunity to participate in the evolution of the autonomous built environment through creating frameworks within which users design.

Fab labs allowed people to modify or "hack" the world around them.  
To produce solutions to local problems.

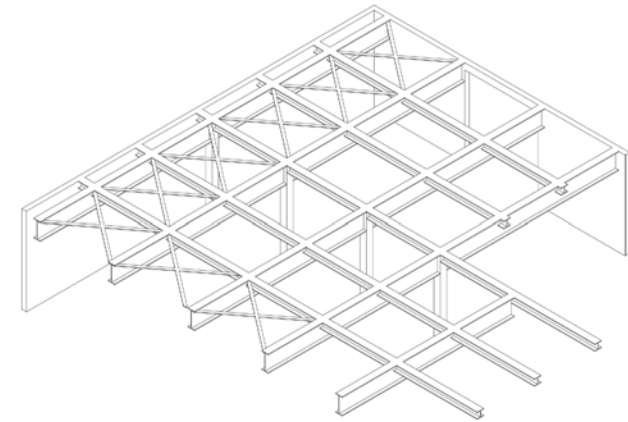
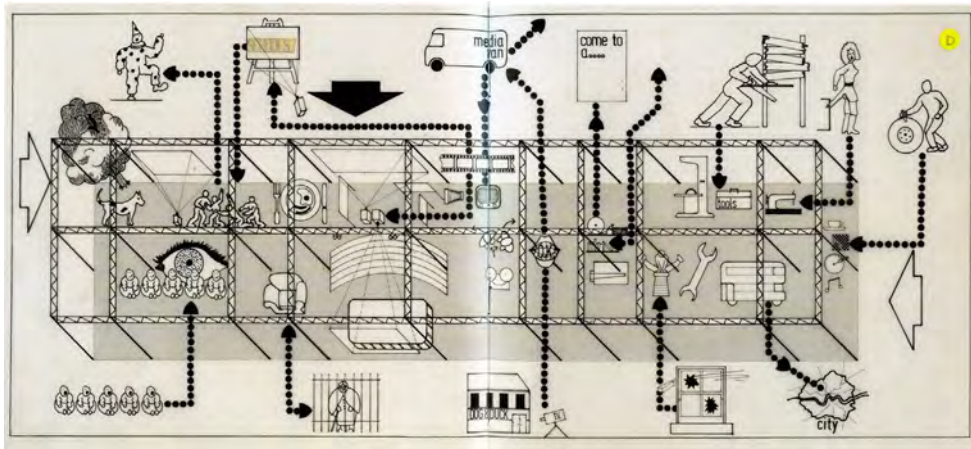
? = STATION

// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN

CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE

site Teufelsberg

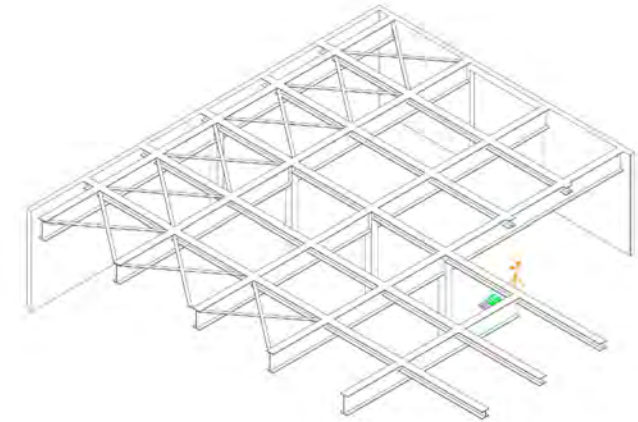
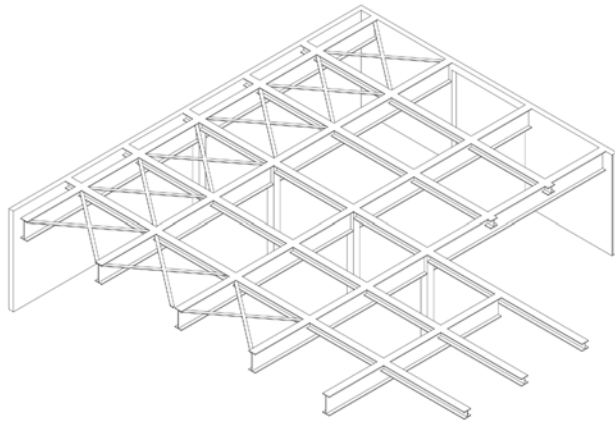


// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN

CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE  
uploading and positioning TOOLS; // mapping Teufelsberg

? start  
? energy + light



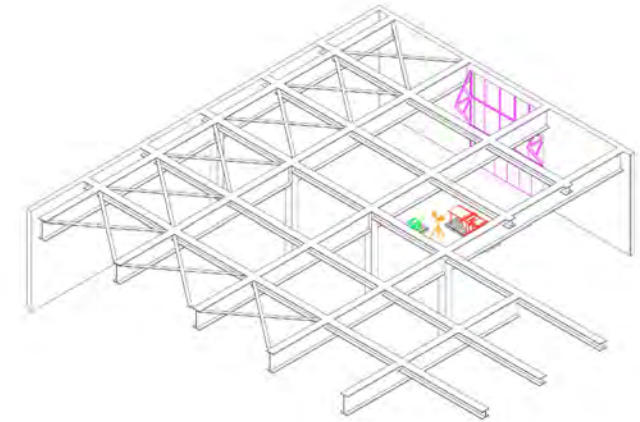
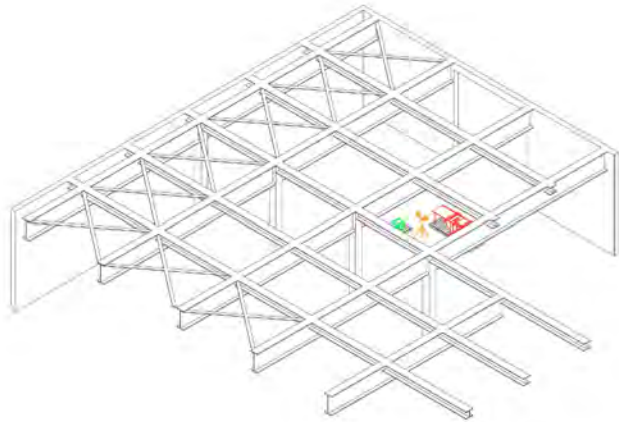


// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN

CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE  
uploading and positioning TOOLS; // mapping Teufelsberg

? skin

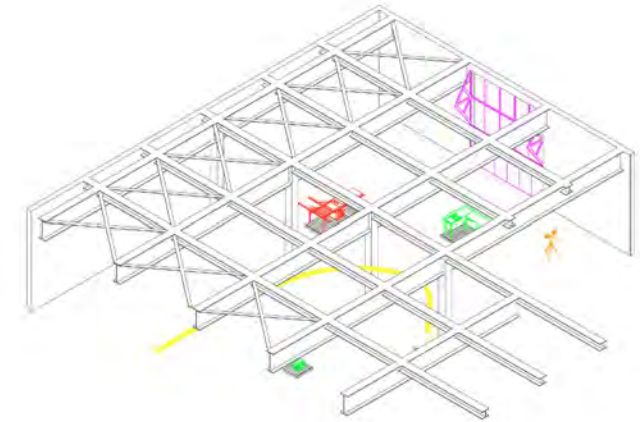
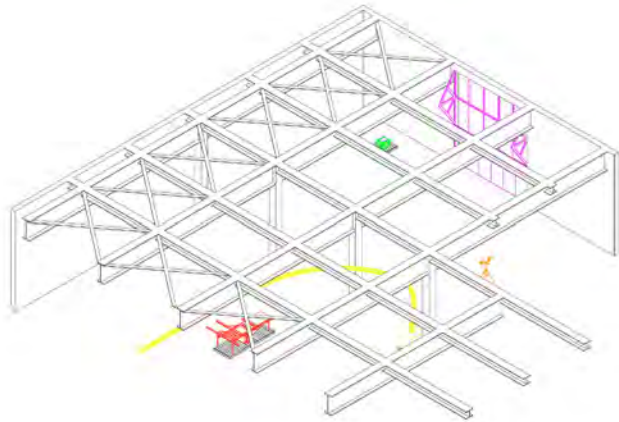


// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN

CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE  
uploading and positioning TOOLS; // mapping Teufelsberg

? heating



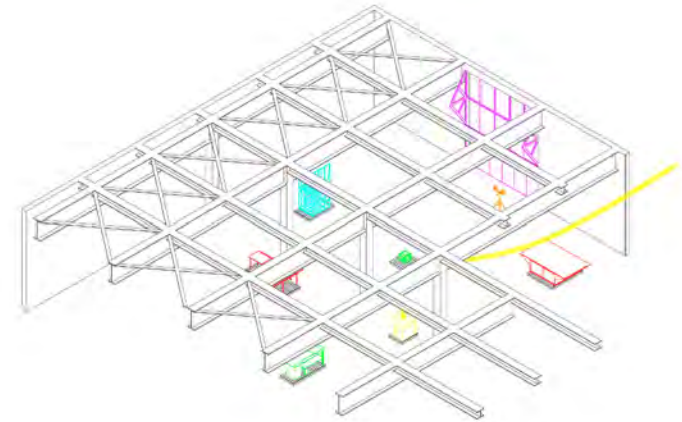
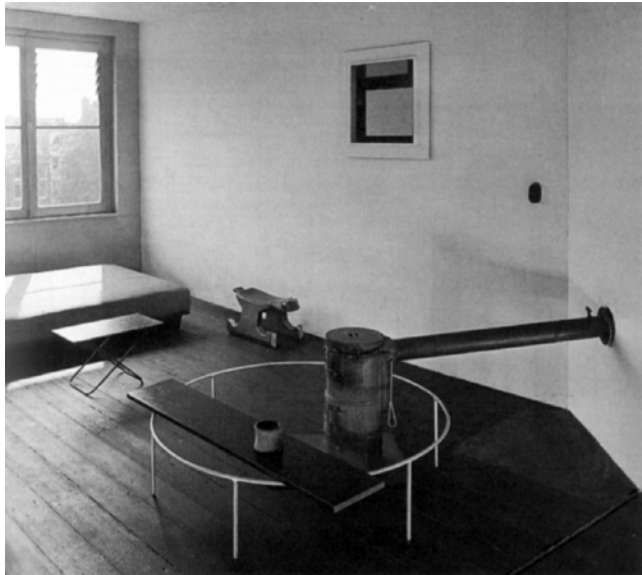


// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN

CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE  
uploading and positioning TOOLS; // mapping Teufelsberg  
// Steel Ring, ALDO VAN EYCK

? positioning

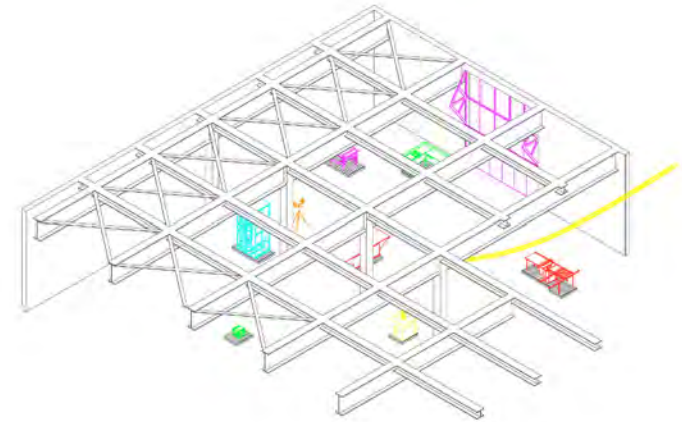
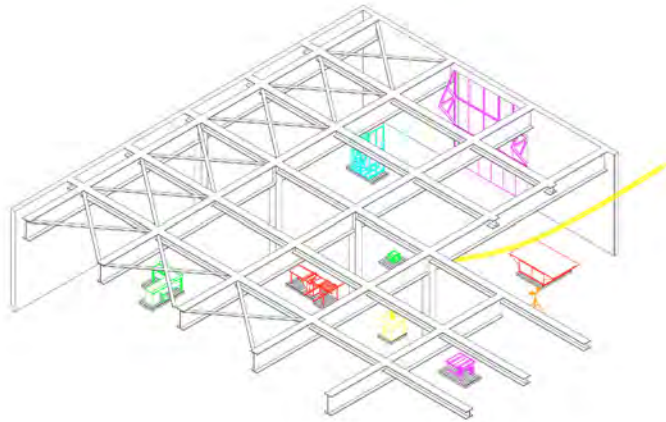


// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN

CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE  
uploading and positioning TOOLS; // mapping Teufelsberg  
// Steel Ring, ALDO VAN EYCK

? positioning

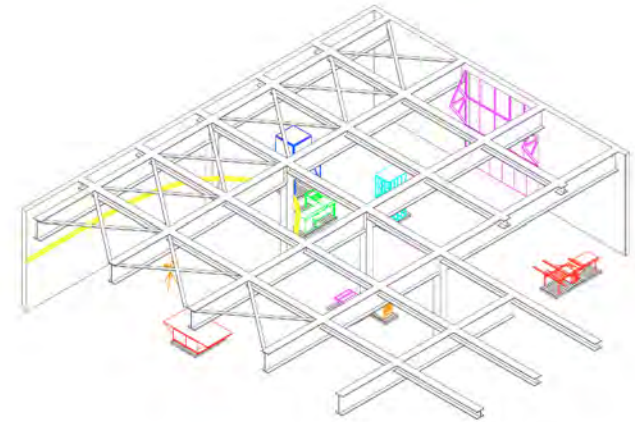
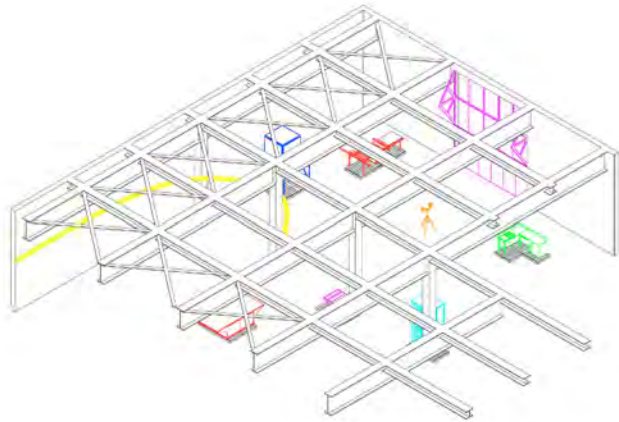


```
// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN
```

```
CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity
```

```
OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE  
  uploading and positioning TOOLS; // mapping Teufelsberg  
  // Steel Ring, ALDO VAN EYCK
```

realtime

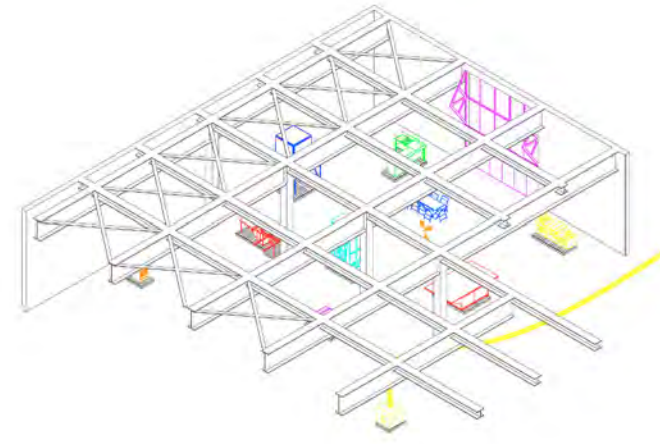
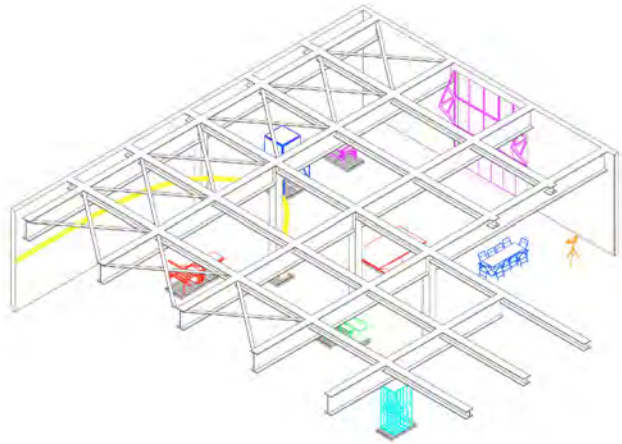


// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN

CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE  
uploading and positioning TOOLS; // mapping Teufelsberg  
// Steel Ring, ALDO VAN EYCK

? future

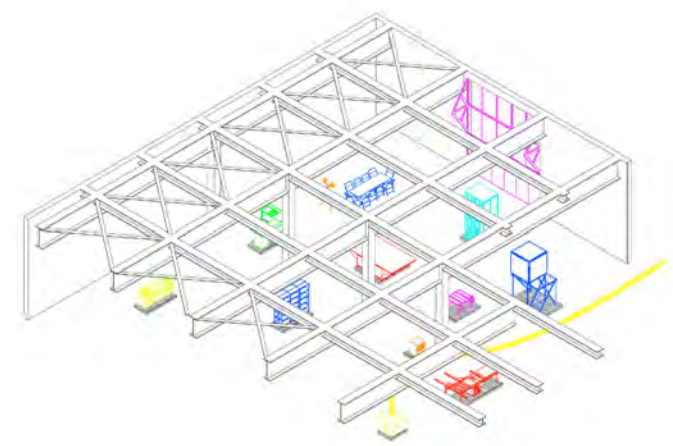
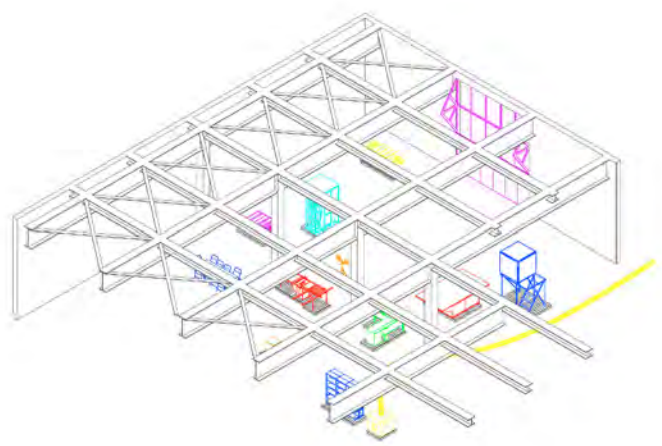


// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN

CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE  
uploading and positioning TOOLS; // mapping Teufelsberg  
// Steel Ring, ALDO VAN EYCK

? future



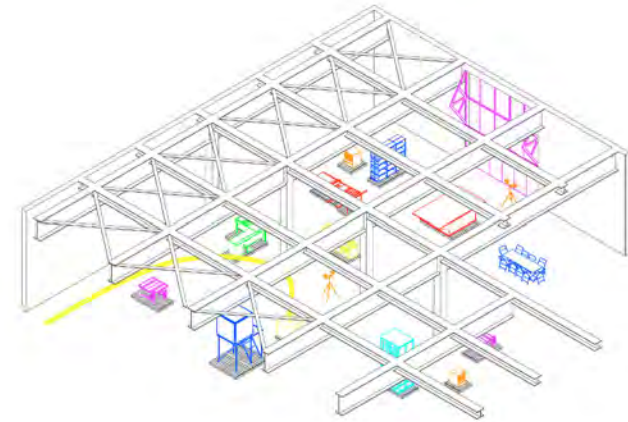
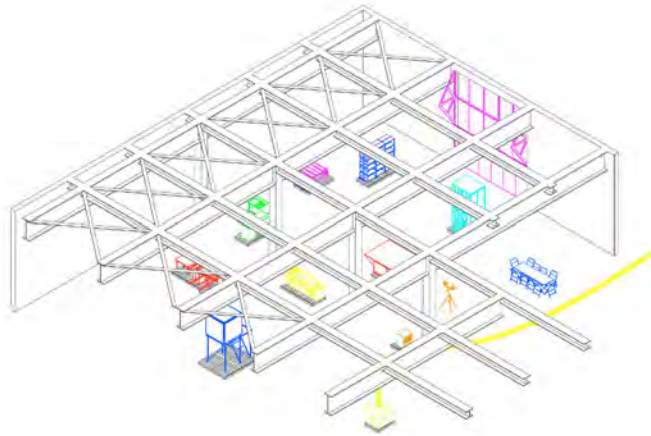


// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN

CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE  
uploading and positioning TOOLS; // mapping Teufelsberg  
// Steel Ring, ALDO VAN EYCK

? future



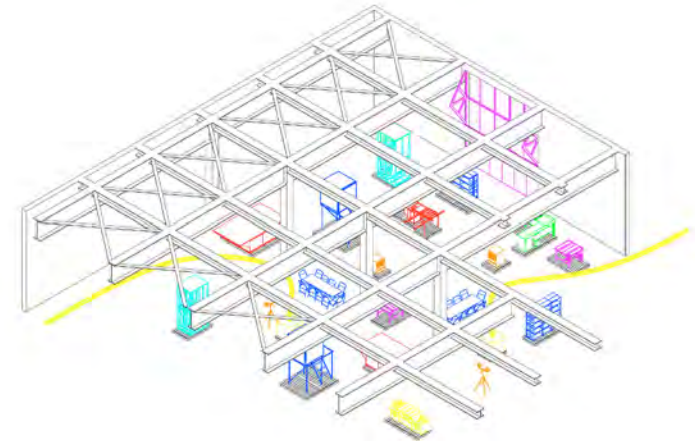
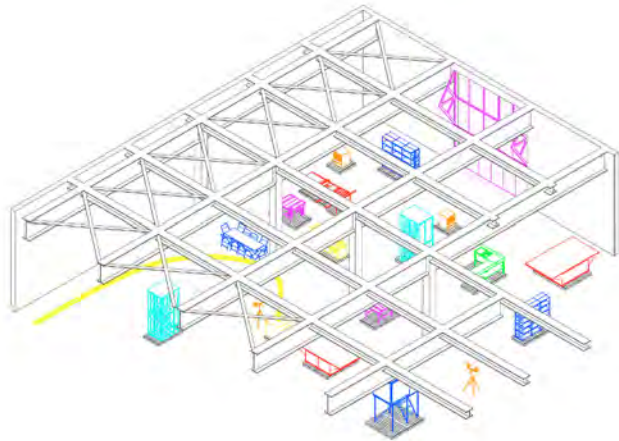


// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN

CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE  
uploading and positioning TOOLS; // mapping Teufelsberg  
// Steel Ring, ALDO VAN EYCK

