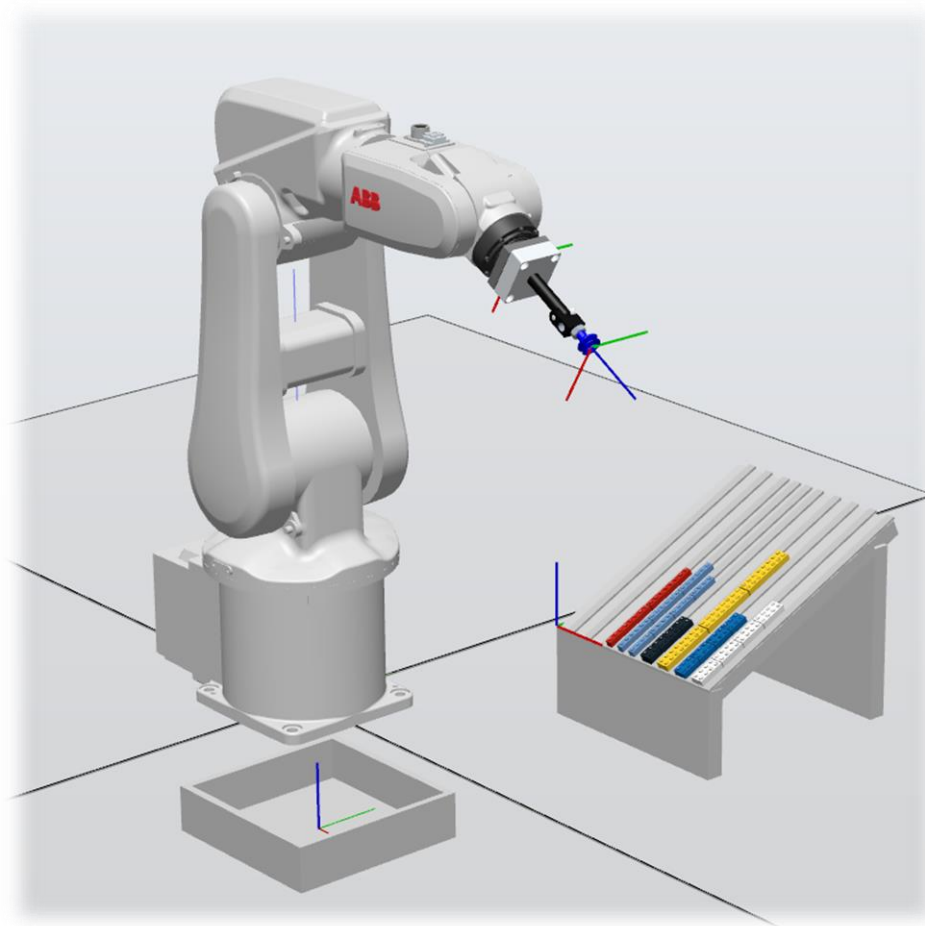


Prvi domaći zadatak

iz predmeta

Robotika i automatizacija



Katarina Petrović 2021/0068

Anja Stojanović 2021/0167

1. Postavka zadatka

Osnovna ideja domaćeg zadatka je pakovanje LEGO kockica u kutije. Tekstom zadatka definisane su igračke koje je moguće sklopiti i mogućnosti njihovog rasporeda po kutijama. U kutije je moguće spakovati 3 igračke tipa 1 ili 2 igračke tipa 2 i jednu igračku tipa 3 ili 2 igračke tipa 3.

Koristeći korisnički interfejs korisnik bira maksimalno 10 igračaka koje mu robot pakuje u kutije. Robot treba da ispoštuje ograničenja po pitanju brzine i zone, kao i da radi u skladu sa digitalnim (I/O) signalima koji se odnose na vakuumski griper i njegovu aktivnost. Dodatno, korisnik treba da prati da li se ispred robota u svakom trenutku nalaze odgovarajuće prazne kutije.

