
Alice
Loading...
version 3.4.0.0+build123

Program Alice za izradu 3D animacija

Alice 3.exe

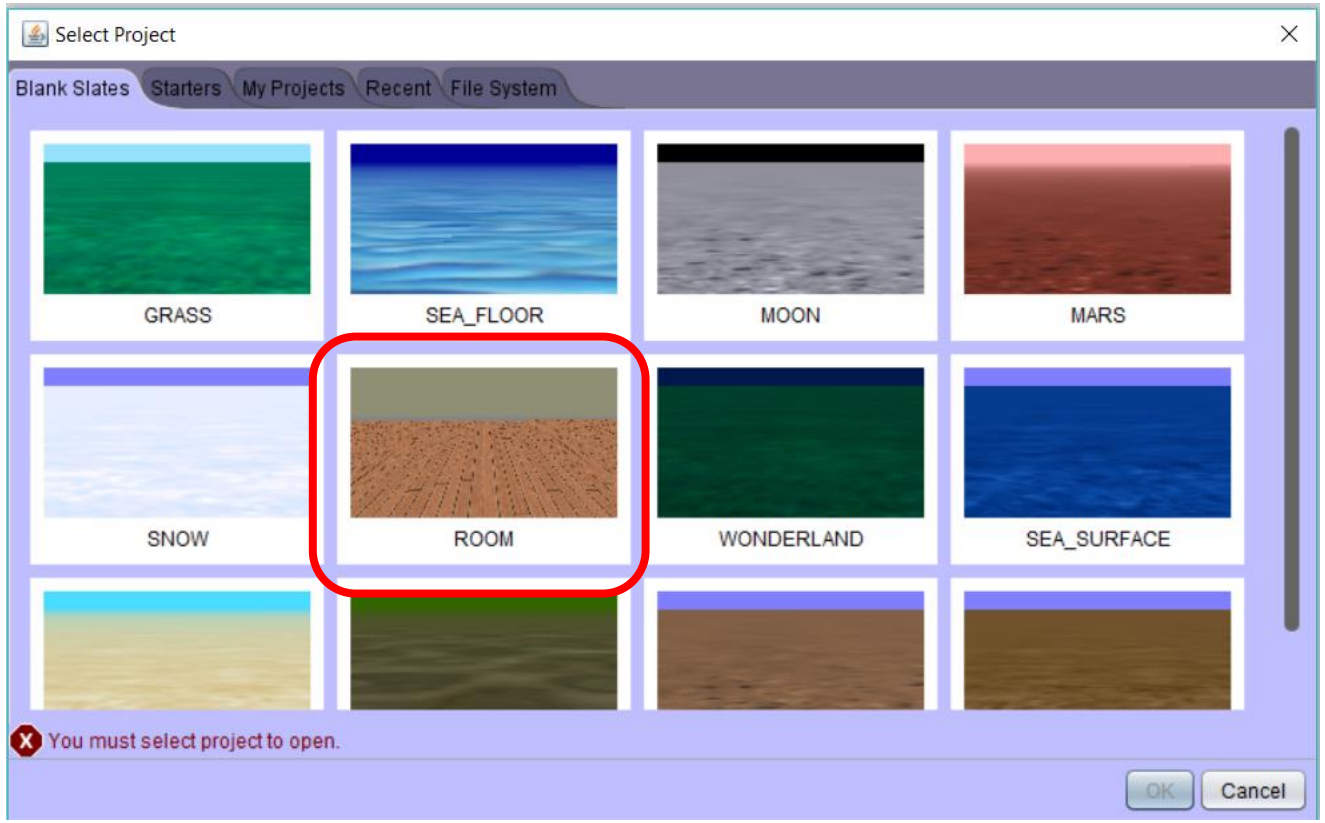
www.alice.org

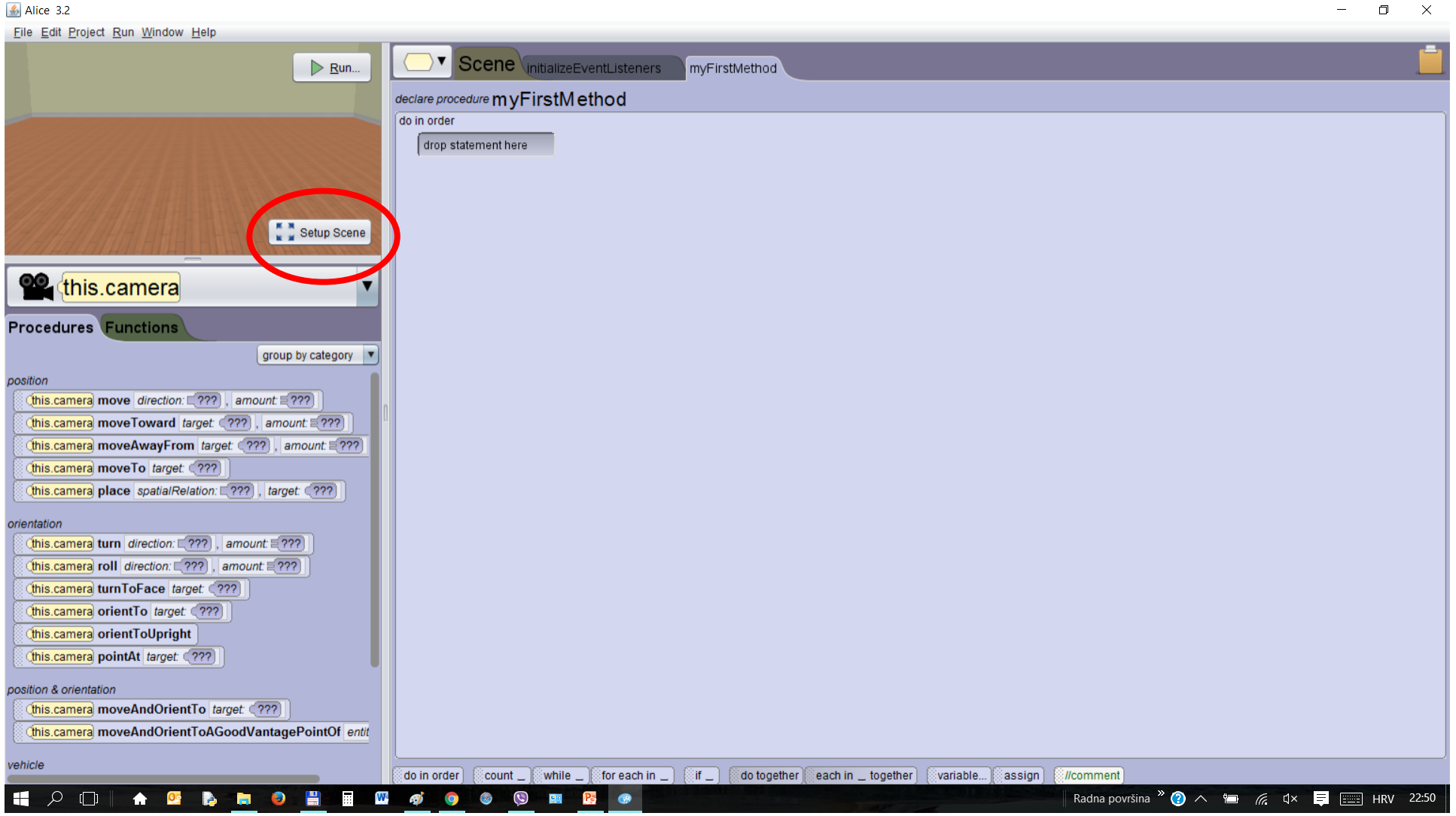
Pogledajmo primjer programa u Alice

SortiranjeHRV.a3p

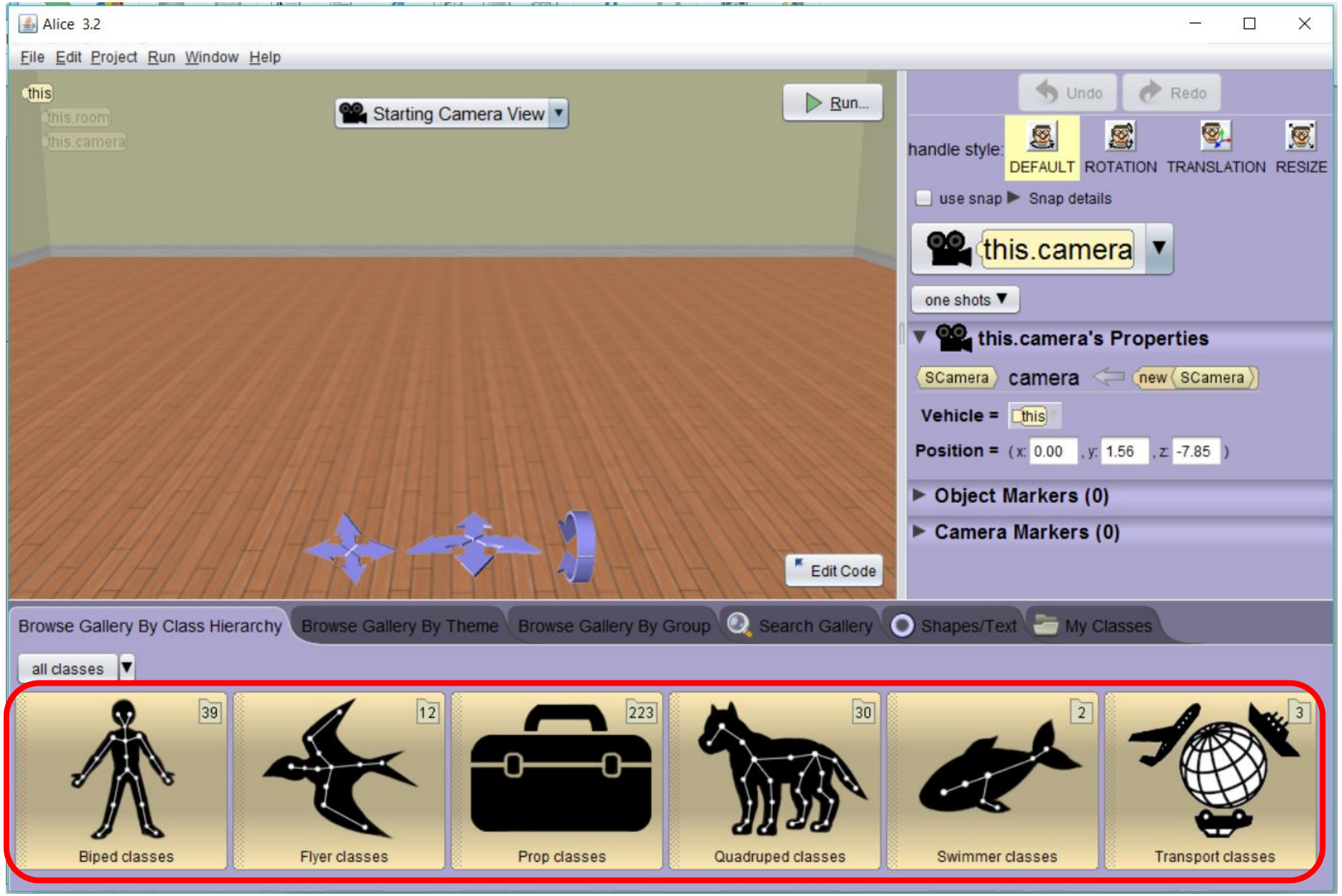
Namještanje i organizacija scene

Na početku rada odabiremo vrstu pozadine našeg projekta:




Alice 3.2
File Edit Project Run Window Help
Run...
Setup Scene
this.camera
Procedures
Functions
group by category
position
this.camera move direction: ??? , amount: ???
this.camera moveToward target: ??? , amount: ???
this.camera moveAwayFrom target: ??? , amount: ???
this.camera moveTo target: ???
this.camera place spatialRelation: ??? , target: ???
orientation
this.camera turn direction: ??? , amount: ???
this.camera roll direction: ??? , amount: ???
this.camera turnToFace target: ???
this.camera orientTo target: ???
this.camera orientToUpright
this.camera pointAt target: ???
position & orientation
this.camera moveAndOrientTo target: ???
this.camera moveAndOrientToAGoodVantagePointOf entit
vehicle
Scene
initializeEventListeners
myFirstMethod
declare procedure myFirstMethod
do in order
drop statement here
do in order
count _
while _
for each in _
if _
do together
each in _ together
variable...
assign
//comment
Radna površina
HRV
22:50

U scenu umećemo statične i dinamične objekte iz izbornika na dnu:

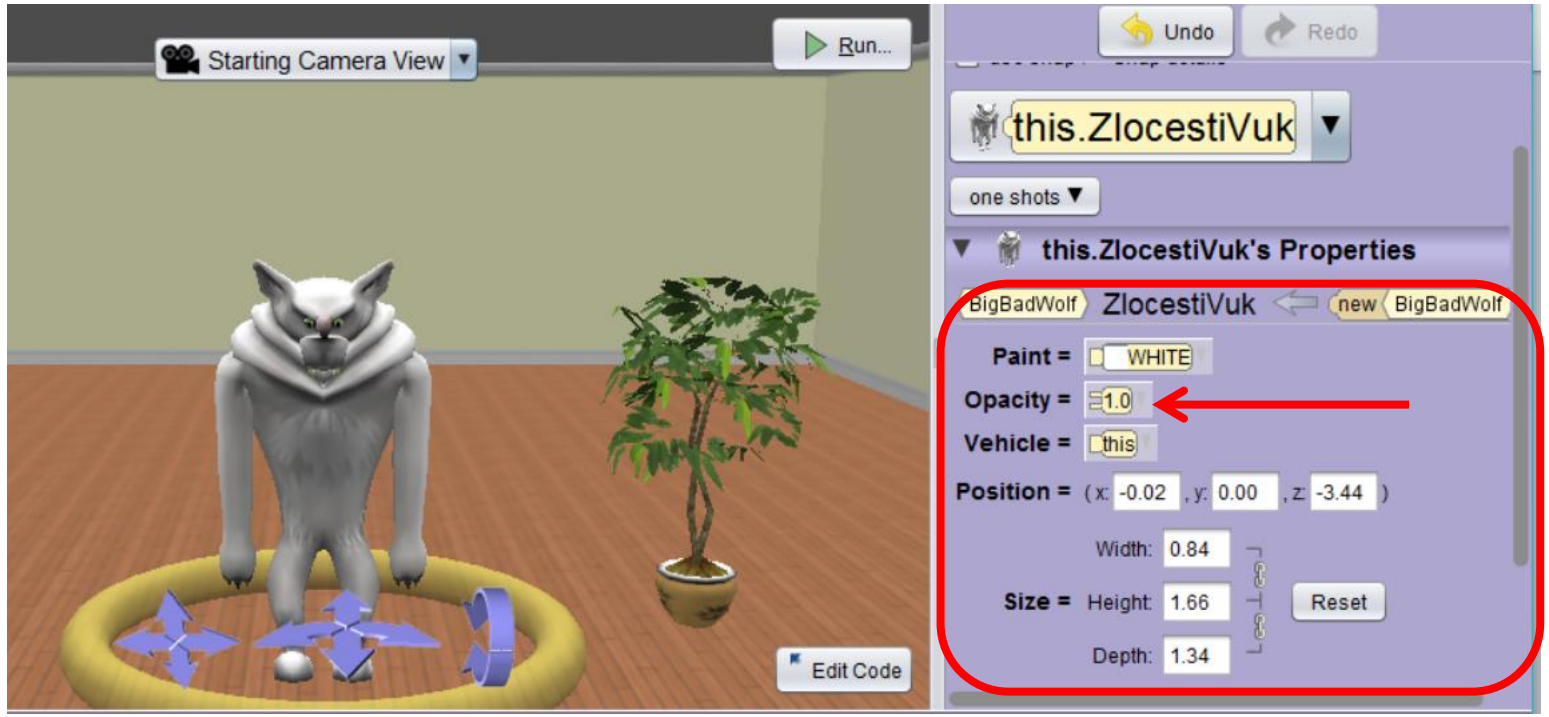


Umetanje objekata

Objekt možemo umetnuti na dva načina:

1. ***dvoklikom*** (dolazi u centar scene i dajemo mu JEDINSTVENO ime)
2. ***povlačenjem*** na mjesto na sceni (i davanjem JEDINSTVENOG imena)


Starting Camera View
Run...
Undo
Redo
handle style:
DEFAULT
ROTATION
TRANSLATION
RESIZ
this.Fikus
one shots
this.Fikus's Properties
FigTree
Fikus
new
FigTree
Paint =
WHITE
Opacity =
1.0
Vehicle =
this
Position =
(x: -1.70 , y: 0.00 , z: -2.99)
Width:
1.04
Edit Code

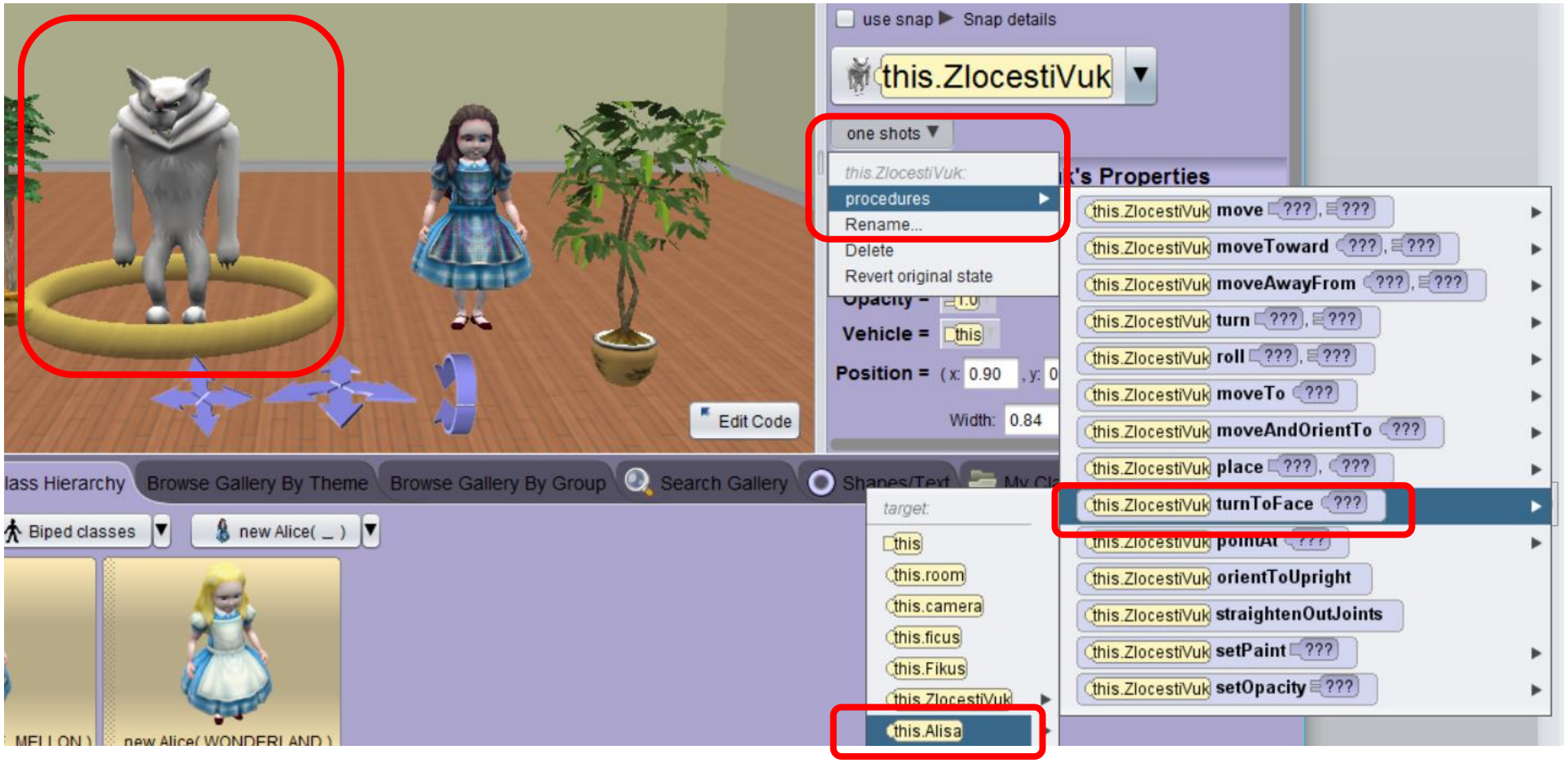


Spremajte često!

Pogledi kamere:

Vuk se okreće i gleda u Alisu:

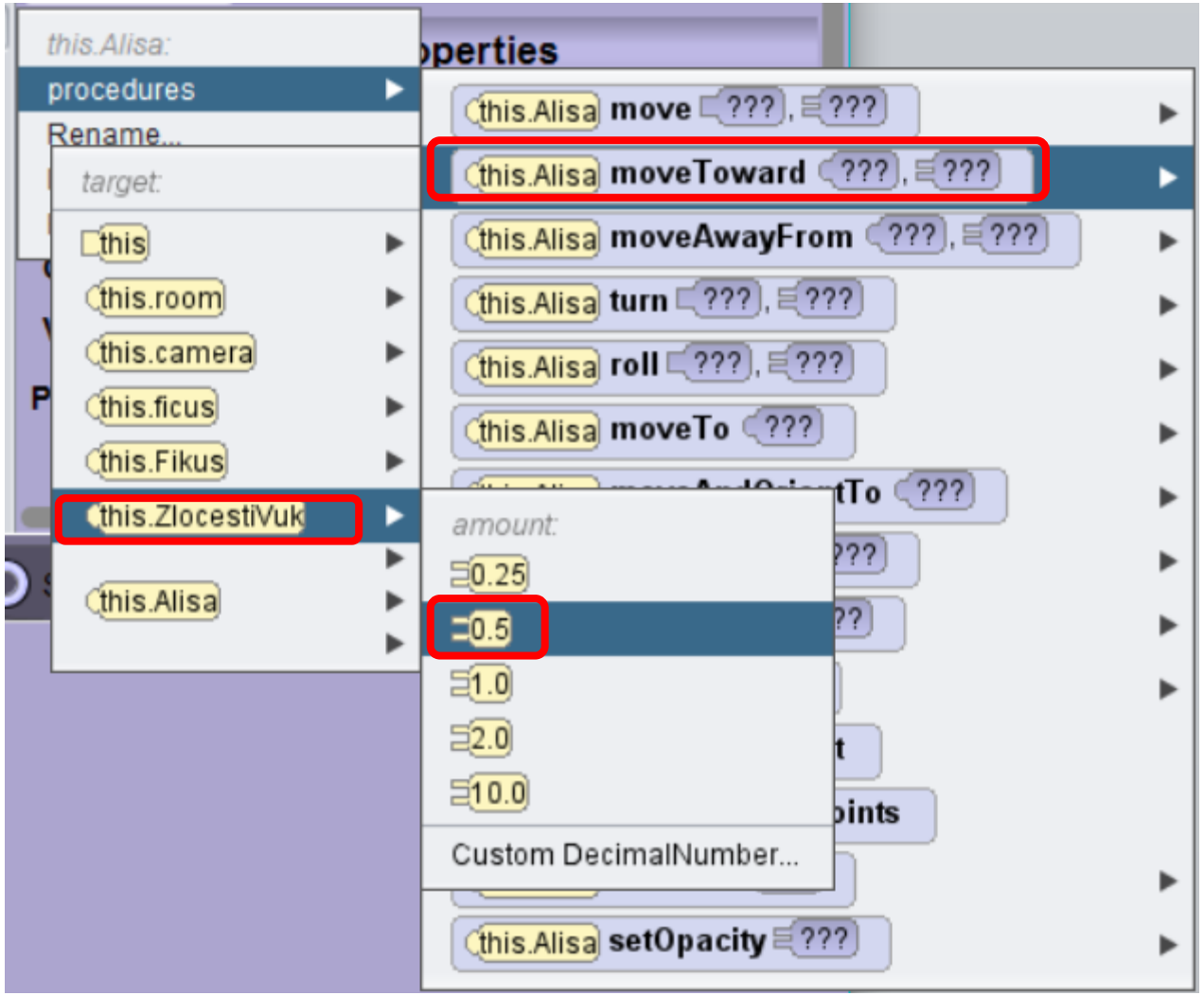
one shots procedure služe za namještanje scene




this
this.room
this.camera
this.ficus
this.Fikus
this.ZlocestiVuk
this.Alisa
Starting Camera View
Run...
Edit Code

Okrenite Alisu prema vuku:

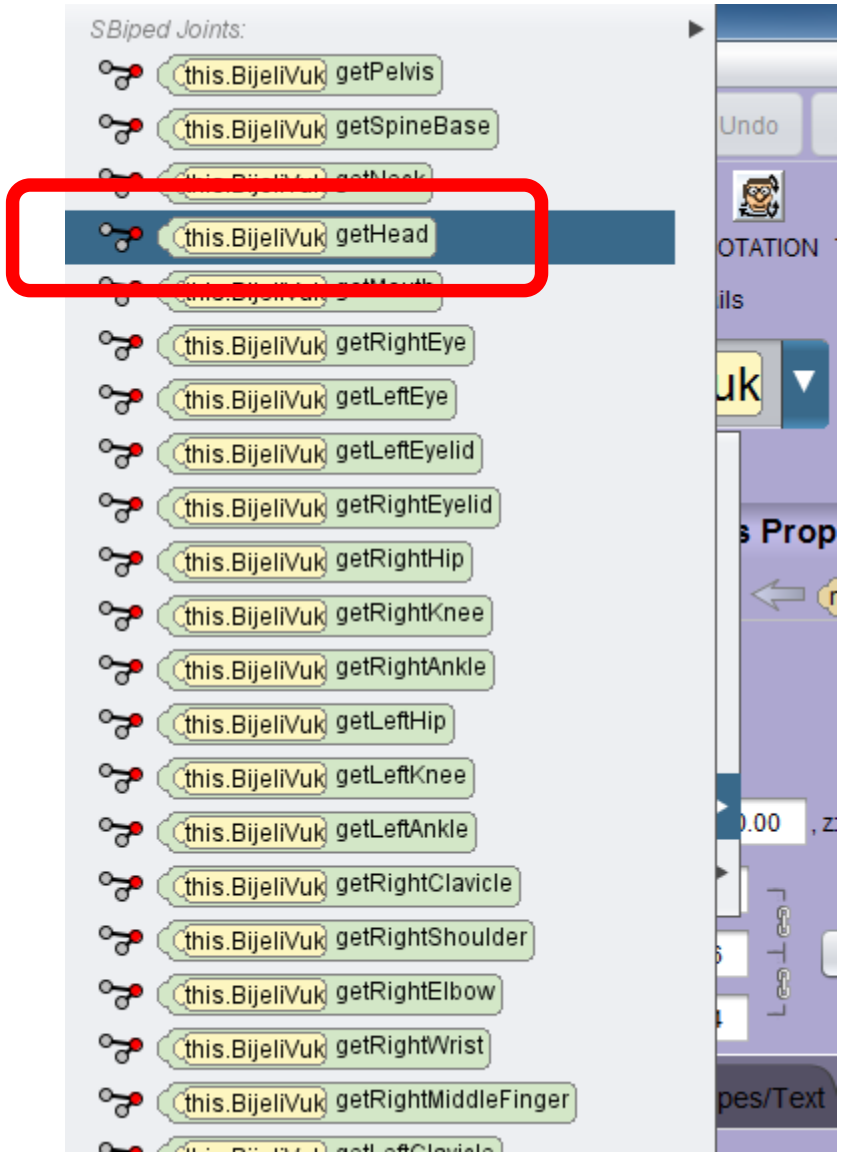
Približite Alisu pola koraka prema zločestom vuku:


this.Alisa:
procedures
Rename...
target:
this
this.room
this.camera
this.ficus
this.Fikus
this.ZlocestiVuk
this.Alisa
amount:
0.25
0.5
1.0
2.0
10.0
Custom DecimalNumber...
this.Alisa move ??? , ???
this.Alisa moveToward ??? , ???
this.Alisa moveAwayFrom ??? , ???
this.Alisa turn ??? , ???
this.Alisa roll ??? , ???
this.Alisa moveTo ???
this.Alisa setOpacity ???