? peak

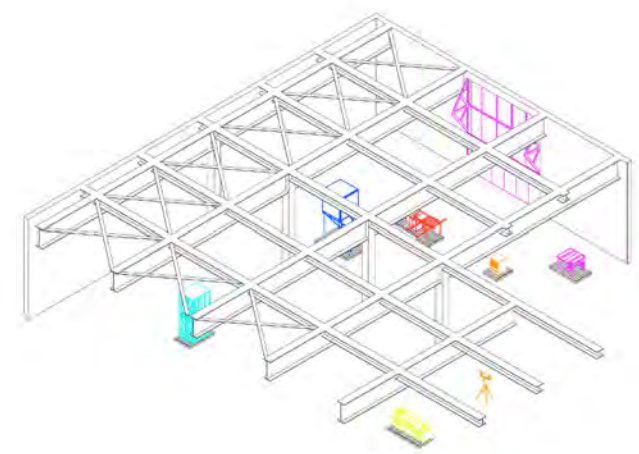
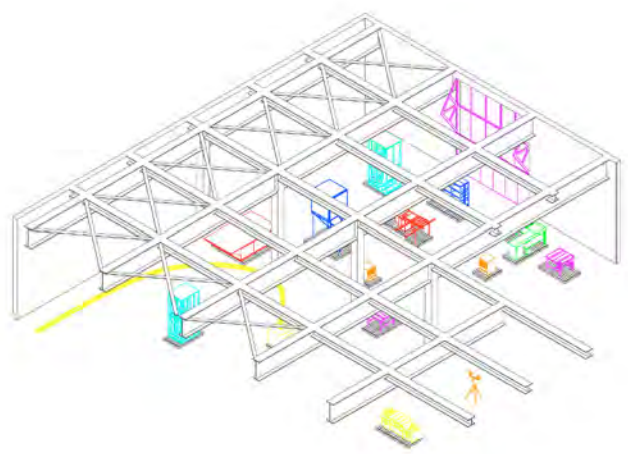


// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN

CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE  
uploading and positioning TOOLS; // mapping Teufelsberg  
// Steel Ring, ALDO VAN EYCK

? depression

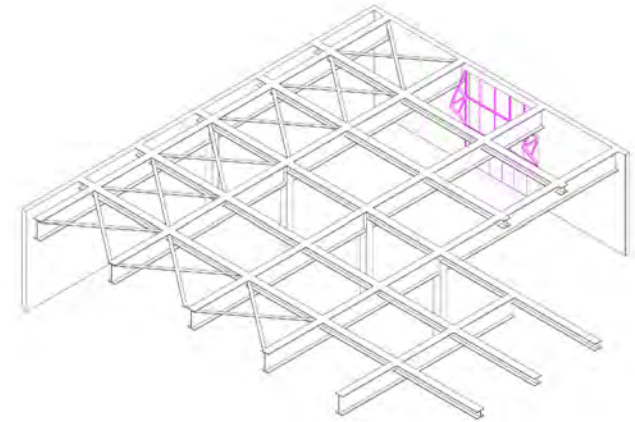


```
// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN
```

```
CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity
```

```
OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE  
  uploading and positioning TOOLS; // mapping Teufelsberg  
  // Steel Ring, ALDO VAN EYCK  
  // Playtime, JAQUES TATI
```

```
? end
```



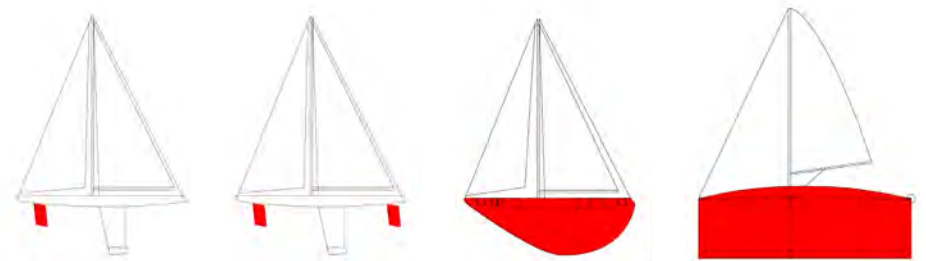
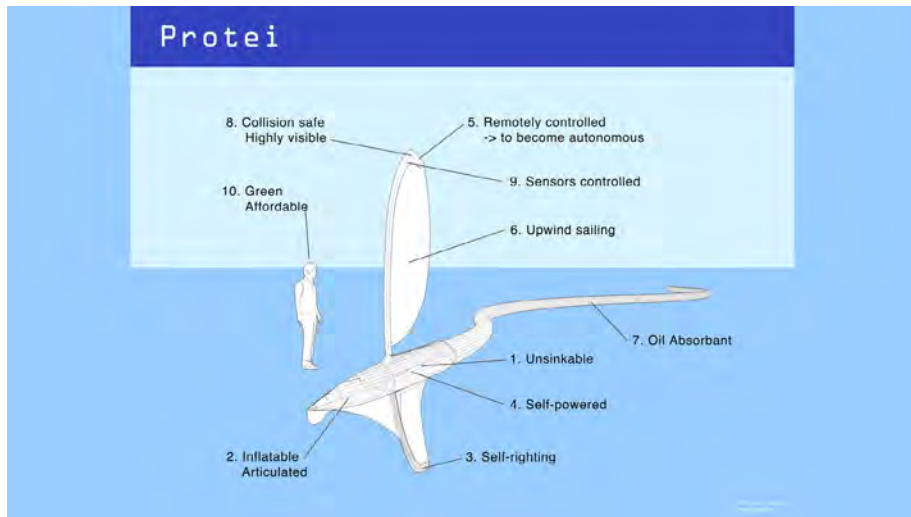
```
// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN
```

```
CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity
```

```
OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE  
  uploading and positioning TOOLS; // mapping Teufelsberg  
    // Steel Ring, ALDO VAN EYCK  
    // Playtime, JAQUES TATI  
}
```

```
(FIELD) STATION < problems - solution; // Protei Sailboat
```

series of prototypes



// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN

CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE  
uploading and positioning TOOLS; // mapping Teufelsberg  
// Steel Ring, ALDO VAN EYCK  
// Playtime, JAQUES TATI

}

(FIELD) STATION < problems - solution; // Protei Sailboat  
investigation - knowledge; // Drone Survival Guide  
// necessity of context

guide - manual





```
// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN
CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE
  uploading and positioning TOOLS; // mapping Teufelsberg
  // Steel Ring, ALDO VAN EYCK
  // Playtime, JAQCUES TATI
}

(FIELD) STATION < problems - solution; // Protei Sailboat
  investigation - knowledge; // Drone Survival Guide
  // necessity of context

void setup() {
  0 CONTEXT
  CODE: start (one time only) FIELD STATION 0 as open hardware; // Open Source Ecology, MARICIN JAKUBOWSKI
```



catalogue



**cabinet**  
2200 x 600 x 2000



**gate**  
4800 x 300 x 6500



**site lamp**  
400 WATT



**fireplace**  
1000 x 1000 x 1200



**worktable**  
3500 x 2000 x 800



**kitchenette**  
2200 x 1600 x 800



**watertank**  
2800 x 1900 x 4500



**woodcabinet**  
1400 x 1100 x 3000



**saw table**  
3000 x 1700 x 600



**workbench**  
1100 x 1100 x 600



**toolbox**  
600 x 600 x 700



```
// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN
CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

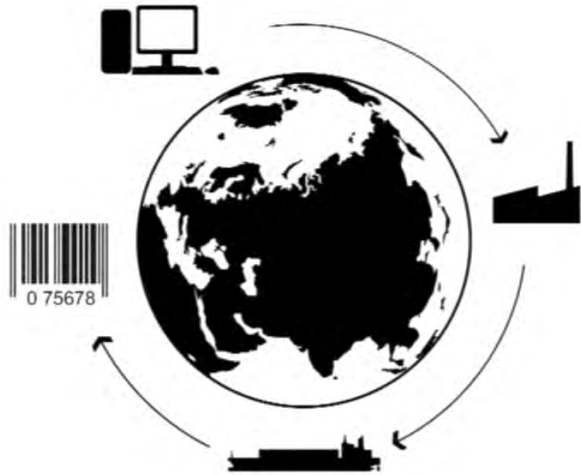
OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE
  uploading and positioning TOOLS; // mapping Teufelsberg
                                  // Steel Ring, ALDO VAN EYCK
                                  // Playtime, JAQUES TATI
}

(FIELD) STATION < problems - solution; // Protei Sailboat
            investigation - knowledge; // Drone Survival Guide
            // necessity of context

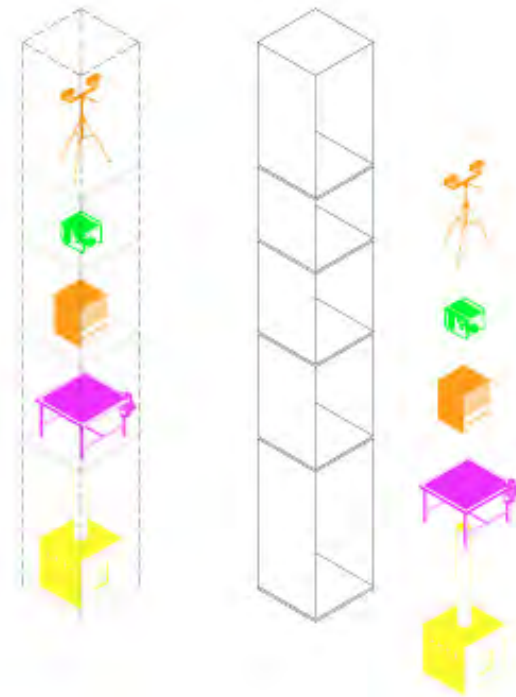
void setup() {
  0 CONTEXT

CODE: start (one time only) FIELD STATION 0 as open hardware; // Open Source Ecology, MARICIN JAKUBOWSKI

INPUT = FIELD STATION 0 = COMPONENTS; // WikiHouse, ALISTAIR PARVIN
      // it's easier to ship recipes than cakes and biscuits
```



