

“Darwin Game” with SPIKE Prime



Introduction

Whoever stands last in the arena with his/her robot in the "Darwin Game" is the winner. Different tactics can lead to the goal. The teams can show their creativity in building the robot and programming and often there are surprising winners.

The teacher can adjust the rules so that each team can successfully participate in the challenge. For example: giving basemodel and program to the students or limit it to less parts).

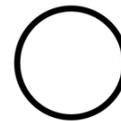
Preparation

Playing field «Arena»:

A smooth, white (or bright) rectangular or round playing arena on the floor, which is closed by a contrasting border.

Size of playing field (recommendation)

- 2-6 SPIKE: rectangle: 1m x 1m or circle: d=1m
- 7-12 SPIKE: rectangle 1,5m x 1,5m or circle: d=1,5m

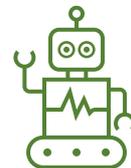


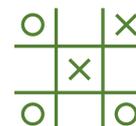
Materials:

- 6m Black adhesive tape 2-5cm wide
- If there is no corresponding floor: solid paper, light-colored wooden board, backside of FLL or WRO
- SPIKE Prime Set and programming device (PC, Tablet, ..) per team
- Building and programming instructions digital or on paper for each team.
- A4 or A5 paper

Robot specifications

- Each team has a SPIKE set at their disposal.
- No part of the robot may exceed the footprint of an A5 paper.
- Only parts of the own SPIKE set may be used.
- The LEGO parts may not be modified.
- No glue, tape, cable ties or other materials may be used.





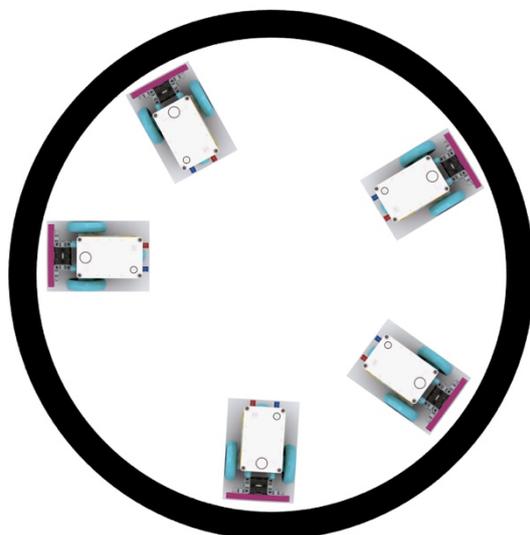
Rules of the game

- The winner is the last person to enter the arena.
- Every robot that leaves the arena (sensor outside the black border) is eliminated.
- Any robot that falls over (cannot continue) is eliminated.
- Robots that do not move or only move in one place are eliminated.
- Robots, which are destructive (the other
- Damage robot) are eliminated.
- Robots that damage the underground are excluded.
- The robots must drive autonomously (no remote control or program change during the game)
- After a round, the teams are given a few minutes for repairs or optimization.

The Game

1. all robots are placed in the ring facing to the border. All robots must first move outwards.
2. on the command "3-2-1-LEGO" the teams start the robots (via button or app)
3. the game continues until only one robot is left in the arena.
4. the game will be interrupted if the active robots cannot move any more (e.g. got stuck) The remaining robots in the game are placed back on the outline and the game is started with the command "3-2-1-LEGO".
5. the robot name of the winning team is written down.

Starting position:



Double lesson 90'

Introduction 5':

- Greeting
- What is the game "Darwin" about
- (Darwin has researched which species prevail)
- Sequence of the double lesson
- Rules for the model

Construction and test phase 45':

- Building a model according to the rules
- Testing in the game arena
- Name assignment of the robot by the team
- Time announcement to inform the teams (with 5min reserve)

Game 25':

- Rules of the game
- 3-5 game rounds
- Optional conversion phases max. 3min

Award Ceremony 5':

- Winning team and all participants receive feedback

Tidy Up and Reflection 10':

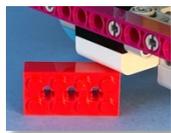
- Robots are dismantled and sorted into the boxes
- Reflection

General tips

1. by specifying a basic model and basic programming, even inexperienced teams can take part in the challenge
2. for advanced SuS the general rules for robots are sufficient.
3. 15min before the start of the Challenge all teams should have successfully tested their robot in the arena. If a team does not have a roadworthy robot at this time, it should use the basic model and basic programming.
4. the teams should give their robot a name already in the construction phase.
5. if there are discussions when placing the robots in the arena, the winning team must first place the robot on the line
6. all spectators must keep enough distance to the arena so that no feet touch the robots.
7. the teacher should announce in time when the challenge will start. 10min, 5min. Experience has shown that a buffer of 5min has to be included here. I.e. start with the 10min announcement 15min before the planned challenge.