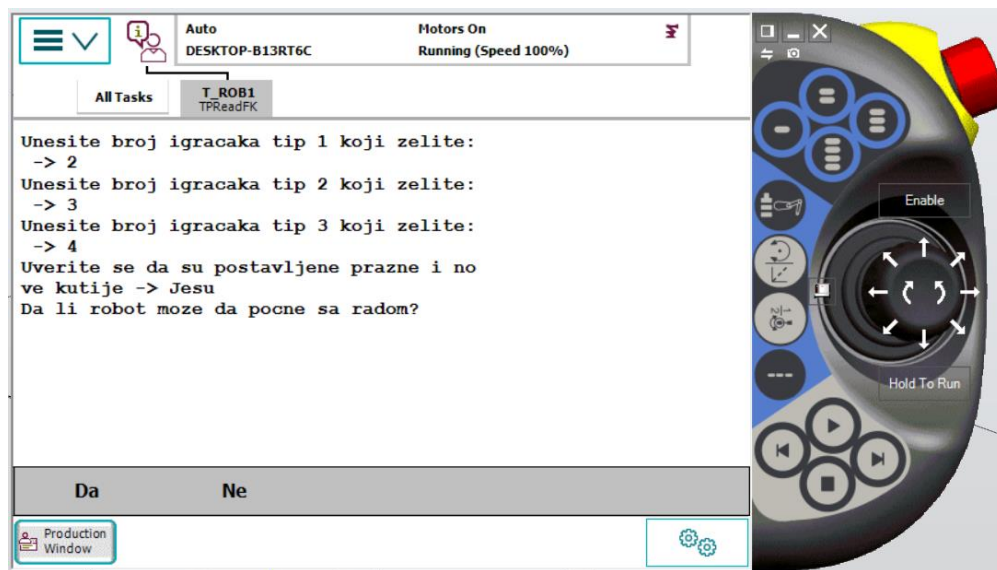
2. Iz ugla korisnika

Korisnički interfejs realizovan je pomoću **FlexPendant-a**.

U prvom koraku, korisnik bira broj igračaka svakog tipa koji želi. Program zatim proverava da li je ukupan broj igračaka veći od 10 i ukoliko jeste signalizira grešku porukom "Mozete spakovati maksimalno 10 igračaka".

U sledećem koraku, potrebno je da korisnik potvrdi da se ispred robota nalaze nove i prazne kutije, kao i da da dozvolu robotu da počne sa radom.

Na slici ispod je prikazan **izgled interfejsa** nakon ovih koraka:

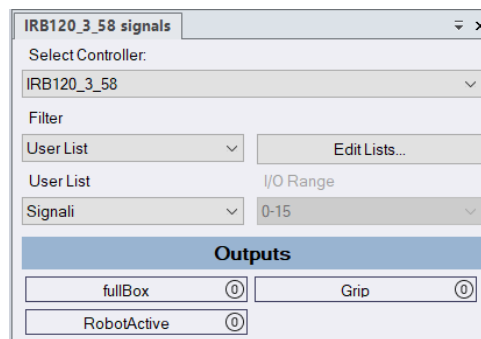


Nakon što robot dobije dozvolu da počne sa radom, on kreće sa pakovanjem kutija.

Korisniku je omogućeno praćenje **digitalnih signala** *fullBox*, *RobotActive* i *Grip* (što se može videti na slici ispod) koji mu daju bolji uvid u rad i aktivnost robota.

U toku rada, nakon što je spakovana kutija i robot došao u Home poziciju, na korisničkom interfejsu se pojavljuje poruka koja signalizira korisniku da zameni kutiju. Nakon što korisnik da signal da je to uradio, robot nastavlja pakovanje.

Kada su sve kutije spakovane, na FlexPendant-u se pojavljuje poruka "Robot je završio sa radom! Pokupite kutije".