first batch of cookies



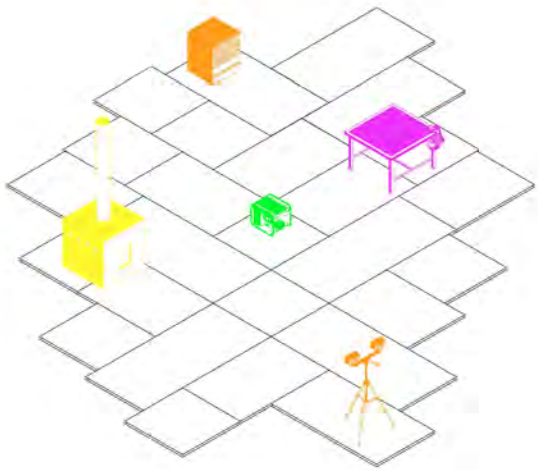
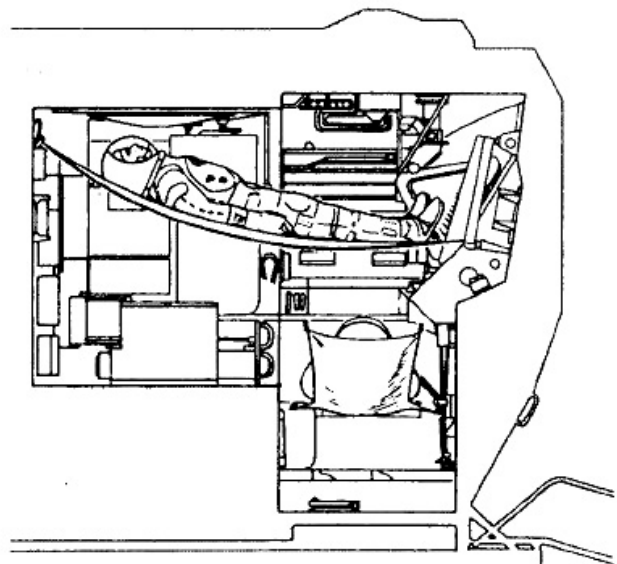
```
// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN
CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE
  uploading and positioning TOOLS; // mapping Teufelsberg
  // Steel Ring, ALDO VAN EYCK
  // Playtime, JAQUES TATI
}

(FIELD) STATION < problems - solution; // Protei Sailboat
  investigation - knowledge; // Drone Survival Guide
  // necessity of context

void setup() {
  0 CONTEXT
  CODE: start (one time only) FIELD STATION 0 as open hardware; // Open Source Ecology, MARICIN JAKUBOWSKI
  INPUT = FIELD STATION 0 = COMPONENTS; // WikiHouse, ALISTAIR PARVIN
  // it's easier to ship recipes than cakes and biscuits
  COMPONENTS: first suggestion of TOOLS; // Apollo Lunar Module
}
```

dissolves

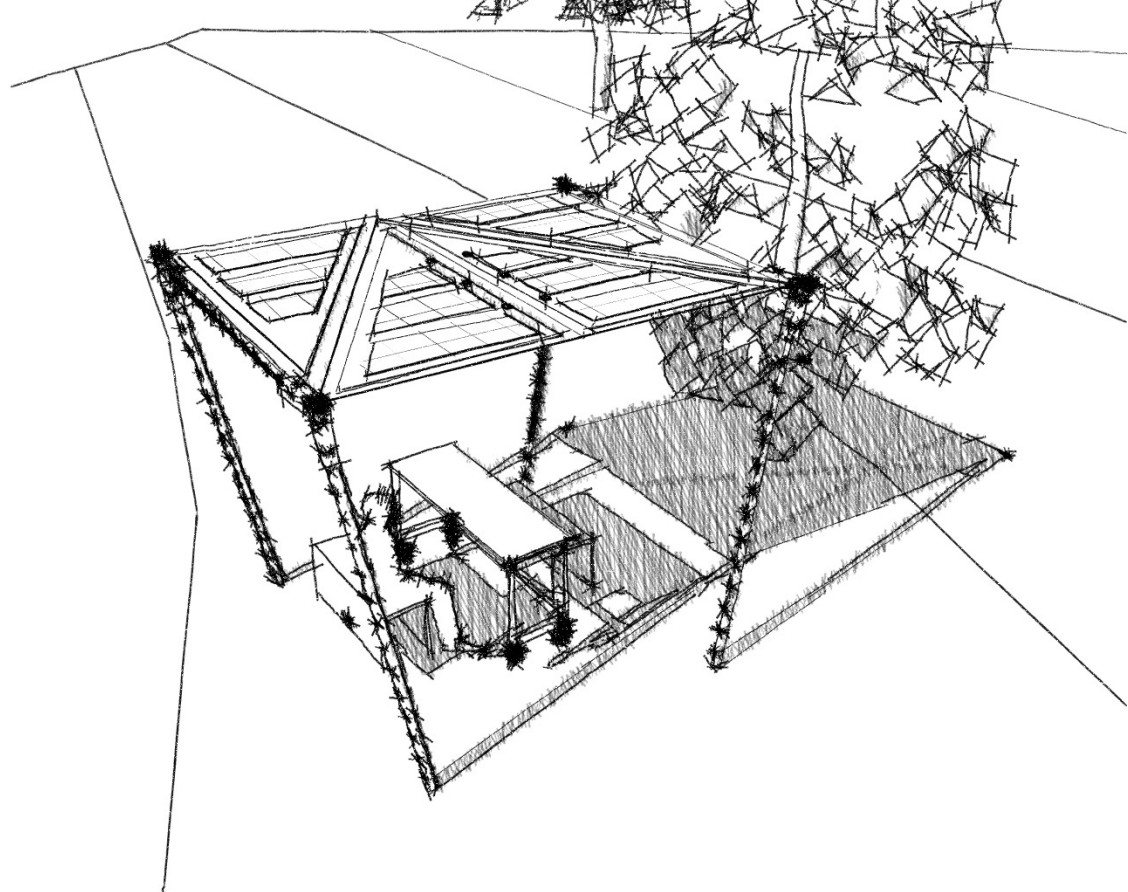


```
// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN
CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE
  uploading and positioning TOOLS; // mapping Teufelsberg
  // Steel Ring, ALDO VAN EYCK
  // Playtime, JAQUES TATI
}

(FIELD) STATION < problems - solution; // Protei Sailboat
  investigation - knowledge; // Drone Survival Guide
  // necessity of context

void setup() {
  0 CONTEXT
  CODE: start (one time only) FIELD STATION 0 as open hardware; // Open Source Ecology, MARICIN JAKUBOWSKI
  INPUT = FIELD STATION 0 = COMPONENTS; // WikiHouse, ALISTAIR PARVIN
  // it's easier to ship recipes than cakes and biscuits
  COMPONENTS: first suggestion of TOOLS; // Apollo Lunar Module
}
```



```
// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN
```

```
CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity
```

```
OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE  
  uploading and positioning TOOLS; // mapping Teufelsberg  
    // Steel Ring, ALDO VAN EYCK  
    // Playtime, JAQUES TATI  
}
```

```
(FIELD) STATION < problems - solution; // Protei Sailboat  
  investigation - knowledge; // Drone Survival Guide  
  // necessity of context
```

```
void setup() {
```

