

3. NALOGA-IGRICA

1. Ideja iz rezervacije seminarske naloge:

Na zaslonu se bo uporabniku izrisal nekakšen loparček in na njem nekakšna žogica. Nad žogico in loparjem bomo imeli postavljene nekakšne kocke, ki jih bo moral uporabnik s pomočjo žogice podreti. Žogica se bo od kock odbijala nazaj proti dnu, uporabnik pa jo bo moral uloviti s pomočjo loparja. Če je ne bo ulovil, bo igre konec in bo izpadel kot poraženec. Uporabnik bo igro zmagal takrat, ko bo uspešno podrl vse kocke, tako da žogice ne bo nikoli zgrešil.

2. Besedilo naloge:

Ko uporabnik odpre igrico ŽogaLopar, se mu na zaslonu izriše 22 črnih pravokotnikov, lopar, žogica in tla. Naloga uporabnika je, da mora s pomočjo premikanja loparja ujeti žogico in kar se da najbolje usmeriti proti pravokotnikom. Lopar premika s pomočjo drsenja miške levo in desno.

Na zaslonu ima polje v katerem se seštevajo točke, ki jih je dosegel. Vsak posamezen uničen pravokotnik mu prinese eno točko (vsi pravokotniki so ovrednoteni enako). Vseh možnih točk je 22. Če uporabnik zgreši žogico in žogica pade na tla se igra konča in v oblaku se izpiše naslednje obvestilo: Nisi ujel žogice! ☹ Poskusi znova!.

Igra je sestavljena iz dveh stopenj. Stopnji se razlikujeta v postavitvi pravokotnikov in hitrosti premikanja žogice.

Ko uporabnik uspešno zaključi prvo stopnjo, lahko nadaljuje drugo stopnjo. Pri tem se upoštevajo točke iz prve stopnje.

Igro je uporabnik uspešno zaključil, ko žogica uniči vse pravokotnike iz prve in druge stopnje, takrat se mu v oblaku izpiše naslednje obvestilo: Čestitke! Zadel si vse pravokotnike! 😊 Če uporabnik zgreši žogico v drugi stopnji lahko nadaljuje samo z drugo stopnjo.

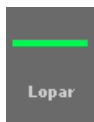
3. Prikaz igre v živo:

Na zaslonu se uporabniku izrišejo naslednji elementi igrice:

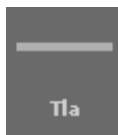
- žogica



- lopar



- tla

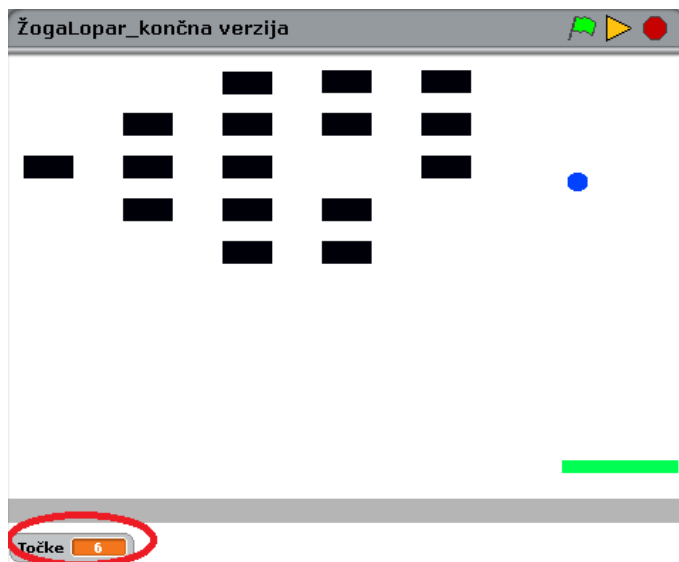


- 22 pravokotnikov črne barve, ki se jih mora žogica dotakniti, da se uničijo



1. Primer »igre v živo«:

Na spodnji zaslonski sliki je prikaz igrice, ko je žogica v prvi stopnji zadela in uničila šest pravokotnikov in se približuje loparju. V okvirčku, ki je del prikaznega okna, vidimo polje kjer se seštevajo točke (1 točka predstavlja en uničen pravokotnik, ki ga je zadela žogica). Kot sem že omenila je žogica uničila natanko šest pravokotnikov, zato je uporabnik dosegel 6 točk). Na sliki lahko vidimo vse zgoraj naštet elemente.



2. Primer »igre v živo«:

V tem primeru je uporabnik zgrešil žogico in igra se je končala. V oblaku se je izpisalo naslednje obvestilo: Nisi ujel žogice! ☹️ Poskusi znova!. V okvirčku kjer se izpisujejo točke lahko vidimo koliko točk je dosegel uporabnik, oziroma koliko pravokotnikov je uničil s pomočjo žogice.

