

TETRIS





Úvod



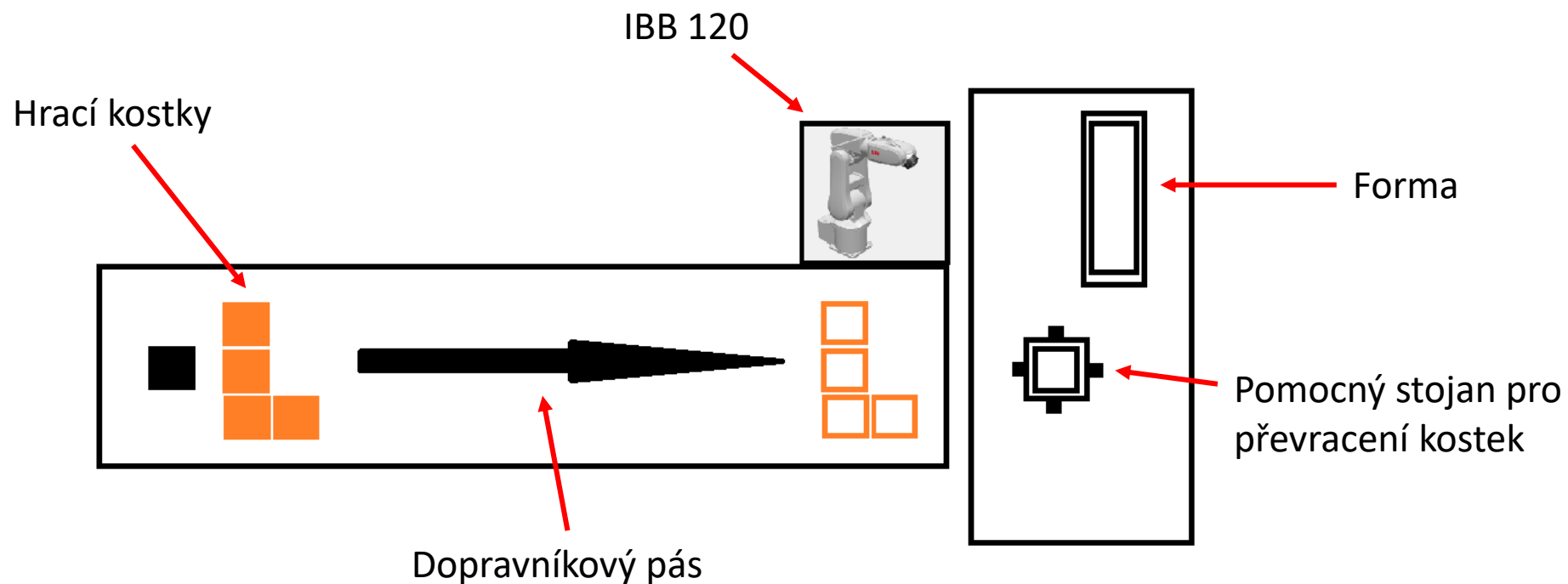
Inspirace

- Robotický stolní fotbal

- Tetris
 - 1984
 - Alexej Pažitnov – Sovětská akademie věd
 - Electronika 60



Schéma pracoviště





Vizuální část

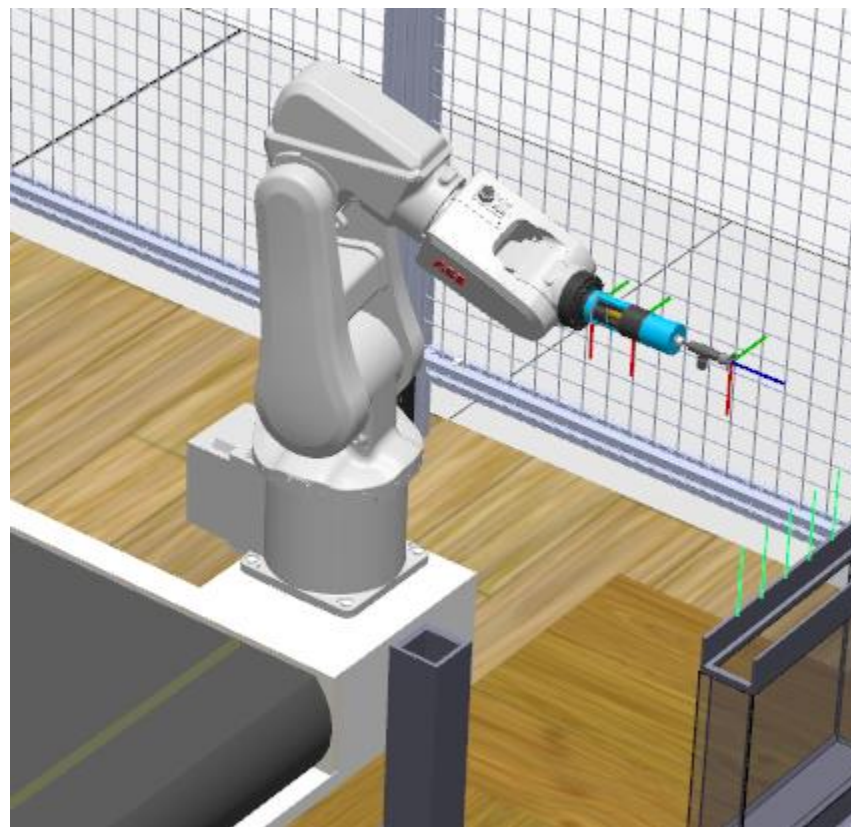




- Aktivní prvky – dopravníkový pás, robotické rameno, 6 kostek, hrací forma, stojan k manipulaci
- Pasivní prvky prostředí