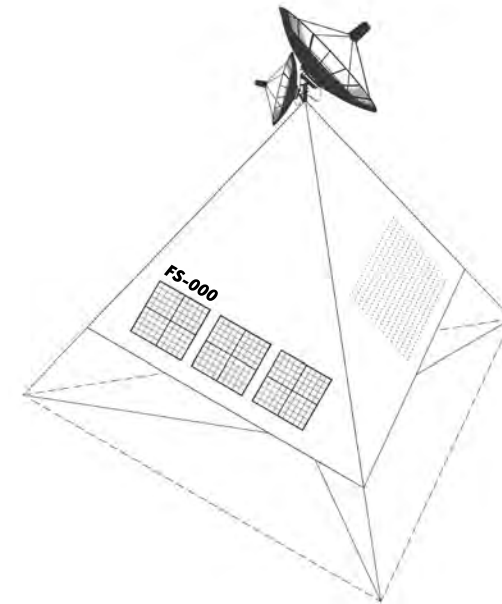
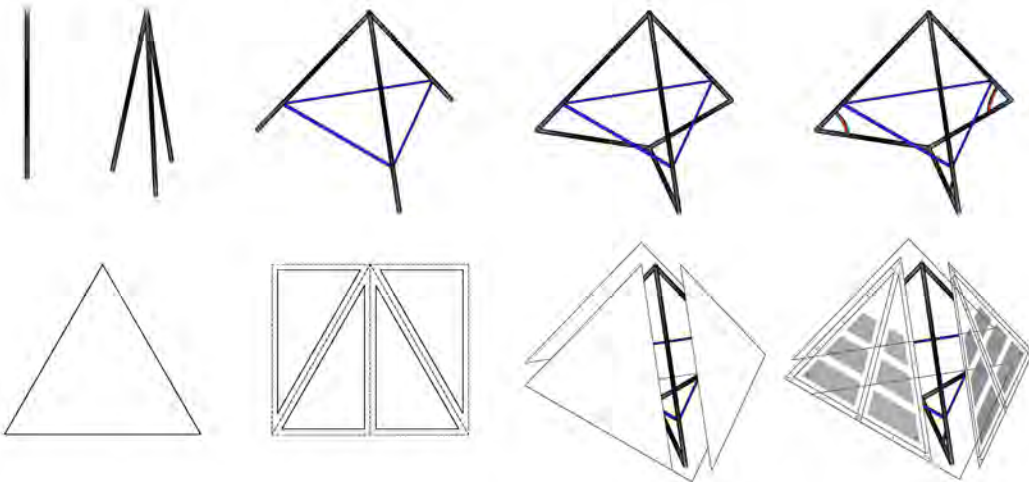
```
  0 CONTEXT
```

```
CODE: start (one time only) FIELD STATION 0 as open hardware; // Open Source Ecology, MARICIN JAKUBOWSKI
```

```
INPUT = FIELD STATION 0 = COMPONENTS; // WikiHouse, ALISTAIR PARVIN  
  // it's easier to ship recipes than cakes and biscuits
```

```
COMPONENTS: first suggestion of TOOLS; // Apollo Lunar Module
```

```
}
```





```

// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN
CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE
  uploading and positioning TOOLS; // mapping Teufelsberg
                                  // Steel Ring, ALDO VAN EYCK
                                  // Playtime, JAQCUES TATI
}

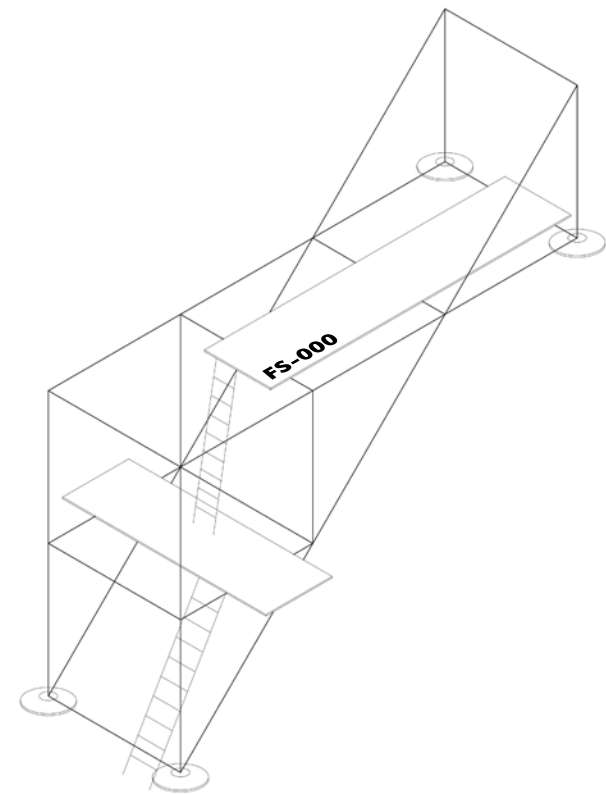
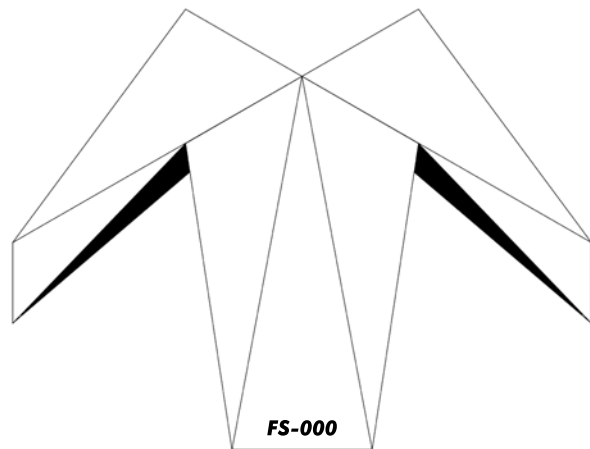
(FIELD) STATION < problems - solution; // Protei Sailboat
            investigation - knowledge; // Drone Survival Guide
            // necessity of context

void setup() {
  0 CONTEXT
CODE: start (one time only) FIELD STATION 0 as open hardware; // Open Source Ecology, MARICIN JAKUBOWSKI

INPUT = FIELD STATION 0 = COMPONENTS; // WikiHouse, ALISTAIR PARVIN
      // it's easier to ship recipes than cakes and biscuits

COMPONENTS: first suggestion of TOOLS; // Apollo Lunar Module
}

