

by Maximilian Wöhrl (2024) for MSXdev24

Introduction

ROBO RACE is a turn-based programming game in which three players (human or computer) are controlling their robot. The goal is to be the first one to reach a number of checkpoints marked by colored flags on a factory floor. The first robot to collect all flags in numbered order wins the game.

Hardware requirements

MSX1 computer with 64k of RAM

Optional: one or two gamepads / joysticks

The program is freeware

Getting started

The game starts on the main menu screen where you can choose players' colors, the map you will be playing on, the number of flags (which directly affects the game length) and weather or not the robots shoot lasers at each other.

By pressing the joystick button or spacebar, you get to the next menu where you can assign a controller to each robot, or have it controlled by the computer's AI. Pressing a button once again starts the game.

Programming your robot

At the start of each round, you get up to nine programming cards from which you have to program a sequence of five moves. These cards are:



Move 1, 2 or 3 squares forward




Move 1 square backwards




Turn left or right by 90 degrees



Turn by 180 degrees

Once all five registers are filled, you can commit your sequence by pressing , and the next player gets to program their robot.

You may press  to scroll the map and have a look around. Pressing the button once again gets you back to programming.

When all three robots are programmed, the execution phase begins.

Program execution

The programmed moves are executed sequentially. The cards priority determines the order in which the robots move. As a rule of thumb, moving cards have a higher priority than rotating cards, and the more squares a card moves, the higher its priority. However, priority ranges overlap, so do not rely too much on this rule and be prepared for some chaos!

After each robot has executed their move, the factory elements take their respective action, which are described in the next paragraph. When all five registers are executed, the next programming round commences.

Collisions

If a robot moves into a square that is already occupied by another robot, this robot is shifted onto the next field (unless it hits a wall). This might mess up the whole programming and a robot might end up somewhere completely different than intended, especially when some factory elements come into play. So, prepare for some chaos and Schadenfreude!

Factory elements

The following factory elements are triggered in order:



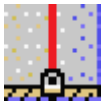
Express conveyor: moves a robot in designated direction, also when standard conveyors move.



Standard conveyor: moves (and rotates) a robot after the express conveyors have moved.



Turntable: Rotates a robot by 90 degrees in the indicated direction.



Laser or double laser: fires a laser if a robot stands in its beam. The robot takes 1 damage per hit (see paragraph "Taking damage").



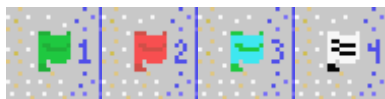
Repair or double repair: if a move ends on one of these squares, a robot gets 1 or 2 damage repaired. If a robot finishes the 5th move on a double repair square AND has no damage, it gains an extra life.

Robot lasers (optional)

After the factory elements have triggered, all robots fire their lasers if they have another robot in their line of sight. This causes 1 damage point to the robot hit by the laser.

Checkpoints

Moving over the flag square that is currently on, you gain the flag of the current color. Now the next flag color is on. The first robot who gains the 4th flag (or less if selected) wins the game. Furthermore, a flag square behaves like a single repair square.



Taking damage

Initially, a robot has 9 hit points and therefore gets 9 cards dealt on the start of the programming phase. For each damage a robot takes, one less card is dealt. If the number of cards dealt gets below 5, the top registers are locked and will execute the previous move. Locked registers are marked red. A player can use repair squares to remove damage points or he may announce a shutdown (see next paragraph).

Shutdown

A shutdown is used to repair the robot and restore its full health, regardless of the damage taken previously.

Players can announce a shutdown for the next round by pressing



after finishing the program for the current round. The robot will still execute the programmed move sequence for the current round and will then skip the following programming phase. However, a player still gets to choose if he wants to announce another shutdown, or go for a normal programming

round by pressing  or  respectively.

During a shutdown, the robot remains completely passive. It can only be moved by factory elements or other players colliding with it. It can also take damage if hit by a laser.

Dying

If a robot takes its 10th damage point or if it falls off the board or into a pit, it loses a life. Players have three lives initially. A dead robot is respawned at its last checkpoint at the beginning of a new round before the programming phase. Respawned robots start with two damage points.

Mini map

By pressing and holding the fire button for 1 second during the programming phase, you can switch to a mini map view.

Good luck and have fun!

Additional notes

ROBO RACE is also available for the Spectravideo Svi328 in cassette or ROM format.