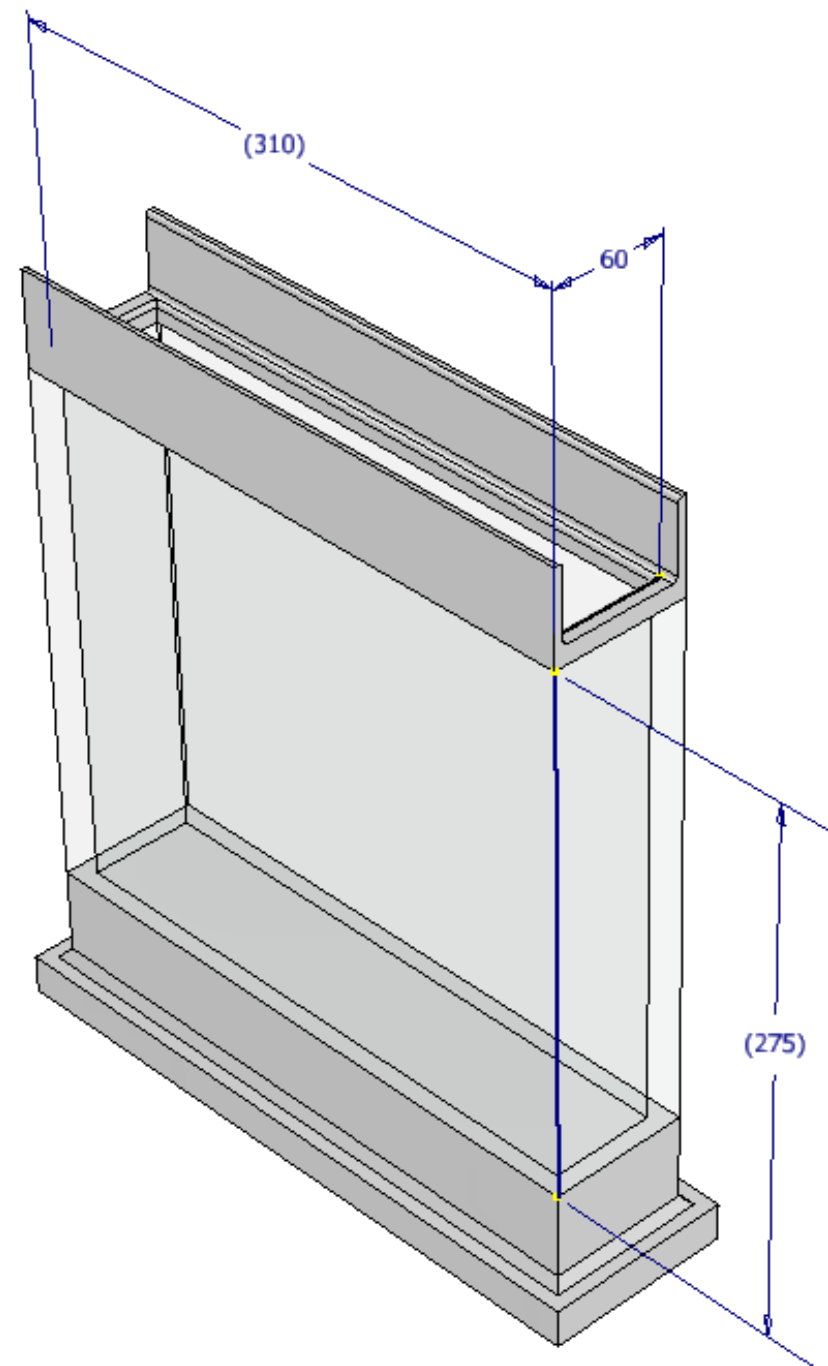
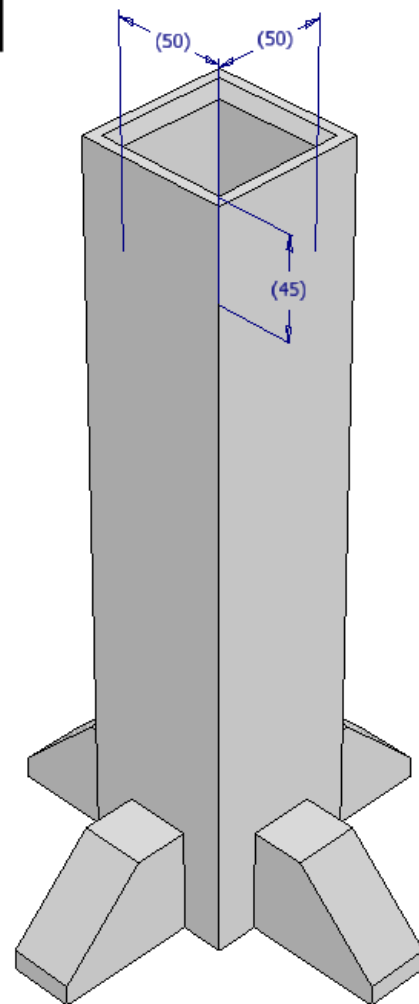
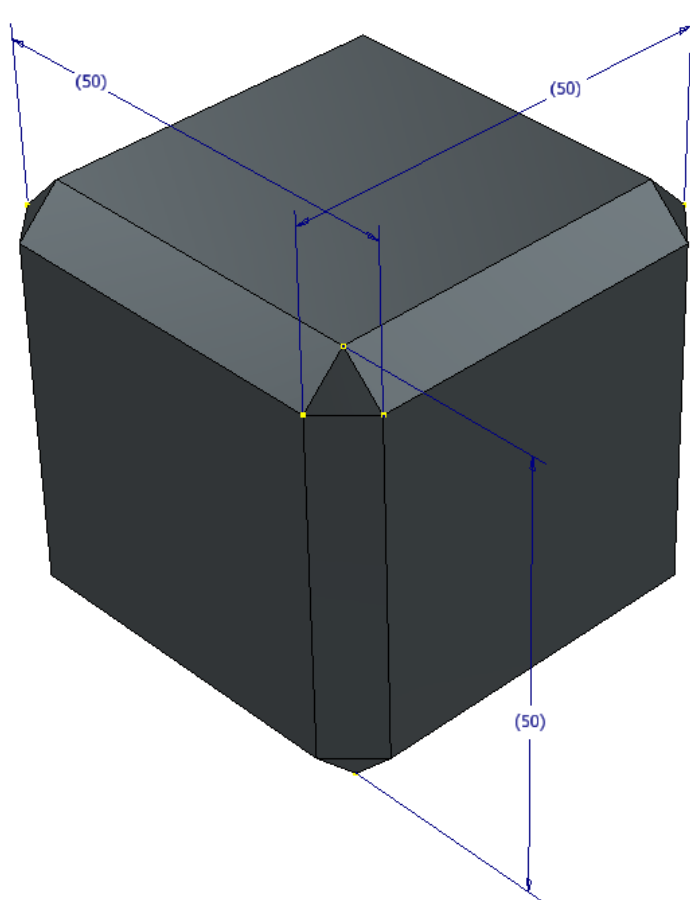