Uničil je sedem pravokotnikov in nato je zgrešil žogico.



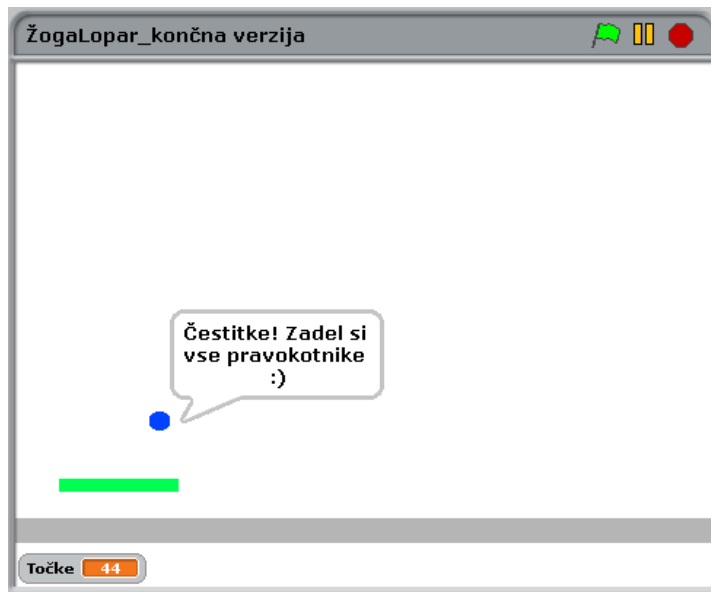
3.Primer »igre v živo«:

Na spodnji zaslonski sliki je prikaz igrice v trenutku, ko je uporabnik v prvi stopnji uspešno uničil vse pravokotnike in dosegel vse možne točke v prvi stopnji. Če želi nadaljevati z drugo stopnjo mora pritisniti SPACE BAR.



4.Primer »igre v živo«:

Na spodnji zaslonski sliki je prikaz igrice v trenutku, ko je uporabnik v prvi in drugi stopnji uspešno uničil vse pravokotnike in dosegel vse možne točke v obeh stopnjah(44 točk). Na zaslonu se mu je izpisalo naslednje obvestilo : Čestitke! Zadel si vse pravokotnike ☺.



4. Postopek pisanja programa in opis ideje postopkov:

Opis postopka pisanja programa se bom lotila na ta način, da bom opisala kodo in idejo pri vsakem objektu posebjaj.

V prejšnem razdelku sem navedla vse elemente/objekte, ki sestavljajo igrice.

- **Pravokotnik**

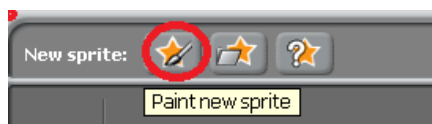


Pravokotniki sestavljajo pomemben del igrice. Naloga uporabnika je, da s pomočjo loparja in primerne ciljanja zadane čim več pravokotnikov, pri tem pa mora paziti, da mu žogica ne pade na tla. Uporabnik uspešno zaključi igrico takrat, ko uniči vse pravokotnike.

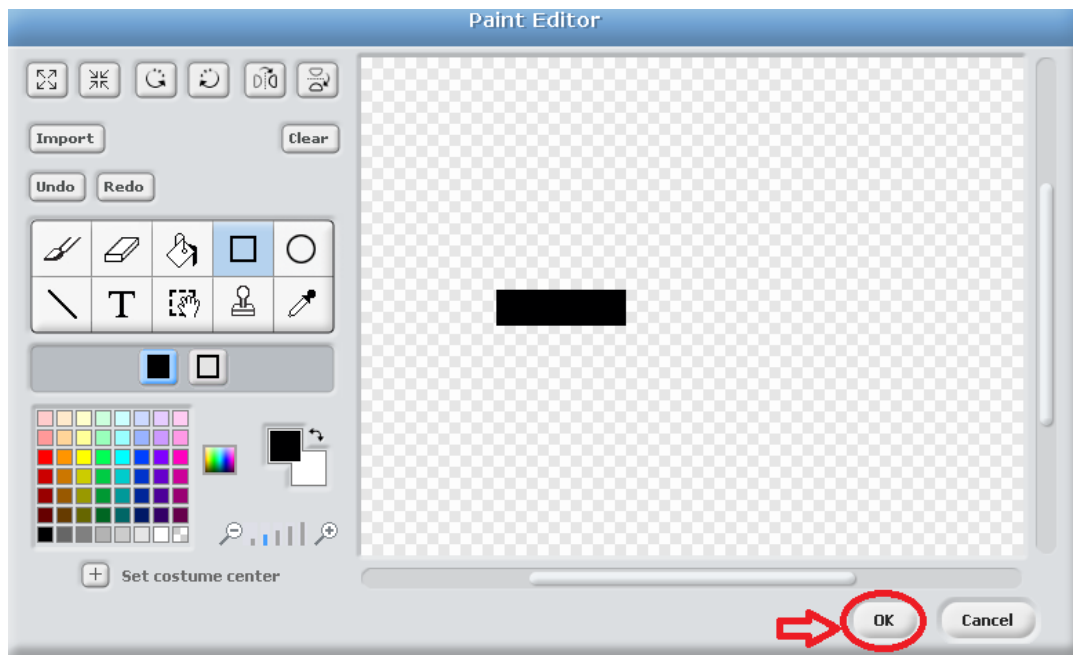
Sedaj pa bom navedla kako sem ustvarila en pravokotnik in nato še preostale. Pojasnila bom kodo, ki sem jo napisala.