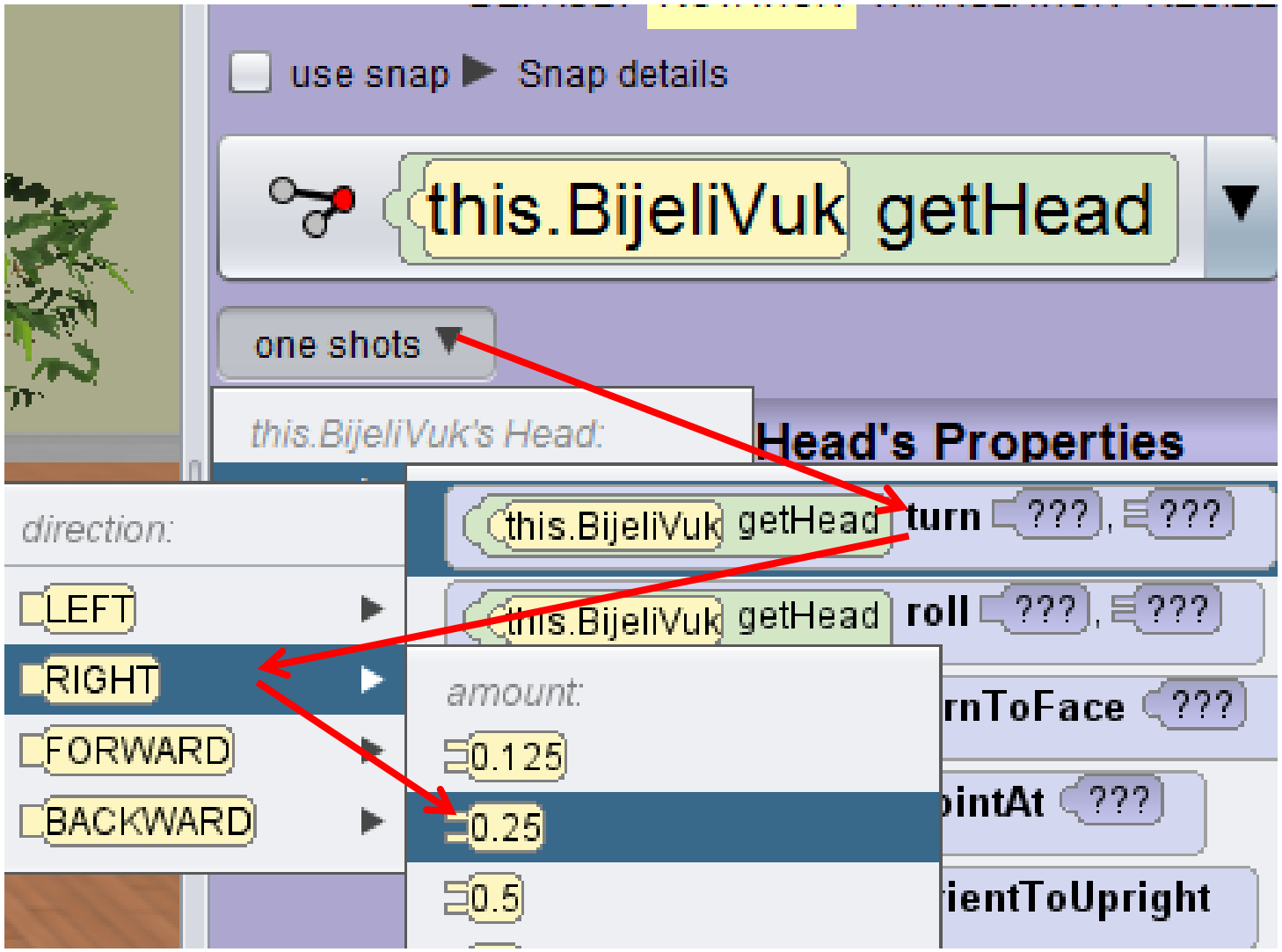
Rukovanje:

getRightShoulder - turn left 0.125

Vuk okreće glavu prema nama:


SBiped Joints:
this.BijeliVuk getPelvis
this.BijeliVuk getSpineBase
this.BijeliVuk getHead
this.BijeliVuk getRightEye
this.BijeliVuk getLeftEye
this.BijeliVuk getLeftEyelid
this.BijeliVuk getRightEyelid
this.BijeliVuk getRightHip
this.BijeliVuk getRightKnee
this.BijeliVuk getRightAnkle
this.BijeliVuk getLeftHip
this.BijeliVuk getLeftKnee
this.BijeliVuk getLeftAnkle
this.BijeliVuk getRightClavicle
this.BijeliVuk getRightShoulder
this.BijeliVuk getRightElbow
this.BijeliVuk getRightWrist
this.BijeliVuk getRightMiddleFinger
Undo
OTATION
ils
uk
s Prop
pes/Text


Alice 3.2 \\Laptop\c\A\Prezentacije 15-16\8\Alice\Alisa i vuk1.a3p
File Edit Project Run Window Help
Starting Camera View
Run...
this.BijeliVuk
Undo
Redo
handle style:
DEFAULT
ROTATION
TRANSLATION
RESIZE
use snap
Snap details
this.BijeliVuk getHead
one shots
this.BijeliVuk's Head:
Head's Properties
this.BijeliVuk getHead turn ???, ???
this.BijeliVuk getHead roll ???, ???
this.BijeliVuk getHead turnToFace ???
this.BijeliVuk getHead pointAt ???
this.BijeliVuk getHead orientToUpright
Edit Code
Browse Gallery By Class Hierarchy
Browse Gallery By Theme
Browse Gallery By Group
Search Gallery
Shapes/Text
My Classes
all classes
39
Biped classes
12
Flyer classes
223
Prop classes
30
Quadruped classes
2
Swimmer classes
Transport cla

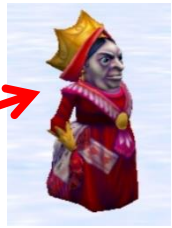

use snap
Snap details
this.BijeliVuk getHead
one shots
this.BijeliVuk's Head:
Head's Properties
direction:
LEFT
RIGHT
FORWARD
BACKWARD
this.BijeliVuk getHead turn ??? , ???
this.BijeliVuk getHead roll ??? , ???
rnToFace ???
ointAt ???
ientToUpright
amount:
0.125
0.25
0.5

Dijelovi tijela se mogu pomicati u početni položaj i pomoću miša i krugova za rotaciju:

move – up down left right forward backward – **objekt se POKREĆE**
turn – left right forward backward – **objekt ne mijenja mjesto (stajalište) – okreće se i naklanja**

CRVENO
turn
forward - backward

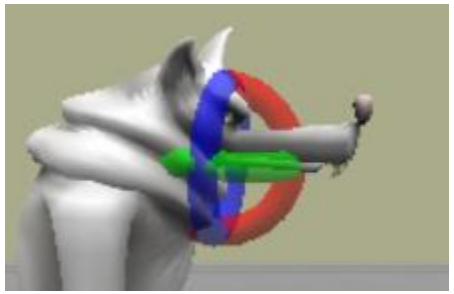
ZELENO
turn
left - right



PLAVO
roll
left - right

roll – left right – **objekt ne mijenja mjesto ni usmjerenje**

Vuku se mogu otvoriti usta:



turn forward 0.125

Zadaci:

1. Postavite dvije osobe na scenu (odaberite odjeću, izraz lica, boju i frizuru kose)
2. Okrenite ih jednog prema drugome
3. Mogućnost naredbi (move, turn, roll)
4. Odabir ramena (shoulder – dizanje ruke)
5. Odabir noge (hip – turn backward 0.125)
6. Od svake vrste po dva objekta – na scenu

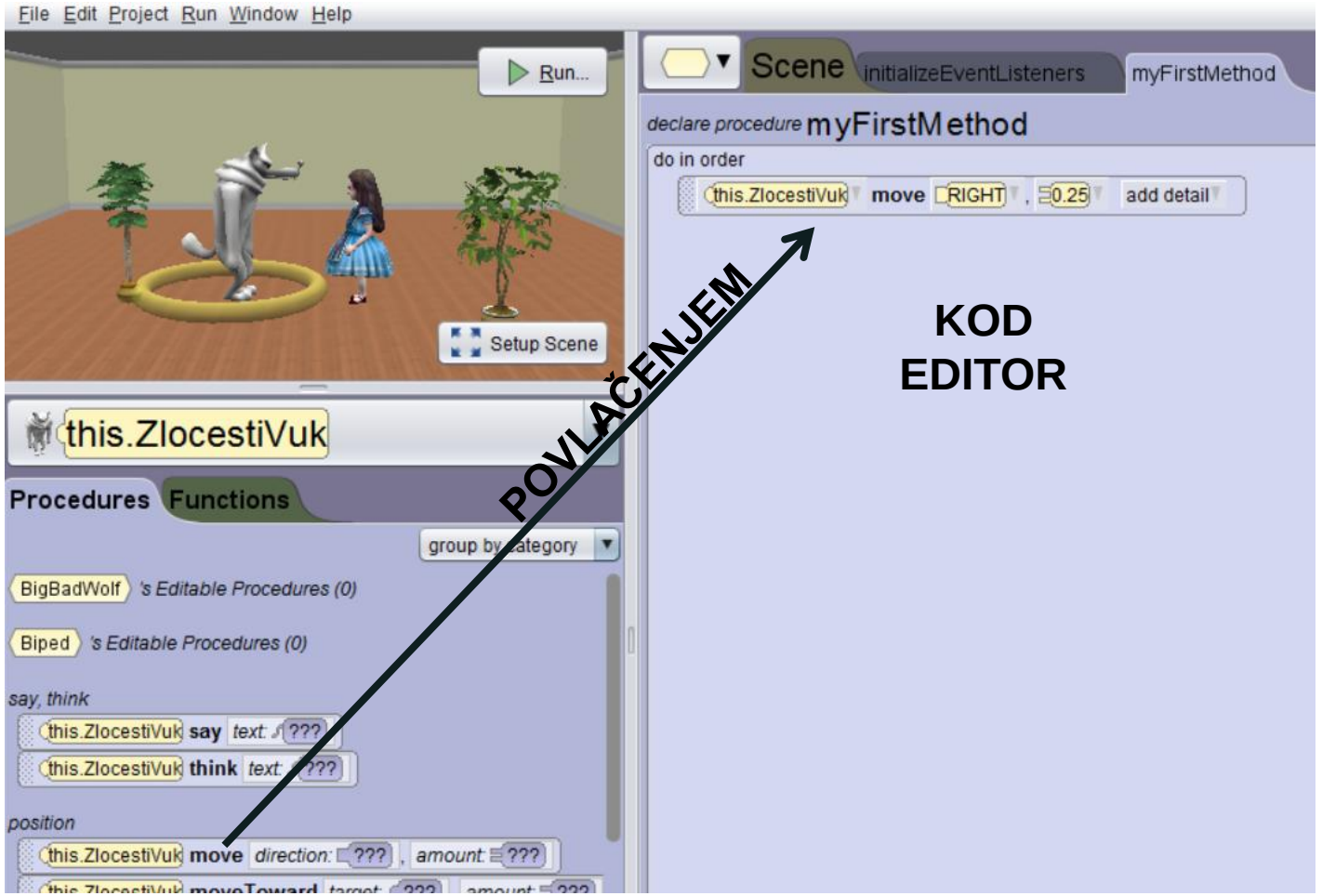
Procedure i funkcije

(metode)

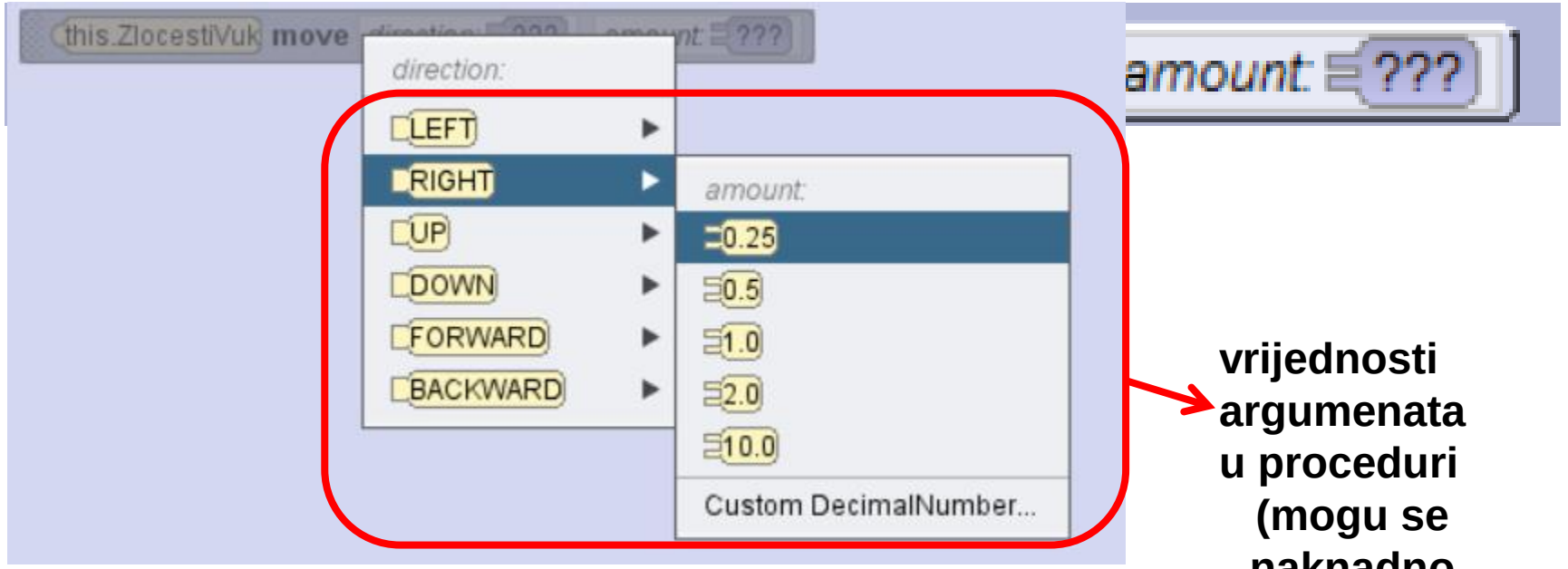
Procedura je dio programskog koda koji definira kako bi se objekt trebao ponašati.

Funkcija je dio programskog koda koji nadopunjava odabranu proceduru (namješta i određuje vrijednost).

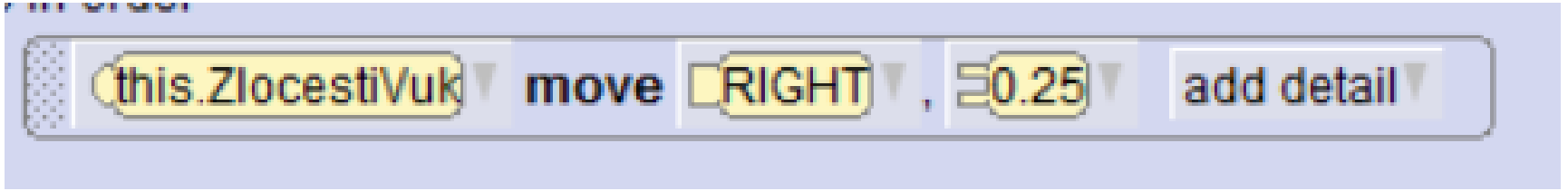
Procedura


File Edit Project Run Window Help
Run...
Setup Scene
this.ZlocestiVuk
Procedures
Functions
group by category
BigBadWolf 's Editable Procedures (0)
Biped 's Editable Procedures (0)
say, think
this.ZlocestiVuk say text: ???
this.ZlocestiVuk think text: ???
position
this.ZlocestiVuk move direction: ??? , amount: ???
Scene
initializeEventListeners
myFirstMethod
declare procedure myFirstMethod
do in order
this.ZlocestiVuk move RIGHT , 0.25 add detail
POVLAČENJEM
KOD
EDITOR

Nakon povlačenja procedure u kod editor, pojavljuje se izbornik vrijednosti argumenata u proceduri:



vrijednosti argumenata u proceduri (mogu se naknadno mijenjati)



Objekt (ili njegov dio) možemo micati u 6 smjerova:

- gore (up)
- dolje (down)
- lijevo (left)
- desno (right)
- naprijed (forward)
- nazad (backward)

Procedure	Description
Move	**Pomiče objekt u jednom od 6 mogućih smjerova.**
Move Toward	**Pomiče objekt prema drugom objektu.**
Move Away From	**Odmiče objekt od drugog objekta.**
Move To	**Pomiče objekt iz njegove trenutne pozicije u poziciju ciljnog (navedenog) objekta.**
Move and Orient To	**Pomiče objekt iz njegove trenutne pozicije u poziciju navedenog objekta i namješta orijentaciju objekta istu kao ciljnog (navedenog).**
Delay	**Zaustavlja pomicanje objekta za određen broj sekundi. Kašnjenje se može upotrijebiti za usporavanje animacije.**
Say	**Izrada oblačića s tekstom kojeg objekt izgovara.**

Izrada procedura

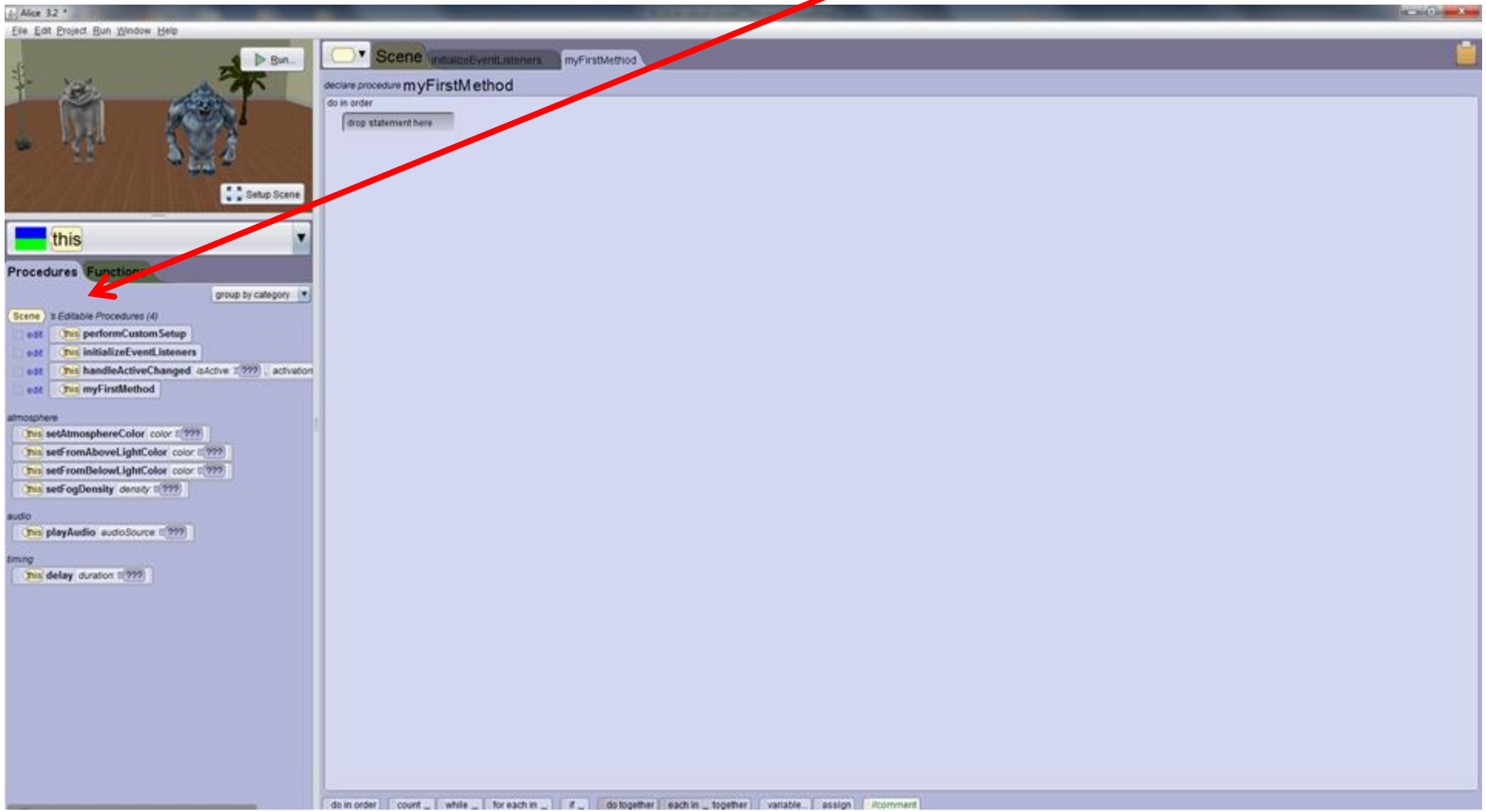
Procedura je dio programskog koda koji definira kako bi se objekt trebao ponašati.

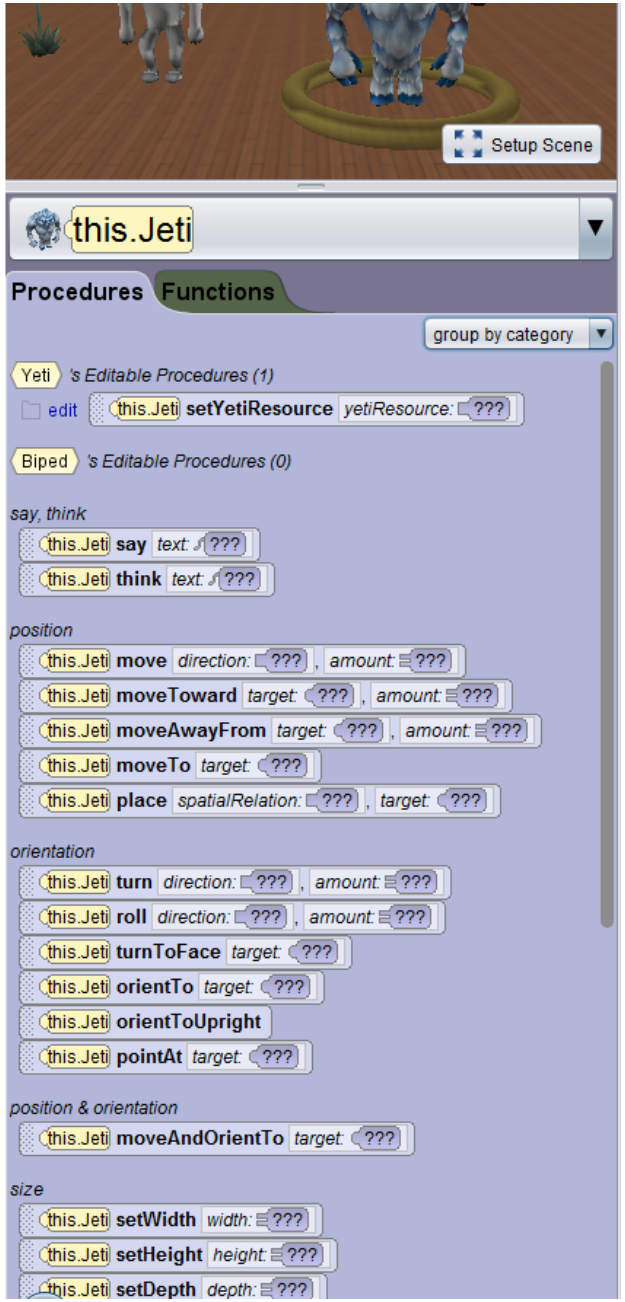
Do sada smo namještali scenu, a sad slijedi programiranje.