Robot



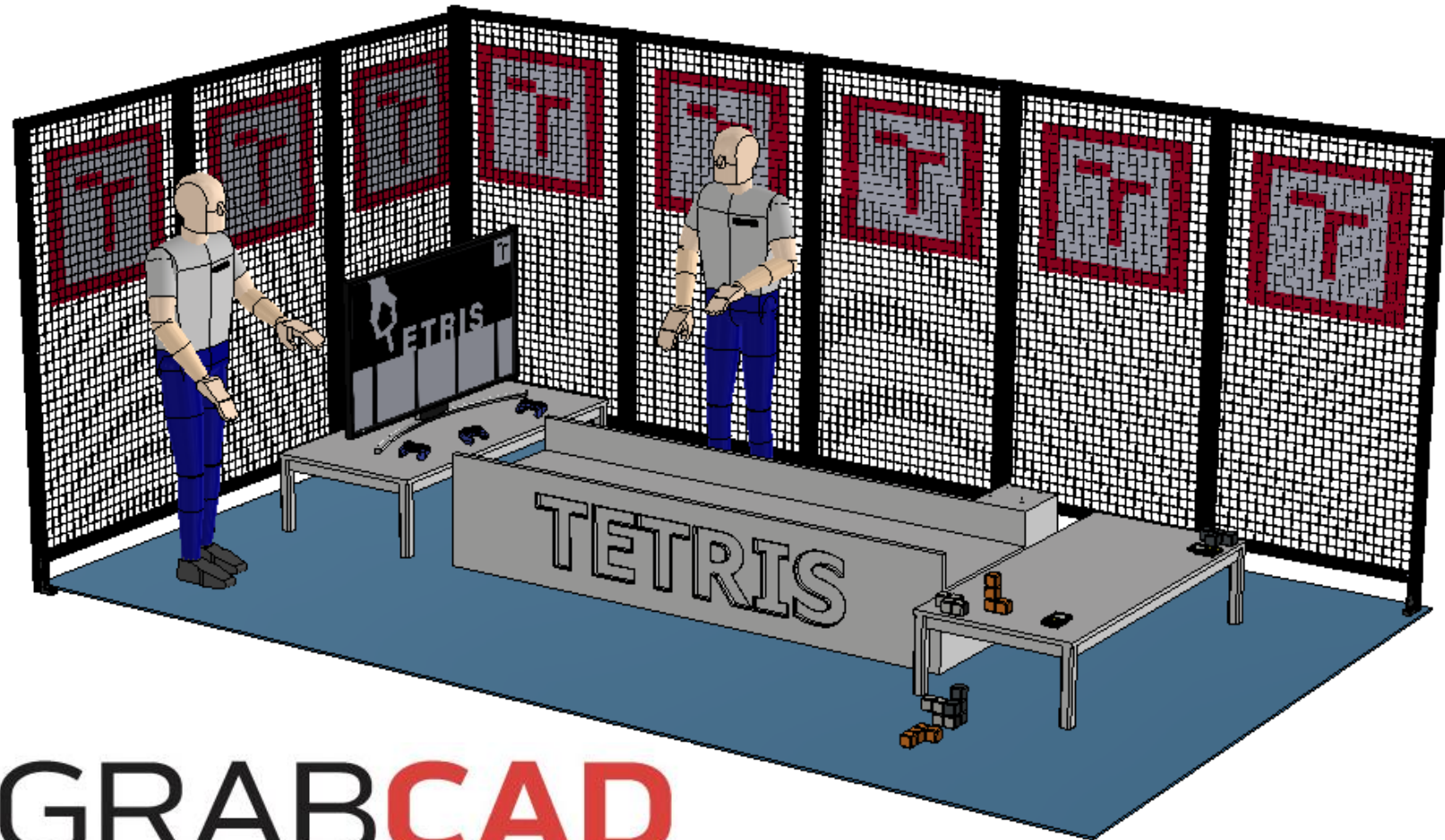
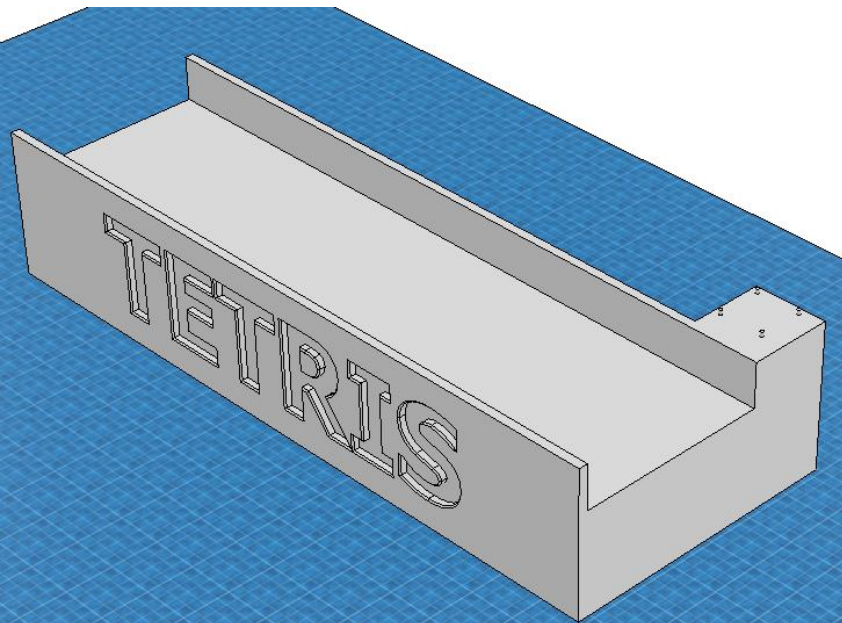
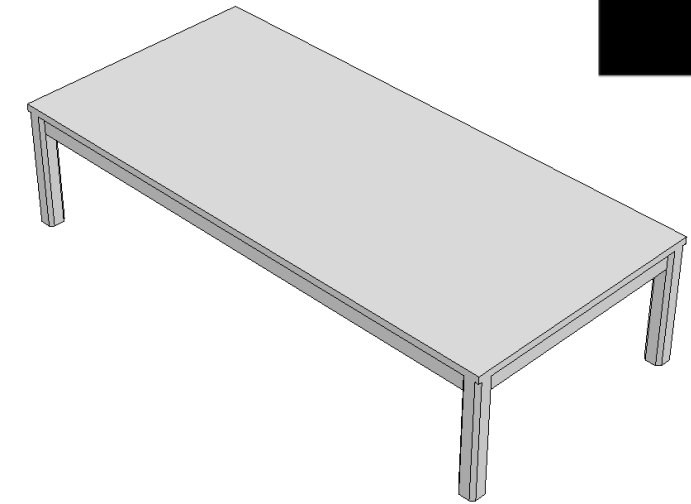
IRB 120 s přísavkou