Najprej sem s pomočjo risanja ustvarila pravokotnik na naslednji način:

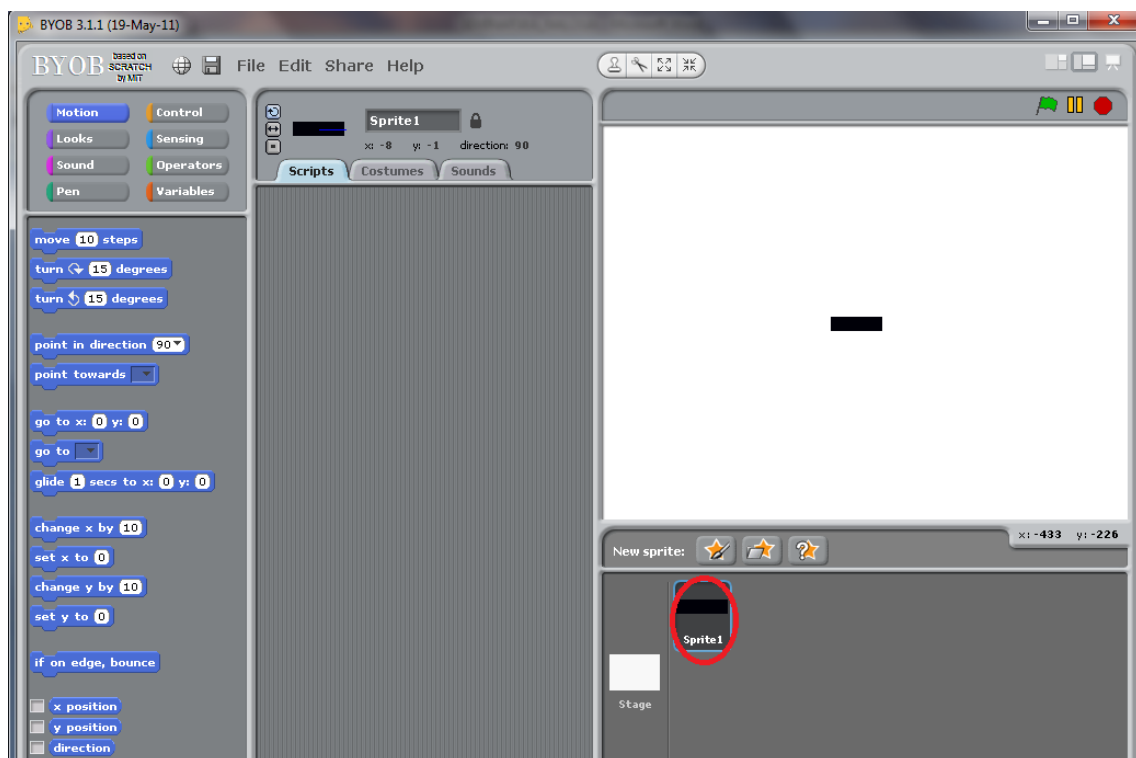
V vrstici ki ima na izbiro rubrike za ustvarjanje novega objekta, sem si izbrala možnost Paint new sprite.



Nato se mi je odprlo spodnje okno, kjer sem ustvarila pravokotnik.



Ko sem pritisnila gumb OK, se mi je objekt prikazal v oknu, ki je namenjen za objekte in na belem zaslону, kjer bomo kasneje igrali igrico.

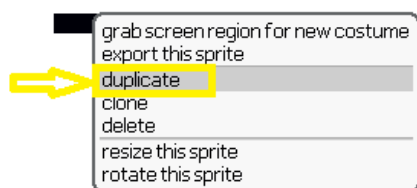


Ker sem želela da ima objekt ime sem objekt Sprite1 preimenovala v Pravokotnik.





Ker je smisel da imamo večje število pravokotnikov, sem iz nastalega pravkotnika ustvarila kopije in jih poljubno zložila po zaslonu.