Pokrenite Alice i organizirajte scenu:

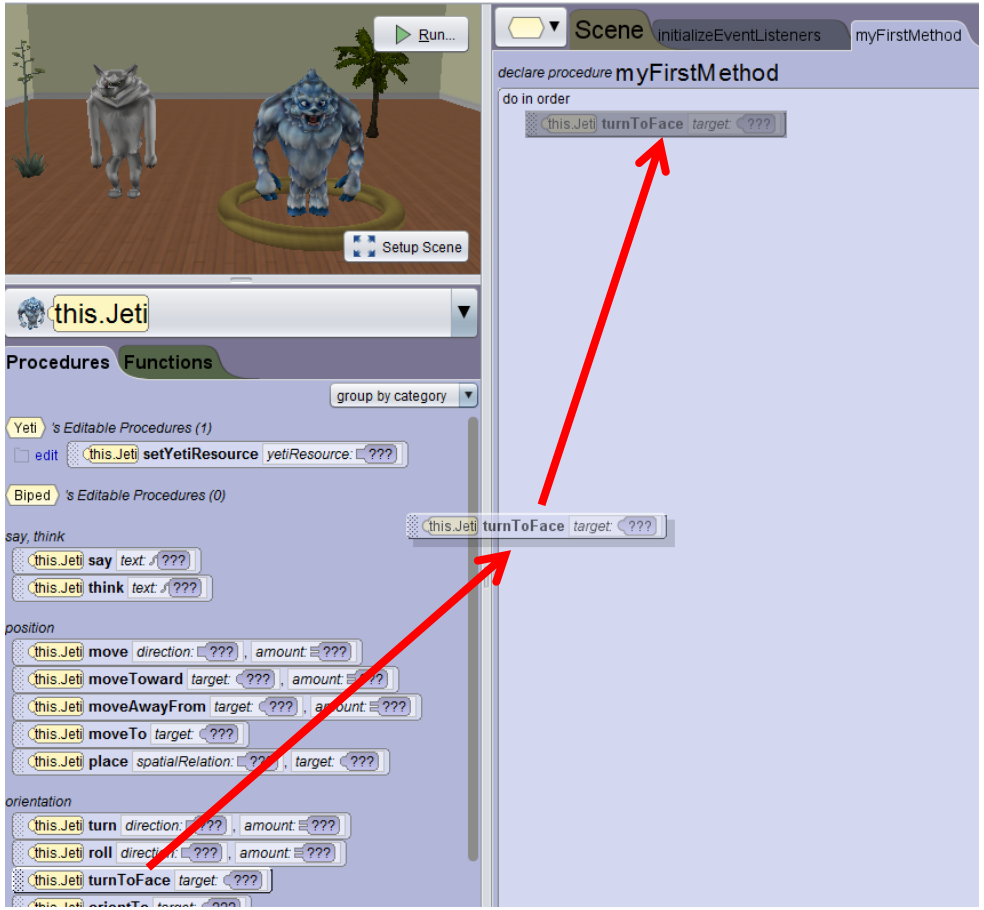
Klikom na **Edit Code** dolazimo u prostor za pisanje programa (pokretanje likova).

U ovom prostoru koristimo gotove procedure za svaki objekt pojedinačno.
Označimo objekt na koji će se odnositi i odaberemo proceduru:





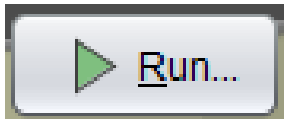
Procedure se odabiru povlačenjem u desni dio ekrana:



Vuk i Jeti pričaju:

ISPROBAJTE!

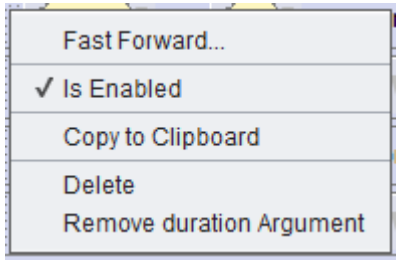
Tipkom

Run 

pokrećemo proceduru.

Desnim klikom maknemo kvačicu ispred

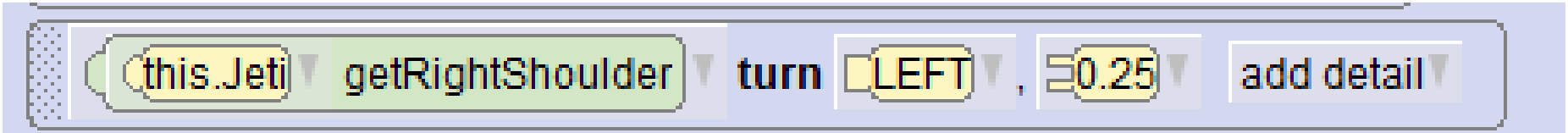
Is Enabled



pa se taj redak ne izvršava!

Vuk i Jeti mašu rukama:

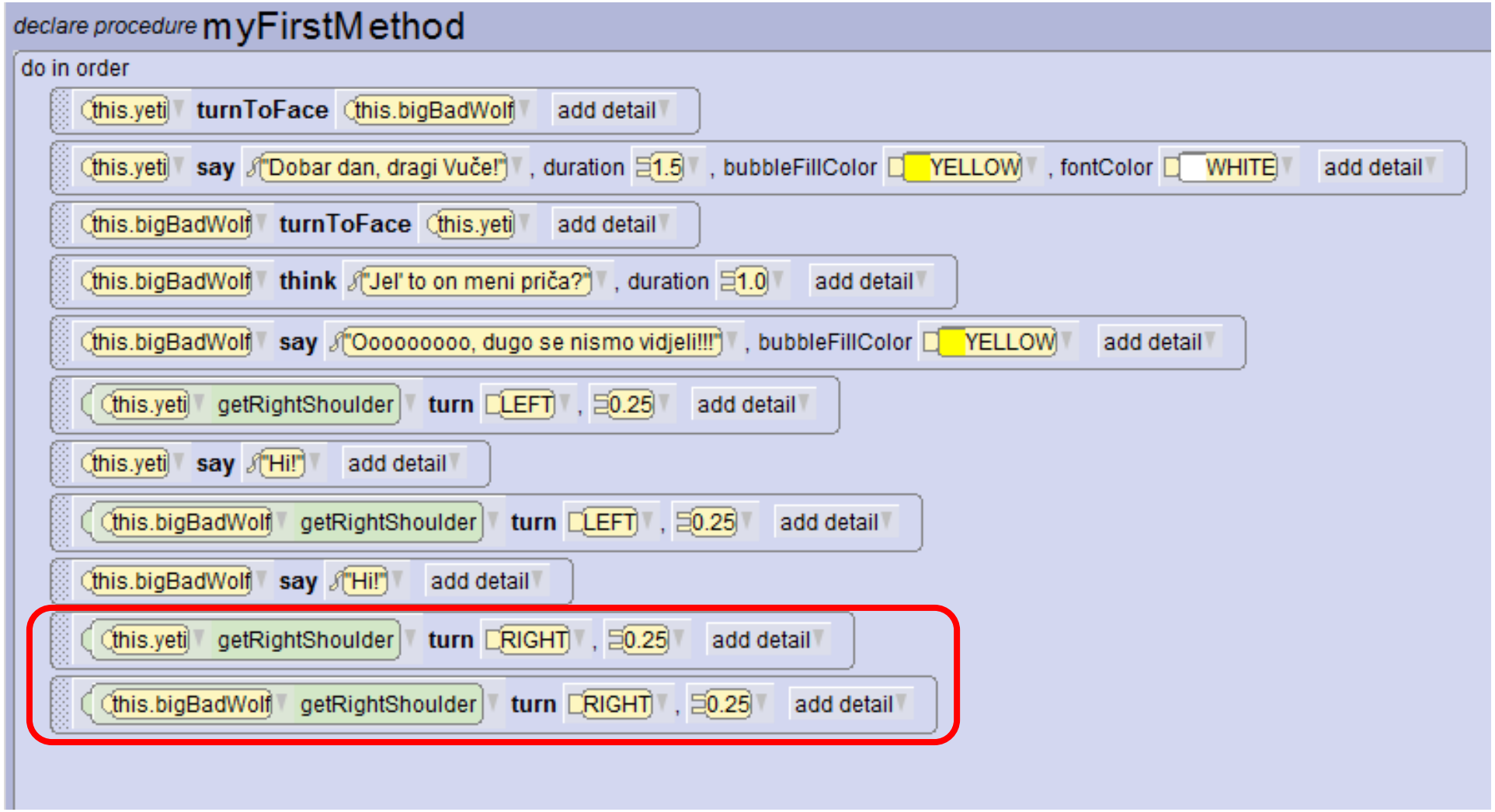

Run...
Setup Scene
this.Jeti
Procedures
Functions
group by category
this.Jeti getRightHip
this.Jeti getRightIndexFinger
this.Jeti getRightIndexFingerKnuckle
this.Jeti getRightShoulder
Scene
initializeEventListeners
declare procedure myFirstMethod
do in order
this.Jeti turnToFace this.Vuk add detail
this.Jeti say "Dobar dan, dragi Vuče!"
this.Vuk turnToFace this.Jeti add detail
this.Vuk think "Jel on to meni priča?"
this.Vuk say "OOOOO, dugo se nismo vidjeli!!!"
this.Jeti turn LEFT 0.25 add detail


this.Jeti getRightShoulder turn LEFT 0.25 add detail

Kopiranjem i lijepljenjem dovršite program:

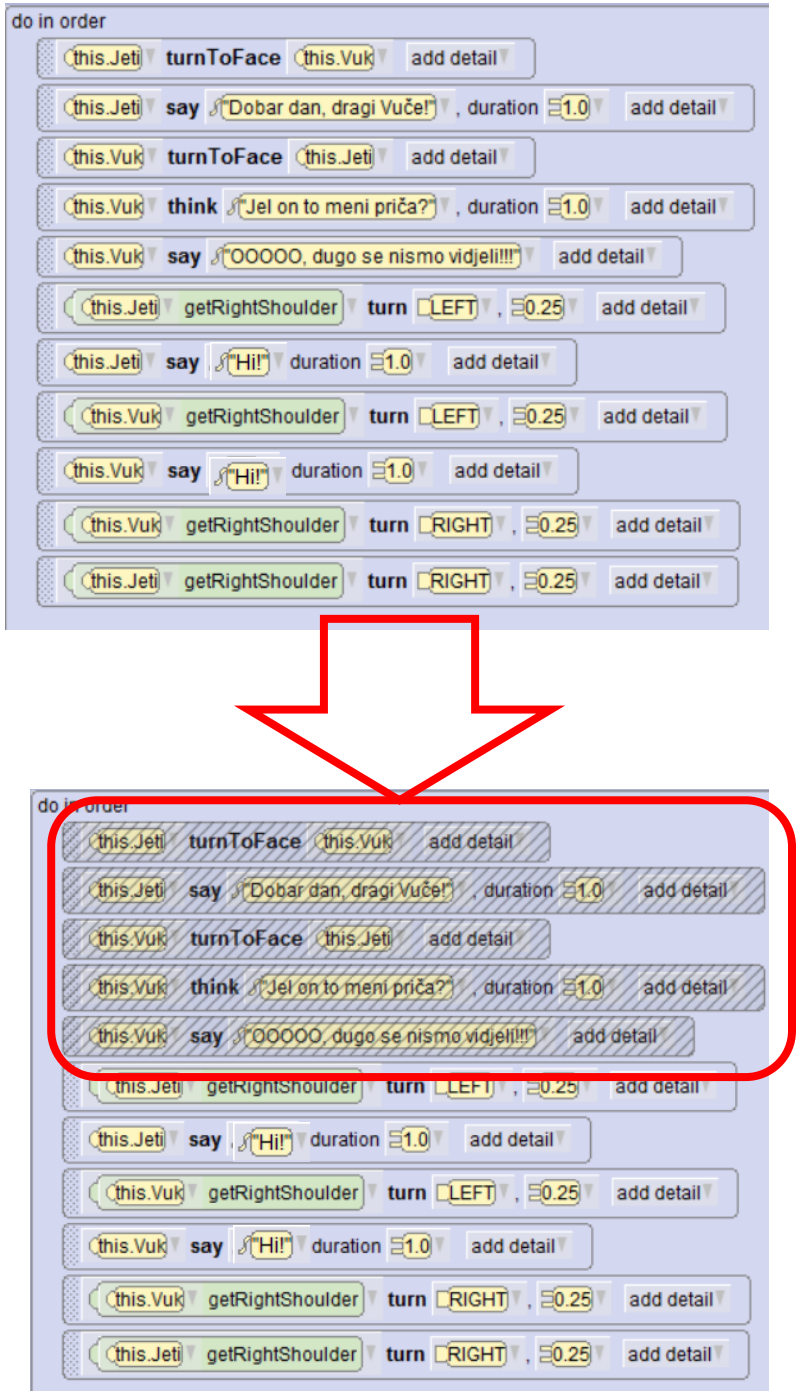
Kopiranje:
Ctrl+povlačenje ili
desni klik

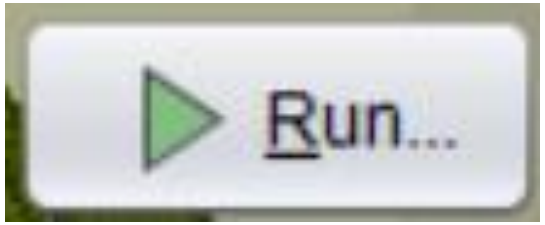
Na kraju im spustite ruke:

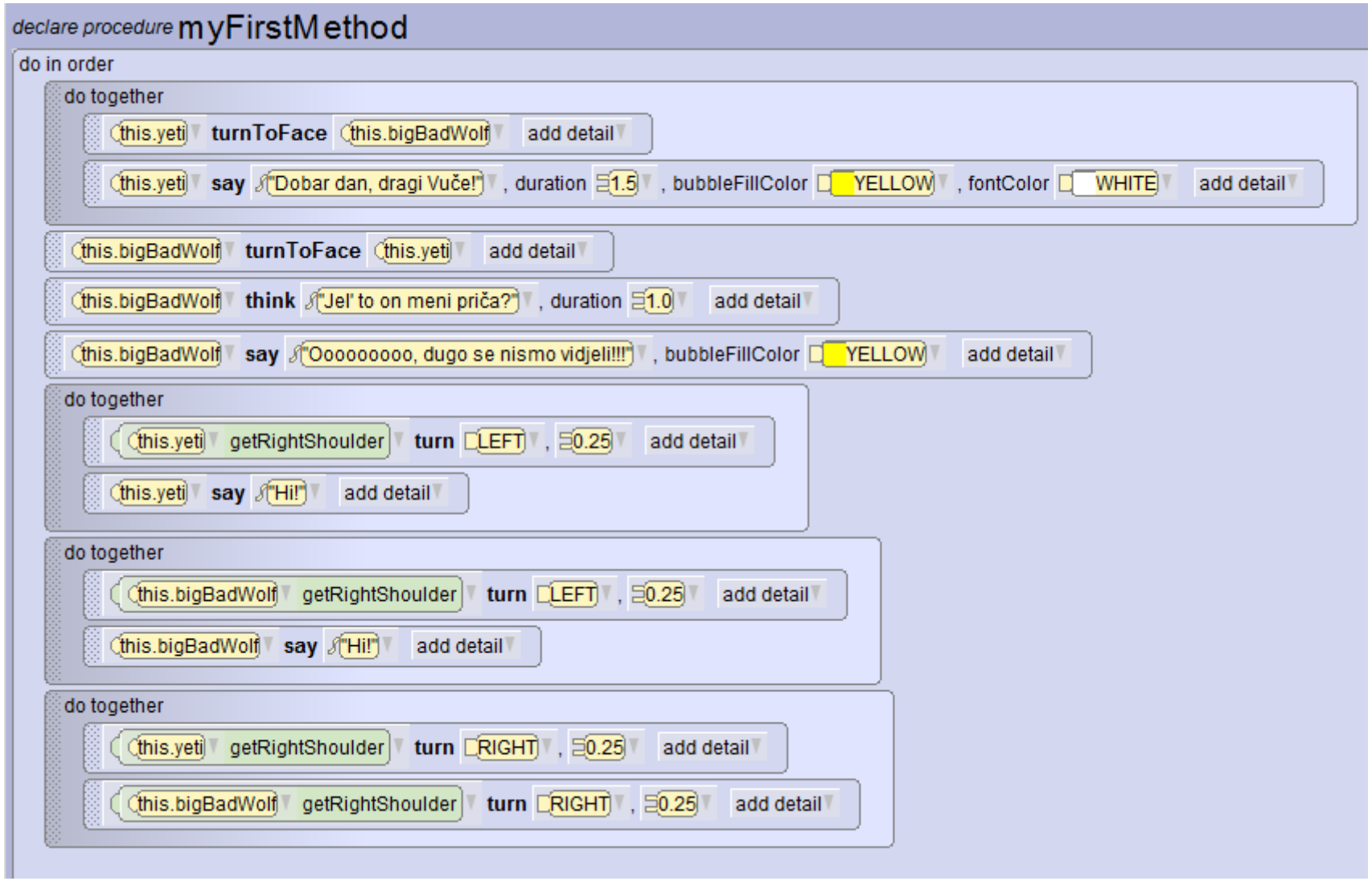


Pojedini dijelovi se mogu trenutno i onemogućiti:

desnim klikom

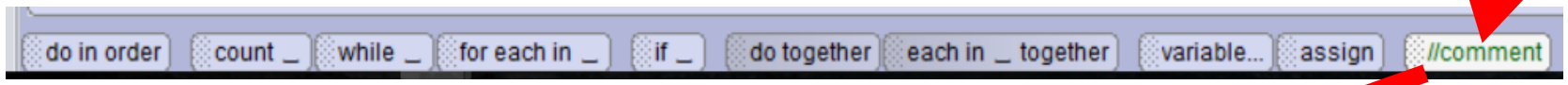

do in order
this.Jeti turnToFace this.Vuk add detail
this.Jeti say "Dobar dan, dragi Vuče!" , duration 1.0 add detail
this.Vuk turnToFace this.Jeti add detail
this.Vuk think "Jel on to meni priča?" , duration 1.0 add detail
this.Vuk say "OOOOO, dugo se nismo vidjeli!!!" add detail
this.Jeti getRightShoulder turn LEFT , 0.25 add detail
this.Jeti say "Hi!" duration 1.0 add detail
this.Vuk getRightShoulder turn LEFT , 0.25 add detail
this.Vuk say "Hi!" duration 1.0 add detail
this.Vuk getRightShoulder turn RIGHT , 0.25 add detail
this.Jeti getRightShoulder turn RIGHT , 0.25 add detail
this.Jeti turnToFace this.Vuk add detail
this.Jeti say "Dobar dan, dragi Vuče!" , duration 1.0 add detail
this.Vuk turnToFace this.Jeti add detail
this.Vuk think "Jel on to meni priča?" , duration 1.0 add detail
this.Vuk say "OOOOO, dugo se nismo vidjeli!!!" add detail
this.Jeti getRightShoulder turn LEFT , 0.25 add detail
this.Jeti say "Hi!" duration 1.0 add detail
this.Vuk getRightShoulder turn LEFT , 0.25 add detail
this.Vuk say "Hi!" duration 1.0 add detail
this.Vuk getRightShoulder turn RIGHT , 0.25 add detail
this.Jeti getRightShoulder turn RIGHT , 0.25 add detail


Run...

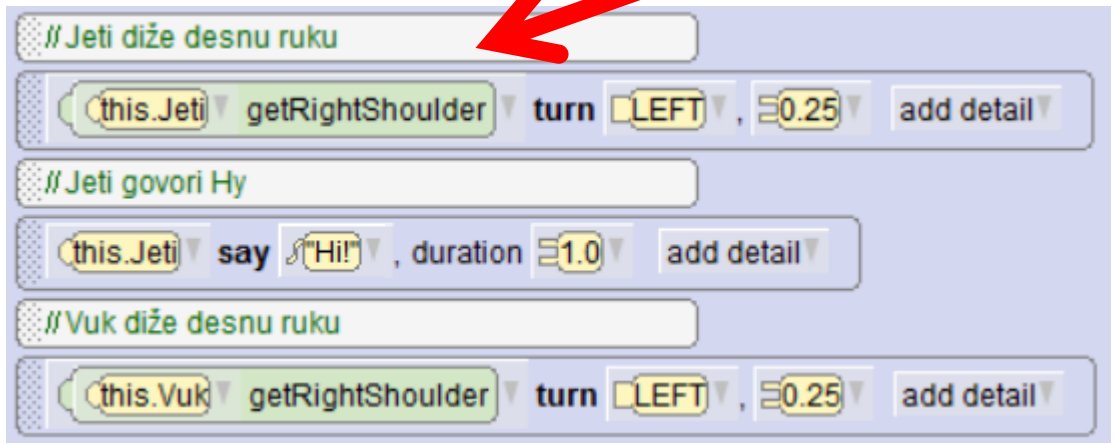

declare procedure myFirstMethod
do in order
do together
this.yeti turnToFace this.bigBadWolf add detail
this.yeti say "Dobar dan, dragi Vuče!" , duration 1.5 , bubbleFillColor YELLOW , fontColor WHITE add detail
this.bigBadWolf turnToFace this.yeti add detail
this.bigBadWolf think "Jel' to on meni priča?" , duration 1.0 add detail
this.bigBadWolf say "Ooooooooo, dugo se nismo vidjeli!!!" , bubbleFillColor YELLOW add detail
do together
this.yeti getRightShoulder turn LEFT , 0.25 add detail
this.yeti say "Hi!" add detail
do together
this.bigBadWolf getRightShoulder turn LEFT , 0.25 add detail
this.bigBadWolf say "Hi!" add detail
do together
this.yeti getRightShoulder turn RIGHT , 0.25 add detail
this.bigBadWolf getRightShoulder turn RIGHT , 0.25 add detail

Umetanje komentara

Za opis programa možemo koristiti komentare koji opisuju što u programu slijedi



umećemo povlačenjem



Komentari ne utječu na rad programa i mogu se umetnuti u bilo koje mjesto u programu!

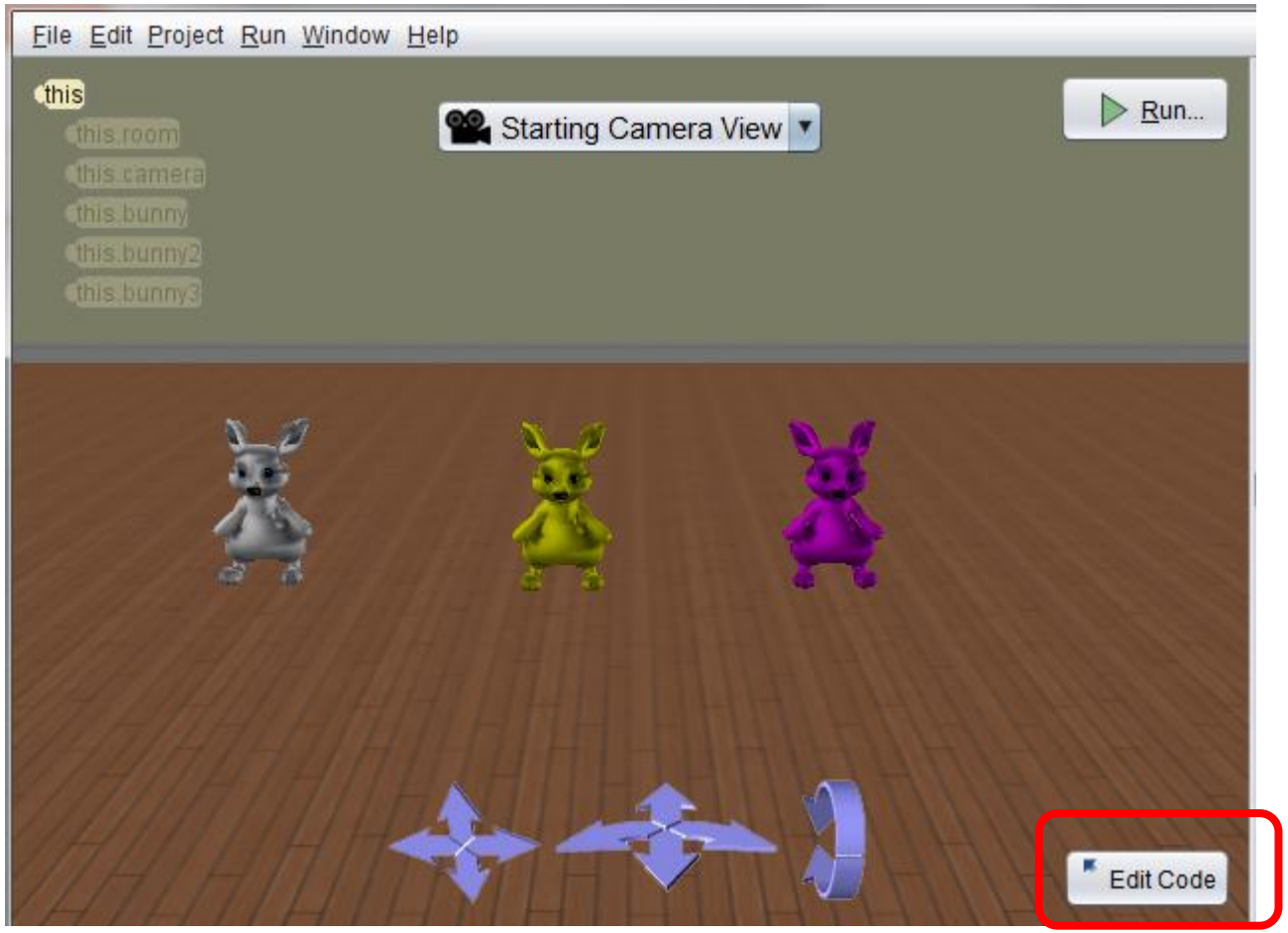
Namjestite scenu sa svoja odabrana dva lika i dva stabla.