Modely v Inventoru





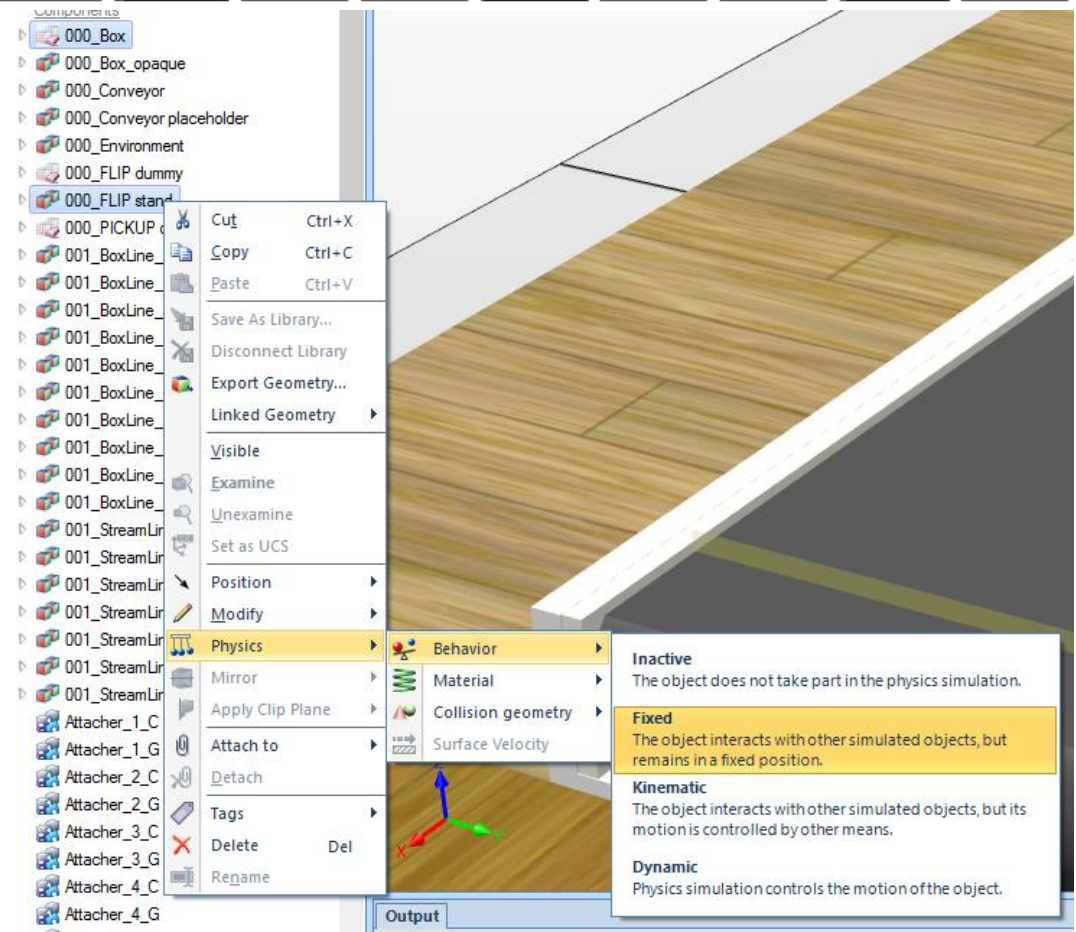
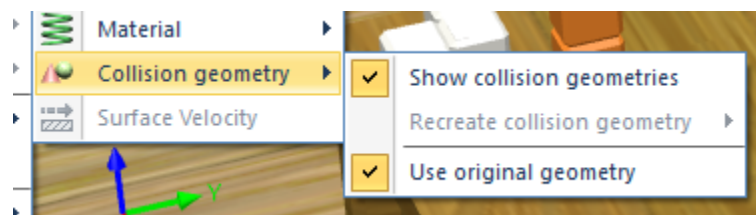
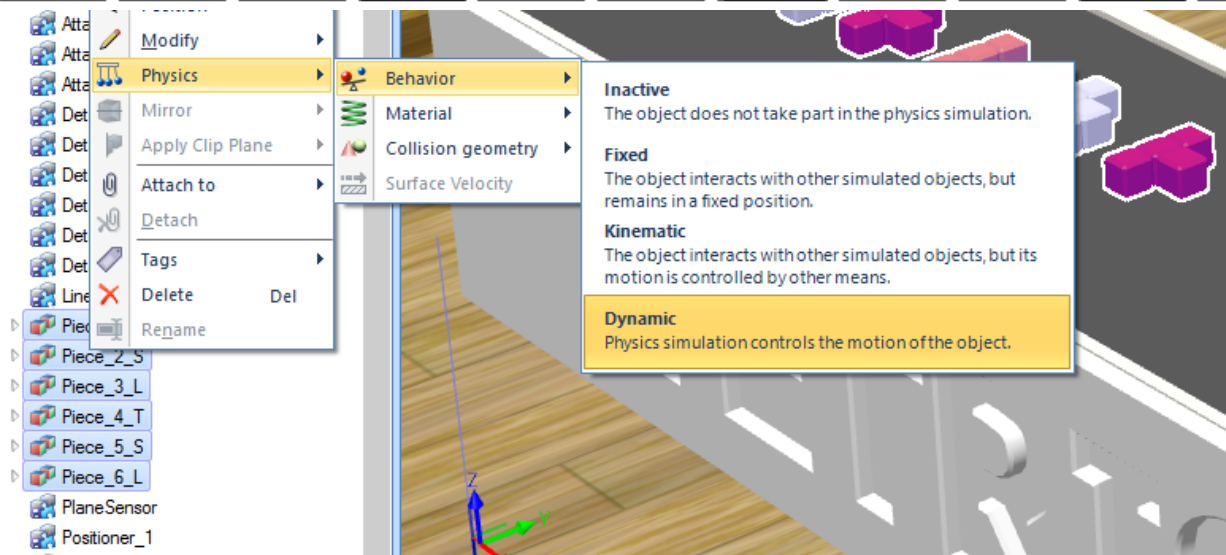
Modely v Inventoru