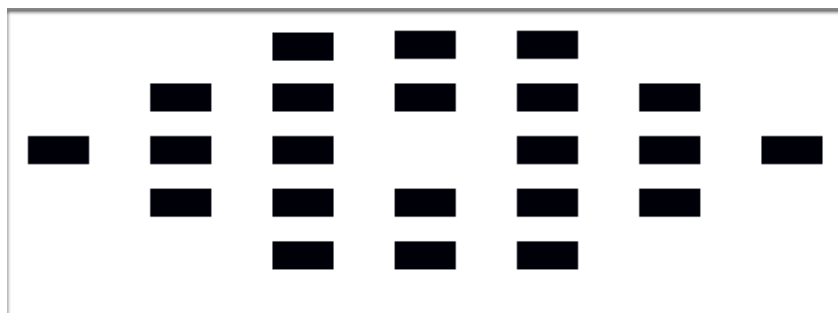


Pravokotnike sem na začetku poljubno razporedila po zaslonu. Ker sem hotela, da so smiselno razporejeni po zaslonu sem vsakemu zapisala ustrezne koordinate. Kordinate sem si najprej preračunala po peš poti.

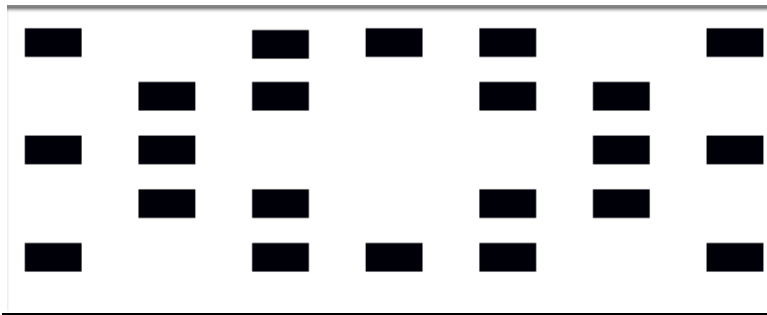
Postavitev pravokotnikov je v obeh stopnjah različno, zato sem v vsaki stopnji posebej nastavila koordinate.

Na spodnjih dveh zaslonskih slikah je prikaz postavitve pravokotnikov v obeh stopnjah.

1. Postavitev pravokotnikov v prvi stopnji:



2. Postavitev pravokotnikov v drugi stopnji:



Skupno sem ustvarila 22 pravokotnikov, vsakega sem preimenovala v pravokotnik in mu dodala zaporedno številko ustvarjenega pravokotnika.

Pri programu BYOB moramo za vsak objekt posebej zapisati kodo. Kodo ki sem jo zapisala pri pravokotniku bom obrazložila po korakih. Vsi pravokotniki imajo identično kodo, razlikujejo se le v postavitvi na zaslonu. Ker je 21 pravokotnikov kopija prvega, sem samo pri prvem napisala kodo in koda se je pri shranjevanju samodejno prekopirala v ostalih 21.

Kodo sem napisala po naslednjem premisleku:

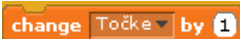
- Vsak pravokotnik ima ustrezne krajevne koordinate, ki določajo njegov položaj na zaslonu.
- Vsak pravokotnik je prikazan na zaslonu vse do trenutka, ko ga zadane žogica, takrat izgine iz zaslona. S pomočjo ukaza show, se pravokotnik prikaže na zaslonu. S pomočjo ukaza hide, pravokotnik izgine v trenutku, ko se ga dotakne žogica.
- Ko žogica uniči pravokotnik, se sproži zvok in v okvirčku, kjer so zabeležene točke, se točka poveča za 1.

Na spodnji zaslonski sliki je prikazana koda, ki jo bom ustrezno pokomentirala.



Funkcija ki je zapisana se začne izvajati v trenutku, ko začnemo igrati igrico, oziroma pritisnemo .

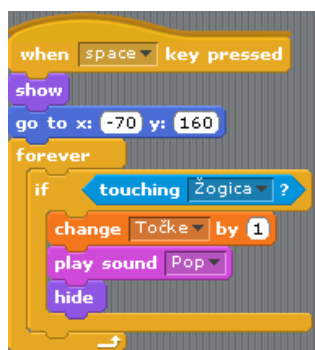
S pomočjo ukaza show, se na zaslonu prikaže pravokotnik. Vsakemu pravokotniku s pomočjo ukaza , določimo krajevne koordinate.

Če se pravokotnika dotakne žogica, se uporabniku poveča točka za ena , sproži se zvok Pop  in pravokotnik izgine (to dosežemo s pomočjo ukaza  hide).

Na zgornji zaslonski sliki vidimo zapisano kodo, ki jo imajo vsi pravokotniki, ki se nahajajo v prvi stopnji. V drugi stopnji imajo vsi liki identično kodo, razlikujejo se samo v krajevnih koordinatah.