```



```
// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN
CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

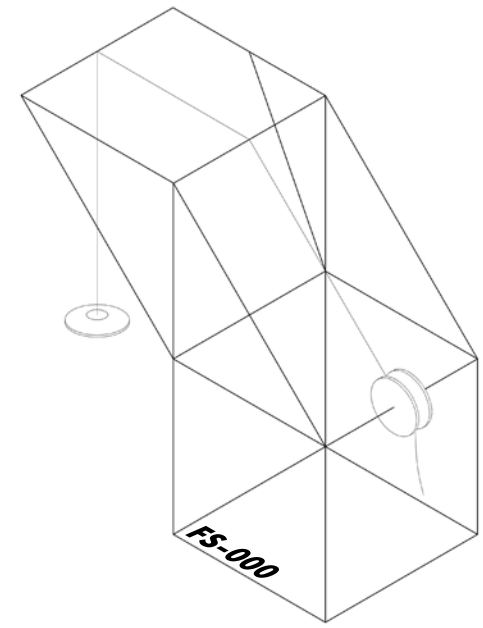
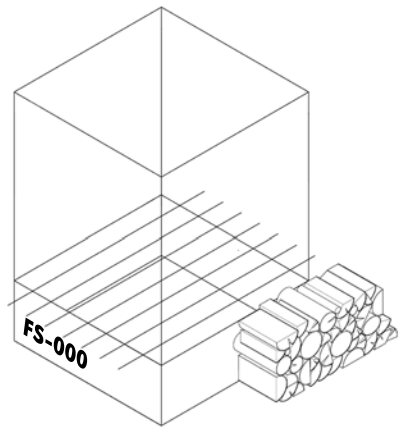
OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE
  uploading and positioning TOOLS; // mapping Teufelsberg
  // Steel Ring, ALDO VAN EYCK
  // Playtime, JAQUES TATI
}

(FIELD) STATION < problems - solution; // Protei Sailboat
  investigation - knowledge; // Drone Survival Guide
  // necessity of context

void setup() {
  0 CONTEXT
CODE: start (one time only) FIELD STATION 0 as open hardware; // Open Source Ecology, MARICIN JAKUBOWSKI

INPUT = FIELD STATION 0 = COMPONENTS; // WikiHouse, ALISTAIR PARVIN
  // it's easier to ship recipes than cakes and biscuits

COMPONENTS: first suggestion of TOOLS; // Apollo Lunar Module
}
```



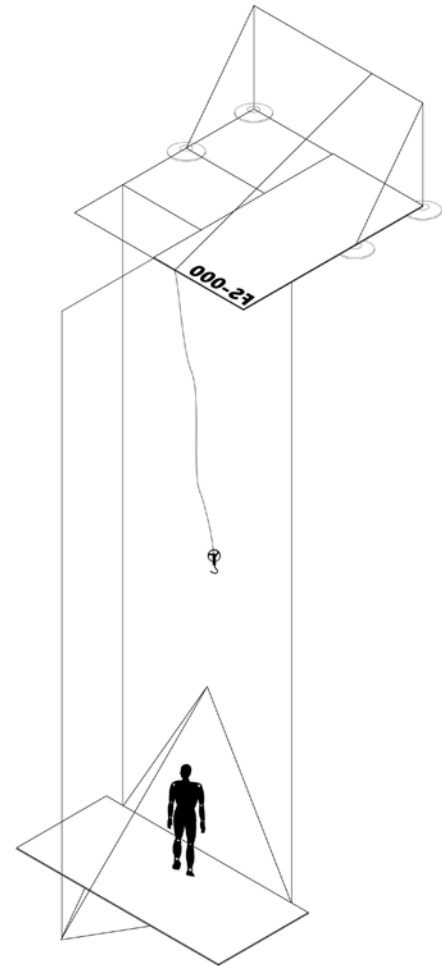
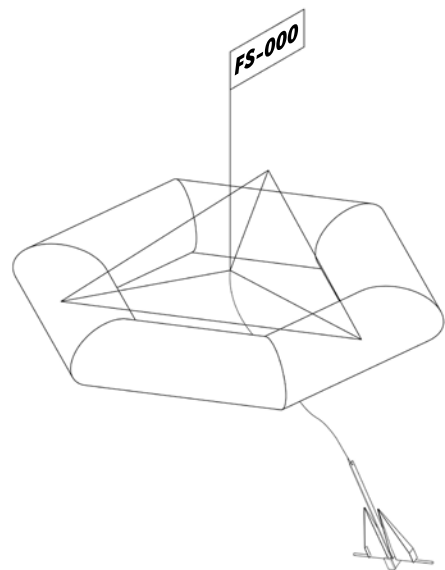


```
// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN
CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE
  uploading and positioning TOOLS; // mapping Teufelsberg
  // Steel Ring, ALDO VAN EYCK
  // Playtime, JAQUES TATI
}

(FIELD) STATION < problems - solution; // Protei Sailboat
  investigation - knowledge; // Drone Survival Guide
  // necessity of context

void setup() {
  0 CONTEXT
  CODE: start (one time only) FIELD STATION 0 as open hardware; // Open Source Ecology, MARICIN JAKUBOWSKI
  INPUT = FIELD STATION 0 = COMPONENTS; // WikiHouse, ALISTAIR PARVIN
  // it's easier to ship recipes than cakes and biscuits
  COMPONENTS: first suggestion of TOOLS; // Apollo Lunar Module
}
```



```
// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN
CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE
  uploading and positioning TOOLS; // mapping Teufelsberg
                                  // Steel Ring, ALDO VAN EYCK
                                  // Playtime, JAQCUES TATI
}

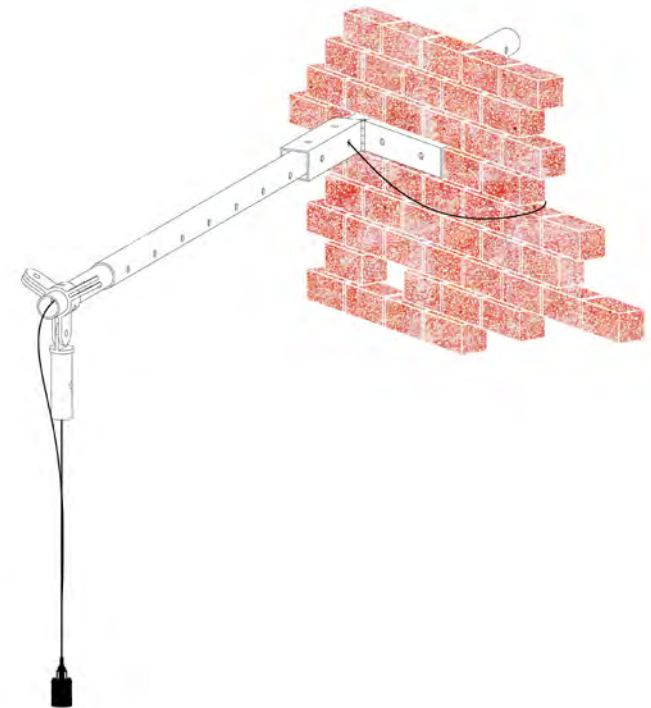
(FIELD) STATION < problems - solution; // Protei Sailboat
               investigation - knowledge; // Drone Survival Guide
               // necessity of context

void setup() {
  0 CONTEXT
  CODE: start (one time only) FIELD STATION 0 as open hardware; // Open Source Ecology, MARICIN JAKUBOWSKI

  INPUT = FIELD STATION 0 = COMPONENTS; // WikiHouse, ALISTAIR PARVIN
        // it's easier to ship recipes than cakes and biscuits

  COMPONENTS: first suggestion of TOOLS; // Apollo Lunar Module
}

void loop() {
+ CONTEXT: FIELD STATION (n);
```



```

// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN
CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE
  uploading and positioning TOOLS; // mapping Teufelsberg
                                // Steel Ring, ALDO VAN EYCK
                                // Playtime, JAQUES TATI
}

(FIELD) STATION < problems - solution; // Protei Sailboat
              investigation - knowledge; // Drone Survival Guide
              // necessity of context

void setup() {
  0 CONTEXT

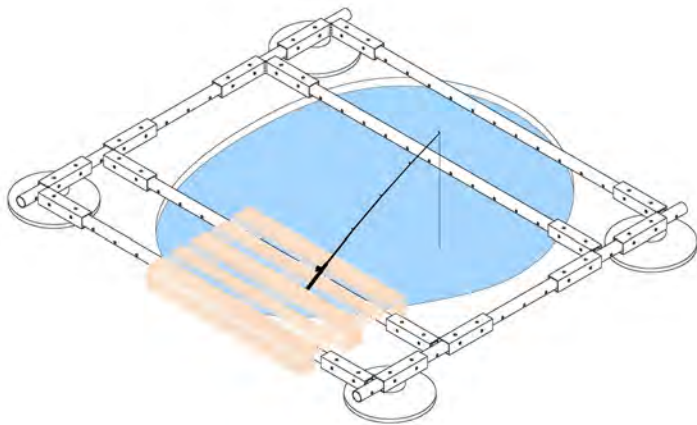
CODE: start (one time only) FIELD STATION 0 as open hardware; // Open Source Ecology, MARICIN JAKUBOWSKI

  INPUT = FIELD STATION 0 = COMPONENTS; // WikiHouse, ALISTAIR PARVIN
        // it's easier to ship recipes than cakes and biscuits

  COMPONENTS: first suggestion of TOOLS; // Apollo Lunar Module
}

void loop() {
+ CONTEXT: FIELD STATION (n);