GRABCAD



Fyzika prostředí

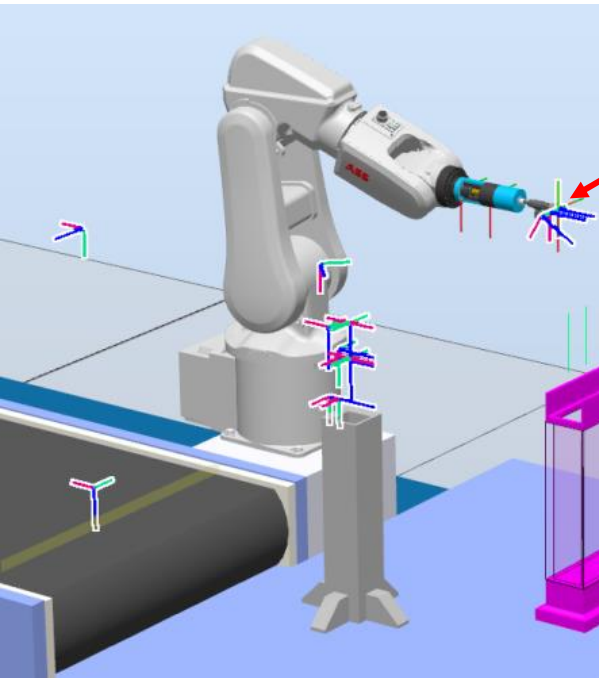




Logika prostředí

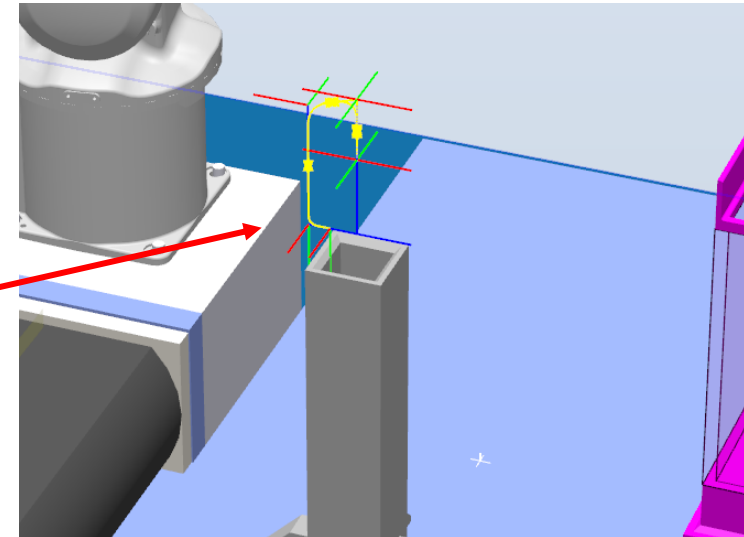


Trasy

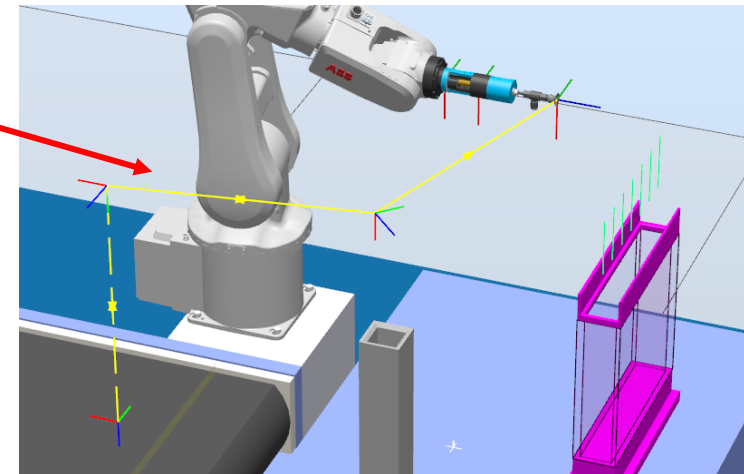


Targety – příklad

Trasa pro *flip* funkci



Trasa pro *pick up* funkci



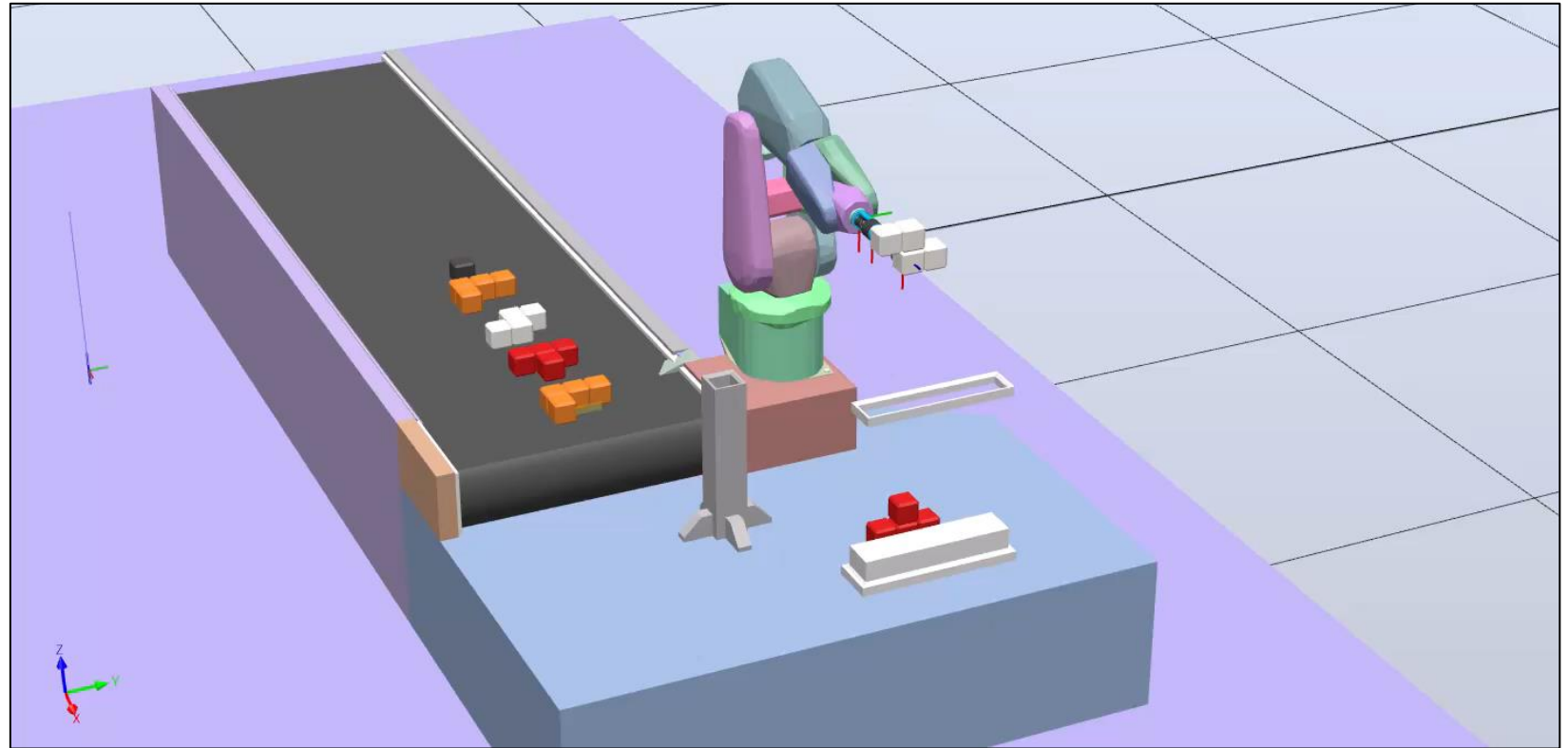
Trasy



PROC TestPath()

```
MoveL D_000,v1000,fine,smc_gripper\WObj:=wobj0;  
MoveL D_000,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL P_1,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL P_2,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveJ P_3,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveJ P_2,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL P_1,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL D_000,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_1,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_2,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_3,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveJ F_4,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_5,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_6,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_5,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_5R,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_6R,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_5,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_4,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_3,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_2,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_1,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_1R,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveJ D_000,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL D_000,v1000,fine,smc_gripper\WObj:=wobj0;  
MoveL D_000,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL D_000,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL D_090,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL D_180,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL D_270,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL HOME,v1000,z100,smc_gripper\WObj:=wobj0;
```

ENDPROC





Program





Proměnné, resety

```
VAR num CASE_NO := 0;  
VAR NUM rof := 0;  
VAR NUM rot := 0;  
VAR NUM offst_x := 0;  
VAR NUM offst_y := 0;  
VAR NUM offst_z := 0;  
VAR NUM moved := 0;  
VAR NUM WHERE_AM_I := 0;  
VAR NUM md := 0;
```

```
PROC ResetEnv()  
  
    ! Zastavit dopravník  
    SetDO DO_CONV_MOVE, 0;  
  
    ! Odpojit kostky  
    PulseDO DO_DET_1;  
    PulseDO DO_DET_2;  
    PulseDO DO_DET_3;  
    PulseDO DO_DET_4;  
    PulseDO DO_DET_5;  
    PulseDO DO_DET_6;  
  
    ! Vrátit kostky na výchozí pozici  
    PulseDO DO_POS_1;  