Na začetku ne uporabimo ukaza  kot v prvi stopnji, ampak uporabimo ukaz , ker smo nadaljevali igrico v drugo stopnjo.

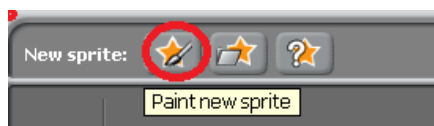
Na spodnji zaslonski sliki je prikaz kode za pravokotnik, ki se nahaja v drugi stopnji igrice.



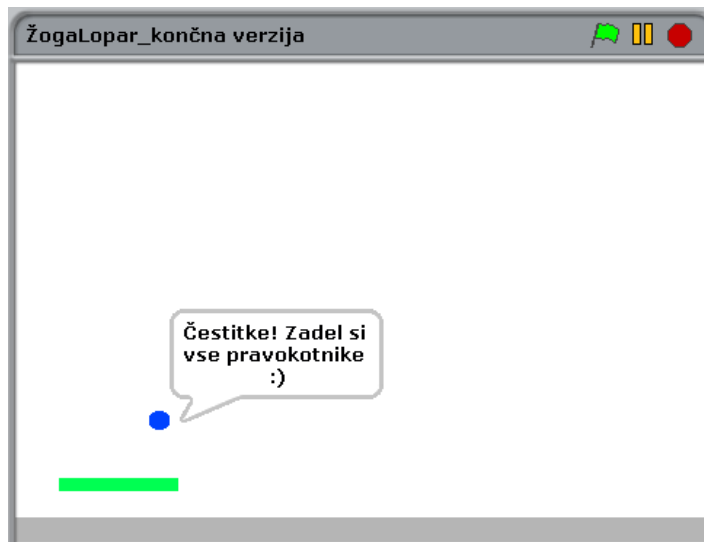
• Tla

Tla imajo pri igrici vlogo meje, kamor žogica ne sme pasti. Če žogica pade na tla, se igra zaključi. Tla sem narisala po istem postopku kot pravokotnik.

V vrstici, ki ima na izbiro rubrike za ustvarjanje novega objekta, sem si izbrala možnost `Paint new sprite`.



Nato se mi je odprlo spodnje okno, kjer sem ustvarila tla. Tla so na zaslonu pod loparjem in so sive barve. Tla sem iz Sprite premenovala v Tla.



• Lopar

Lopar sem ustvarila na identičen način kot tla in pravokotnik. Lopar v igrici ima vlogo lovljenja žogice. Uporabnik lopar po zaslonu premika s pomočjo drsenja miške v levo ali v desno stran.

Spodaj bom obrazložila kodo, ki sem jo zapisala pri objektu Lopar. Najprej pa kratki premislek.

- Začetni položaj loparja se bo nahajal nekje pod žogico.
- Na začetku bo lopar sprožil obvestilo o tem, da naj bo uporabnik pripravljen za začetek igre.
- Ko se bo žogica dotaknila loparja, bo lopar sprožil določen zvok.
- Lopar se bo premikal vodoravno glede na drsenje miške.
- Paziti moramo, da lopar ne gre skozi rob zaslona, kjer se izvaja igrice, pri tem bomo uporabili poseben ukaz.

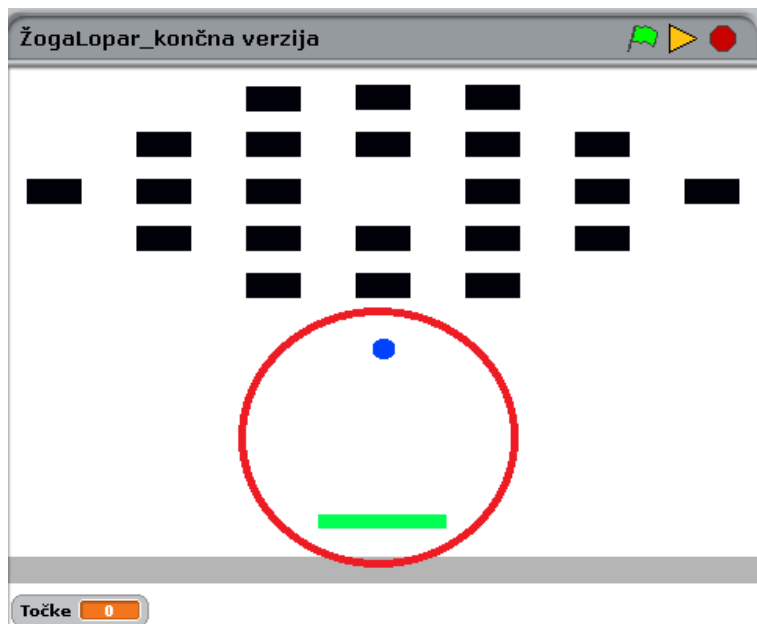
Lopar se pojavi v prvi in drugi stopnji igrice, zato sem morala zapisati kodo pri vsaki stopnji posebej. Razliki med kodama bom pojasnila po tem, ko bom obrazložila kodo, ki sem jo zapisala pri prvi stopnji.