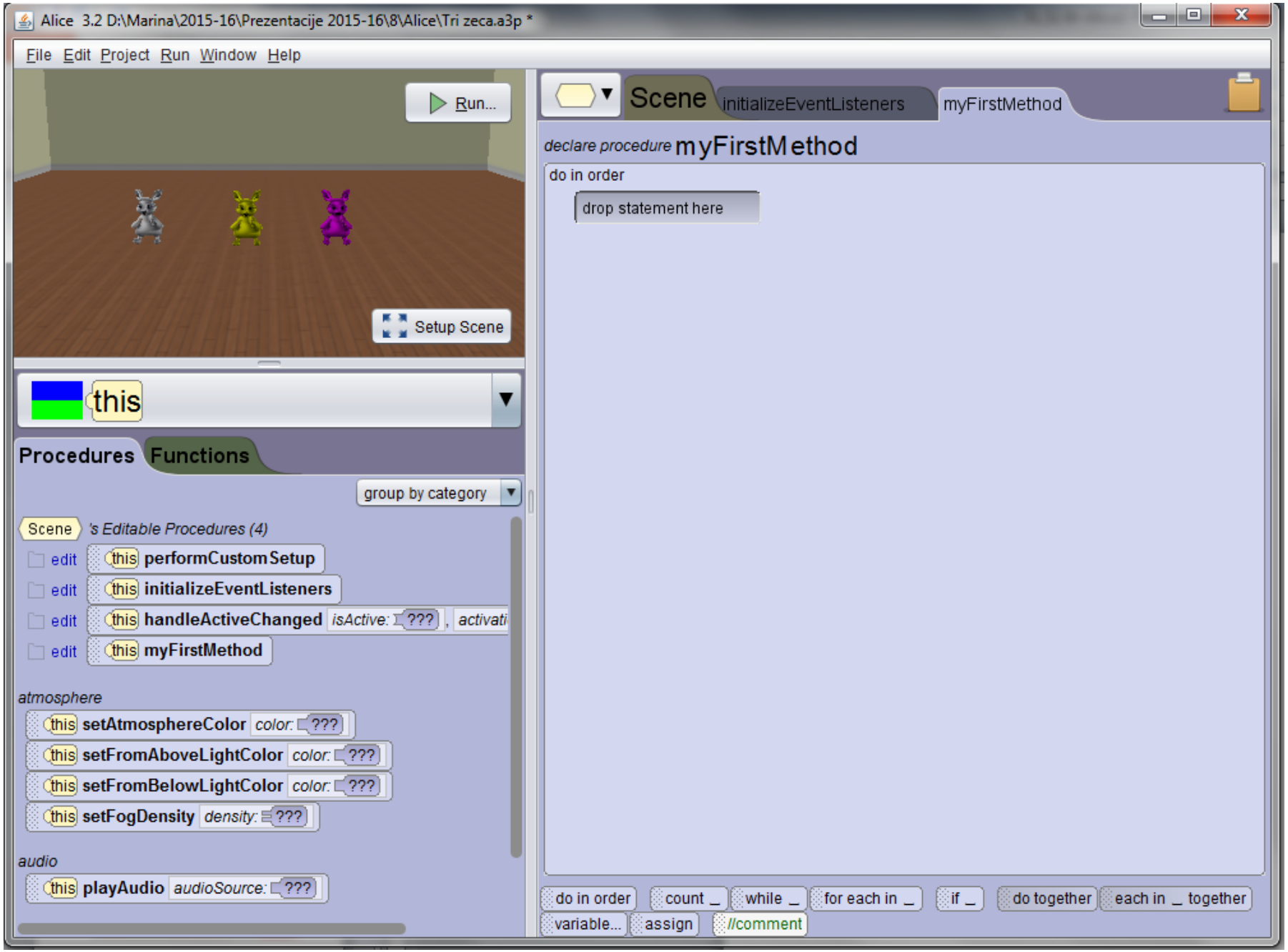
Neka se likovi okrenu jedan prema drugome, pruže ruke kao da se rukuju (početnu scenu namjestite tako da su blizu) i neka nešto kažu.

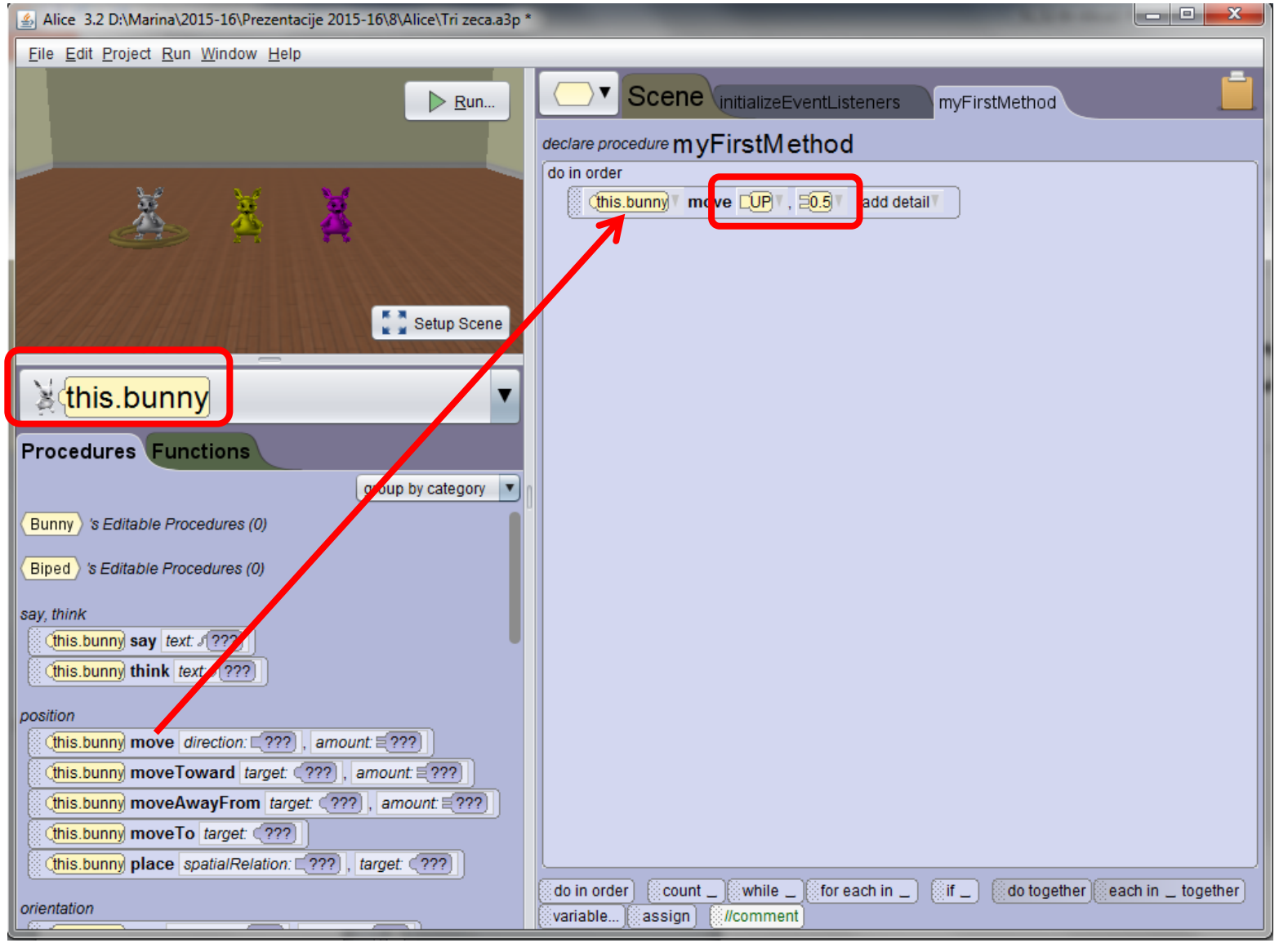
Tri zečića skaču

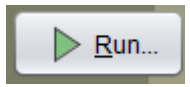
Organizirajte scenu

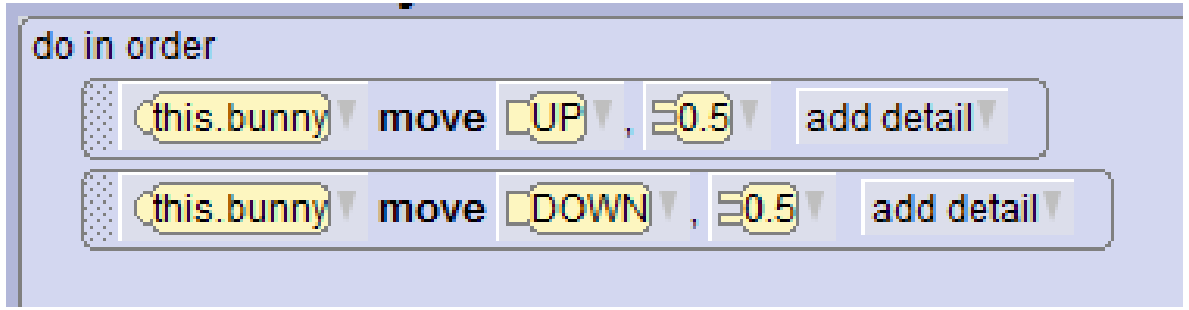
Napišimo programski kod



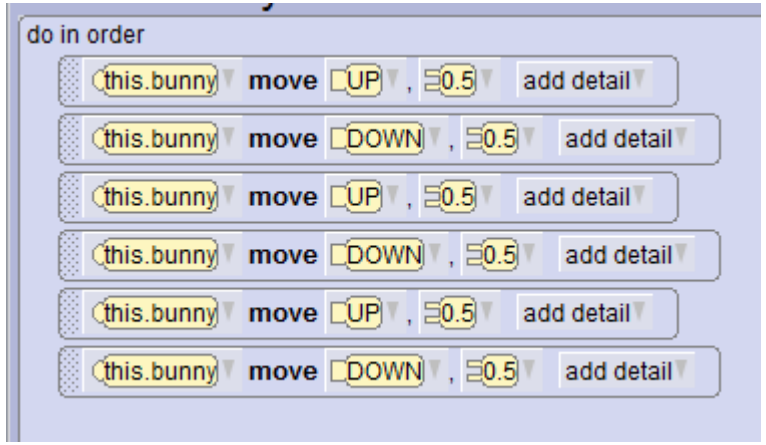

Alice 3.2 D:\Marina\2015-16\Prezentacije 2015-16\8\Alice\Tri zeca.a3p *
File Edit Project Run Window Help
Run...
Setup Scene
this
Procedures
Functions
group by category
Scene 's Editable Procedures (4)
edit this performCustomSetup
edit this initializeEventListeners
edit this handleActiveChanged isActive: ??? , activati
edit this myFirstMethod
atmosphere
this setAtmosphereColor color: ???
this setFromAboveLightColor color: ???
this setFromBelowLightColor color: ???
this setFogDensity density: ???
audio
this playAudio audioSource: ???
Scene
initializeEventListeners
myFirstMethod
declare procedure myFirstMethod
do in order
drop statement here
do in order
count _
while _
for each in _
if _
do together
each in _ together
variable...
assign
//comment


Alice 3.2 D:\Marina\2015-16\Prezentacije 2015-16\8\Alice\Tri zeca.a3p *
File Edit Project Run Window Help
Run...
Setup Scene
this.bunny
Procedures
Functions
group by category
Bunny 's Editable Procedures (0)
Biped 's Editable Procedures (0)
say, think
this.bunny say text: ???
this.bunny think text: ???
position
this.bunny move direction: ??? , amount: ???
this.bunny moveToward target: ??? , amount: ???
this.bunny moveAwayFrom target: ??? , amount: ???
this.bunny moveTo target: ???
this.bunny place spatialRelation: ??? , target: ???
orientation
Scene
initializeEventListeners
myFirstMethod
declare procedure myFirstMethod
do in order
this.bunny move UP , 0.5 add detail
do in order
count _
while _
for each in _
if _
do together
each in _ together
variable...
assign
//comment

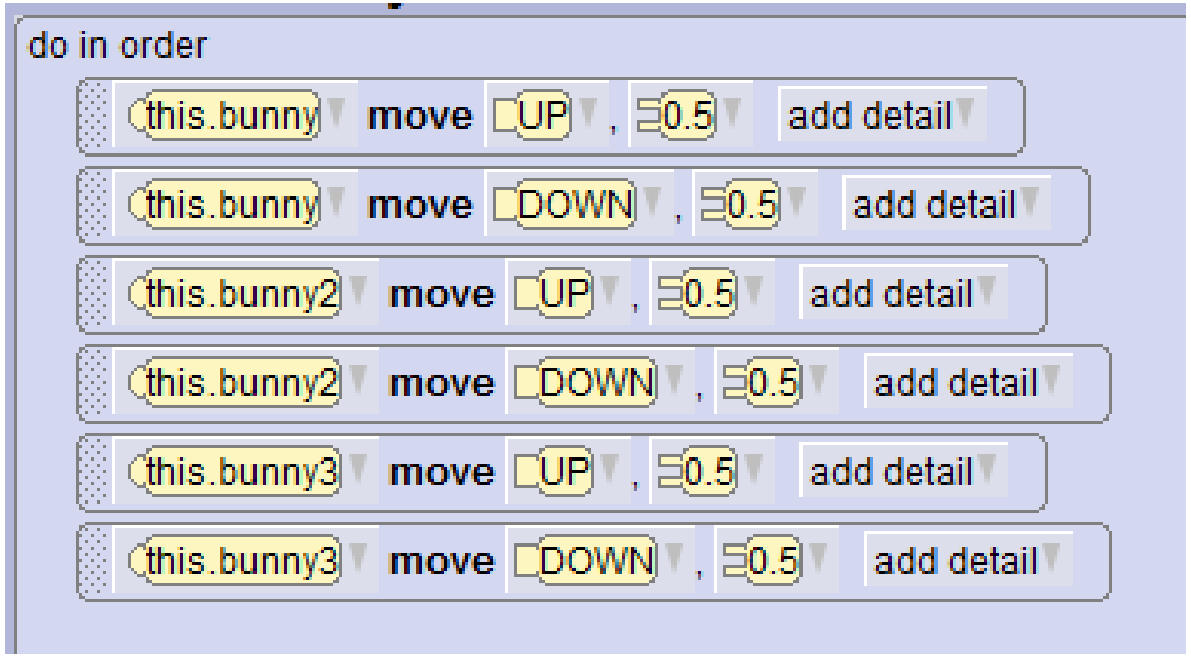

Run...



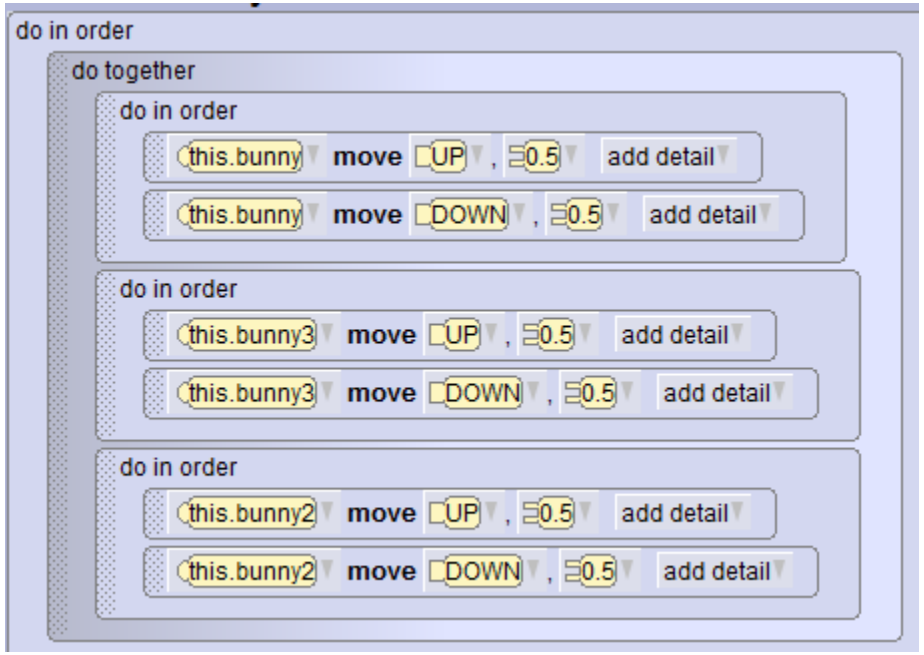
Kopirajmo (Ctrl+povlačenje):

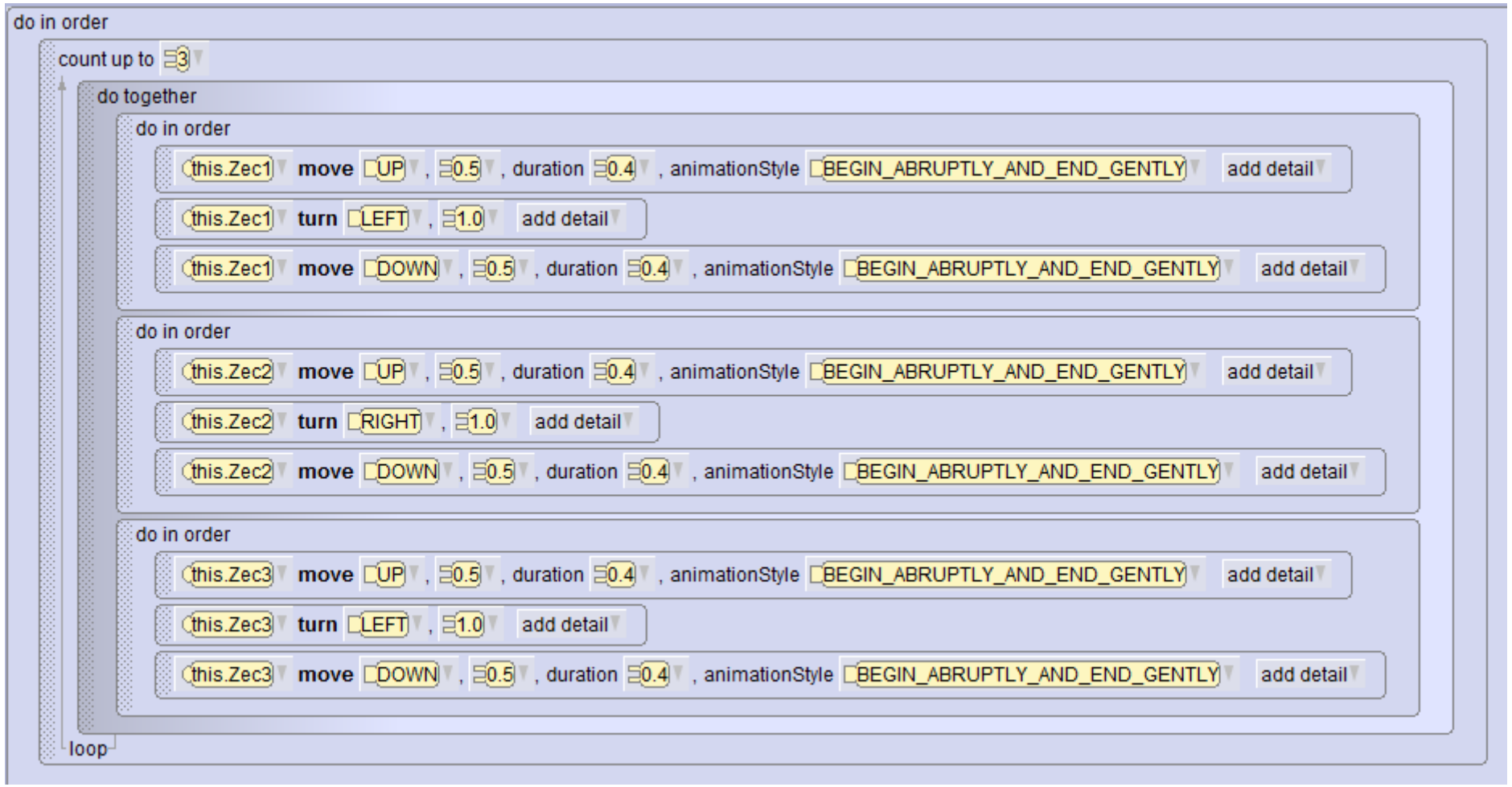


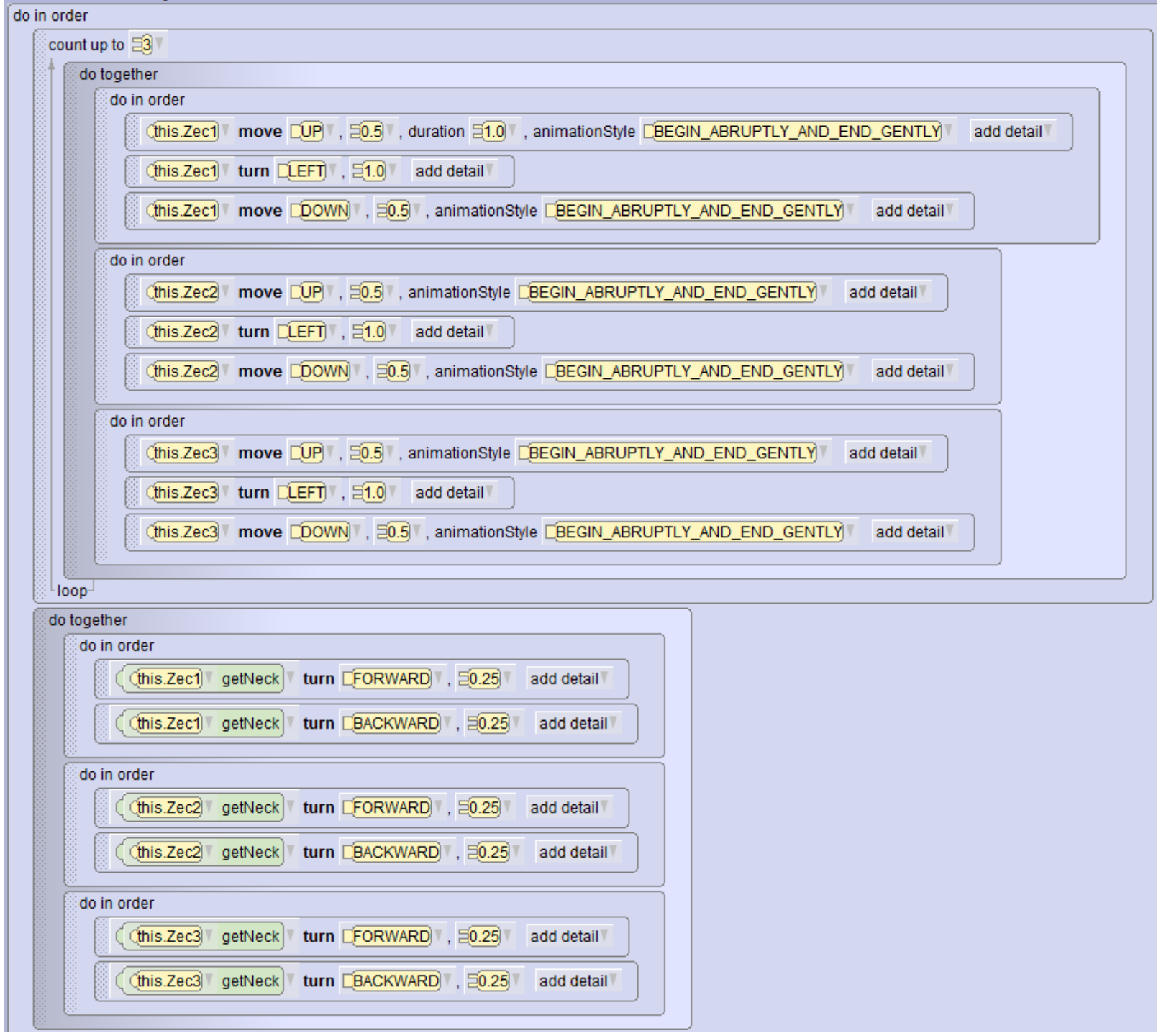
Promijenimo i pokrenimo



Pokušajte promijeniti u




do in order
count up to 3
do together
do in order
this.Zec1 move UP , 0.5 , duration 0.4 , animationStyle BEGIN_ABRUPTLY_AND_END_GENTLY add detail
this.Zec1 turn LEFT , 1.0 add detail
this.Zec1 move DOWN , 0.5 , duration 0.4 , animationStyle BEGIN_ABRUPTLY_AND_END_GENTLY add detail
do in order
this.Zec2 move UP , 0.5 , duration 0.4 , animationStyle BEGIN_ABRUPTLY_AND_END_GENTLY add detail
this.Zec2 turn RIGHT , 1.0 add detail
this.Zec2 move DOWN , 0.5 , duration 0.4 , animationStyle BEGIN_ABRUPTLY_AND_END_GENTLY add detail
do in order
this.Zec3 move UP , 0.5 , duration 0.4 , animationStyle BEGIN_ABRUPTLY_AND_END_GENTLY add detail
this.Zec3 turn LEFT , 1.0 add detail
this.Zec3 move DOWN , 0.5 , duration 0.4 , animationStyle BEGIN_ABRUPTLY_AND_END_GENTLY add detail
loop


do in order
count up to 3
do together
do in order
this.Zec1 move UP , 0.5 , duration 1.0 , animationStyle BEGIN_ABRUPTLY_AND_END_GENTLY add detail
this.Zec1 turn LEFT , 1.0 add detail
this.Zec1 move DOWN , 0.5 , animationStyle BEGIN_ABRUPTLY_AND_END_GENTLY add detail
do in order
this.Zec2 move UP , 0.5 , animationStyle BEGIN_ABRUPTLY_AND_END_GENTLY add detail
this.Zec2 turn LEFT , 1.0 add detail
this.Zec2 move DOWN , 0.5 , animationStyle BEGIN_ABRUPTLY_AND_END_GENTLY add detail
do in order
this.Zec3 move UP , 0.5 , animationStyle BEGIN_ABRUPTLY_AND_END_GENTLY add detail
this.Zec3 turn LEFT , 1.0 add detail
this.Zec3 move DOWN , 0.5 , animationStyle BEGIN_ABRUPTLY_AND_END_GENTLY add detail
loop
do together
do in order
this.Zec1 getNeck turn FORWARD , 0.25 add detail
this.Zec1 getNeck turn BACKWARD , 0.25 add detail
do in order
this.Zec2 getNeck turn FORWARD , 0.25 add detail
this.Zec2 getNeck turn BACKWARD , 0.25 add detail
do in order
this.Zec3 getNeck turn FORWARD , 0.25 add detail
this.Zec3 getNeck turn BACKWARD , 0.25 add detail