    PulseDO DO_POS_2;  
    PulseDO DO_POS_3;  
    PulseDO DO_POS_4;  
    PulseDO DO_POS_5;  
    PulseDO DO_POS_6;  
    PulseDO DO_POS_7;  
  
    ! Vrátit robota na výchozí pozici  
    MoveJ HOME,v200,fine,smc_gripper\WObj:=wobj0;  
  
ENDPROC
```

```
PROC ResetParams()  
  
    !md := 0;  
    rof := 0;  
    rot := 0;  
    offst_x := 0;  
    offst_y := 0;  
    offst_z := 0;  
    SetDO DO_FLIPPED, 0;  
    moved := 0;  
  
ENDPROC
```

Startovní funkce

```
PROC Start()
```

```
! Napojit kostky na dopravnik
```

```
PulseDO DO_ATT_1_C;
```

```
PulseDO DO_ATT_2_C;
```

```
PulseDO DO_ATT_3_C;
```

```
PulseDO DO_ATT_4_C;
```

```
PulseDO DO_ATT_5_C;
```

```
PulseDO DO_ATT_6_C;
```

```
! Rozpohybovat dopravnik
```

```
SetDO DO_CONV_MOVE, 1;
```

```
ENDPROC
```

```
PROC main()
```

```
TEST CASE_NO
```

```
CASE 0:
```

```
ResetEnv;
```

```
Start;
```

```
WaitDI DI_INPOS, 1;
```

```
CASE_NO := 1;
```

```
CASE 1:
```

```
Pickup_1;
```

```
rot_or_flip_1;
```

```
MoveJ D_000,v1000,z100,smc_gripper\Wobj:=wobj0;
```

```
CASE_NO := 2;
```

```
CASE 2:
```

```
Pickup_2;
```

```
rot_or_flip_2;
```

```
MoveJ D_000,v1000,z100,smc_gripper\Wobj:=wobj0;
```

```
CASE_NO := 3;
```

```
CASE 3:
```

```
Pickup_3;
```

```
rot_or_flip_3;
```

```
MoveJ D_000,v1000,z100,smc_gripper\Wobj:=wobj0;
```

```
CASE_NO := 4;
```

```
CASE 4:
```

```
Pickup_4;
```

```
rot_or_flip_4;
```

```
MoveJ D_000,v1000,z100,smc_gripper\Wobj:=wobj0;
```

```
CASE_NO := 5;
```

```
CASE 5:
```

```
Pickup_5;
```


Funkce pickup

```
PROC Pickup_1()  
  MoveJ P_1,v1000,fine,smc_gripper\WObj:=wobj0;  
  MoveL P_2,v1000,fine,smc_gripper\WObj:=wobj0;  
  MoveJ P_3,v1000,fine,smc_gripper\WObj:=wobj0;  
  PulseDO DO_DET_1;  
  WaitTime 0.25;  
  PulseDO DO_ATT_1_G;  
  MoveJ P_2,v1000,z200,smc_gripper\WObj:=wobj0;  
  MoveL P_1,v1000,fine,smc_gripper\WObj:=wobj0;  
  MoveL D_000,v1000,fine,smc_gripper\WObj:=wobj0;  
ENDPROC
```

```
PROC main()
```

```
  TEST CASE_NO
```

```
  CASE 0:
```

```
    ResetEnv;  
    Start;  
    WaitDI DI_INPOS, 1;  
    CASE_NO := 1;
```

```
  CASE 1:
```

```
    Pickup_1;  
    rot_or_flip_1;  
    MoveJ D_000,v1000,z100,smc_gripper\WObj:=wobj0;  
    CASE_NO := 2;
```

```
  CASE 2:
```

```
    Pickup_2;  
    rot_or_flip_2;  
    MoveJ D_000,v1000,z100,smc_gripper\WObj:=wobj0;  
    CASE_NO := 3;
```

```
  CASE 3:
```

```
    Pickup_3;  
    rot_or_flip_3;  
    MoveJ D_000,v1000,z100,smc_gripper\WObj:=wobj0;  
    CASE_NO := 4;
```

```
  CASE 4:
```

```
    Pickup_4;  
    rot_or_flip_4;  
    MoveJ D_000,v1000,z100,smc_gripper\WObj:=wobj0;  
    CASE_NO := 5;
```

```
  CASE 5:
```

```
    Pickup_5;
```