Na spodnji zaslonski sliki je prikaz kode za lopar, ki se nahaja v prvi stopnji igrice.



Funkcija ki je zapisana se začne izvajati v trenutku, ko začnemo igrati igrico, oziroma pritisnemo . Začetni položaj pravokotnika določimo s pomočjo ukaza `go to x: 0 y: 0`.

Ker želimo, da je začetni položaj žogice in loparja tak kot je prikazan na spodnji zaslonski sliki, moramo upoštevati, da imata lopar in žogica enako x koordinato in različno y koordinato. Ker se lopar nahaja pod žogico bo imel manjšo y koordinato.



Tik pred začetkom igrice se iz loparja »izpiše« obvestilo uporabniku, da naj bo pripravljen na začetek igranja. Obvestilo ki se izpiše je naslednje: Si pripravljen!. Obvestilo ustvarimo s pomočjo ukaza

`say Hello! for 2 secs`.

V igrici moramo lopar premikati toliko časa, dokler uspešno s pomočjo žogice ne uničimo vseh pravokotnikov (ne dosežemo 22 točk) in dokler se nam ne izpiše obvestilo o tem da smo uspešno zaključili igro.

Zanka poteka vse do trenutka, ko je število točk manjše od 22. Zanko sem definirala tako, ker se mi je zgodilo, da sem igro uspešno zaključila in da sem dobila obvestilo o uspešnem zaključku, vendar je žogica tik po koncu igre padla na tla in se mi je pokazalo obvestilo Nisi ujel žogice! ☹️ Poskusi znova! (kar pomeni da sem hkrati izgubila in zmagala, zaradi tega protislovja sem uporabila zgornjo zanko!).

S pomočjo ukaza `set x to mouse x` sem uporabniku omogočila, da premika lopar s pomočjo drsenja miške in da se položaj loparja spreminja samo po x osi.

Ker pa se mora lopar gibati samo po zaslonu, kjer se izvaja igrica sem uporabila ukaz

`if on edge, bounce`,

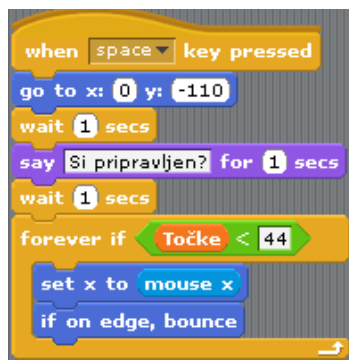
ki omogoča, da se lopar dotakne roba in da ne gre skozi rob zaslona.

Na spodnji zaslonski sliki je prikaz kode, ki sem jo zapisala za lopar, ko bomo nadaljevali igrico v drugo stopnjo.

Na začetku ne uporabimo ukaza  kot v prvi stopnji, ampak uporabimo ukaz

, ker smo nadaljevali igrico v drugo stopnjo.

Zanka poteka vse do trenutka, ko je število točk manjše od 44. V prvi in drugi stopnji je enako število pravokotnikov, vendar se točke iz prve stopnje prištejejo k točkam, ki jih uporabnik doseže v drugi stopnji. Zanko sem ponovno definirala tako, ker se mi je zgodilo, da sem igro uspešno zaključila in da sem dobila obvestilo o uspešnem zaključku, vendar je žogica tik po koncu igre padla na tla in se mi je pokazalo obvestilo Nisi ujel žogice! ☹️ Poskusi znova! (kar pomeni da sem hkrati izgubila in zmagala, zaradi tega protislovja sem uporabila zgornjo zanko!).



• Žogica

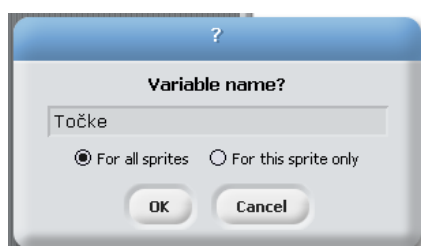
Žogico sem ustvarila na identičen način kot tla, pravokotnik in lopar. Vloga žogice pri igrici je, da jo uporabnik s pomočjo loparja ujame, ta pa se odbije od loparja pod naključnim kotom, ki sem ga definirala pri odboju žogice od loparja.

Ko žogica zadane določen pravokotnik se uporabniku v okvirčku za točke prišteje 1 točka. Točke sem definirala na naslednji način.

- V rubriki *Variables*, sem si izbrala možnost *Make a variable*.



- Odprla se mi je spodnja možnost, kamor sem definirala novo spremenljivko **Točke**. Ko sem pritisnila gumb **OK**, se je nova spremenljivka dodala med ostale spremenljivke v okvirčku.