Ponovite današnji zadatak s tri pande:

skoče u zrak, okrenu se oko sebe i vrate se na pod

Vještica i Troll

Odabiremo pozadinu i namještamo scenu:

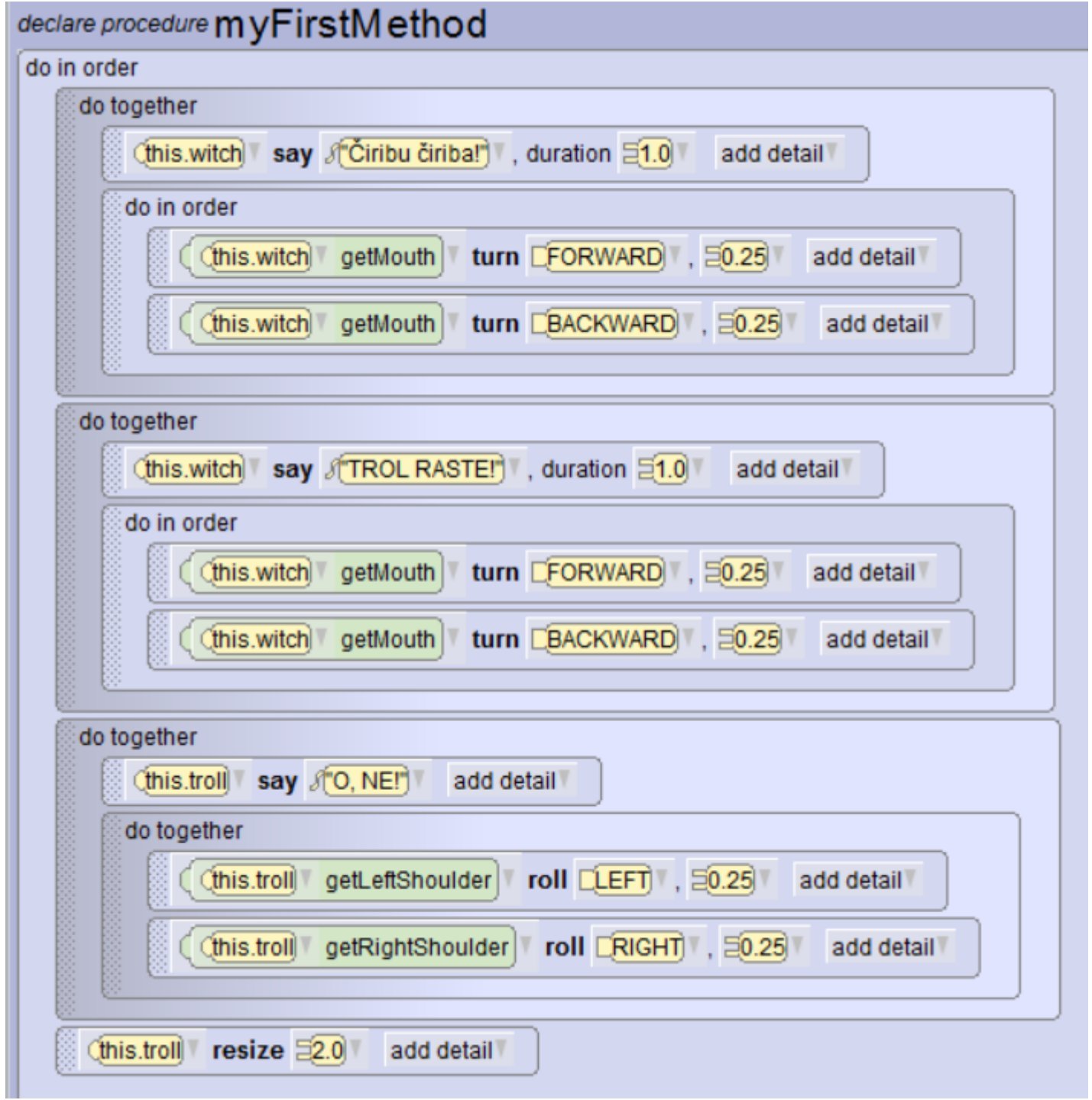
Odaberemo **SWAMP** pozadinu.

I objekte:

- Troll
- Witch
- 2 Pixies
- Cauldron
- Spell book
- Potion
- Magic stone

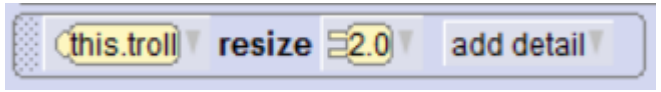
Namjestimo scenu

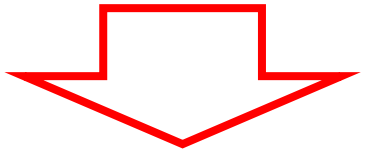
- Troll
- Witch
- 2 Pixies
- Cauldron
- Spell book
- Potion
- Magic stone

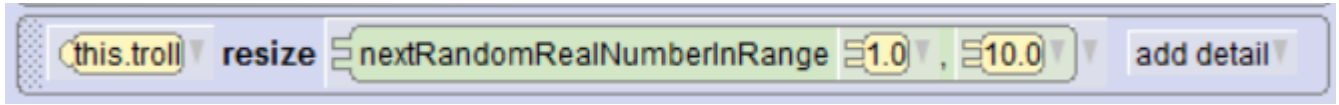

declare procedure myFirstMethod
do in order
do together
this.witch say "Čiribu čiriba!" , duration 1.0 add detail
do in order
this.witch getMouth turn FORWARD , 0.25 add detail
this.witch getMouth turn BACKWARD , 0.25 add detail
do together
this.witch say "TROL RASTE!" , duration 1.0 add detail
do in order
this.witch getMouth turn FORWARD , 0.25 add detail
this.witch getMouth turn BACKWARD , 0.25 add detail
do together
this.troll say "O, NE!" add detail
do together
this.troll getLeftShoulder roll LEFT , 0.25 add detail
this.troll getRightShoulder roll RIGHT , 0.25 add detail
this.troll resize 2.0 add detail

Promjenimo u random povećavanje Trola

(Trol će se svaki puta različito povećati)

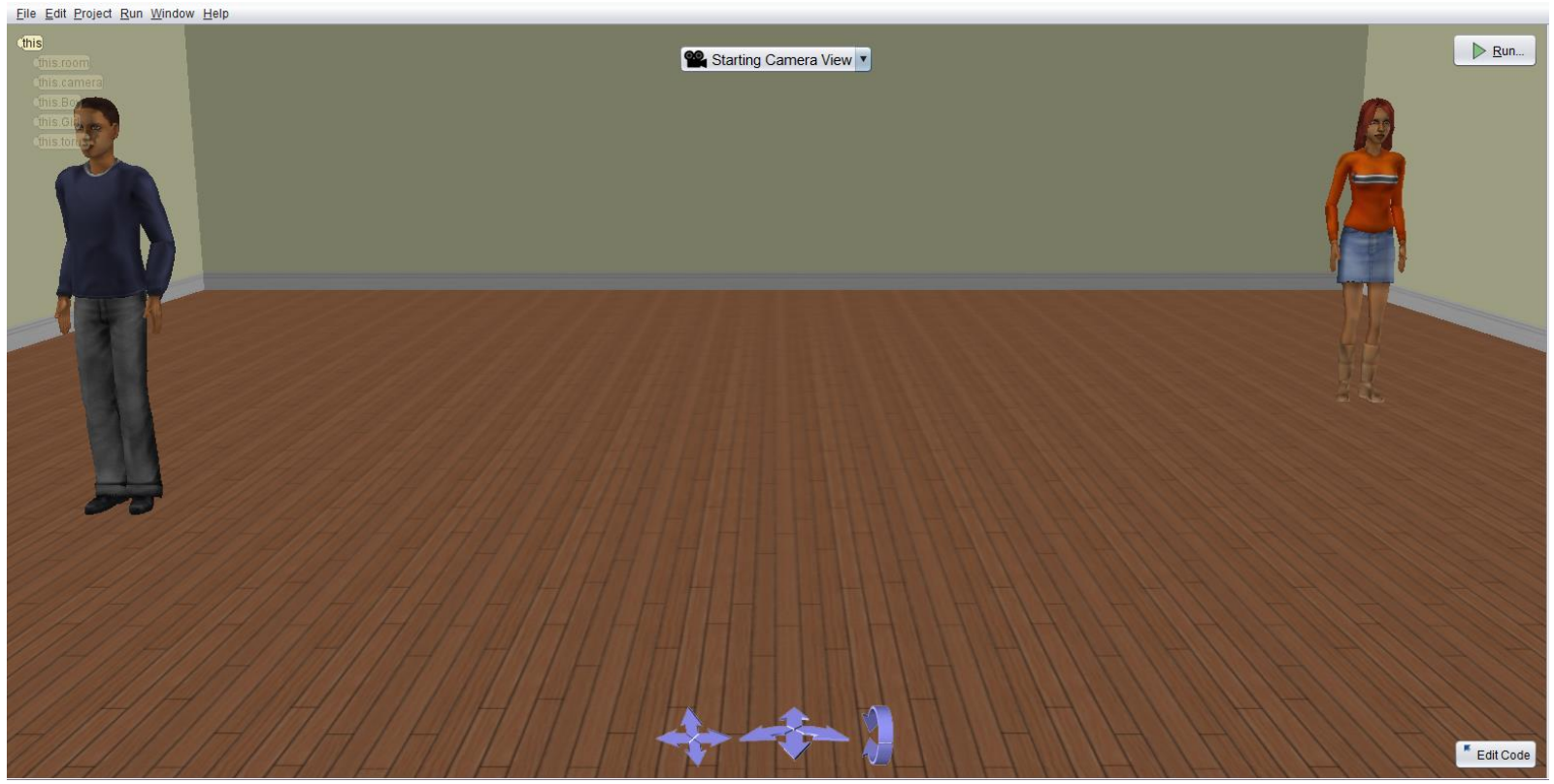




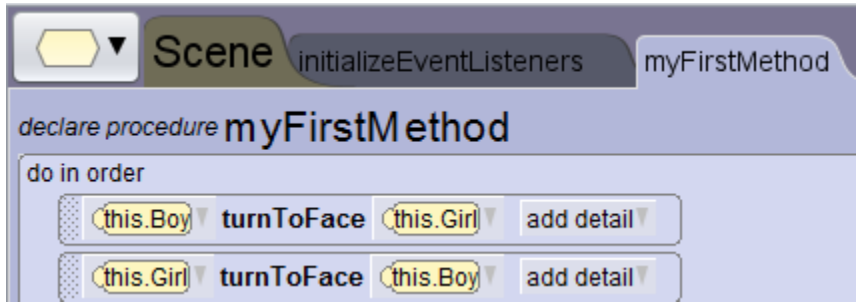


Djevojka i mladić se primiču i pričaju kad se sretnu

Odaberite likove i postavite ih u početni položaj na scenu



Likovi se okreću jedan prema drugome



Likovi se kreću dok se ne sretnu

Odabiremo naredbu

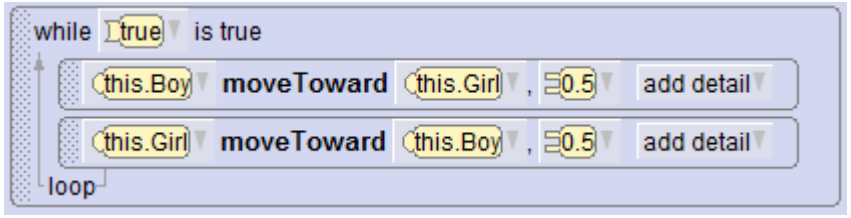
„dok se ne sretnu, kreću se prema naprijed”

Koristimo naredbu **WHILE** **(DOK)** – dok se ne sretnu, pomiču se prema naprijed:

ŠTO SE DOGAĐA?

Ne dolaze točno na istu poziciju!

i

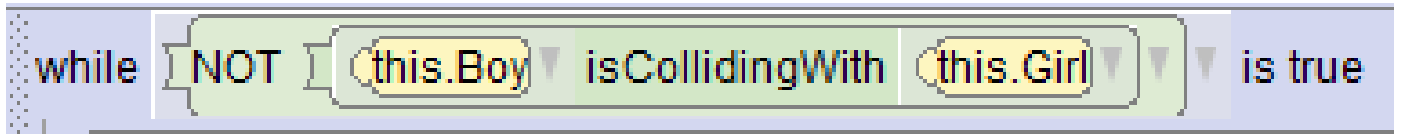


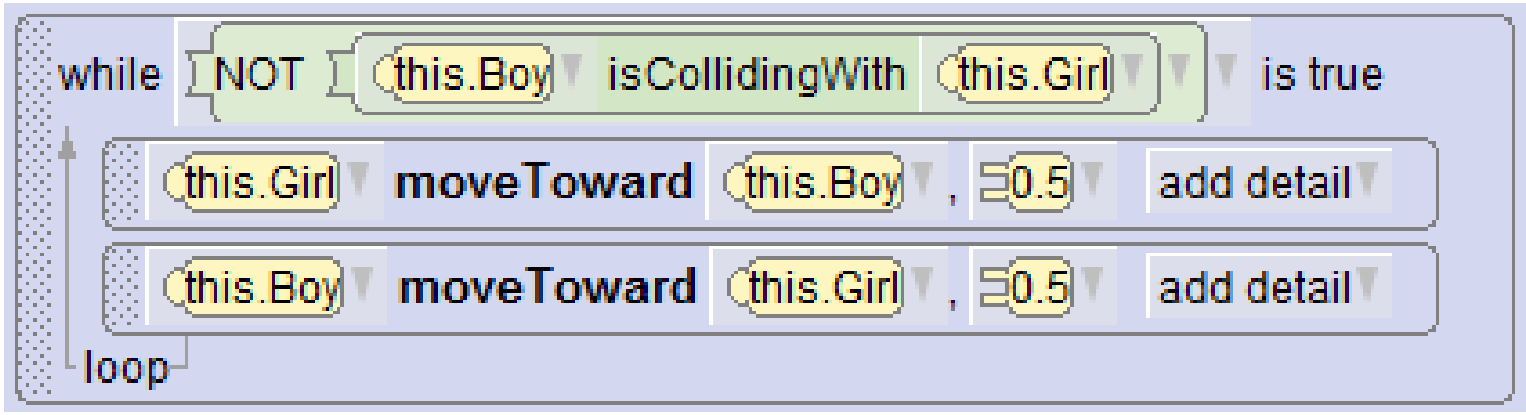
pa pokušajte!

Ovo kretanje prema drugom liku moramo ograničiti na trenutak kad se sretnu

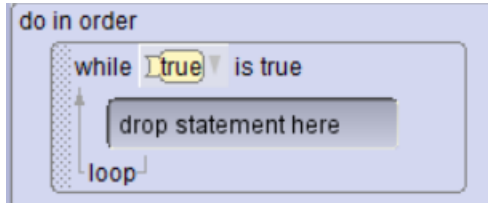
Trebamo reći: „krećite se jedan prema drugome dok se ne sudarite (dok ne kolidirate)”:

collision=sudar

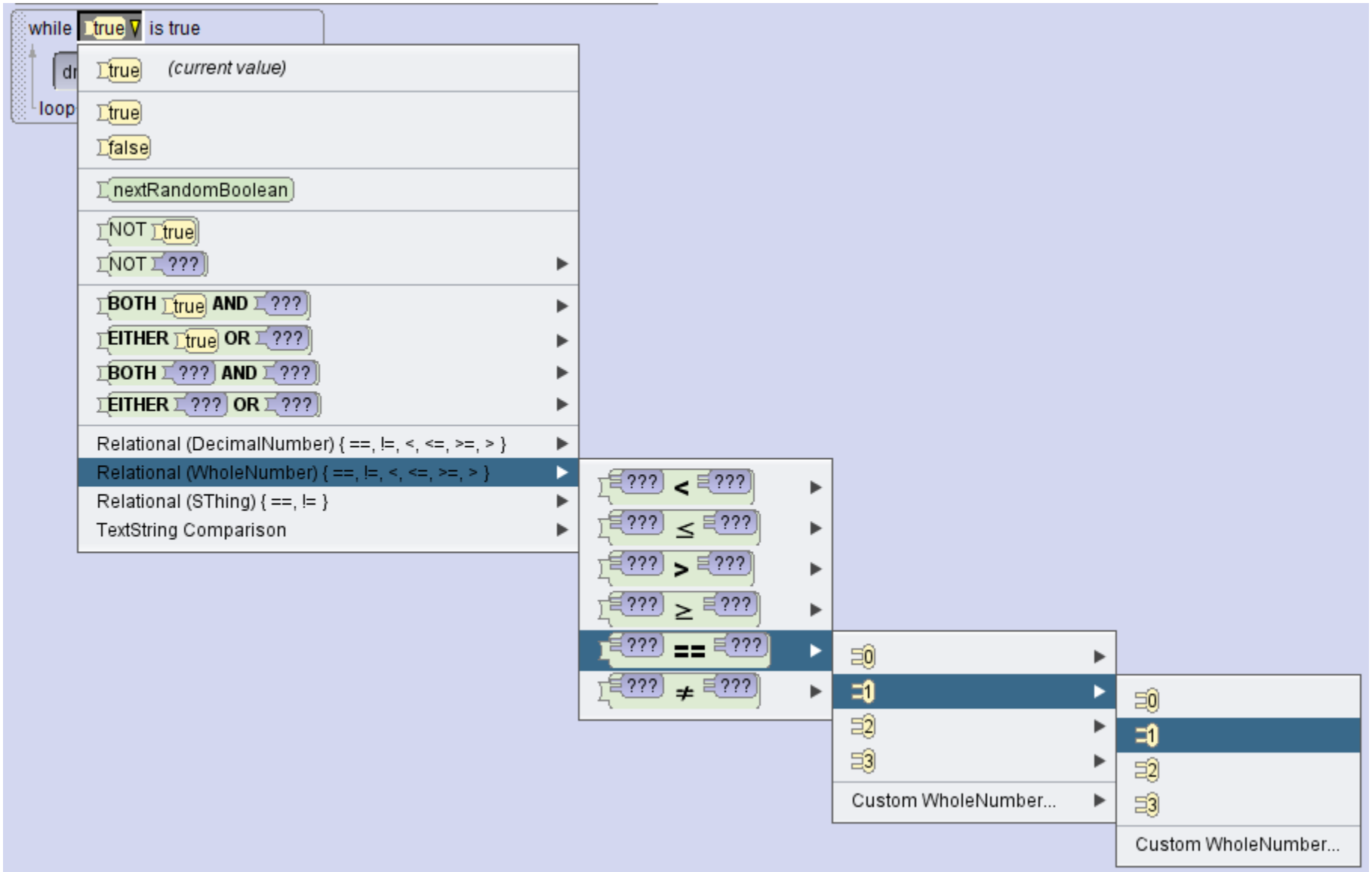



while
NOT
this.Boy
isCollidingWith
this.Girl
is true
this.Girl
moveToward
this.Boy
0.5
add detail
this.Boy
moveToward
this.Girl
0.5
add detail
loop

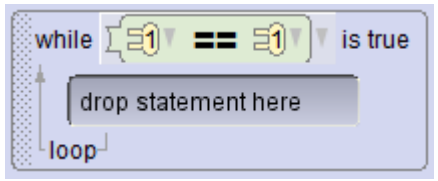
1.korak


do in order
while
true
is true
drop statement here
loop

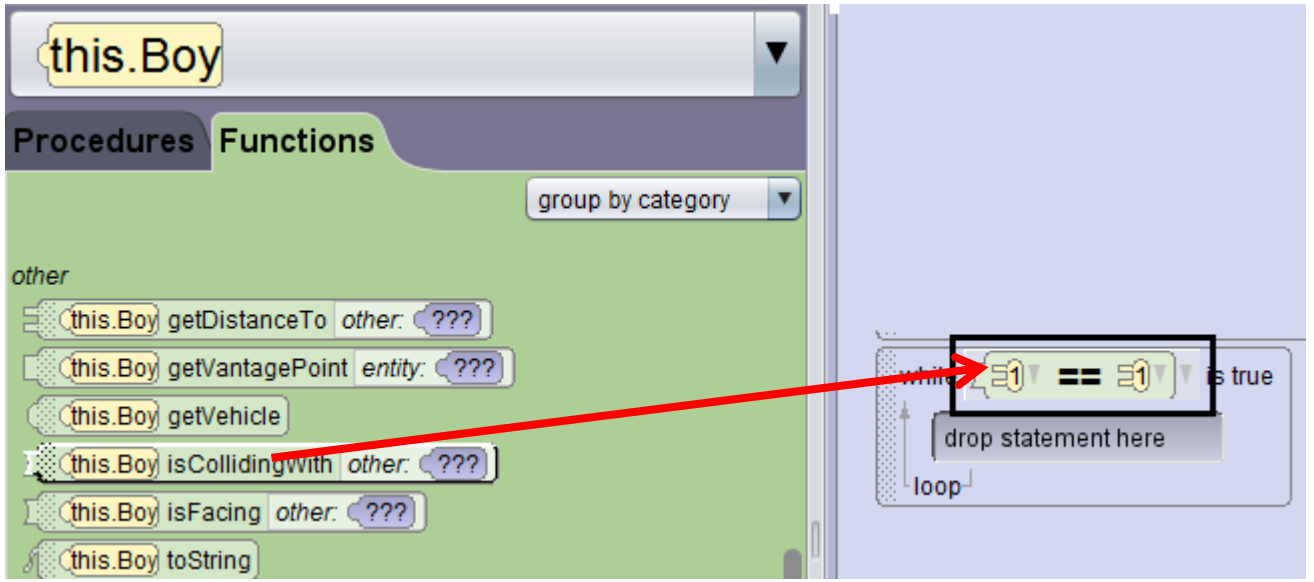
2.korak


while true is true
loop
true (current value)
true
false
nextRandomBoolean
NOT true
NOT ???
BOTH true AND ???
EITHER true OR ???
BOTH ??? AND ???
EITHER ??? OR ???
Relational (DecimalNumber) { ==, !=, <, <=, >=, > }
Relational (WholeNumber) { ==, !=, <, <=, >=, > }
Relational (SThing) { ==, != }
TextString Comparison
??? < ???
??? ≤ ???
??? > ???
??? ≥ ???
??? == ???
??? ≠ ???
0
1
2
3
Custom WholeNumber...
0
1
2
3
Custom WholeNumber...