Funkce rot_or_flip

```
PROC rot_or_flip_1()

ResetParams;
WHILE rof = 0 DO

    WaitUntil DI_ROT90 = 1 OR DI_FLIP = 1 OR DI_MOVE_RIGHT = 1 OR DI_MOVE_LEFT = 1 OR DI_DROP = 1 OR DI_GET_IN_POS = 1;

    IF DI_ROT90 = 1 THEN
        rotate_90;
        moved := 1;
        WaitTime 1;

    ELSEIF DI_GET_IN_POS = 1 THEN
        Get_in_position;

    ELSEIF DI_FLIP = 1 THEN
        between_rotate_or_move_and_flip;
        Flip_1;
        InvertDO DO_Flipped;
        MoveJ D_000,v1000,z100,smc_gripper\WObj:=wobj0;
        WaitTime 1;

        !vynuluje parametry rotace a movementu, polohovani nastava od poc. parametru po flipu
        WHERE_AM_I := 0;
        offst_x := 0;
        offst_y := 0;
        offst_z := 0;
        rot := 0;
        rof := 0;

    ELSEIF DI_MOVE_RIGHT = 1 THEN
        move_right;
        moved := 1;
        WaitTime 1;

    ELSEIF DI_MOVE_LEFT = 1 THEN
        move_left;
        moved := 1;
        WaitTime 1;

    ELSEIF DI_DROP = 1 THEN
        WaitTime 0.25;
        PulseDO DO_DET_1;
        rof := 1;

    ENDIF

ENDWHILE

ENDPROC
```

```
PROC main()

TEST CASE_NO

CASE 0:
    ResetEnv;
    Start;
    WaitDI DI_INPOS, 1;
    CASE_NO := 1;

CASE 1:
    Pickup_1;
    rot_or_flip_1;
    MoveJ D_000,v1000,z100,smc_gripper\WObj:=wobj0;
    CASE_NO := 2;

CASE 2:
    Pickup_2;
    rot_or_flip_2;
    MoveJ D_000,v1000,z100,smc_gripper\WObj:=wobj0;
    CASE_NO := 3;

CASE 3:
    Pickup_3;
    rot_or_flip_3;
    MoveJ D_000,v1000,z100,smc_gripper\WObj:=wobj0;
    CASE_NO := 4;

CASE 4:
    Pickup_4;
    rot_or_flip_4;
    MoveJ D_000,v1000,z100,smc_gripper\WObj:=wobj0;
    CASE_NO := 5;

CASE 5:
    Pickup_5;
```

Funkce get_in_position

```
PROC Get_in_position()
```

```
TEST WHERE_AM_I
```

```
CASE 0: ! pozice NAHORE
```

```
offst_x := 0;  
offst_z := 0;
```

```
TEST rot
```

```
CASE 000:
```

```
MoveJ Offs(D_000, offst_x, offst_y, offst_z), v1000, fine, smc_gripper\WObj:=wobj0;  
rof := 0;
```

```
CASE 090:
```

```
MoveJ Offs(D_090, offst_x, offst_y, offst_z), v1000, fine, smc_gripper\WObj:=wobj0;  
rof := 0;
```

```
CASE 180:
```

```
MoveJ Offs(D_180, offst_x, offst_y, offst_z), v1000, fine, smc_gripper\WObj:=wobj0;  
rof := 0;
```

```
CASE 270:
```

```
MoveJ Offs(D_270, offst_x, offst_y, offst_z), v1000, fine, smc_gripper\WObj:=wobj0;  
rof := 0;
```

```
ENDTEST
```

```
WaitTime 1;
```

```
WHERE_AM_I := 1;
```

```
CASE 1: ! pozice DOLE
```

```
offst_x := 119.538;  
offst_z := -160.5;
```

```
TEST rot
```

```
CASE 000:
```

```
MoveJ Offs(D_000, offst_x, offst_y, offst_z), v1000, fine, smc_gripper\WObj:=wobj0;  
rof := 0;
```

```
CASE 090:
```

```
MoveJ Offs(D_090, offst_x, offst_y, offst_z), v1000, fine, smc_gripper\WObj:=wobj0;  
rof := 0;
```

```
CASE 180:
```

```
MoveJ Offs(D_180, offst_x, offst_y, offst_z), v1000, fine, smc_gripper\WObj:=wobj0;  
rof := 0;
```

```
CASE 270:
```

```
MoveJ Offs(D_270, offst_x, offst_y, offst_z), v1000, fine, smc_gripper\WObj:=wobj0;  
rof := 0;
```

```
ENDTEST
```

```
WaitTime 1;
```

```
WHERE_AM_I := 0;
```

```
ENDTEST
```

```
ENDPROC
```



Horizontální posun

Obdobně pro levou stranu

```
PROC move_right()

TEST rot
CASE 000:
  if_move_right_horiz;
  MoveJ Offs(D_000, offst_x, offst_y, offst_z), v200, fine, smc_gripper\WObj:=wobj0;
  rof := 0;

CASE 090:
  if_move_right_combo;
  MoveJ Offs(D_090, offst_x, offst_y, offst_z), v200, fine, smc_gripper\WObj:=wobj0;
  rof := 0;

CASE 180:
  if_move_right_horiz;
  MoveJ Offs(D_180, offst_x, offst_y, offst_z), v200, fine, smc_gripper\WObj:=wobj0;
  rof := 0;

CASE 270:
  if_move_right_comboII;
  MoveJ Offs(D_270, offst_x, offst_y, offst_z), v200, fine, smc_gripper\WObj:=wobj0;
  rof := 0;

ENDTEST

ENDPROC
```