```



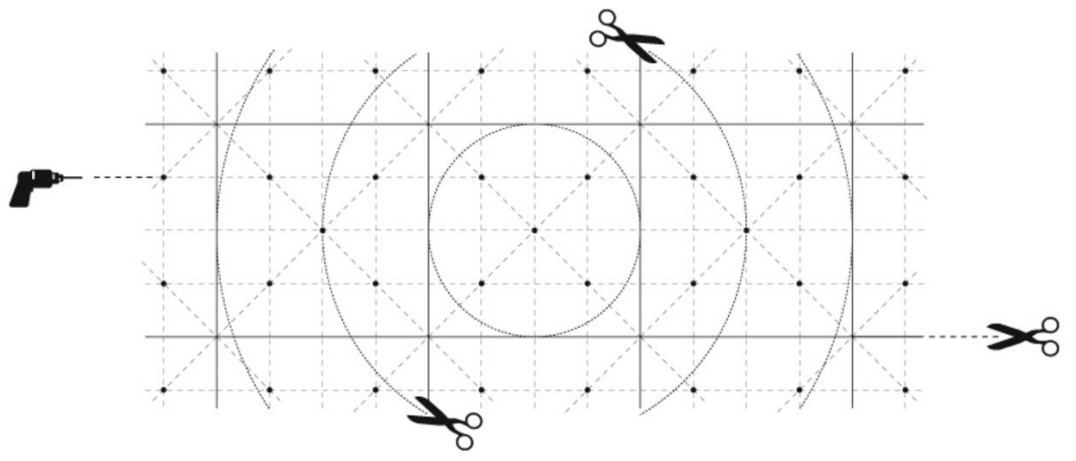
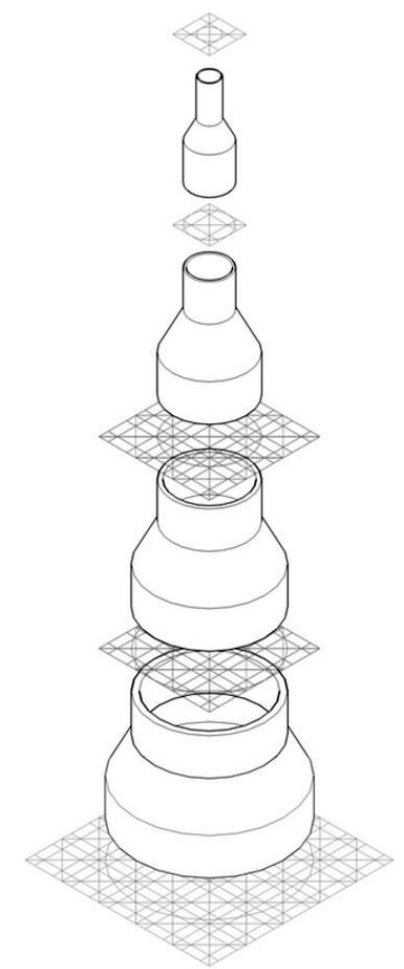
```
// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN
CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE
  uploading and positioning TOOLS; // mapping Teufelsberg
  // Steel Ring, ALDO VAN EYCK
  // Playtime, JAQUES TATI
}

(FIELD) STATION < problems - solution; // Protei Sailboat
  investigation - knowledge; // Drone Survival Guide
  // necessity of context

void setup() {
  0 CONTEXT
  CODE: start (one time only) FIELD STATION 0 as open hardware; // Open Source Ecology, MARICIN JAKUBOWSKI
  INPUT = FIELD STATION 0 = COMPONENTS; // WikiHouse, ALISTAIR PARVIN
  // it's easier to ship recipes than cakes and biscuits
  COMPONENTS: first suggestion of TOOLS; // Apollo Lunar Module
}

void loop() {
+ CONTEXT: FIELD STATION (n);
FIELD STATION (n+1): addition, mutation, implementation of FIELD STATION (n) // Open Structures
```





```
// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN
CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE
  uploading and positioning TOOLS; // mapping Teufelsberg
  // Steel Ring, ALDO VAN EYCK
  // Playtime, JAQUES TATI
}

(FIELD) STATION < problems - solution; // Protei Sailboat
  investigation - knowledge; // Drone Survival Guide
  // necessity of context

void setup() {
0 CONTEXT

CODE: start (one time only) FIELD STATION 0 as open hardware; // Open Source Ecology, MARICIN JAKUBOWSKI

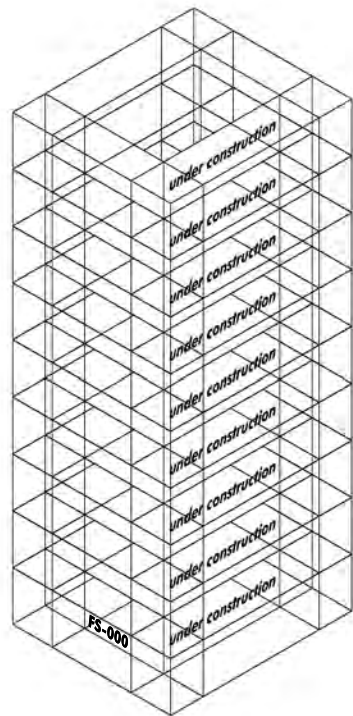
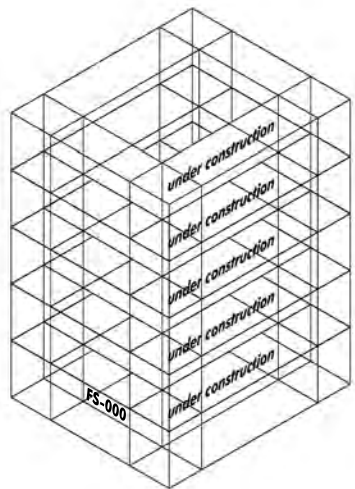
INPUT = FIELD STATION 0 = COMPONENTS; // WikiHouse, ALISTAIR PARVIN
  // it's easier to ship recipes than cakes and biscuits

COMPONENTS: first suggestion of TOOLS; // Apollo Lunar Module
}

void loop() {
+ CONTEXT: FIELD STATION (n);

FIELD STATION (n+1): addition, mutation, implementation of FIELD STATION (n) // Open Structures

If END {
  activation of space?;
}
}
```



```
// THE EXECUTION OF CODE IN SPACE // Self-Replicating Machine, JOHN VON NEUMANN
CODE (field): OPEN SOURCE ARCHITECTURE; learning from the network; // as analogy, as necessity

OUTPUT = the activation of space { // Fun Palace, CEDRIC PRICE
  uploading and positioning TOOLS; // mapping Teufelsberg
  // Steel Ring, ALDO VAN EYCK
  // Playtime, JAQUES TATI
}

(FIELD) STATION < problems - solution; // Protei Sailboat
  investigation - knowledge; // Drone Survival Guide
  // necessity of context

void setup() {
  0 CONTEXT
  CODE: start (one time only) FIELD STATION 0 as open hardware; // Open Source Ecology, MARICIN JAKUBOWSKI
  INPUT = FIELD STATION 0 = COMPONENTS; // WikiHouse, ALISTAIR PARVIN
  // it's easier to ship recipes than cakes and biscuits
  COMPONENTS: first suggestion of TOOLS; // Apollo Lunar Module
}

void loop() {
  + CONTEXT: FIELD STATION (n);
  FIELD STATION (n+1): addition, mutation, implementation of FIELD STATION (n) // Open Structures
  If END {
    activation of space?;
  }
  If notEND {
    FIELD STATION (n+1);
  }
}
```