Točke na začetku igre nastavimo na 0. Ko je uporabnik dosegel v prvi stopnji vseh 22 točk se prva stopnja zaključi in izpiše se naslednje obvestilo: Čestitke! Za naslednjo stopnjo pritisni SPACE BAR.

Na spodnji zaslonski sliki je koda, ki prikazuje na kakšen način sem definirala točke, kdaj in kako se zaključi prva stopnja igrice.



Za drugo stopnjo sem na isti način definirala prištevanje točk. Le da pri drugi stopnji upoštevamo, da se točke prištevajo točkam iz prve stopnje, zato začnemo prištevati točke že k danim 22.

Ko uporabnik uspešno zbere vseh 44 točk, je uspešno zaključil igro in v okvirčku se mu izpiše naslednje obvestilo: Čestitke! Zadel si vse pravokotnike :)

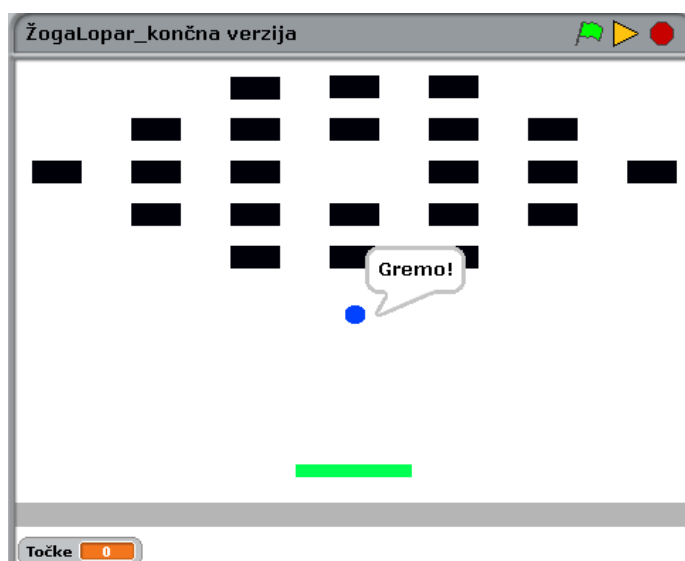
Na spodnji zaslonski sliki imamo izpis kode:



Kot pri ostalih objektih sem tudi pri žogici definirala kodo, s pomočjo katere sem določila kote odboja od loparja, kote odbojev od pravokotnikov, hitrost gibanja, Kodi se pri obeh stopnjah razlikujeta v nekaterih lastostih. Kodi sem razmislila in napisala na dva različna načina.

Celotno kodo bom razložila po korakih. Žogico sem zaradi lažje orientacije postavila v koordinatno izhodišče, na ta način sem tudi lažje določila začetni položaj loparja. S pomočjo ukaza `go to x: 0 y: 0` sem določila začetni položaj žogice. Na začetku igrice žogica uporabniku »sporoči«
začetek igre z obvestilom Gremo! .

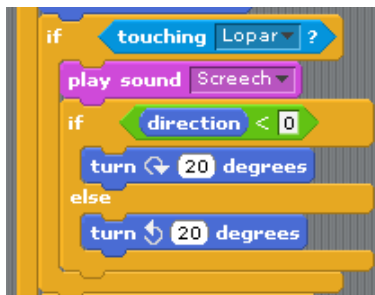
Na spodnji sliki je prikaz žogice na začetku igre.



Žogica na začetku pada proti tlem pod naključno smerjo med 150 in 210. Pri določanju smeri in kotov, ki sem jih uporabila v kodi sem si pomagala s pomočjo ozadja, ki sem ga našla v programu in na katerem je prikaz koordinatnega sistema.

Hitrost žogice je konstantna. Določila sem jo s pomočjo naslednjega ukaza `move 4 steps`. Žogica se giblje samo po vidnem zaslonu, na katerem poteka igrice. Ko pride do roba se odbije nazaj. S pomočjo ukaza se žogica odbije od roba `if on edge, bounce`.

Na spodnji zasloni sliki je prikaz dela kode, ki pove kaj se zgodi z žogico, ko se dotakne loparja.



Ko se žogica dotakne loparja se sproži zvok Screeh. Znotraj zanke sem razlikovala primera kaj se zgodi z žogico, če prileti do loparja z negativne smeri in kaj če prileti s pozitivne smeri.

- Če žogica prileti do loparja z negativne smeri (oziroma z leve strani) se obrne in se od loparja odbije pod kotom 20° v desno.
- Drugi primer nastopi, ko žogica prileti do loparja s pozitivne smeri (oziroma z desne strani) se obrne in se od loparja odbije pod kotom 20° v levo.

Na spodnji zaslonski sliki je prikaz dela kode, ki pove kaj se zgodi z žogico, ko se dotakne tal.