Rotace – 4 polohy efektoru

```
PROC rotate_90()
```

```
TEST rot
```

```
CASE 000:
```

```
MoveJ Offs(D_090, offst_x, offst_y, offst_z),v1000,fine,smc_gripper\WObj:=wobj0;  
WaitTime 0.5;  
rof := 0;  
rot := 090;
```

```
CASE 090:
```

```
MoveJ Offs(D_180, offst_x, offst_y, offst_z),v1000,fine,smc_gripper\WObj:=wobj0;  
WaitTime 0.5;  
rof := 0;  
rot := 180;
```

```
CASE 180:
```

```
MoveJ Offs(D_270, offst_x, offst_y, offst_z),v1000,fine,smc_gripper\WObj:=wobj0;  
WaitTime 0.5;  
rof := 0;  
rot := 270;
```

```
CASE 270:
```

```
MoveJ Offs(D_000, offst_x, offst_y, offst_z),v1000,fine,smc_gripper\WObj:=wobj0;  
rof := 0;  
rot := 000;
```

```
ENDTEST
```

```
ENDPROC
```

```
PROC Flip_1()
```

```
MoveL D_000,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveJ Offs(F_1, 0, 0, 200),v1000,fine,smc_gripper\WObj:=wobj0;  
!MoveL F_1,v1000,fine,smc_gripper\WObj:=wobj0;  
MoveL F_2,v1000,fine,smc_gripper\WObj:=wobj0;  
PulseDO DO_DET_1;  
WaitTime 0.25;  
MoveL F_3,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_4,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_5,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_6,v1000,fine,smc_gripper\WObj:=wobj0;  
PulseDO DO_ATT_1_G;  
WaitTime 0.25;  
MoveL F_5,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_5R,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_6R,v1000,fine,smc_gripper\WObj:=wobj0;  
PulseDO DO_DET_1;  
WaitTime 0.25;  
MoveL F_5,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_4,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_3,v1000,z100,smc_gripper\WObj:=wobj0;  
MoveL F_2,v1000,fine,smc_gripper\WObj:=wobj0;  
PulseDO DO_ATT_1_G;  
WaitTime 0.25;  
MoveL Offs(F_2, 0, 0, 300),v1000,fine,smc_gripper\WObj:=wobj0;
```

```
!MoveL F_1,v1000,z100,smc_gripper\WObj:=wobj0;  
!MoveL F_1R,v1000,z100,smc_gripper\WObj:=wobj0;  
!MoveJ D_000,v1000,fine,smc_gripper\WObj:=wobj0;
```

```
ENDPROC
```



Paměť

```
! offst_y -----
PROC if_move_right_horiz()
  IF offst_y < 50 THEN
    offst_y := offst_y + 50;
  ELSEIF offst_y >= 50 THEN
    offst_y := offst_y + 0;
  ENDIF
ENDPROC

PROC if_move_right_vert()
  IF offst_y < 100 THEN
    offst_y := offst_y + 50;
  ELSEIF offst_y >= 100 THEN
    offst_y := offst_y + 0;
  ENDIF
ENDPROC

PROC if_move_left_horiz()
  IF offst_y > -100 THEN
    offst_y := offst_y - 50;
  ELSEIF offst_y <= -100 THEN
    offst_y := offst_y + 0;
  ENDIF
ENDPROC

PROC if_move_left_vert()
  IF offst_y > -150 THEN
    offst_y := offst_y - 50;
  ELSEIF offst_y <= -100 THEN
    offst_y := offst_y + 0;
  ENDIF
ENDPROC
! -----
```

```
! (not)flipped -----
PROC if_move_right_combo()
  IF DO_FLIPPED = 0 THEN
    if_move_right_vert;
  ELSEIF DO_FLIPPED = 1 THEN
    if_move_right_horiz;
  ENDIF
ENDPROC

PROC if_move_right_comboII()
  IF DO_FLIPPED = 0 THEN
    if_move_right_horiz;
  ELSEIF DO_FLIPPED = 1 THEN
    if_move_right_vert;
  ENDIF
ENDPROC

PROC if_move_left_combo()
  IF DO_FLIPPED = 0 THEN
    if_move_left_vert;
  ELSEIF DO_FLIPPED = 1 THEN
    if_move_left_horiz;
  ENDIF
ENDPROC

PROC if_move_left_comboII()
  IF DO_FLIPPED = 0 THEN
    if_move_left_horiz;
  ELSEIF DO_FLIPPED = 1 THEN
    if_move_left_vert;
  ENDIF
ENDPROC
! -----
```

flipování má vliv na polohu kostky vůči

rameni



Ošetření kolizí

```
! vyhybání se kolizím mezi funkcema II. kategorie (rotate, move) a flipováním  
PROC between_rotate_or_move_and_flip()  
  
    IF moved = 1 THEN  
        MoveJ Offs(D_000, 0, offst_y, 100), v1000,fine,smc_gripper\WObj:=wobj0;  
        MoveL D_000,v1000,z100,smc_gripper\WObj:=wobj0;  
  
        moved := 0;  
    ENDIF  
  
ENDPROC
```

Main funkce

```
CASE 0:
  ResetEnv;
  Start;
  WaitDI DI_INPOS, 1;
  CASE_NO := 1;

CASE 1:
  Pickup_1;
  rot_or_flip_1;
  MoveJ D_000,v1000,z100,smc_gripper\WObj:=wobj0;
  CASE_NO := 2;

CASE 2:
  Pickup_2;
  rot_or_flip_2;
  MoveJ D_000,v1000,z100,smc_gripper\WObj:=wobj0;
  CASE_NO := 3;

CASE 3:
  Pickup_3;
  rot_or_flip_3;
  MoveJ D_000,v1000,z100,smc_gripper\WObj:=wobj0;
  CASE_NO := 4;

CASE 4:
  Pickup_4;
  rot_or_flip_4;
  MoveJ D_000,v1000,z100,smc_gripper\WObj:=wobj0;
  CASE_NO := 5;

CASE 5:
  Pickup_5;
  rot_or_flip_5;
  MoveJ D_000,v1000,z100,smc_gripper\WObj:=wobj0;
  CASE_NO := 6;

CASE 6:
  Pickup_6;
  rot_or_flip_6;
  MoveJ D_000,v1000,z100,smc_gripper\WObj:=wobj0;
  CASE_NO := 7;

CASE 7:

  CASE_NO := 8;

CASE 8:

  CASE_NO := 9;

CASE 9:

  CASE_NO := 10;

CASE 10:
  WaitTime 5;
  ResetEnv;
  ResetParams;

  Break;
```

ENDTEST