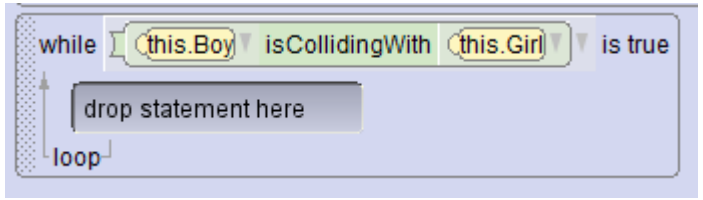
3. korak



4. korak

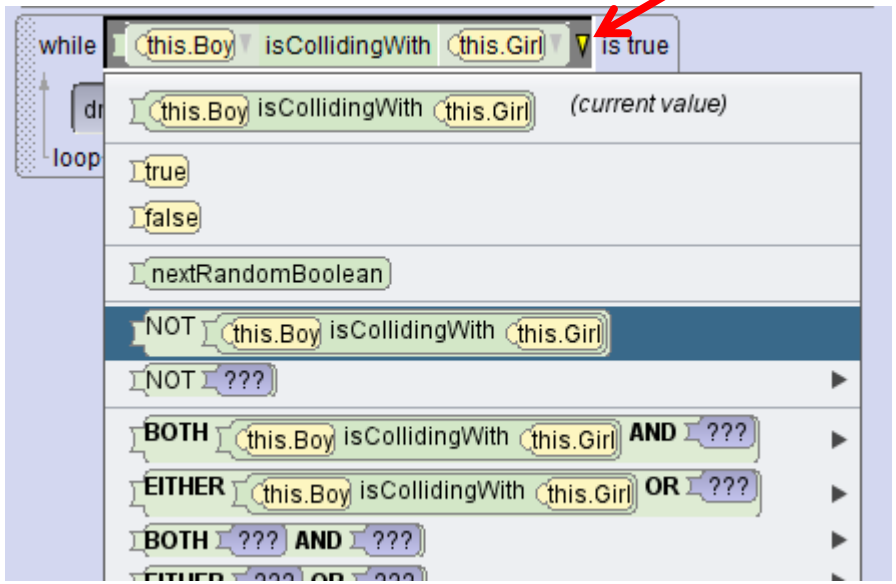


imamo:

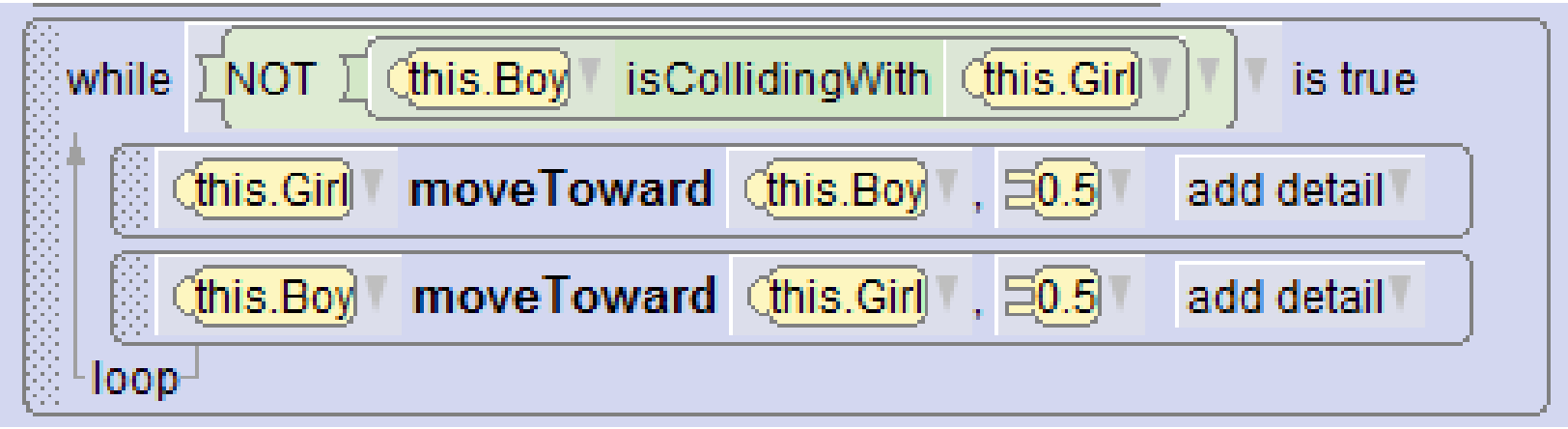


pazite, koju strelicu padajućeg izbornika odaberete!

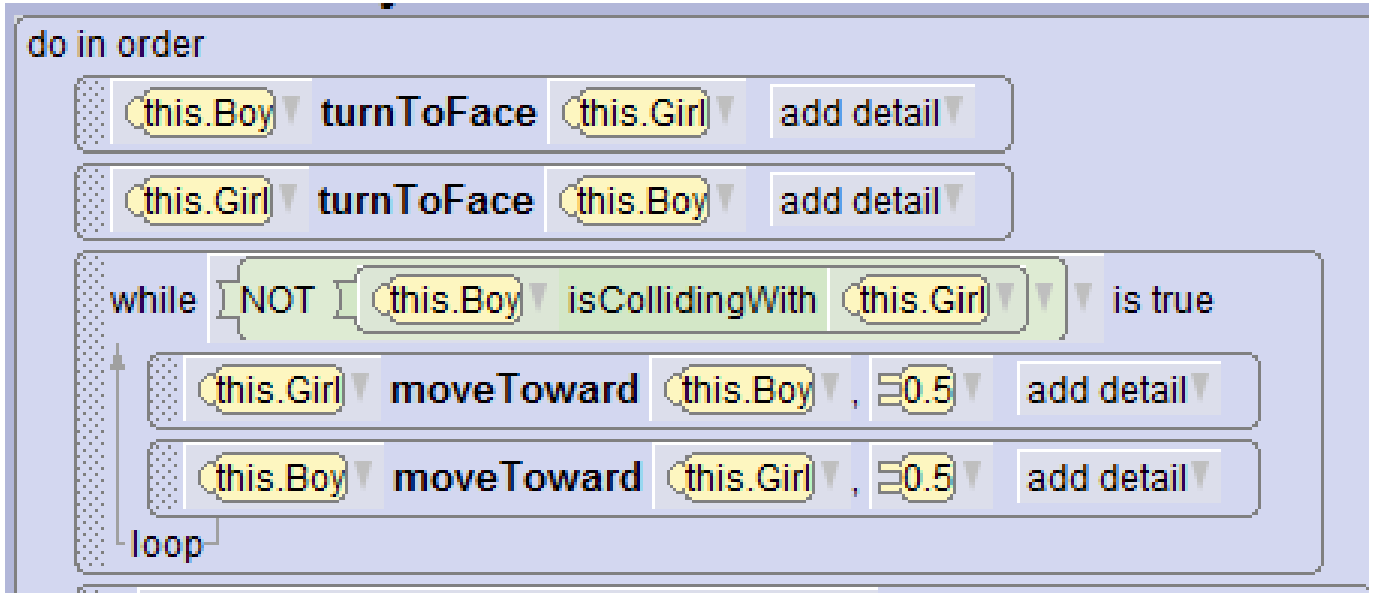
5.korak, umećemo negaciju:



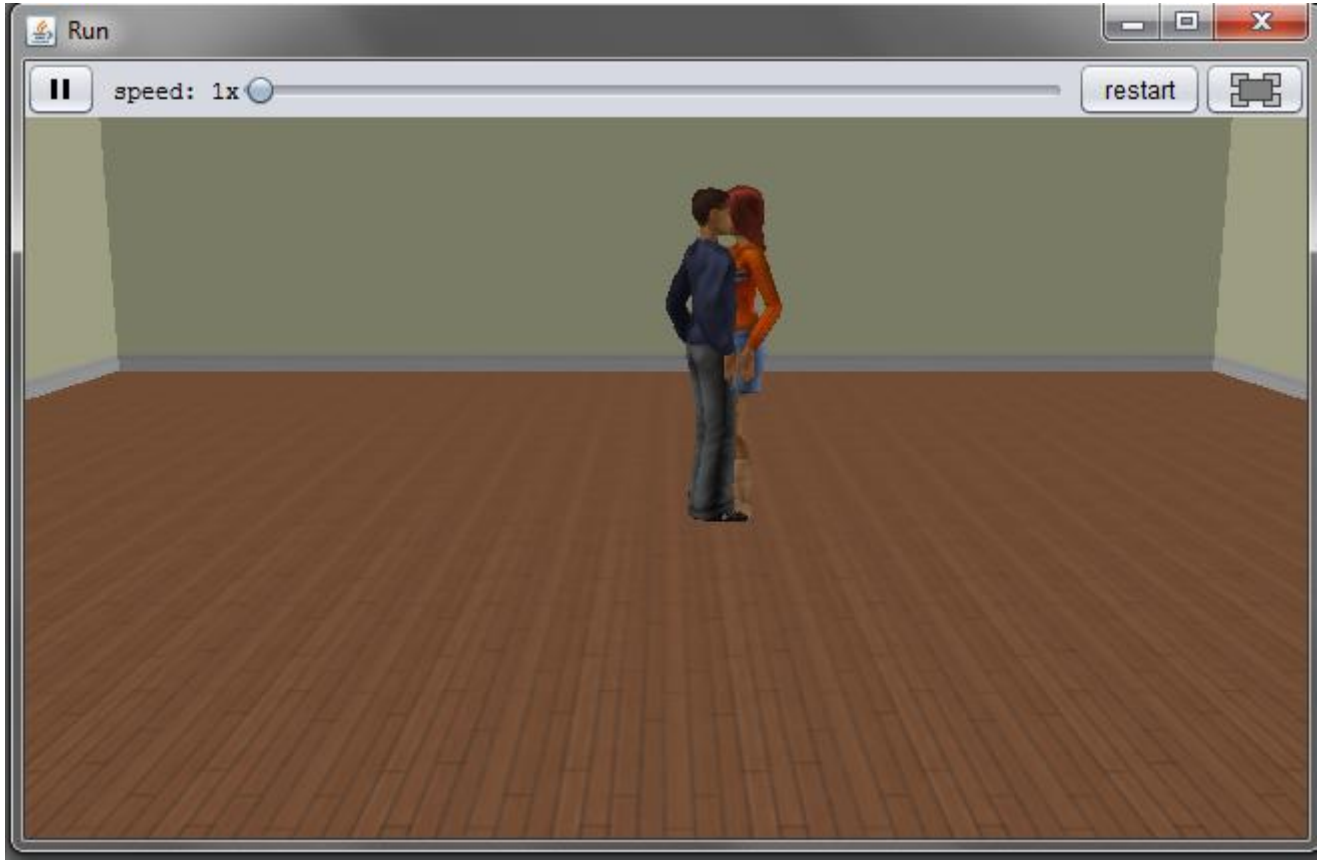
Sad pokrenimo jednog prema drugome dok se ne sretnu:



Do sada imamo program:



Pokrenimo ga…



Moramo malo to korigirati…

Moramo ograničiti njihovo kretanje

Kad i dođu jedan do drugoga, prekrit će se i to neće biti dobro. Oko djevojke ćemo zato napraviti jedan krug i mladić dolazi samo do ruba kruga.

Krug (torus)

napravimo

nevidljivim.

(Opacity=0)

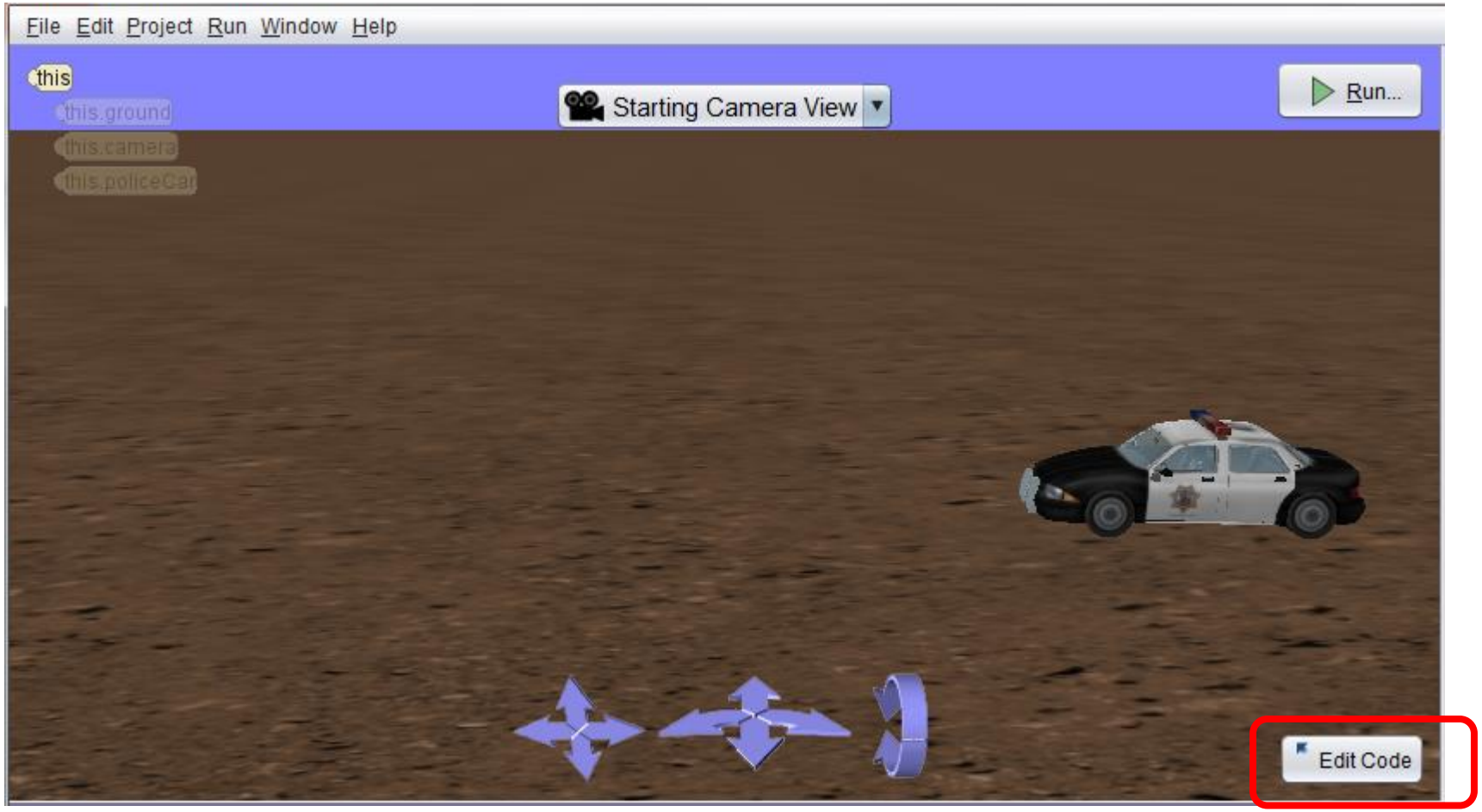
Krug će se kretati zajedno s djevojkom.


do in order
this.Boy turnToFace this.Girl add detail
this.Girl turnToFace this.Boy add detail
while NOT this.Boy isCollidingWith this.torus is true
do together
this.torus moveToward this.Boy , 0.25 add detail
this.Girl moveToward this.Boy , 0.25 add detail
this.Boy moveToward this.Girl , 0.5 add detail
loop

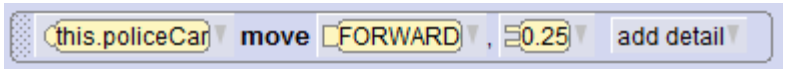
Kad se sretnu, popričaju:

Auto vozi

Organizirajte scenu i postavite auto



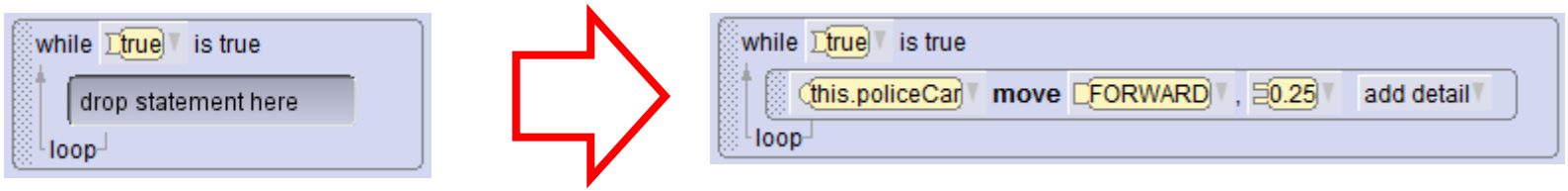
Napišimo kod za kretanje auta



zadovoljni?

Pokrenimo auto i krećimo ga ulijevo polako, na dulje vrijeme

Reći ćemo: **DOK (while)** je nešto istinito, kreći auto:

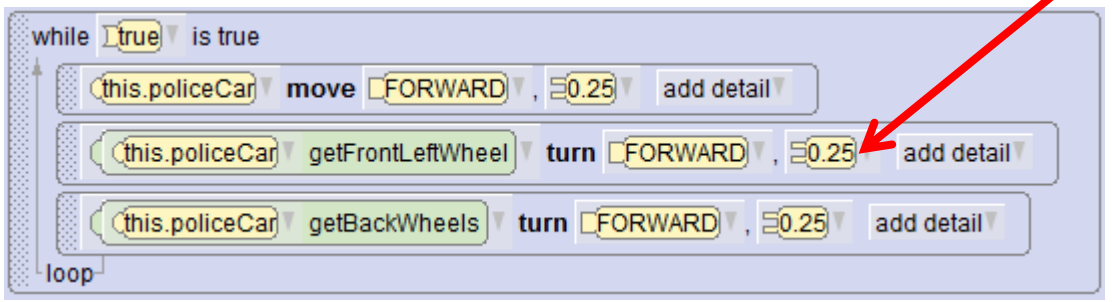


Ako ne želimo vožnju zauvijek, možemo koristiti nevidljivi cilindar kao prepreku – važnost postavljanja scene!

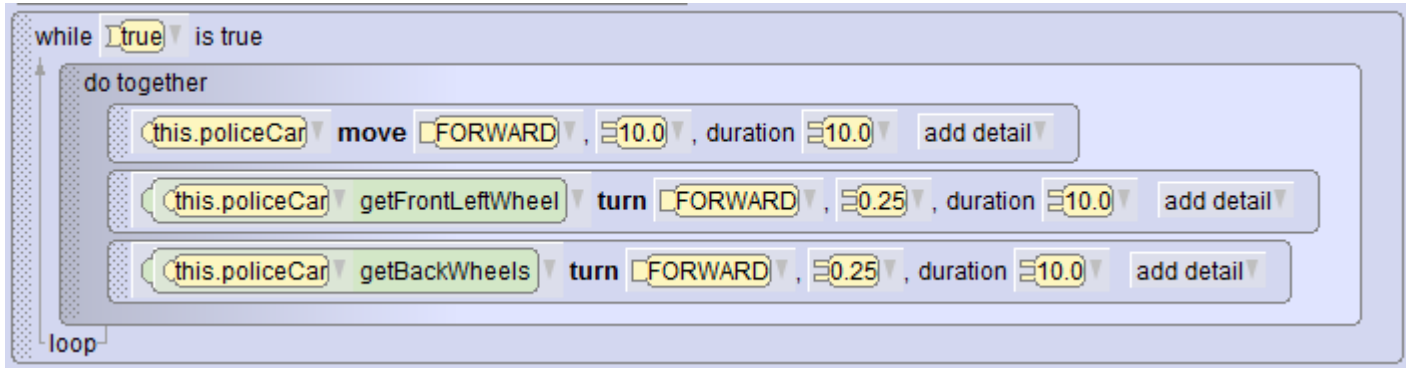
1. Postavimo cilindar
2. Auto dovedemo do njega
3. Provjerimo kamerama
4. Auto maknemo unazad za 10
5. Cilindar napravimo nevidljivim!

Mogli bi pokrenuti i kotače!

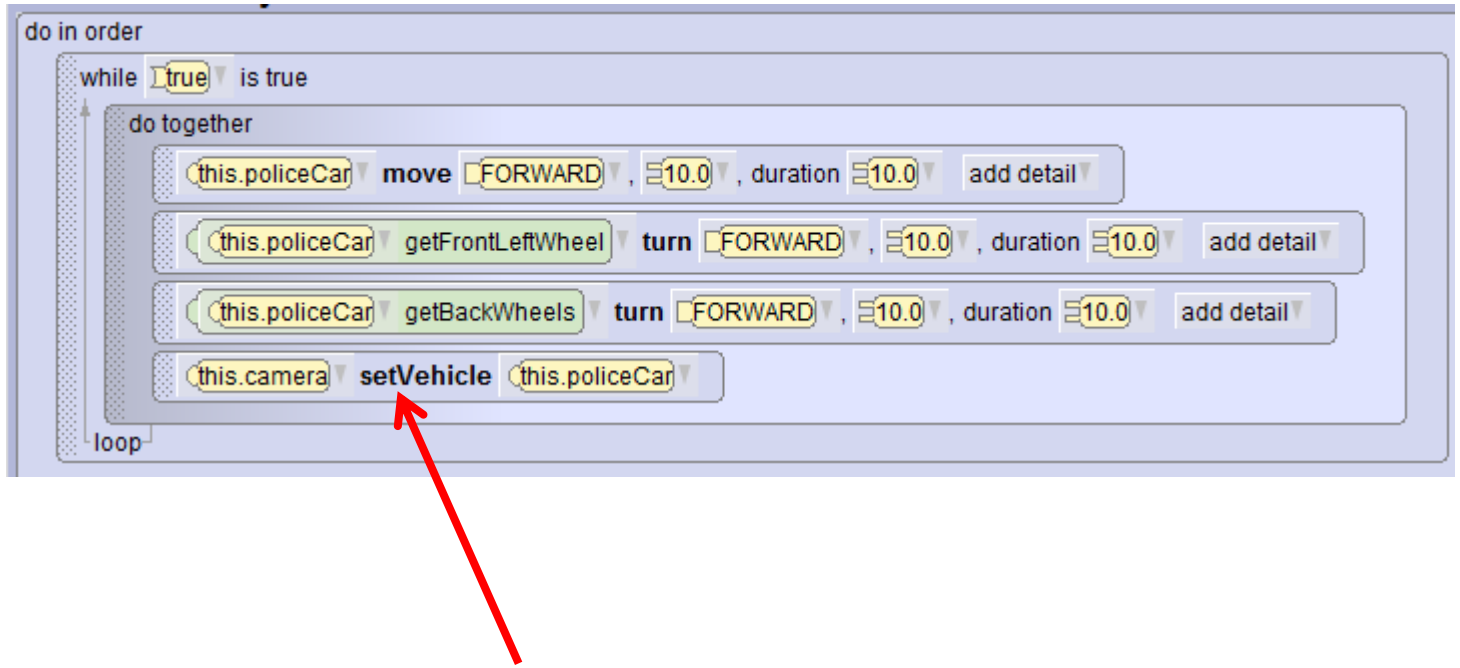
broj rotacija kotača



Nekako nisu sinhronizirani (prvo se miče auto, zatim prvi kotači, pa zadnji!) pokreti, to ćemo popraviti tako da svi zajedno u isto vrijeme krenu:



Pokrenite kameru da snima i prati kretanje auta:



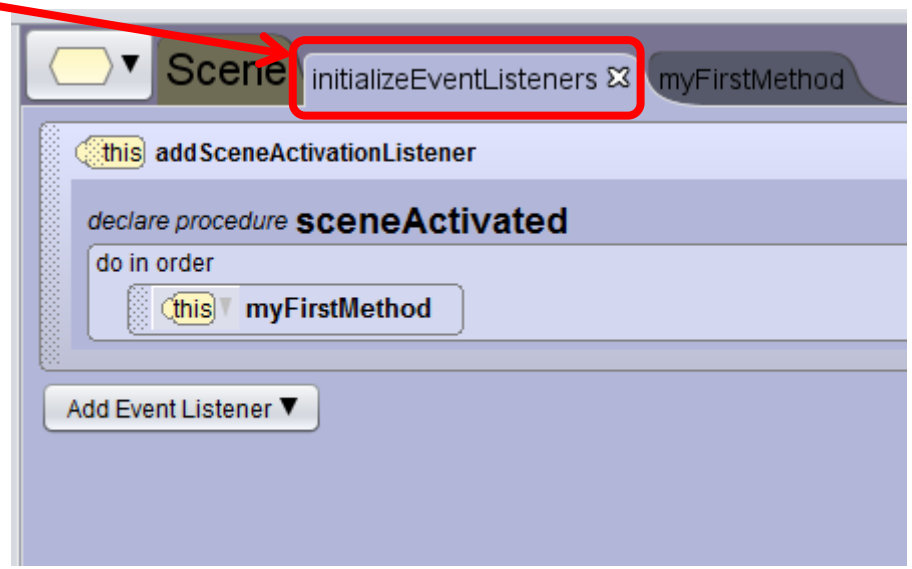
prati kretanje auta („podloga se miče”)

Riba pliva

Organizirajmo scenu

Napišimo kod za kretanje ribe kursorskim strelicama

Program „osluškuje”
pritisak tipkovnice



Event Listeners

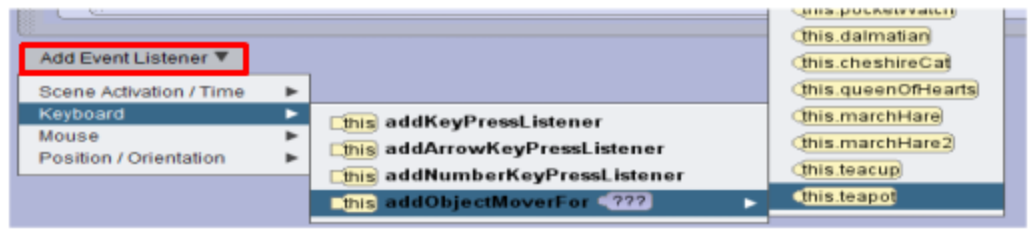
- Event listeners are procedures in the Scene class that listen for keyboard input while the animation is running.
- Keyboard keys can be programmed to:
  - Move an object up or down when certain keys are pressed.
  - Move an object forward, backward, left, and right using the arrow keys.
  - Make an object perform an action, such as speak or disappear.