Znotraj zanke sem razlikovala dva primera in sicer:

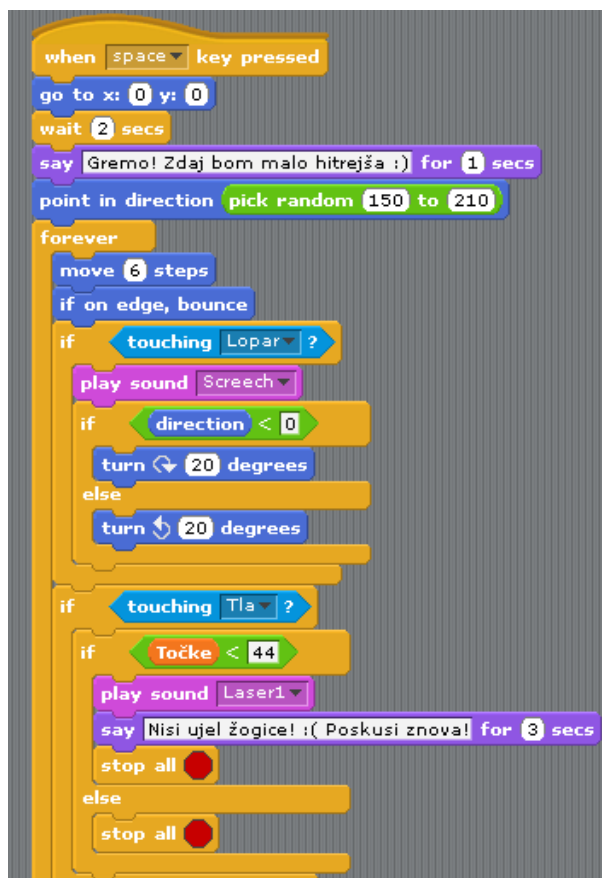
- Če žogica pade na tla predno je zadela vse pravokotnike, oziroma predno je uporabnik dosegel vse možne točke, se igra zaključi in uporabniku se izpiše obvestilo Nisi ujel žogice! ☹️ Poskusi znova! . Ko se dotakne žogica tal se sproži zvok Laser1.
- Če žogica nikoli ne pade na tla, je igra uspešno zaključena.

Na spodnji zaslonski sliki je prikaz dela kode, ki pove kaj se zgodi z žogico, ko se dotakne pravokotnika. V programu sem za vsak pravokotnik zapisala identično zanko.



Smer odboja od pravokotnikov sem določila tako, da sem v kodo zapisala kar nekaj primerov smeri odboja. Za zgornje smeri sem se odločila, ker je igrice bolj dinamična.

Na spodnji zaslonski sliki je prikaz kode za žogico, ko smo ustvarili drugo stopnjo. Razmislek je bil popolnoma identičen kot pri kodi, ki sem jo zapisala za prvo stopnjo, vendar pa moram vseeno navesti razlike.



V drugi stopnji je žogica hitrejša za dve stopnji. Na začetku igrice uporabnik prejme obvestilo o spremembi hitrosti žogice (Gremo! Zdaj bom malo hitrejša! 😊).

V tej kodi nisem zapisala zanke za vsak pravokotnik posebj, ampak sem s pomočjo logičnega operatorja or , najprej naštel vse možne izbire pravokotnikov, ki se jih lahko dotakne žogice in nato zapisala kako žogica spremeni smer.

Prikaz zapisa kode je na spodnji zaslonski sliki.



5. Nekaj idej za nadgradnjo igrice:

Ideja za nastalo igrico sem dobila od igrice, ki jo imam na svojem računalniku in jo še vedno rada igram. Moja izvedba igrice je samo osnovna ideja za takšno igrico.

Spodaj bom navedla nekaj idej, kako bi lahko nadgradila igro:

- Da se posamezen pravokotnik uniči, ga mora žogica zadeti trikrat.
- Hitrost žogice bi se tekom igre stopnjevala, zato bi moral biti uporabnik zelo hiter in iznajdliv.
- Poleg točk bi se uporabniku na zaslonu izpisoval še čas trajanja igre. Ko bi pretekel določen čas bi se igra samodejno zaključila in uporabnik bi izpadel kot poraženec. Zato bi moral biti pozoren kam bi tekom igre usmerjal žogico, da bi ta zadela kar se da največ pravokotnikov.
- Pravokotnike bi lahko ovrednotila z različnimi točkami.
- Če bi žogica zadela tla, se igra ne bi zaključila, ampak bi se lopar malo skrajšal.
- Imamo dve žogici.
- Če bi uporabnik ob koncu igre dosegel vse možne točke, bi se mu za nagrado prikazala določena animacija, ki bi jo spremljala glasba (npr. Maček bi lovil miško, ob tem pa bi se predvajala glasba).

