



DEEP IN THE UNIVERSE

After the final battle, humanity is gone, and you may be the last living being in this part of the Universe. Space stations and spaceships are still there in the darkness, unmanned, maintained by pre-programmed robots. And they have data: your only clue on the way to find remaining humans.

The game

You cannot enter ships and you cannot control your robots directly. The only way to finish the task lies in the **art of programming**.

- Let droids board the ship...
- ...collect the data...
- ...and destroy everything that stands in the way.

YOUR MISSION

In each mission, your task is to find terminals, download the data, and bring them back. For that, you may use up to four robots. They cannot do anything on their own, so you need to command them to win. Click on the robot in the base or use keys 1-4 to select the robot. Write a chain of command; see the PROGRAMMING section below for details. You have an infinite number of tries, your dropship can create as many robots as you need. Once you start the mission, you can speed up or slow down time with + and – keys and watch the action through the eyes of any robot by pressing the appropriate number key. Do not bother to clean up: once you win the mission, robots will get back to the base (actually, they self-destruct and will be created anew, but let's not talk about it).

Battle Droid



Programmable battle droid. Fast, and versatile. Can detect enemies and shoot laser beams on long distances. Holds up to 1 PB of data.

Guard Bot



Multi-purpose guarding bot. Slow, but strong. Will destroy anything with a short-range EMP pulse once it gets close enough.

The Terminal



Encrypted data are held in terminals. Your robot will hack into it and decrypt the data on the fly. It needs only a few seconds of time.

The Databank



The data need to be brought back to the databank in your dropship. Only one robot can access the databank at one time.

PROGRAMMING

The task of each mission is to bring the requested amount of data back to the base and upload it to the databank. For that you need to program up to four robots to enter, fight, defend, retrieve, and cooperate. [Hide details...](#)

Basic commands:

- **MOVE:** make one step ahead
- **TURN LEFT / RIGHT:** rotate 90 degrees
- **DOWNLOAD:** download 1 data packet from the terminal
- **UPLOAD:** upload 1 data packet to the databank
- **WAIT:** wait for a short while
- **SHOOT:** shoot ahead (enemy robots needs two shots to die)
- **OPERATE:** use the switch in front of the robot (to open the door etc.)

Conditions and loops:

- **IF** cond [ELSE]: executes only if true (IF enemy: SHOOT – shoots once if there is a target)
- **WHILE** cond: executes while condition is true (WHILE target: SHOOT – shoots till the target is destroyed or gone)
- Possible conditions (there is also a negative version of each):
 - **WALL AHEAD:** true, when the wall is on the adjacent field (WHILE not wall ahead: MOVE – moves to the end of the corridor)
 - **ENEMY:** true, when a target is visible, i.e. in the direct line of sight (WHILE target: SHOOT – shoot until the target is destroyed or out of sight)
 - **ALIVE:** always valid, always true (WHILE alive: MOVE – moves to the end of the world or beyond)
 - **MARK PRESENT:** true, when standing on the mark of given color (WHILE not red mark present: MOVE – move to the mark)
 - **SIGNAL ACTIVE:** true, when someone started the signal (WHILE not signal alpha active: TURN LEFT – turns left until someone starts the signal)

Marking and signaling:

- **PLACE MARK:** places one mark (stamp, beeper) of a given color on the current field
- **TURN signal ON / OFF:** turns the signal on or off (once started, the signal will be active forever or until someone turns it off)
- **SWITCH SIGNAL:** turns off the signal that was turned on and vice versa
- **WAIT FOR SIGNAL:** waits and does nothing until someone starts the signal

WHILE cycle is meant to encapsulate a part of the code. At the beginning, the condition is checked; if it is true, the code will execute and once it's done, the condition is checked again. If it's false, the code is skipped.

The code editor itself is meant to work without any prior knowledge or skill in programming. The code can be easily edited by arrow keys or console controller, but the full keyboard support is included (and keyboard is recommended to use). The easiest way to control the editor is by simply writing:

1. Open the robot's coding console by pressing number of the droid.
2. Press Enter to insert an empty line.
3. By pressing Enter again, you open a list of available commands.
- or -
4. By starting to write, you can shorten the list to find the wanted command. So, if you want to move robot ahead, just press Enter and start to type MOVE

Copy & Paste works on block of codes: select IF or WHILE and press Ctrl+C or Ctrl+X (cut). Select opening INITIALIZE command to copy whole program.

Example code snippets:

Run to wall:

```
while(NOT WALL)
  Move
end
```

Turn at mark:

```
while(NOT RED MARK)
  Move
end
TurnRight
```

Wait for enemy; destroy:

```
while(NOT ENEMY)
end
while(ENEMY)
  Shoot
end
```

Video #1: Programming Guide 1

This is part of “Automatica Stream”. Viktor, a designer of the game, explains the very basics of Battle Droid programming on a simple mission.



Video #2: Programming Guide 2

The second part of “Automatica Stream” shows how to beat a quite complicated mission. Many important tricks can be learned here.