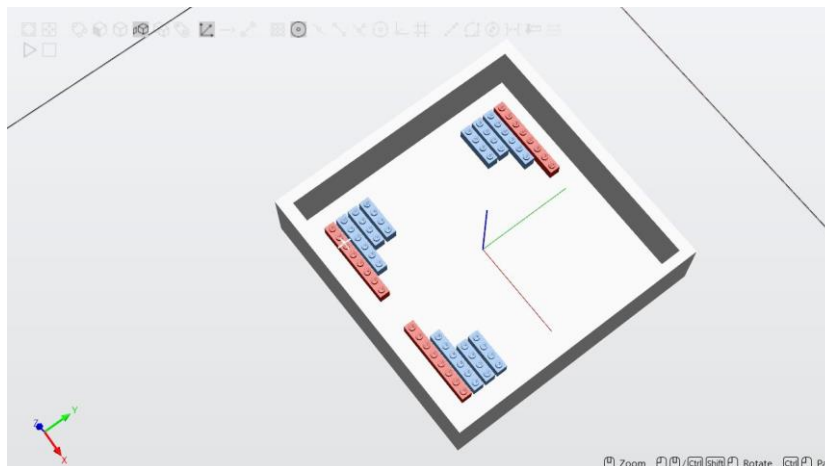


3. Proces izrade domaćeg zadatka

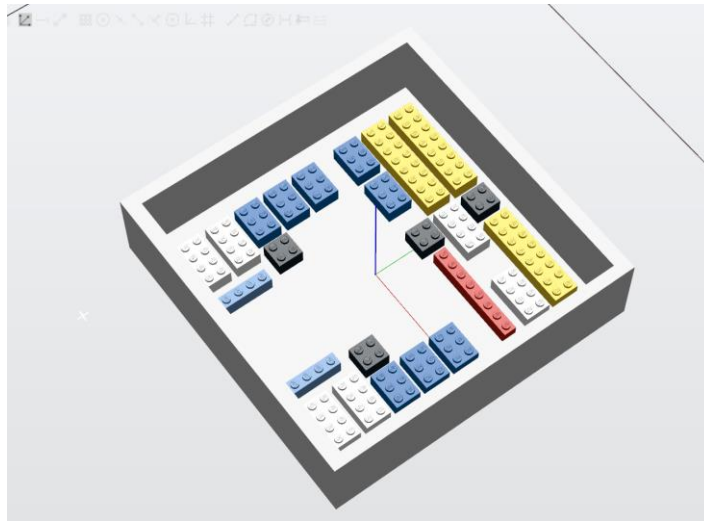
Tekstom zadatka date su 3 moguće kombinacije pakovanja kutija koje smo mi nazvale ***KutijaTip1***, ***KutijaTip2*** i ***KutijaTip3***. Za svaku od njih kreiran je zaseban Workobject sa odgovarajućim targetima.

Kako bi kreirale targete, potrebno je bilo kopirati kockice sa rampe i postaviti ih na optimalan način u kutiji. Pored toga, kreirale smo i pomoćne targete 70mm iznad željenih pozicija kockica u kutiji, kako bi robotu te pozicije bile pristupačne. U nastavku su prikazane kutije sa raspoređenim kockicama.

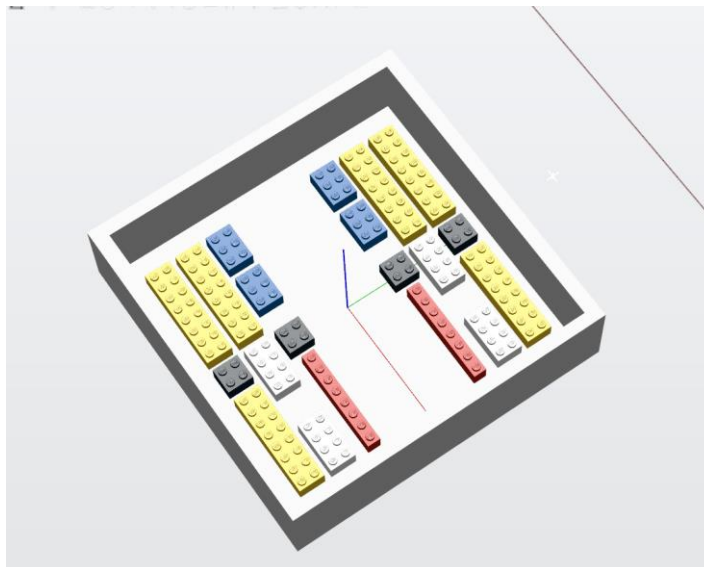
KutijaTip1 (3 igračke tipa 1):



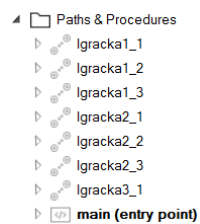
KutijaTip2 (2 igračke tipa 2 i 1 igračka tipa 3):



KutijaTip3 (2 igračke tipa 3):



Nakon targeta, bilo je potrebno kreirati **putanje** kojim će se robot kretati pri sklapanju svake igračke. Sve moguće putanje robota prikazane su u delu Paths & Procedures (što se vidi na slici).



Putanja Igracka2_3 odnosi se i na igračku tipa 3 iz kutije 3.

Putanje robota su napravljene pomoću **RAPID koda** u kome se nalaze i svi kreirani targeti.

Robot kreće iz Home pozicije, nakon čega odlazi do rampe i uzima odgovarajuću kockicu (tada se može videti aktivacija Grip signala), a zatim je nosi do kutije i spušta na odgovarajuću poziciju (tada se griper deaktivira). Tokom uzimanja i ostavljanja kockice, postavljeno je vreme čekanja neophodno da robot uzme ili ostavi kockicu.

Kada su sve putanje napravljene, realizovan je algoritam na osnovu kog robot radi u komunikaciji sa korisnikom.

U nastavku je prikazan **kod sa objašnjenjima**:

```
158
159 PROC main()
160     VAR num ok;
161     VAR num igrackeTip1;
162     VAR num igrackeTip2;
163     VAR num igrackeTip3;
164     VAR num pomocnaTip1;
165     VAR num pomocnaTip2;
166     VAR num pomocnaTip3;
167     VAR num kutijaPrazna;
168     VAR num dozvola;
169
170     TReadNum igrackeTip1, "Unesite broj igračaka tip 1 koji želite:";
171     TReadNum igrackeTip2, "Unesite broj igračaka tip 2 koji želite:";
172     TReadNum igrackeTip3, "Unesite broj igračaka tip 3 koji želite:";
173
174     IF (igrackeTip1 + igrackeTip2 + igrackeTip3) > 10 THEN
175         TWrite "Mozete spakovati maksimalno 10 igračaka";
176         RETURN;
177     ENDIF
178
179     TReadFK kutijaPrazna, "Uverite se da su postavljene prazne i nove kutije", "Jesu", "", "", "", "";
180     SetDO fullBox, 0;
181
182     TReadFK dozvola, "Da li robot može da počne sa radom?", "Da", "Ne", "", "", "";
183
184     IF dozvola <> 1 THEN
185         RETURN;
186     ENDIF
187     SetDO RobotActive, 1;
188
189     pomocnaTip1 := igrackeTip1;
190     pomocnaTip2 := igrackeTip2;
191     pomocnaTip3 := igrackeTip3;
```

Na početku smo definisale promenljive i uspostavile komunikaciju sa korisnikom kako bi nam dao željeni broj igračaka, neophodne dozvole i signale da bi robot izvršavao glavni deo programa. Napravile smo pomoćne promenljive koje su u nastavku bile potrebne radi znanja o tome koliko je u kom trenutku igračaka spakovano.

Ideja za pakovanje je sledeća: Pošto igračke tip 1 ne kombinujemo sa drugim tipovima igračaka, na početku pakujemo njih sve dok ih ima. Zatim, da bi bilo što više potpuno punih kutija, pakujemo drugu kombinaciju kutije kako bi iskoristile maksimalno igračaka tipa 2 i 3. Na kraju pakujemo preostale igračke tipa 3.

U toku procesa, robot u svakom potrebnom trenutku proverava da li su ispred njega postavljene nove i prazne kutije i manipuliše odgovarajućim digitalnim signalima.

```
!krećemo pakovanje kutija
!tip 1 pakujemo sve dok ih ima
WHILE pomocnaTip1 <> 0 DO

    IF fullBox = 1 THEN
        TReadFK kutijaPrazna, "Uverite se da je postavljena prazna kutija", "Jeste", "", "", "", "";
        setDO fullBox, 0;
        WaitTime 0.5;
        SetDO RobotActive, 1;
    ENDIF

    Igracka1_1;
    pomocnaTip1 := pomocnaTip1 - 1;
    IF pomocnaTip1 = 0 THEN
        setDO fullBox, 1;
        SetDO RobotActive, 0;
        GOTO BREAK_POINT1;
    ENDIF

    Igracka1_2;
    pomocnaTip1 := pomocnaTip1 - 1;
    IF pomocnaTip1 = 0 THEN
        setDO fullBox, 1;
        SetDO RobotActive, 0;
        GOTO BREAK_POINT1;
    ENDIF

    Igracka1_3;
    pomocnaTip1 := pomocnaTip1 - 1;
    setDO fullBox, 1;
    SetDO RobotActive, 0;

ENDWHILE
BREAK_POINT1:

WHILE pomocnaTip2 <> 0 DO
    IF fullBox = 1 THEN
        TReadFK kutijaPrazna, "Uverite se da je postavljena prazna kutija", "Jeste", "", "", "", "";
        setDO fullBox, 0;
        WaitTime 0.5;
        SetDO RobotActive, 1;
    ENDIF

    Igracka2_1;
    pomocnaTip2 := pomocnaTip2 - 1;

    IF pomocnaTip2 <> 0 THEN
        Igracka2_2;
        pomocnaTip2 := pomocnaTip2 - 1;
    ENDIF

    IF pomocnaTip3 <> 0 THEN
        Igracka2_3;
        pomocnaTip3 := pomocnaTip3 - 1;
    ENDIF

    SetDO fullBox, 1;
    SetDO RobotActive, 0;
ENDWHILE
```

```

WHILE pomocnaTip3 <> 0 DO

    IF fullBox = 1 THEN
        TReadFK kutijaPrazna, "Uverite se da je postavljena prazna kutija", "Jeste", "", "", "", "";
        setDO fullBox, 0;
        WaitTime 0.5;
        SetDO RobotActive, 1;
    ENDIF

    Igracka2_3;
    pomocnaTip3 := pomocnaTip3 - 1;
    IF pomocnaTip3 = 0 THEN
        setDO fullBox, 1;
        SetDO RobotActive, 0;
        GOTO BREAK_POINT2;
    ENDIF

    Igracka3_1;
    pomocnaTip3 := pomocnaTip3 - 1;
    setDO fullBox, 1;
    SetDO RobotActive, 0;

ENDWHILE
BREAK_POINT2:

TPWrite "Robot je zavrrio sa radom! Pokupite kutije";
SetDO RobotActive, 0;

```