Program debugging:

- If the robot moves backwards instead of forwards, the letters must be swapped when defining the motors by clicking the port letter again.
- If the robot does not move at all, the ports must be checked.
- If the robot does not recognize the black line:
 - o If the port is not correct.
 - o If the color sensor is mounted too low or too high. Ideally a 2x4Block fits just underneath.



Building Instructions:



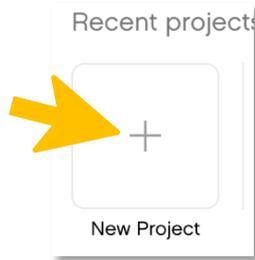
Download-Link: <https://phobricks.ch/ressourcen/>

Block-Program SPIKE Prime Basemode «Stay in the Field»:

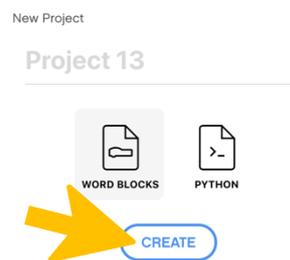
1. Open SPIKE App



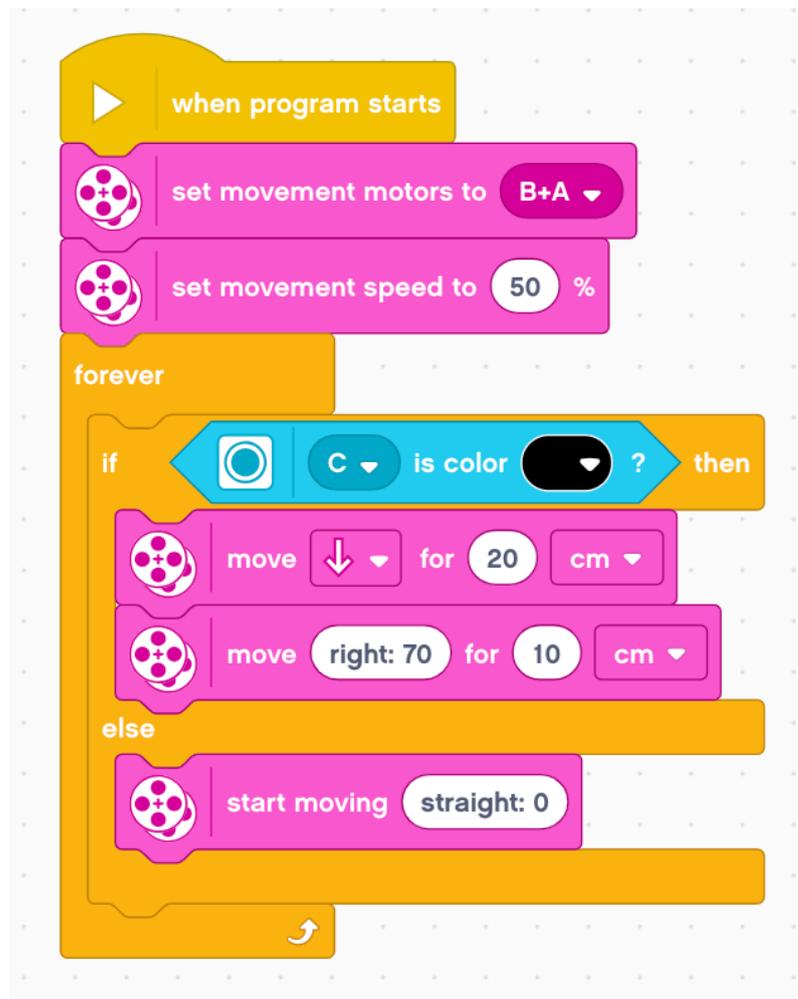
2. New Project:



3. Create new word blocks project:



4. Program:

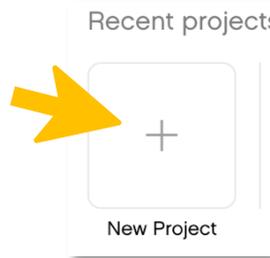


Python Program SPIKE Prime Basemodel «Stay in the Field» (Advanced users):

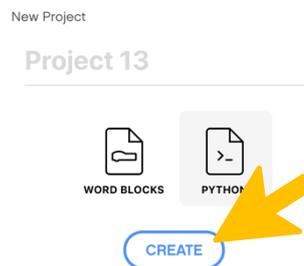
1. Open SPIKE App:



2. New Project



3. Create new python project:



4. Program:

```
1 from spike import PrimeHub, ColorSensor, MotorPair
2
3 myHub = PrimeHub()
4 myColorSensorC = ColorSensor('C')
5 myMotorPair = MotorPair('B', 'A')
6
7 while True:
8     color = myColorSensorC.wait_for_new_color()
9     if color == 'black':
10         myMotorPair.move(20, 'cm', 0, -50)
11         myMotorPair.move(10, 'cm', 70, 50)
12     else:
13         myMotorPair.start(0, 50)
```