Types of Event Listeners

- There are four types of event listeners available in Alice 3:
  - Scene Activation/Time
  - Keyboard
  - Mouse
  - Position/Orientation

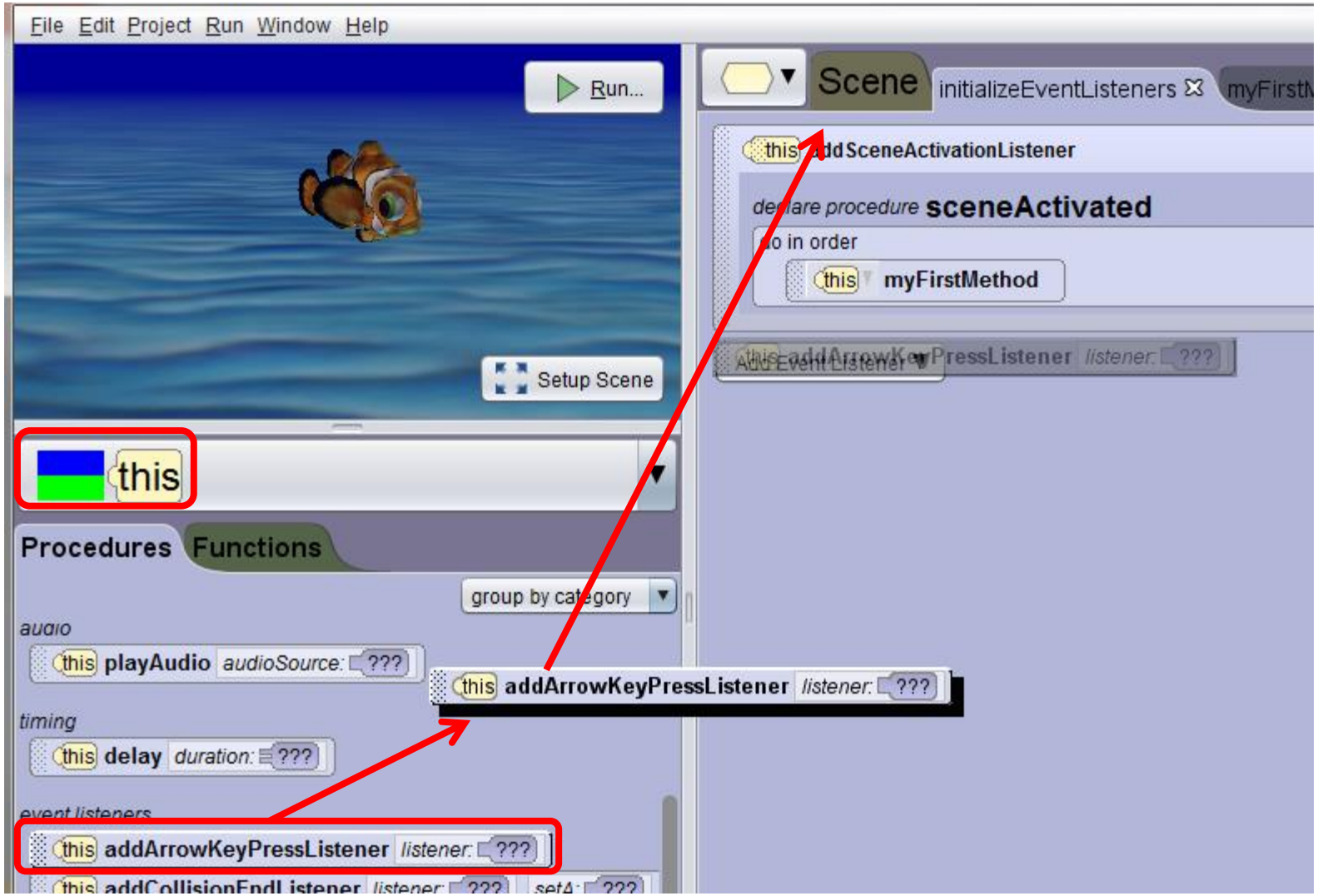
Steps to Move Objects Using Arrow Keys

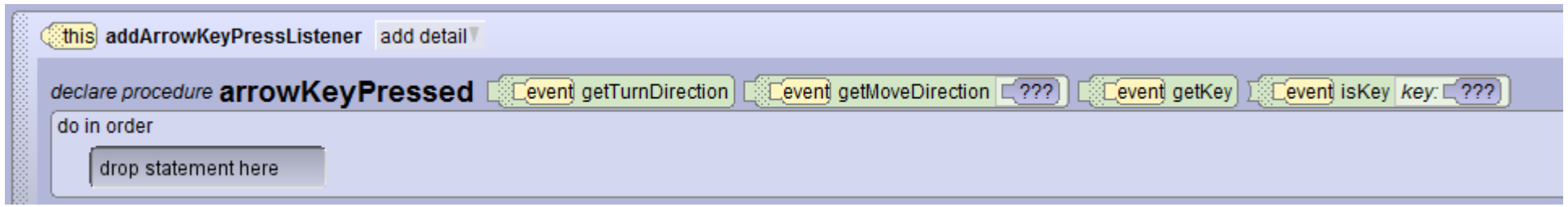
- Select the Add Event Listener drop-down menu.
- Select Keyboard.
- Select addObjectMoverFor.
- Select the entity, or object, to control.



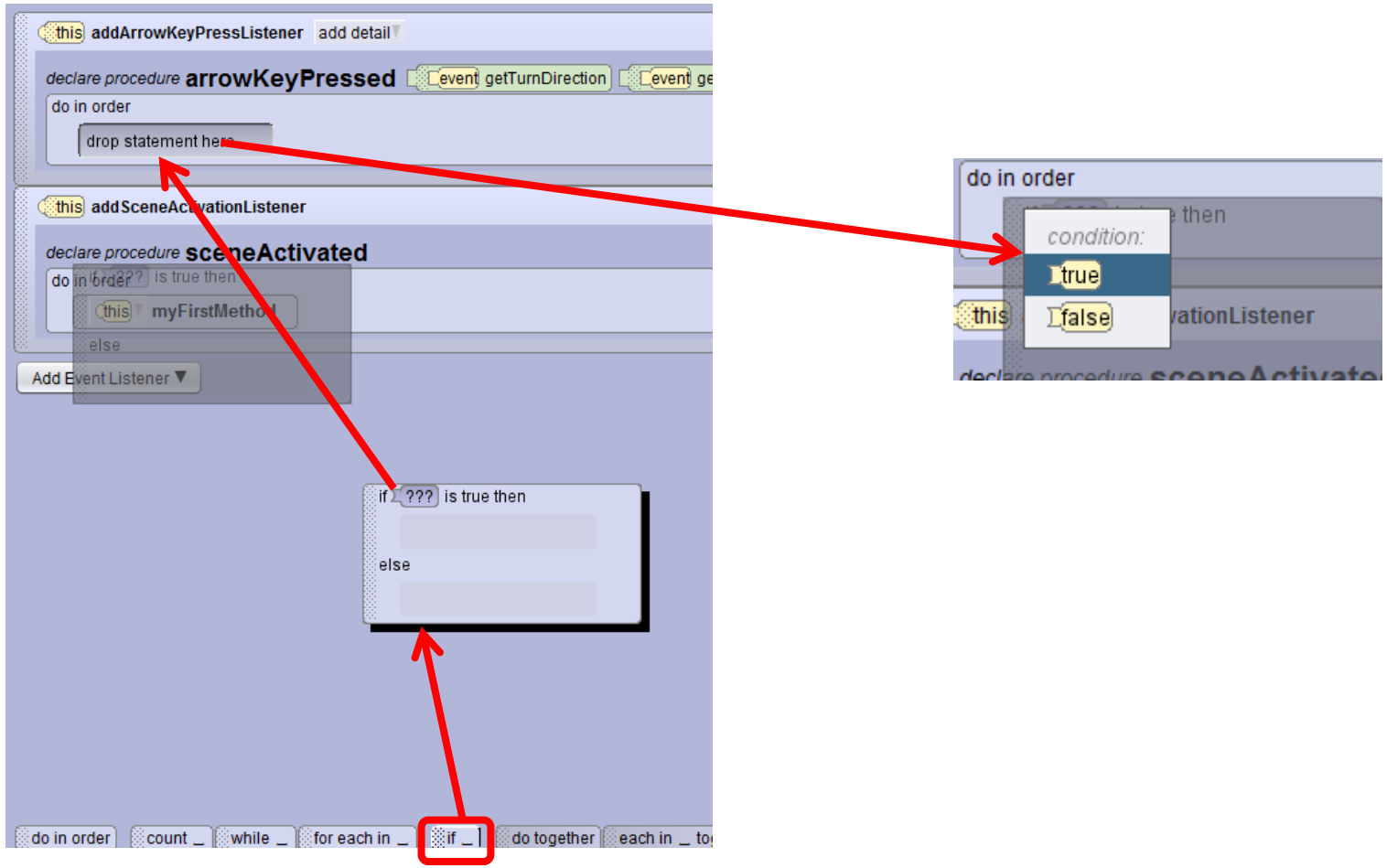
- Creates the following line of code

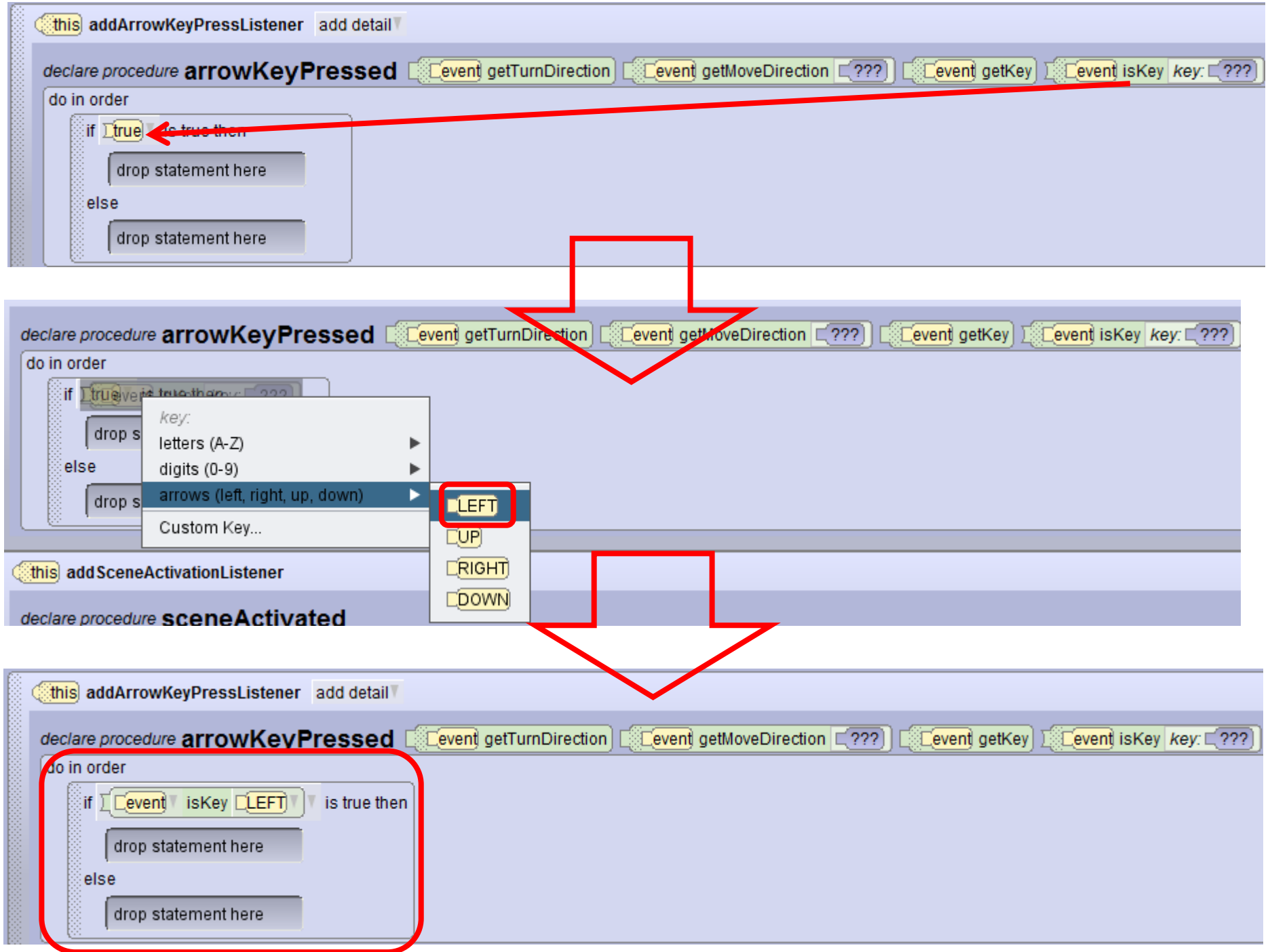
Umećemo addArrowKeyPressListener proceduru




this addArrowKeyPressListener add detail
declare procedure arrowKeyPressed event getTurnDirection event getMoveDirection ??? event getKey event isKey key: ???
do in order
drop statement here

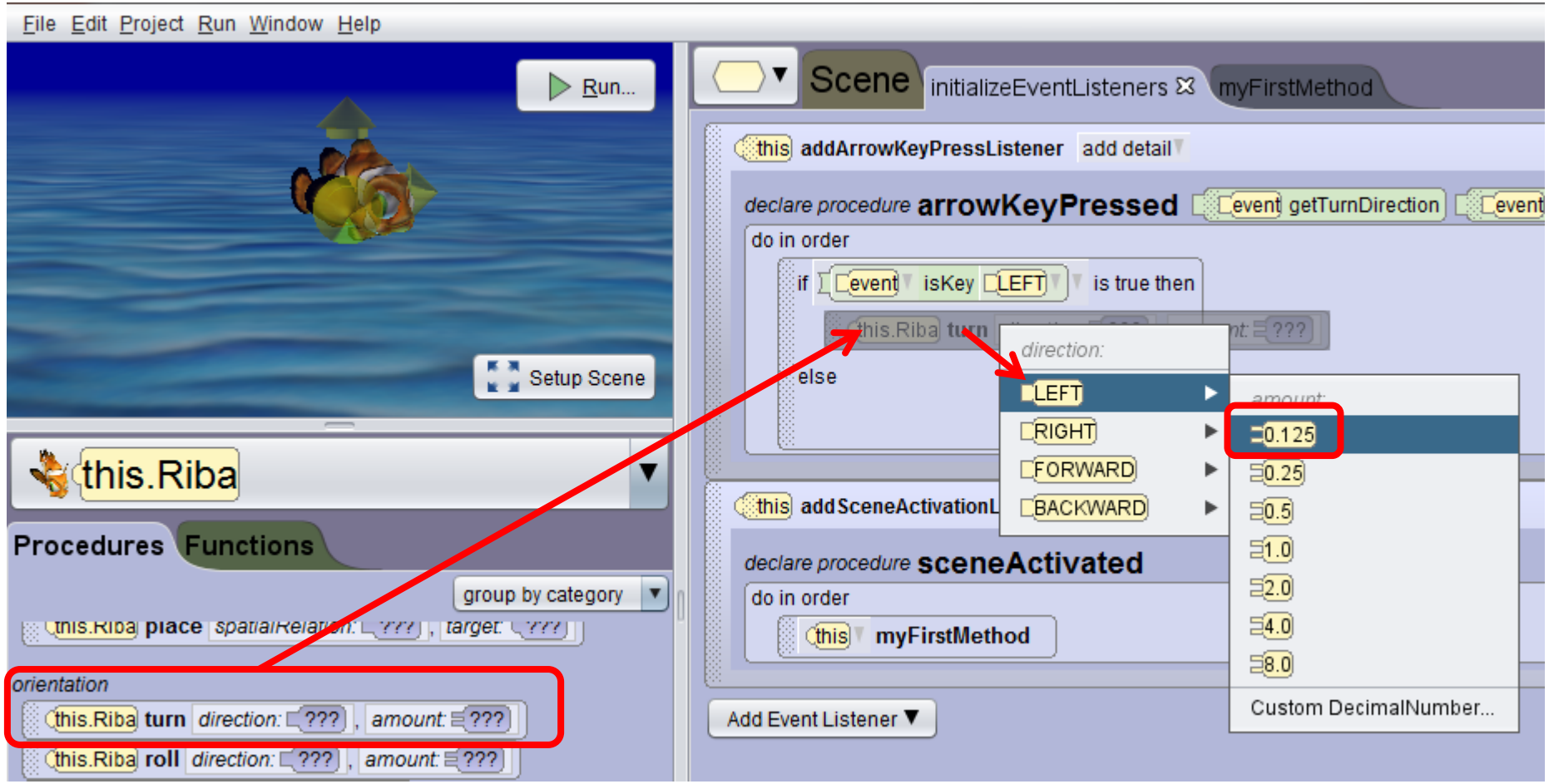
Sada koristimo naredbu IF…THEN sa dna ekrana



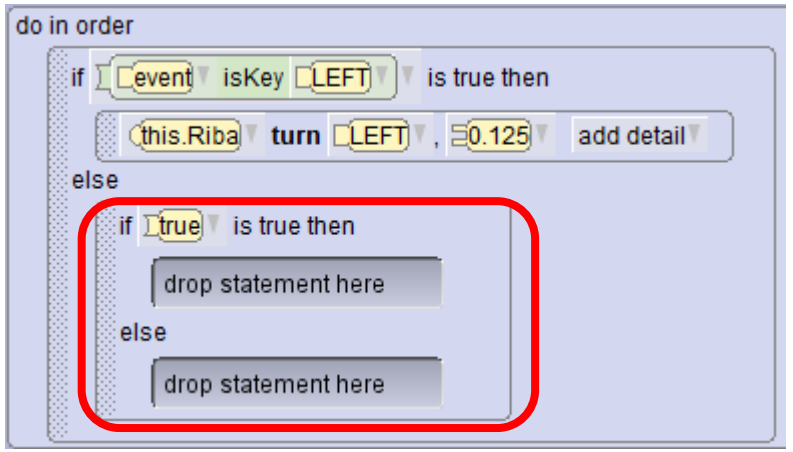

this addArrowKeyPressListener add detail
declare procedure arrowKeyPressed event getTurnDirection event getMoveDirection ??? event getKey event isKey key: ???
do in order
if true is true then
drop statement here
else
drop statement here
declare procedure arrowKeyPressed event getTurnDirection event getMoveDirection ??? event getKey event isKey key: ???
do in order
if true
drop s
else
drop s
key:
letters (A-Z)
digits (0-9)
arrows (left, right, up, down)
Custom Key...
LEFT
UP
RIGHT
DOWN
this addSceneActivationListener
declare procedure sceneActivated
this addArrowKeyPressListener add detail
declare procedure arrowKeyPressed event getTurnDirection event getMoveDirection ??? event getKey event isKey key: ???
do in order
if event isKey LEFT is true then
drop statement here
else
drop statement here

Odaberemo ribu

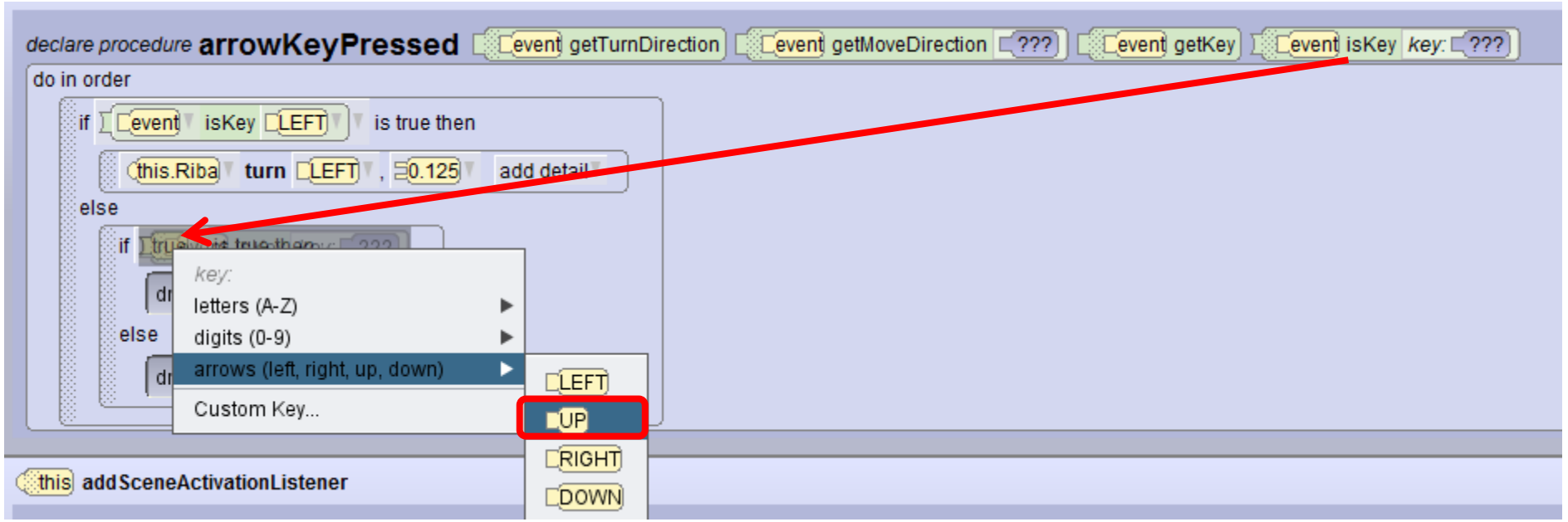
Odaberemo radnju okretanja (turn) ulijevo za 0.125 jedinica



Dodamo novu **if** naredbu iza **ELSE**

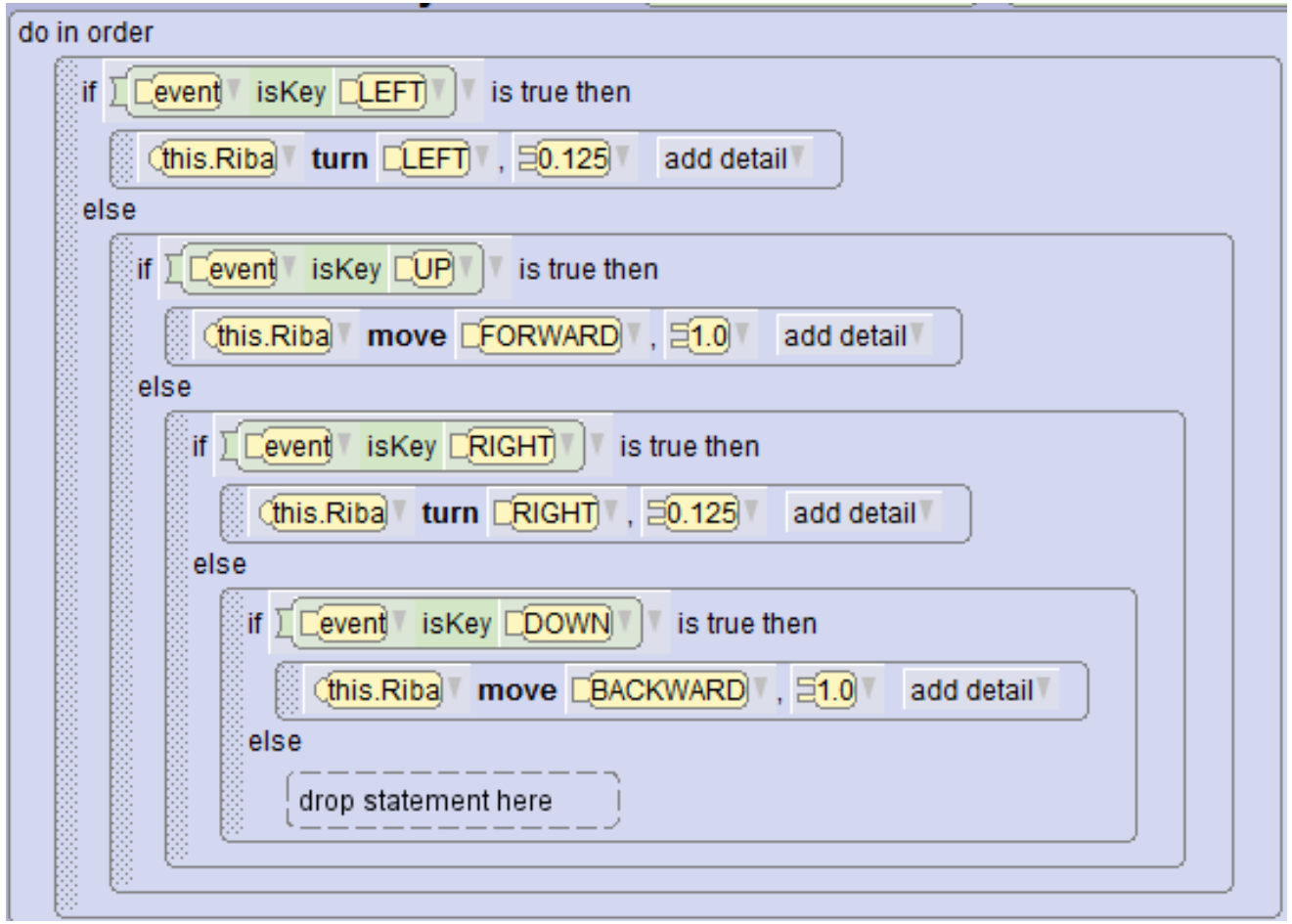


Ako pritisnemo na tipku GORE (UP)

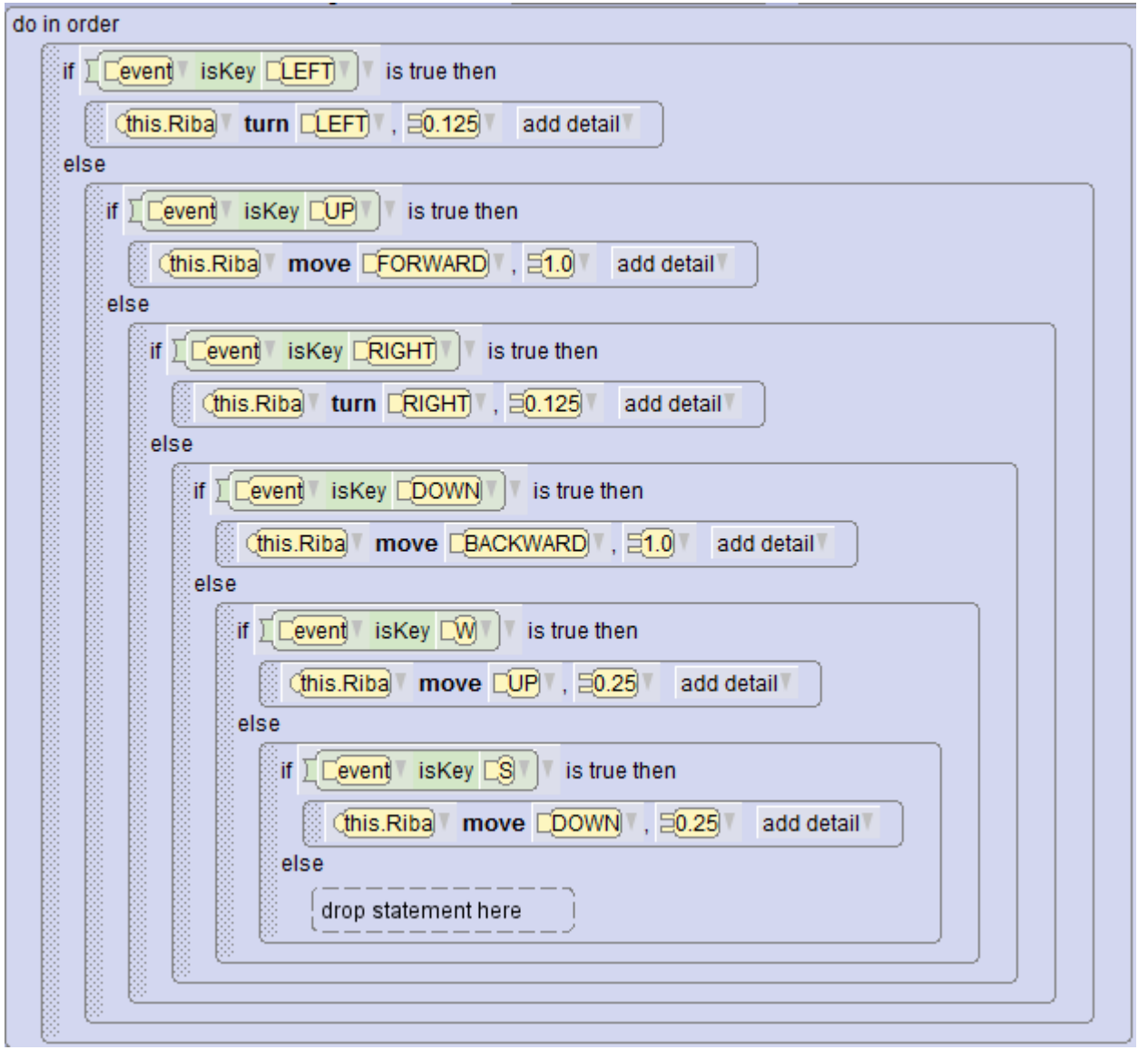


Definiramo pritisak na tipku GORE (up) i **pomak** naprijed za 1

Dovršite:

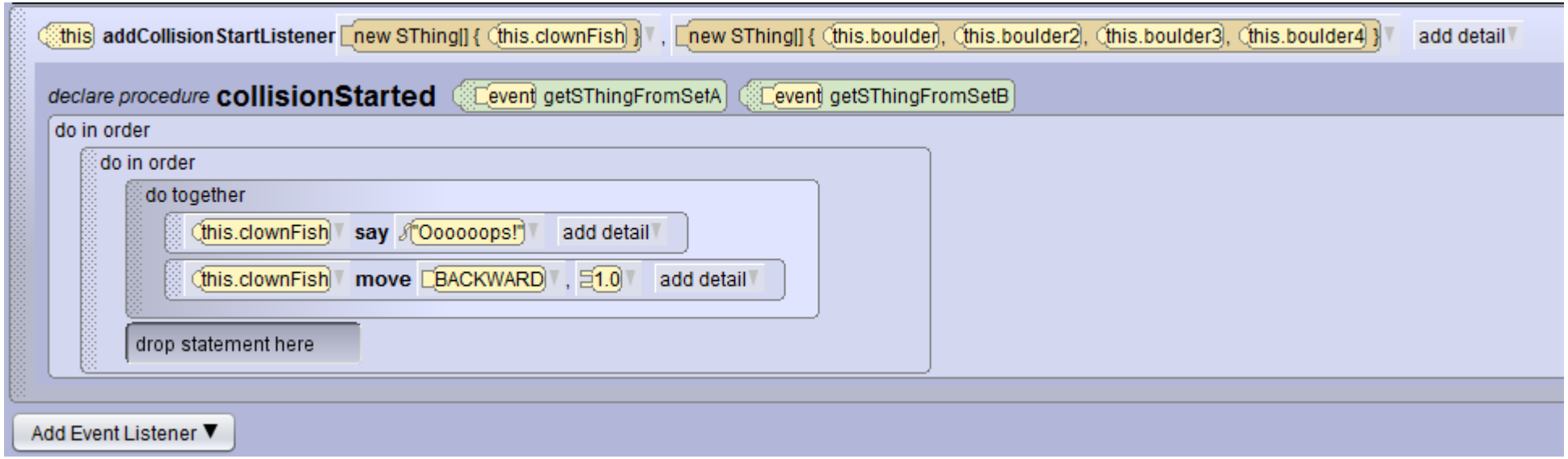

do in order
if event isKey LEFT is true then
this.Riba turn LEFT , 0.125 add detail
else
if event isKey UP is true then
this.Riba move FORWARD , 1.0 add detail
else
if event isKey RIGHT is true then
this.Riba turn RIGHT , 0.125 add detail
else
if event isKey DOWN is true then
this.Riba move BACKWARD , 1.0 add detail
else
drop statement here

Dodajte još pomicanje GORE-DOLJE slovima w i s




Starting Camera View

Postavite neke prepreke pa pomičite ribu između njih - pokušajte je i odbiti ako dođe do prepreke!



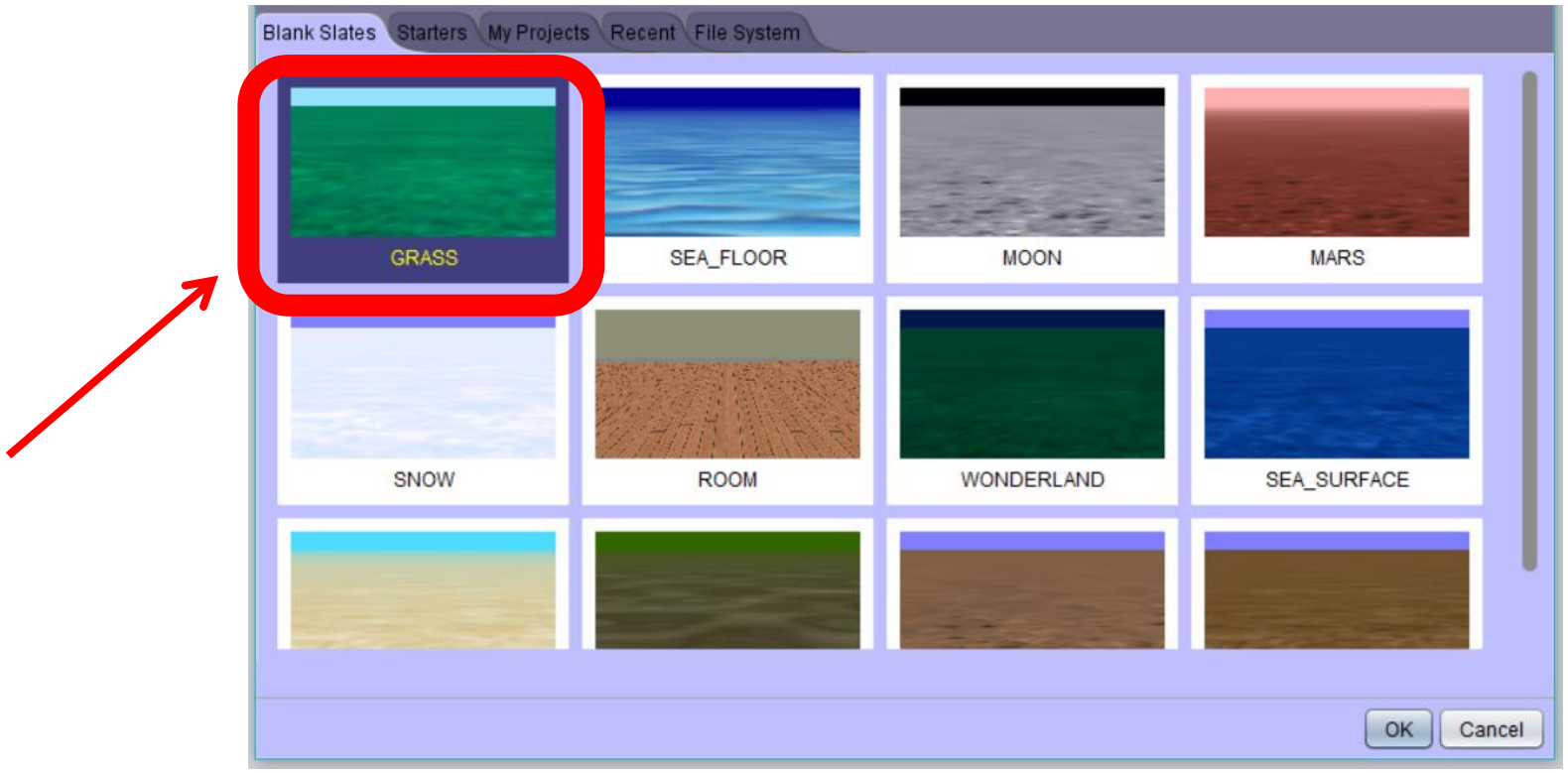
Izrada jednostavne igrice „Skupljanje gljiva”

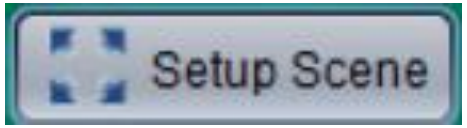
Skupljanje gljiva

1. Izradit ćemo igru s Alisom koja skuplja gljive
2. Kursorskim strelicama kontroliramo Alisu
3. Dvije gljive su na livadi
4. Kada Alisa dođe do gljive, gljiva nestane, Alisa promijeni veličinu
5. Na kraju se vraća na početnu poziciju

Pokrenimo Alice 3

Za pozadinu odabiremo travu:



Idemo na obradu scene → 

Izradite scenu sa sljedećim likovima:

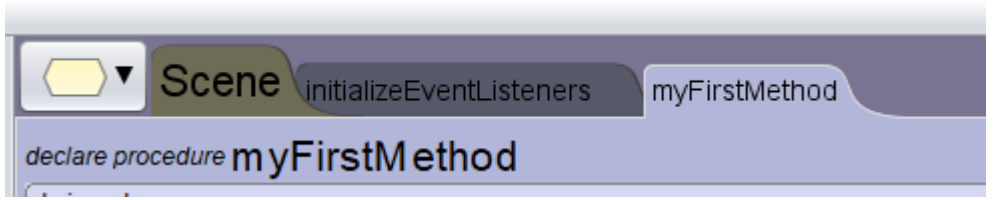
PAZITE: <u>SVAKA GLJIVA MORA IMATI DRUGO IME!!!</u>

Nakon što namjestimo likove na sceni, oko svakog lika napravimo nevidljivi torus.

Prvo kažimo da se Alisa neprestano miče prema naprijed (0.25 i brzinom 0.25) sve dok ne pokupi sve gljive.
Gljiva je „pokupljena” kad joj je Opacity jednak nuli.
Na kraju se Alisa vraća u početnu poziciju
Kad pokupi Gljivu1, smanji se 4 puta, a kad pokupi Gljivu2, poveća se 4 puta.

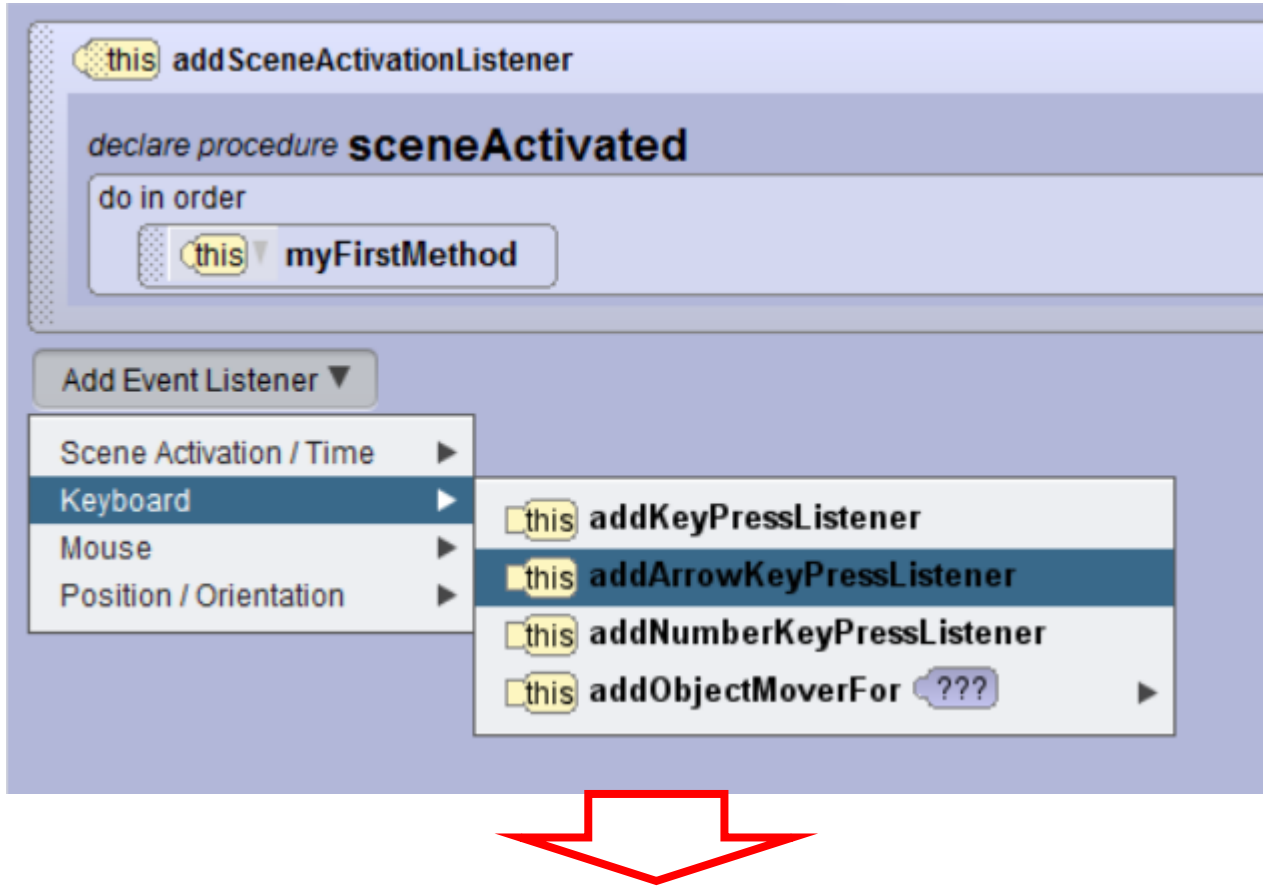
Napravimo i tri nevidljiva torusa: oko gljiva i oko Alise.

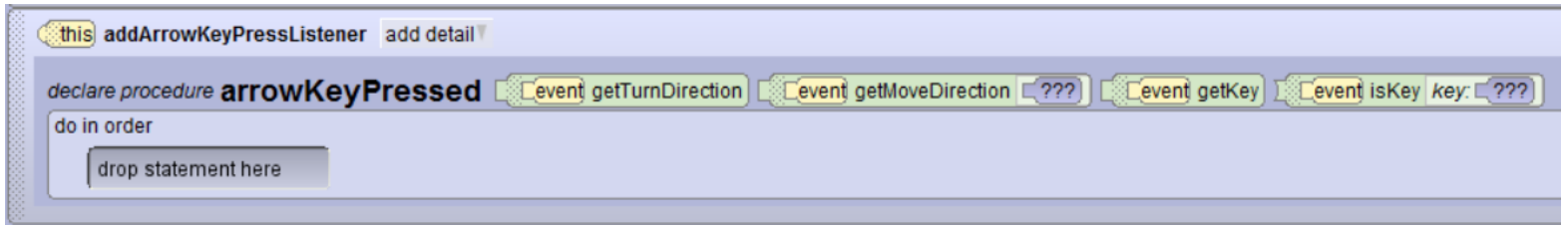
Početni program



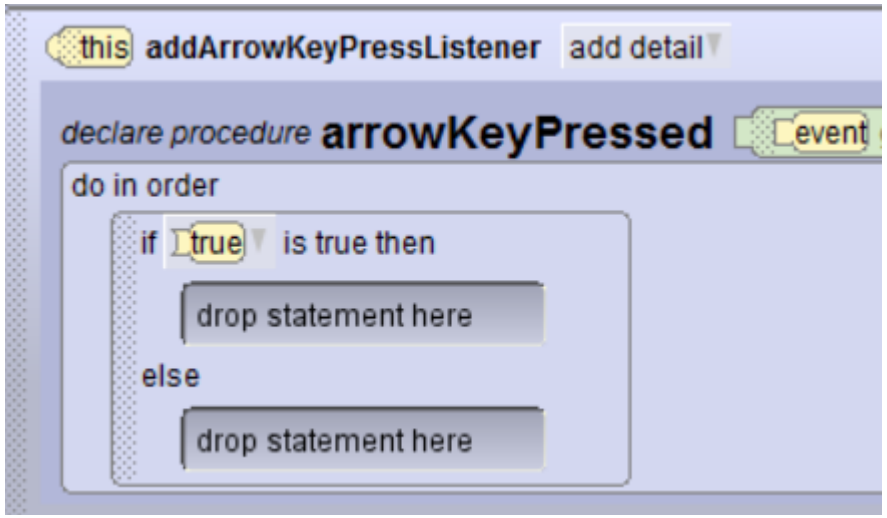
Kako to napraviti?
Krenimo koracima:

Pokrećimo je tipkama lijevo i desno:

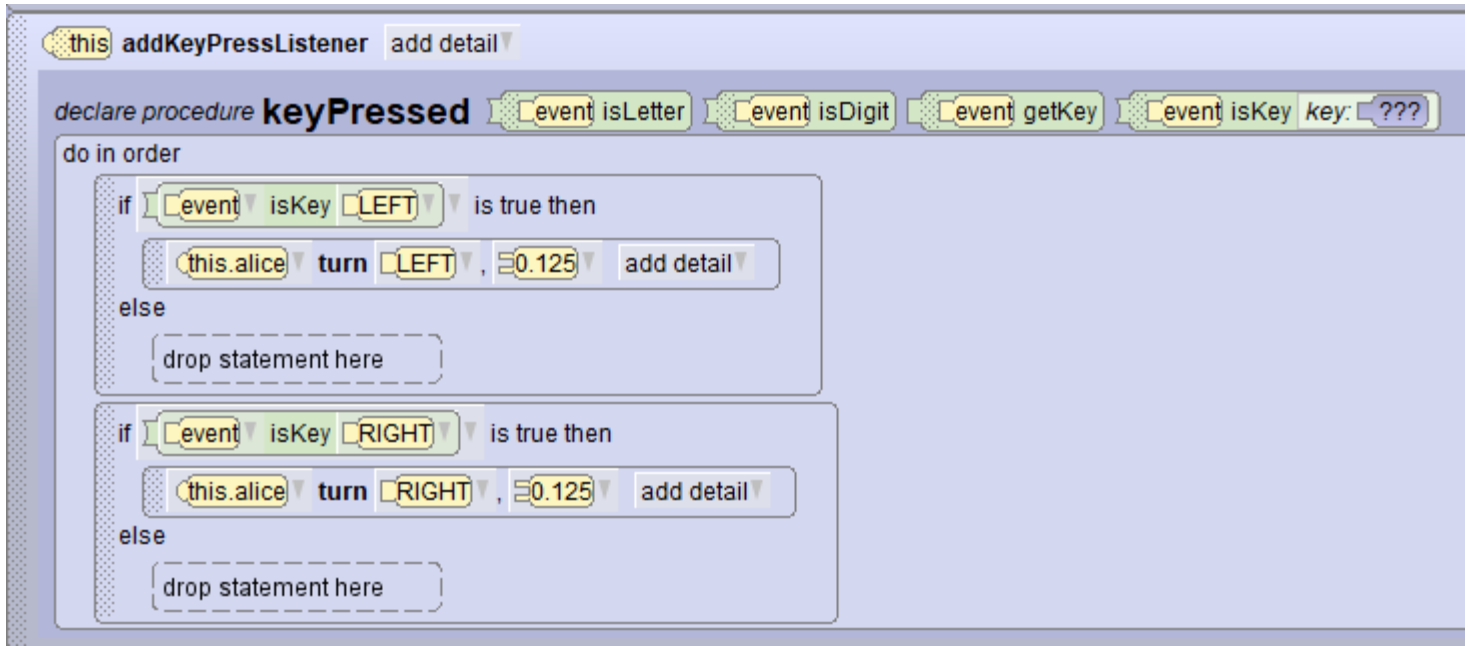




dodajemo if_



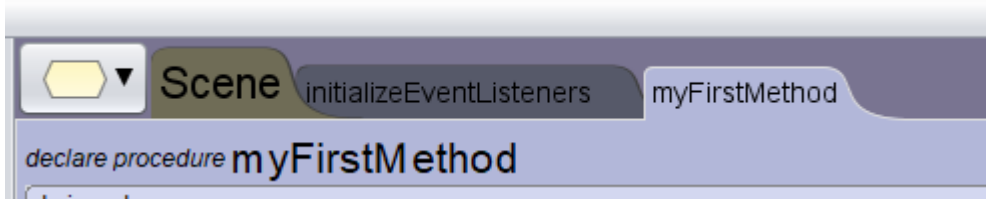
krećemo se samo lijevo - desno

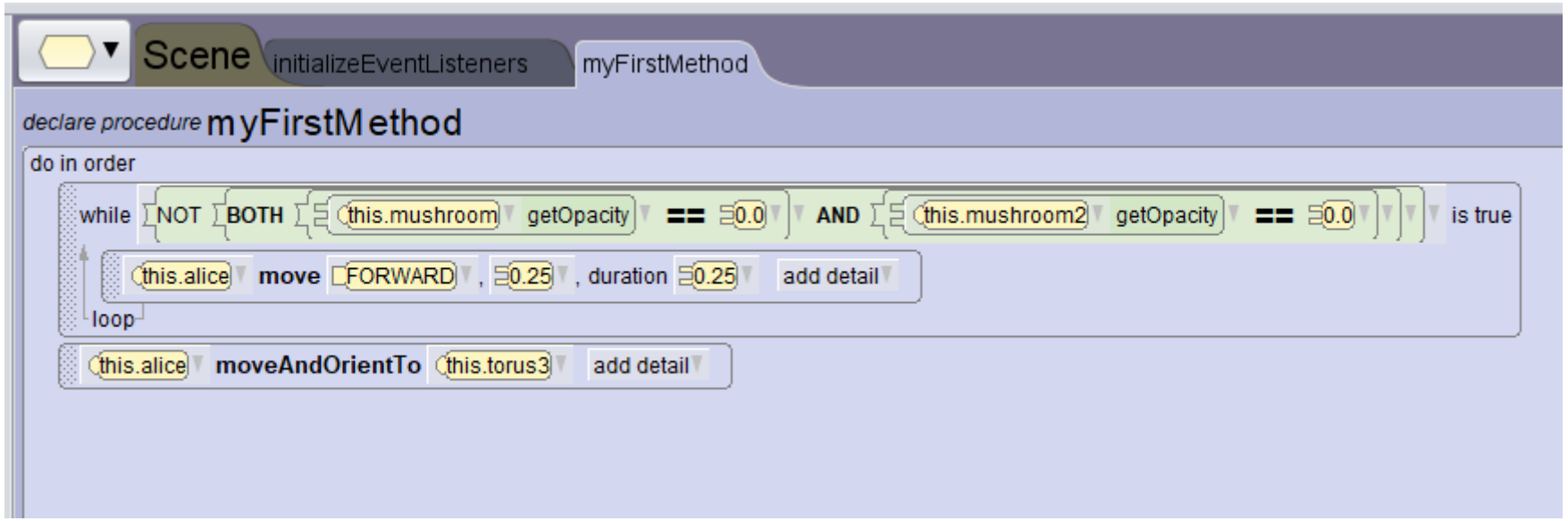


Alisa se miče sve dok se ne zadovolji uvjet

(postane istinit) da su obje gljive nevidljive!

Početni program





Sada moramo opisati što će se dogoditi kad Alisa dođe do gljive. Gljiva bi trebala nestati. (vidljivost „Opacity” joj se smanji na nulu)

Kad se dodirne s nekim od torusa, naraste ili se smanji i gljiva postane nevidljiva

Alisa se vraća na početak kad su sve gljive ubrane:

Čekamo dok vidljivost svih gljiva ne postane nula: