

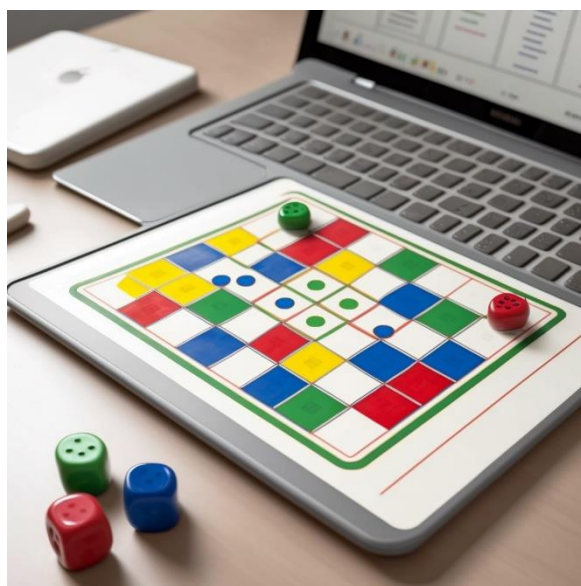


I. osnovna šola Celje
Vrunčeva ulica 13
3000 Celje

Računalničar-ne-jezi-se

Računalništvo

RAZISKOVALNA NALOGA



Avtor:
Rudi Rodič

Mentor:
Žan Močivnik, prof. mat. in rač.

Celje, marec 2023

Zahvala:

Zahvaljujem se vsem, ki so pripomogli k nastanku te raziskovalne naloge. Pri raziskovalni nalogi mi je bil v največjo pomoč mentor Žan Močivnik, ki mi je bil vedno na voljo za morebitna vprašanja, za kar sem mu zelo hvaležen. Zahvaljujem se tudi vodstvu I. osnovne šole Celje, ki mi je omogočilo raziskovalno nalogo.

Povzetek

Osnova raziskovalne naloge je igra *človek ne jezi se*, ki je dobro poznana vsem otrokom. Pri nalogi sem želel s svojim računalniškim znanjem zapisati igro kot računalniški program. Uporabil sem okolje Scratch, ki je enostavno in omogoča bločno in grafično programiranje. Namen naloge je bil izdelati grafično podobo igre, kjer se upoštevajo vse lastnosti in pravila igre. Raziskovalna naloga predstavlja zasnovo programa in metode do končnega izdelka.

Ključne besede: Scratch, človek ne jezi se, programiranje, namizna igra, računalniška igra.

Summary

We all played a board game as children. One of my favourites is Ludo, which is well known to all children. In my research assignment I wanted to see if I had enough computer skills to program the game graphically as a computer program. I used the Scratch environment, which is simple and allows both block and graphical programming. The aim of the assignment was to create a graphical representation of the game, taking into account all the features and rules of a board game. The research task presents the design of the program and the programming methods up to the final product.

Keywords: Scratch, Ludo, programming, board game, computer game.

Kazalo vsebine

1	Uvod	1
1.1	Opis raziskovalnega problema in namen	1
1.2	Hipoteze	1
1.3	Raziskovalne metode.....	1
1.4	Orodja in pripomočki.....	2
2	Teoretični del.....	3
2.1	Zgodovina iger	3
2.1.1	Začetki družabnih iger	3
2.1.2	Zgodovina nekaterih sodobnih družabnih iger.....	3
2.1.3	Namizne igre s kockami.....	4
2.2	Človek-ne-jezi-se	4
2.2.1	Nastanek.....	4
2.2.2	Vrste igre človek ne jezi se	5
2.2.3	Podlaga	6
2.2.4	Potek Igre.....	6
3	Osrednji del	7
3.1	Okolje Scratch.....	7
4	Razvoj programa.....	9
4.1	Zasnova ideje.....	9
4.2	Igralno polje.....	10
4.3	Kocka	10
4.4	Igralne figure	15
5	Zapis dnevnika in uvedba sprememb	19
5.1	Ime projekta	20
6	Sklepne ugotovitve	20
7	Zaključek.....	21
8	Viri in literatura	22

Kazalo slik

Slika 1: Igralna plošča Ludo	5
Slika 2: Slika originalne igralne plošče	5
Slika 3: Slika poznane igralne plošče	5
Slika 4: Ena izmed različic igre – Game of Headache	6
Slika 5: Izgled programa	7
Slika 6: Skica plošče	9
Slika 7: Igralno polje	10
Slika 8: Izris igralne kocke	11
Slika 9: Izgled igralne kocke	11
Slika 10: Pogoji ob metu kocke in urejanje zaporedja igralcev	12
Slika 11: Primer igralnega polja in igralca, ki je na vrsti	12
Slika 12: Prvi del programa za preverjanje barve – met kocke	13
Slika 13: Drugi del kode za preverjanje barve – met kocke	13
Slika 14: Program preveri, kdaj se igra zaključi	14
Slika 15: Slika figur	15
Slika 16: Začetek posamezne figure	15
Slika 17: Program za postavitve figure na polju	15
Slika 18: Program za postavitev figure glede na koordinate x in y	16
Slika 19: Premik figure na začetno polje	16
Slika 20: Preverjanje pogojev za premikanje figure	16
Slika 21: Pogoji, kdaj zbijamo figuro	18

Kazalo tabel

Tabela 1: Dnevnik programiranja – sledenje spremembam	19
---	----

1 Uvod

1.1 Opis raziskovalnega problema in namen

Kot otrok sem zelo rad igral namizne igre. Ena izmed teh je bila tudi človek ne jezi se. Gre za zelo enostavno otroško igro, kjer je potrebno nekaj sreče pri metu kocke in razvijanje strategije, katero od štirih figur boš uporabil za premikanje in izdelovanje svoje strategije.

Ker sem sam tudi zelo navdušen nad programiranjem in sem obiskoval neobvezni izbirni predmet 3 leta, me je zanimalo ali imam dovolj računalniškega znanja, da namizno igro postavim v računalniško okolje. Seveda pri programiranju ni dovolj, da poznaš le pravila igre, ampak je potrebno poleg igralcev »naučiti« tudi računalnik, kako se igra igra. Tako sem se odločil, da bom igro sprogramiral v računalniškem okolju Scratch. Scratch že ponuja grafično programiranje in orodja, zato se bom lahko bolj posvetil samemu programiranju.

1.2 Hipoteze

Za začetek raziskovanja sem si podal sledeči raziskovalni cilj:

Ali imam kot sedmošolec dovolj računalniškega znanja za programiranje računalniške igre človek ne jezi se?

Glede na podano raziskovalno vprašanje sem si postavil hipoteze, ki jih bom kasneje ovrgel ali potrdil.

H 1: Namizna igra je lahko narejena v računalniškem programu.

H 2: Namizno igro je mogoče grafično narediti v poenostavljenem računalniškem okolju, kot je Scratch.

H 3: Računalniško igro bo lahko igralo poljubno število igralcev.

1.3 Raziskovalne metode

Začetki raziskovanja so zelo pomembni. Pri tem sem si pomagal s sledečimi raziskovalnimi metodami:

- sistematični pregled literature,
- metoda možganske nevihte,
- eksperiment,
- metoda študija primera.

Sistematični pregled literature – metoda, s katero sistematično preverimo raziskovalno področje in primerjamo svoje ugotovitve. S to metodo sem preveril zgodovino namiznih iger in različice igre človek ne jezi se.

Metoda možganske nevihte – pripomogla je k raziskovanju različnih problemov in idej, kako namizno igro pretvoriti v računalniški program. S to metodo sem želel pretvoriti namizno igro v računalniški jezik. Iskal sem povezavo med kocko, figuro in igralnim poljem. Po razmišljanju sem si naredil načrt, kako bo celoten program deloval in katera figura bo vodila celoten potek igre – programsko.

Eksperiment – metoda, kjer zapišemo posamezni del programa in ga testiramo. Pri tem ugotavljamo vse možnosti in iščemo napake. Ta metoda je bila zelo pomembna, saj je pri razvijanju programa

potrebnega veliko testiranja. Prav tako ne pridemo do rešitve vedno v prvem koraku, zato lahko rečem, da je programiranje neke vrste eksperimentiranje.

Metoda študija primera – je metoda, kjer se analizira posamezen košček kode in poskuša razumeti kako vpliva na delovanje programa. Pri raziskovanju sem ugotovil, da imam na polju 16 figur, ki se med seboj obnašajo podobno. Po analizi posameznih figur sem ugotovil, da lahko kodo uporabim tudi za ostale figure, le spremeniti moram nekatere posebnosti.

1.4 Orodja in pripomočki

Pri raziskovanju sem potreboval sledeča orodja:

Google Drive je okolje, ki vsebuje urejevalnik dokumentov, ustvarjanje elektronskih prosojnic in ustvarjanje map za odlaganje dokumentov. Zaradi enostavnosti omogoča hitro odlaganje dokumentov preko mobilnih naprav in računalnika.

Scratch je brezplačen in odprtokoden vizualni programski jezik, ki omogoča ustvarjanje interaktivnih animacij, iger in aplikacij s pomočjo blokov kode, ki jih preprosto povlečemo na programsko ploščo. Namenjen je predvsem mlajšim uporabnikom in spodbuja kreativnost, sodelovanje in učenje osnov programiranja. Na voljo je na spletu in za mobilne naprave. Leta 2007 ga je razvil Mitchel Resnick. V njem lahko programiramo video igrice, virtualne instrumente ter drugo.

Office 365 je okolje za urejanje dokumentov, ki ga je razvilo podjetje Microsoft. Paket vsebuje urejevalnik besedil, planer, izdelavo elektronskih prosojnic, shrambo dokumentov in drugo. Pripomoček sem uporabil za pisanje raziskovalne naloge in odlaganje dokumentov

xMinde je program za izdelovanje digitalnih miselnih vzorcev oz. tako imenovan mindemaping. Omogoča vizualni način organiziranja idej in informacij v obliki miselnih vzorcev, ki pomaga pri načrtovanju, raziskovanju, izobraževanju in drugih dejavnostih.

2 Teoretični del

Teoretični del zajema zgodovino nekateri starejših in sodobnih namiznih iger. Predvsem sem se kasneje osredotočil na posamezne igre, ki vsebujejo igralno kocko in figure. Pri tem sem želel najti vzorec med uporabo polja, kocke in figur. Pri iskanju mi je predvsem pomagala literatura.

2.1 Zgodovina iger

2.1.1 Začetki družabnih iger

Poznamo štiri kategorije družabnih iger: dirkalne igre, vesoljske igre, igre lovljenja in igre premikanja.

Arheologi so našli družabne igre iz kamene dobe. Na 49 let starem pokopališču Bašur Höyük na jugovzhodu Turčije so našli vrsto 5000 majhnih izrezljanih poslikanih kamnov. To so najstarejši igralni kosi, ki so jih kdaj našli. Podobni deli so bili najdeni v Siriji in Iraku ter kažejo na družabne igre, ki izvirajo iz Plodnega polmeseca.

Šah je namizna igra med dvema igralcema. Trenutna oblika igre se je pojavila v Španiji v drugi polovici 15. stoletja, potem ko se je razvila iz *chaturange*, podobne, a veliko starejše igre.

Poker je igra s kartami, kjer ljudje igrajo in stavijo na srečo. Mnogi igralci pokra ugotavljajo, da se je današnji poker razvil iz francoske igre *poque* in perzijske igre *as-nas*. Gotovo je, da je bil poker populariziran na ameriškem jugu v 19. stoletju, saj so igre na srečo z rečnimi čolni na reki Mississippi in okoli New Orleansa v 1830-ih letih pomagale razširiti igro.

2.1.2 Zgodovina nekaterih sodobnih družabnih iger

Monopoly, namizna igra za več igralcev, je namenjena spoznavanju ekonomije. V igri se uporabljata dve kocki, ki ju igralci uporabljajo za premikanje po podlagi. Med premikanjem igralci kupujejo in prodajajo posestva in na njih gradijo hiške in hotele (nepremičnine). Igralci pridobivajo denar s pomočjo kupljenih posestev. Ko tuji igralci pridejo na tvoje posestvo, morajo plačati vsoto denarja, ki je zapisana na kartah. Monopoly sega v leto 1903, ko je bila igra namenjena kot izobraževalno orodje za ponazoritev negativnih vidikov koncentracije zemljišč v zasebnih monopolih.

Uno (enka) je ameriška igra s kartami, ki se igra s posebej natisnjenim kompletom kart, na katerih so zapisane številke. Igro je leta 1971 prvotno razvil Merle Robbins v Readingu v Ohio. Ko so njegova družina in prijatelji začeli igrati čedalje več, je porabil 8.000 \$, da je naredil 5.000 kopij igre. Sprva jo je prodajal v svoji brivnici, prodajati pa so jo začela tudi lokalna podjetja. Robbins je pozneje prodal pravice za igro skupini prijateljev, ki jih je vodil Robert Tezak (lastnik pogrebne podjetja v Jolietu) za 50.000 \$ plus licenco (10 centov na igro). Tezak je za trženje igre *uno* ustanovil International Games, Inc, ki je še vedno v zagonu.

"Dungeons & Dragons" (D&D) je priljubljena namizna igra, ki sta jo leta 1974 ustvarila Gary Gygax in Dave Arneson. Igra se igra v skupinah, ki jih sestavljajo igralci, ki predstavljajo like v fiktivnem svetu, ki ga vodi t. i. Dungeon Master (DM). Igralci v D&D uporabljajo kocke in zapletene sisteme pravil, da bi igrali svoje like, ki imajo različne sposobnosti in veščine. Igra spodbuja domišljijo, kreativnost in učenje novih stvari ter spodbuja igralce k raziskovanju in sodelovanju. D&D je postal zelo priljubljen po vsem svetu in je vplival na številne druge igre in izdelke, kot so računalniške igre, knjige in filme. Kocke, ki se uporabljajo v D&D, so različnih velikosti in oblik ter določajo naključne dogodke v igri, na primer izid spopada ali odkritje skritega zaklada.

2.1.3 Namizne igre s kockami

Namizne igre s kockami so priljubljene po vsem svetu in so se igrane že stoletja. Kocke se uporabljajo za ustvarjanje naključnih rezultatov, ki določajo gibanje ali druge pomembne dejavnike igre. Tukaj je nekaj najbolj priljubljenih namiznih iger s kockami:

- Človek ne jezi se – ena najbolj priljubljenih iger, igra se s štirimi figurami, ki se premikajo po igralni plošči z uporabo kocke.
- Monopoly – to je še ena zelo priljubljena igra, ki se igra z uporabo kocke. Igralci kupujejo, prodajajo in trgujejo z nepremičninami, da bi si pridobili bogastvo. Kocka se uporablja za določanje števila mest, ki jih igralec premakne po igralni plošči.
- Yatzy – to je igra, ki se igra s petimi kockami, s katerimi igralci poskušajo doseči določene kombinacije, kot so enake kocke ali zaporedna števila. Cilj igre je doseči čim več točk.
- Prebivalci otoka Catan – ta igra je priljubljena med strategiji in se igra z uporabo šeststranske kocke. Igralci gradijo mesta in ceste ter trgujejo z viri, da bi zbrali dovolj surovin in dosegli zmago.
- Risk – igra je namenjena strategiji in taktiki, igra se s kocko in figurami. Igralci se borijo za nadzor nad svetom in se spopadejo s taktičnimi bitkami, ki jih določi kocka.
- Dungeons & Dragons je namizna igra, ki se igra v skupinah. Igralci predstavljajo like v fiktivnem svetu, ki ga vodi Dungeon Master. Igralne kocke se uporabljajo za določanje naključnih dogodkov v igri, kot so izid spopada ali odkritje skritega zaklada. V igri se uporabljajo različne vrste kock, med katerimi so najbolj pogoste šeststranske kocke (D6), desetstranske kocke (D10), dvanajststranske kocke (D12) in dvajsetstranske kocke (D20). Vsaka kocka ima svoj namen in se uporablja pri različnih situacijah v igri.

Kot lahko opazimo, ima večina iger veliko skupnih iztočnic. Večina jih uporablja podobna igralna polja ali igralne kocke. Namizne igre uporabljajo v večini primerov tudi figure. Dungeons & Dragons je ena izmed posebnih sodobnih iger, ki uporablja različne igralne kocke in temelji na igralčevi domišljiji in predstavah.

2.2 Človek-ne-jezi-se

2.2.1 Nastanek

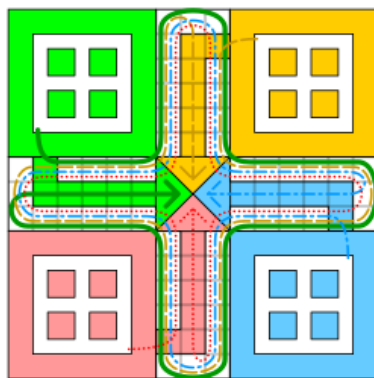
Igra "človek ne jezi se" je ena najbolj priljubljenih družabnih iger na svetu, ki izvira iz Indije. Igra izvira iz igre po imenu *pachisi*, ki so jo več stoletij igrali v Indiji, kjer je nastala v 6. stoletju.

Najnovejši dokaz o razvoju igre so upodobitve desk na jamah Ellora. Prvotna različica je opisana tudi v indijskem epu Mahabharata, v katerem Šakuni z zakletimi kockami premaga Pandave. Na koncu Yudhisthira (najstarejši izmed petih Pandavinih bratov) po izgubi vsega postavi na kocko še svojo ženo Draupadi in izgubi tudi njo. Pandave dobijo vse svoje imetje nazaj, potem ko se Draupadi zaveže, da bo preklela celotno rodovino Kurujev, a se ustavi ob posredovanju Gandharije. Kralj Kurujev Dhritarashtra (mož Gandharije) vidi priložnost, da pomiri Draupadijino jezo in obljubi, da bo vrnil Pandavam vse, kar so izgubili v igri. (Bruning, 2022)

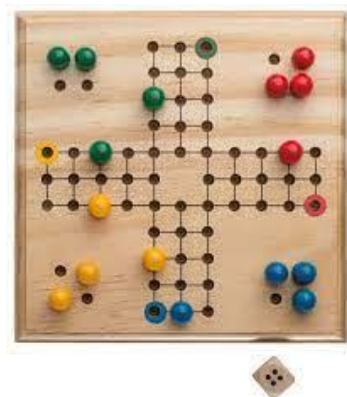
Pachisi izvira iz besede pachisi, kar pomeni 25. Igra je bila leta 1896 prinešena in patentirana v Angliji pod imenom "človek ne jezi se". Od takrat velja za eno izmed najbolj igranih namiznih iger.



Slika 1: Igralna plošča Ludo



Slika 2: Slika originalne igralne plošče



Slika 3: Slika poznane igralne plošče

2.2.2 Vrste igre človek ne jezi se

Človek ne jezi se je po svetu poznana po različnih imenih in pravilih igre:

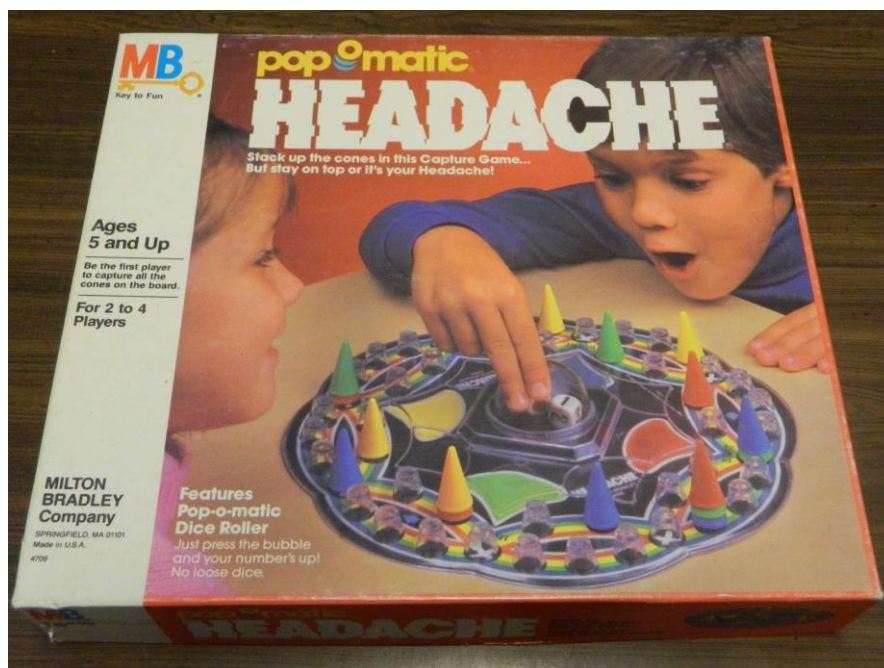
- uckers (britanska),
- pachisi (indijska),
- fia (švedska),
- naglica naredi pacea (švicarska),
- Cờ cá ngựa (vietnamska),
- parchis (španska),
- človek, ne sekiraj se (poljska),
- parques (kolumbijska): vuelta obligada (obvezen ponovni start), cielo robado (ukradena nebesa), de piedra en piedra (od kamna do kamna), con policía (s policistom),

V Sloveniji in Evropi je bolj poznana nemška različica, ki je nastala leta 1914. Pri tem je igra obdržala tudi izvorno ime, ki je bilo prevedeno v različne jezike:

- mens erger je niet (nizozemsko),
- non t'arrabbiare (italijansko),
- človek, ne jezi se (slovensko),
- člověče, nezlob se (češko),
- čovječe, ne ljuti se (hrvaško),
- człowieku, nie irytuj się (poljsko).

Kasneje so se razvile še drugačne različice te igre, da bi popestrile dogajanje. Primer so 3 različice nemškega proizvajalca:

- ne izgubite glave,
- noppit,
- brendi dog.



Slika 4: Ena izmed različic igre – Game of Headache

2.2.3 Podlaga

Območja ob robovih plošče so obarvana s štirimi različnimi barvami (rumena, modra, rdeča in zelena). Ob začetku igre vsak igralec izbere barvo in figure te barve. Podlaga je kvadratna z igralnim prostorom v obliki križa, pri katerem ima vsak krak križa tri stolpce polj, običajno šest na stolpec. Srednji stolpci imajo običajno pet obarvanih kvadratov; ti predstavljajo dom. Šesto obarvano polje, ki ni v domačem stolpcu, je igralčevo začetno polje.

2.2.4 Potek Igre

Vsak igralec ima štiri figure iste barve, ki jih postavi na začetno polje v kotu igralne plošče imenovane dom. Cilj igre je, da igralec prvi pripelje vse štiri svoje figure do ciljnega polja imenovanega hiška v središču igralne plošče.

Vsak igralec igra po vrsti in na vsaki potezi lahko izbere eno izmed svojih figur, ki jo premakne za število polj, ki ga določi s kocko. Če se figura ustavi na polju, na katerem je že figura drugega igralca, se ta figura vrne na začetno polje in mora igralec ponovno začeti s to figuro.

Če pa igralec s svojo figuro doseže ciljno polje, se postavi v hiško in z njo ni več mogoče doseči nobenega drugega polja. Prvi igralec, ki prinese vse svoje figure v ciljno polje, zmaga igro.

Igralci lahko med igro zavirajo napredovanje nasprotnikov in poskušajo odstraniti njihove figure, če pridejo na polje, na katerem že stoji figura nasprotnika. Če igralec vrže šest na kocki, ima pravico do še ene poteze z eno izmed svojih figur.

To so osnovna pravila igre človek ne jezi se, ki jih lahko igralci prilagodijo glede na svoje želje in dogovor. Ko so vse figure iste barve v hiški, se igra zaključi in zmaga igralec, ki je igral s to barvo. Po hišnih pravilih se lahko igra nadaljuje, da drugi igralci končajo in osvojijo 2., 3. ali 4. mesto.

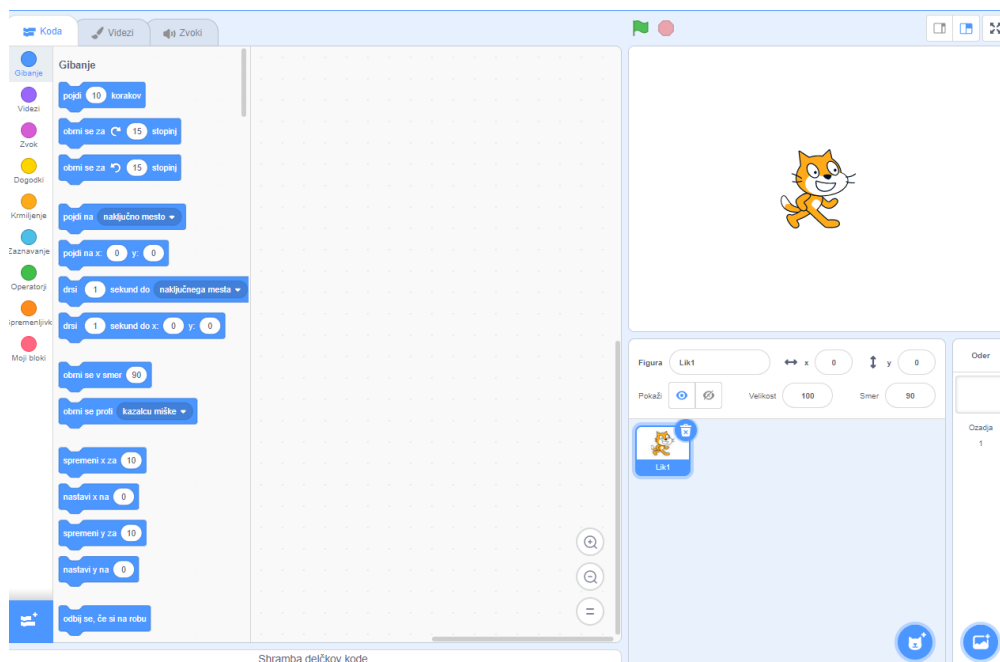
3 Osrednji del

3.1 Okolje Scratch

Scratch je orodje za blokovno programiranje, ki omogoča izdelavo interaktivnih zgodb, iger in animacij. Namenjen je otrokom med 8. in 16. letom, ponuja tudi poenostavljeno verzijo ScratchJr za mlajše otroke. Orodje spodbuja razvoj različnih vidikov razmišljanja, sposobnosti ugotavljanja različnih pristopov k reševanju problemov ter sposobnosti timskega dela, kar je uporabno širše, ne le na področju programiranja. V nalogi predstavljam Scratch s teoretičnega in praktičnega vidika – vsebinske možnosti, primernost za učenje programiranja za otroke, možnost uporabe tako v šoli kot tudi doma ter dostopnost spletne pomoči za uporabo orodja. V nalogi prav tako ugotavljam, da avtorji načine učenja, kot jih ponuja Scratch, prepoznavajo kot pomembne v današnjem času, ko imamo na voljo ogromno informacij in problemov, za katere moramo najti čim boljše rešitve; pri slednjem nam pride prav ravno kritično razmišljanje in sposobnost razmišljanja izven okvirjev.

Scratch je dobil ime po tehniki, ki jo uporabljajo diski jockeyji, imenovani "scratching", pri kateri se vinilne plošče pripnejo skupaj in se z njimi manipulira na gramofonu, da se ustvarijo različni zvočni učinki in glasba. Podobno kot scratching spletno mesto uporabnikom omogoča mešanje različnih medijev – zvok, programiranje, slika itd. (Kernc, 2022)

Vmesnik Scratch je razdeljen na tri glavne dele: območje odra, paleto blokov in območje kodiranja za postavitev in razporeditev blokov v skripte, ki jih je mogoče zagnati s pritiskom na zeleno zastavico ali klikom na samo kodo. Uporabniki lahko ustvarijo tudi lastne kodne bloke, ki bodo prikazani v "Mojih blokih".



Slika 5: Izgled programa

Območje bloka je koordinatni sistem, ki se razteza po x-osi od -200 do 200 in y-osi od -180 do 180. To je področje prikaza našega programiranja. Posamezne like lahko na tem področju skrijemo ali jih z večjo koordinato prestavimo izven vidnega polja. To nam še posebej koristi, kadar programiramo premikanje.

Drugi del je paleta programskih blokov. Ta nam omogoča pester izbor različnih ukazov oz. blokov, ki nam omogočajo programiranje. Posamezno paletu lahko razširimo tudi na druga orodja – svinčnik, micro:bit, video zaznavanje, besedilo v govor, LEGO itd. Osnovni bloki so razdeljeni v posamezne kategorije:

1. Gibanje – omogoča gibanje posamezne figure;
2. Videz – spreminjanje videza in velikosti figure;
3. Zvok – snemanje ali predvajanje zvoka;
4. Dogodki – kako se naj posamezen program začne;
5. Krmiljenje – uporaba zank in pogojev;
6. Zaznavanje – ustvarjanje pomoči pri pogojih;
7. Operatorji – uporaba matematike in združitenih pogojev;
8. Spremenljivke – ustvarjanje lastnih spremenljivk ali seznamov, kamor shranimo podatke;
9. Moji bloki – ustvarjanje podprogramov.

Območje kodiranja je namenjeno in omejeno s figuro ali ozadjem. Kadar ustvarjamo program, se ta vedno zapiše v figuro oz. ukazuje, kako se naj obnaša. Če želimo program pisati na višjem nivoju, moramo uporabljati spremenljivke, saj je to edina možnost, da lahko naredimo prehod informacij med figurami. V Scratchu programiramo tako, da zlagamo figure eno nad drugo in tako se program izvaja vedno od zgoraj navzdol.

Filozofija Scratcha spodbuja deljenje, ponovno uporabo in kombiniranje kode, kot nakazuje slogan ekipe "Zamisli, programiraj, deli". Uporabniki lahko naredijo lastne projekte ali pa se odločijo za "remiks" projekta nekoga drugega. Projekti, ustvarjeni in predelani s programom Scratch, so licencirani pod licenco Creative Commons Attribution-Share Alike. Scratch samodejno pripiše zasluge uporabniku, ki je ustvaril izvirni projekt in program v zgornjem delu projektne strani.

Scratch 2.0 je bil izdan 9. maja 2013. Posodobitev je spremenila videz spletnega mesta in vključevala tako spletni urejevalnik projektov kot urejevalnik brez povezave.

Scratch 3.0 je prvič napovedala ekipa Scratch Team leta 2016. Scratch je bil spisan popolnoma na novo, saj so ustvarjalci uporabili programski jezik Python. Zaradi razširjenosti programskega jezika je Scratch možno povezati z različnimi napravami. Scratch je postal tako priljubljen, da je osnova za marsikaterega proizvajalca. Zadnji veliki gigant, ki uporablja Scratch za programiranje, je LEGO v svojih novih Mindstorm paketih imenovanih Spike.

4 Razvoj programa

4.1 Zasnova ideje

Pred začetkom programiranja je bilo potrebno narediti strategijo. Najprej se je bilo potrebno odločiti, katero polje bom uporabil. Pri tem sem si izdelal skico igralnega polja. Hitro sem ugotovil, da bo za potrebo programiranja potrebno prilagoditi nekatera pravila.

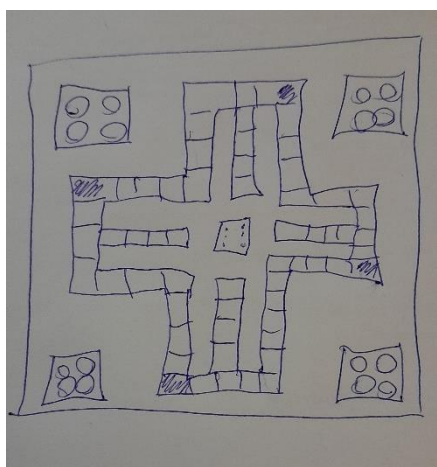
Tako sem se odločil, da bo zaradi načina programiranja igralno polje imelo 40 osnovnih polj, št. 41, 42, 43 in 44 pa bodo predstavljali domovi. Za takšen način sem se odločil, ker mi je bilo lažje seštevati posamezna polja.

Drugi sistem, ki sem ga moral urediti s poljem, je, kako povezati koordinatni sistem oz. izrisano polje s točno lokacijo premikanja figure. Pri tem sem moral paziti, da je najlažje, če polje narišem simetrično, vse ostalo je premikanje po vzorcu.

Na polju sem moral tudi določiti, katera barva bo začela krog in koliko bo igralcev. Pri sami zasnovi sem se odločil, da oštevilčim figure in s klikom na njo se le-ta prestavi na posamezno polje. Prav tako je bilo med seboj potrebno povezati met kocke in premikanje ter uskladiti program s številom 6, ki nam daje pravico do ponovnega meta.

Največjo težavo mi je seveda predstavljala ideja premikanja figur. Ugotovil sem, da bom moral svoje znanje nadgraditi. Pri računalništvu smo do sedaj že uporabljali spremenljivke, a le za zapisovanje konkretnih rezultatov. Tokrat sem moral uporabiti spremenljivke tudi na dodaten način, tako imenovan utež. Utež je posamezna spremenljivka, ki določa pomembnost podane naloge. Uteži sem uporabljal pri spremembi igralca in premikanjih po polju.

Dodatni izziv pri programiranju pa so mi predstavljali sezname. Sezname so struktura podatkov oz. elementov. V mojem primeru sem si v sezname shranjeval pozicije posameznih figur. S pomočjo seznama sem lahko določal, katero figuro je moč premakniti. Z njihovo pomočjo sem tudi enostavno določil zmagovalca.

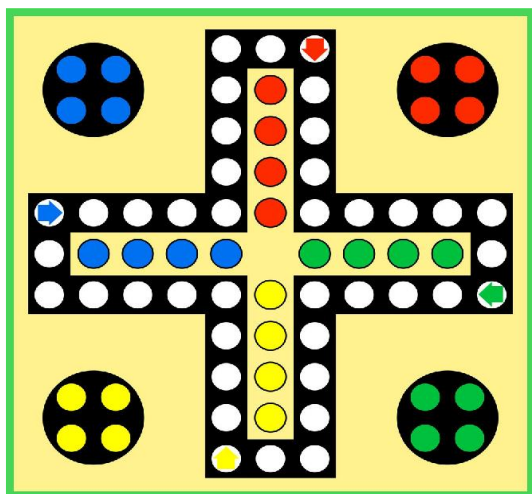


Slika 6 Skica plošče

4.2 Igralno polje

Za potrebo pri programiranju sem si, kot že rečeno, naredil skico, ki je v kasnejših obdobjih prišla še kako prav. V spodnje levo polje (hišo) sem umestil rumeno, v zgornje levo modro, v zgornje desno rdečo in v desno spodnje zeleno barvo. Cela plošča ima štiri krat po štiri polja imenovana hiša, kjer so na začetku shranjene vse figure, po štiri polja s štirimi prostori imenovana dom in 40 osnovnih polj (pot), po katerih se bo igralec premikal do svoje zmage. S puščicami je označeno začetno polje, kamor se postavi figura iz hiše. Polja ni bilo potrebno sprogramirati, ga je pa bilo potrebno postaviti na ustrezna mesta.

Hiša	Začetna pozicija, v kateri se pojavijo figure iste barve, s katerimi igralec igra.
Začetno polje	Prvo polje po metu številke 6 oz. izhod iz začetne pozicije in vključitev v igro.
Pot	40 osnovnih polj, po katerih se igralec premika in sodeluje v igri.
Dom	Končna pozicija posamezne figure, kjer je figura na varnem in se ne premika več.



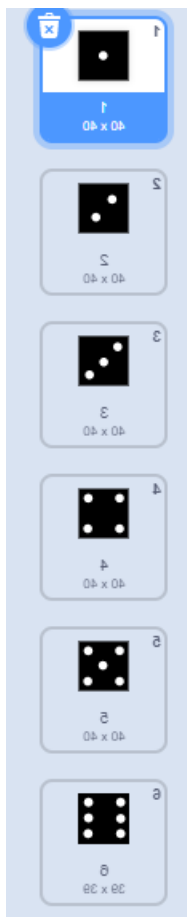
Slika 7: Igralno polje

4.3 Kocka

Kocka je ključni del igre in ukazov, ki jih bomo potrebovali. Ker je to najpomembnejši del, sem se odločil, da bom najprej začel pri tej figuri. Začel sem tako, da sem na črn kvadrat narisal številke, ki povedo, koliko si vrgel. Vsaka številka v figuri je narisana v novem videzu. Ko sem imel vse potrebne slike, sem se odločil za programiranje.

Pri tem sem moral biti pozoren na dve stvari:

1. Odločil sem se, da bom vizualno uredil met kocke.
2. Sprogramirati bom moral naključni met, ki ga bom moral s pomočjo spremenljivke vnesti v ustrezno figuro. Pri tem si bom moral zapolniti korake premikanja figure in njeno pozicijo na igralnem polju.



Slika 8: Izris igralne kocke

Najprej sem uredil izgled kocke.



Slika 9: Izgled igralne kocke

Vsaka igra se začne tako, da primeš kocko in jo vržeš, zato sem kodo kocke začel z ukazom: »Ko je kliknjen ta lik«, kar naredi, da se program kocke izvede, le če se igralec kocke dotakne.

Nato sem določil, da se kocka ob kliku premakne na sredino podlage, saj bi bila tako bolj pregledna. Nato sem hotel dodati občutek vrtenja kocke, kar sem rešil tako, da se videz kocke izvede 10-krat.

Kocka zamenja svoj izgled naključno med številom 1 in 6 in se obrne za 36°.

Po končanem metu kocke program objavi številko, na kateri je pristala, in jo shrani v spremenljivko.



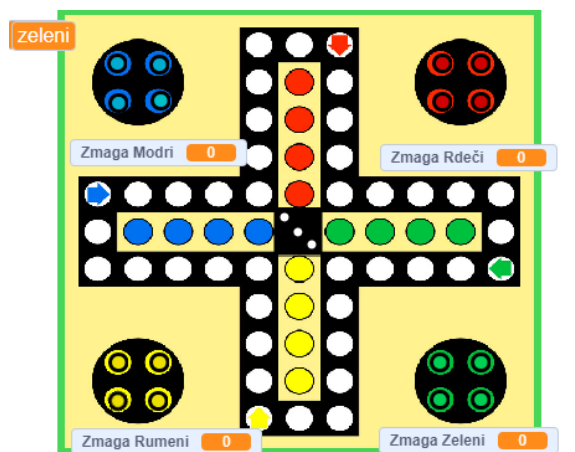
Slika 10: Pogoji ob metu kocke in urejanje zaporedja igralcev

Ker je kocka glavni del vse igre, sem vanjo dodal še kodo za menjavo barve igralca.

V kodo sem dodal pogoj, da se v primeru števila 6 nič ne spremeni. To je eden izmed ključnih pogojev oz. uteži, ki spreminja dinamiko igre. Igralec ima ob metu števila 6 možnost ponovno vreči kocko.

V primeru, da je število različno od 6, pa se program premika v smeri urinega kazalca po polju.

V levem zgornjem kotu programa se vedno zapiše, kateri igralec je na potezi.



Slika 11: Primer igralnega polja in igralca, ki je na vrsti



Slika 12: Prvi del programa za preverjanje barve – met kocke

Za lažje razumevanje programa sem se odločil, da bom podrobneje opisal posamezne dele kode.

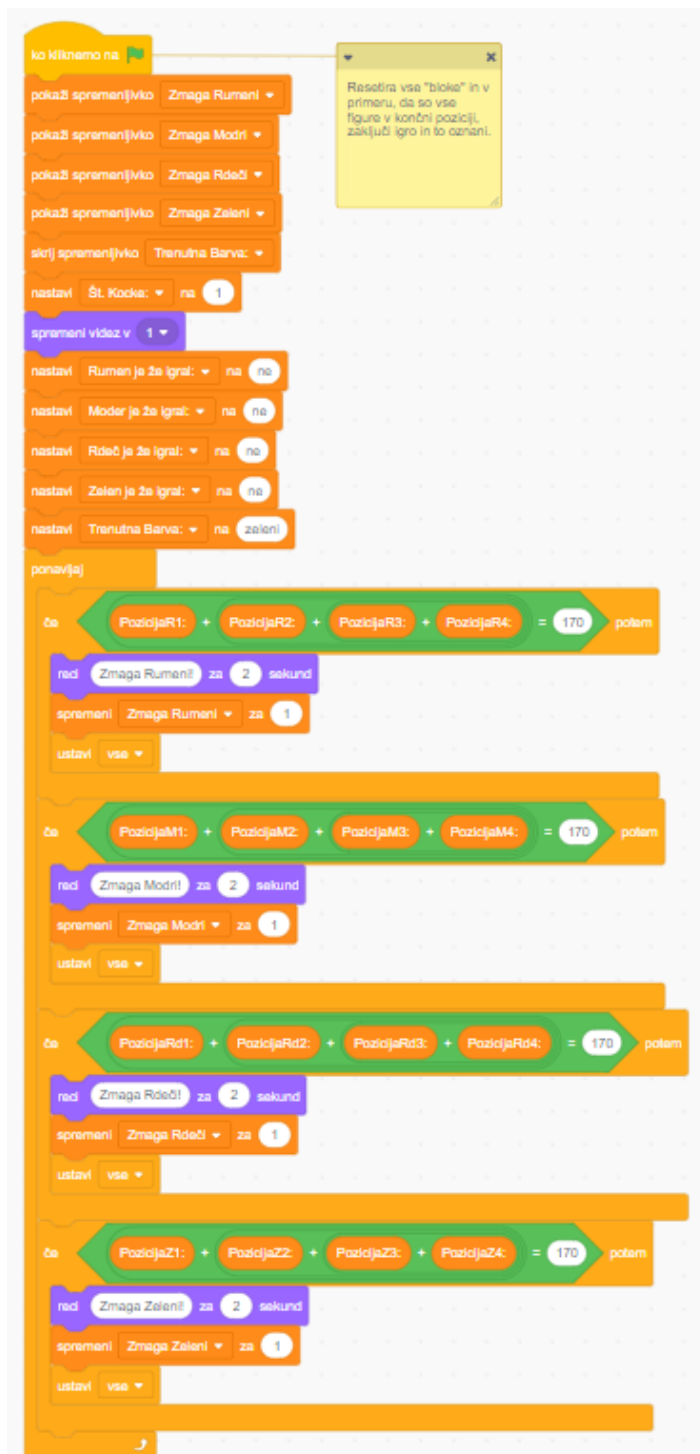
V prvem delu zgornje kode sem napisal program, ki služi temu, da pregleduje, katera barva je na vrsti. V primeru, da pade številka šest, se barva figure ne spremeni. V spremenljivko *Trenutna barva* se zapiše barva igralca, ki je na vrsti.



Slika 13: Drugi del kode za preverjanje barve – met kocke

V drugem delu kode sem naredil kodo, ki preveri naslednji pogoj:

V primeru, da je številka kocke različna od 6, se prestavi na naslednjo barvo v smeri urinega kazalca – grafično, oz. si v programu v krogu sledijo barve rumena, modra, rdeča in zelena.



Slika 14: Program preveri, kdaj se igra zaključi

Zadnja koda v programu pa je koda za zaključek igre. Ob začetku programa ta najprej uredi vse spremenljivke.

Nato program ves čas preverja pozicijo posameznih figur.

Ker vemo, da morajo biti figure v domu na poziciji 41, 42, 43 in 44, je tako njihova vsota 170.

V primeru, da je seštevek pozicij enakih figur enak 170, nam bo program oz. kocka preko sporočila oznanila zmagovalca.

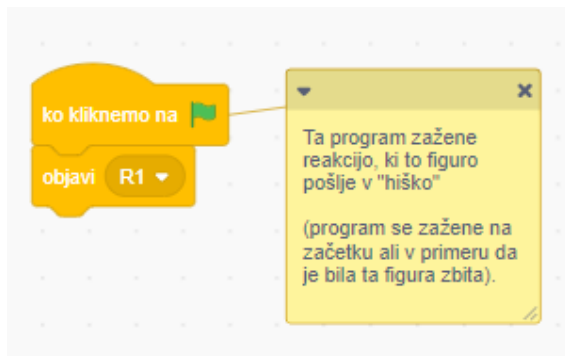
4.4 Igralne figure

Igralne figure imajo med seboj podobno kodo. Načeloma se sklopi kode vežejo na posamezno barvo. Pri barvah sem moral popraviti le nekaj malenkosti, da so figure delovale pravilno.



Slika 15: Slika figur

Igra deluje tako, da se program izvede šele takrat, ko na posamezno figuro kliknemo.



Slika 16: Začetek posamezne figure

Program nad figuro se izvede, ko kliknemo na njo. Vsaki figuri sem dodal svoje ime glede na barvo in pozicijo v hiški. Pri tem se v nadaljevanju izvedeta dva pogoja – kdaj se lahko figura premika naprej in kdaj pride do zbijanja.

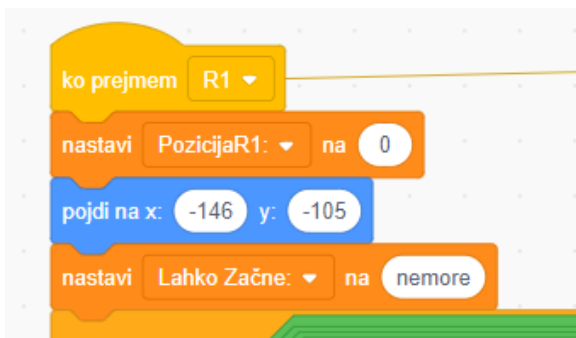


Slika 17: Program za postavitev figure na polju

Prvi pogoj se izvedbe ob dotiku miške na figuro, s katero hoče igralec začeti igro.

Beseda R1 sporoči programu, naj izvede kodo za postavitev figur v hiško in čakanje na številko šest na kocki.

Če je figura pritisnjena in je število na kocki enako šest, potem se izvede drugi del kode za figuro. Pri uresničenem pogoju se v seznam zapiše novo mesto v seznamu od 1 do 44, ki ga uporabim za premikanje po polju.



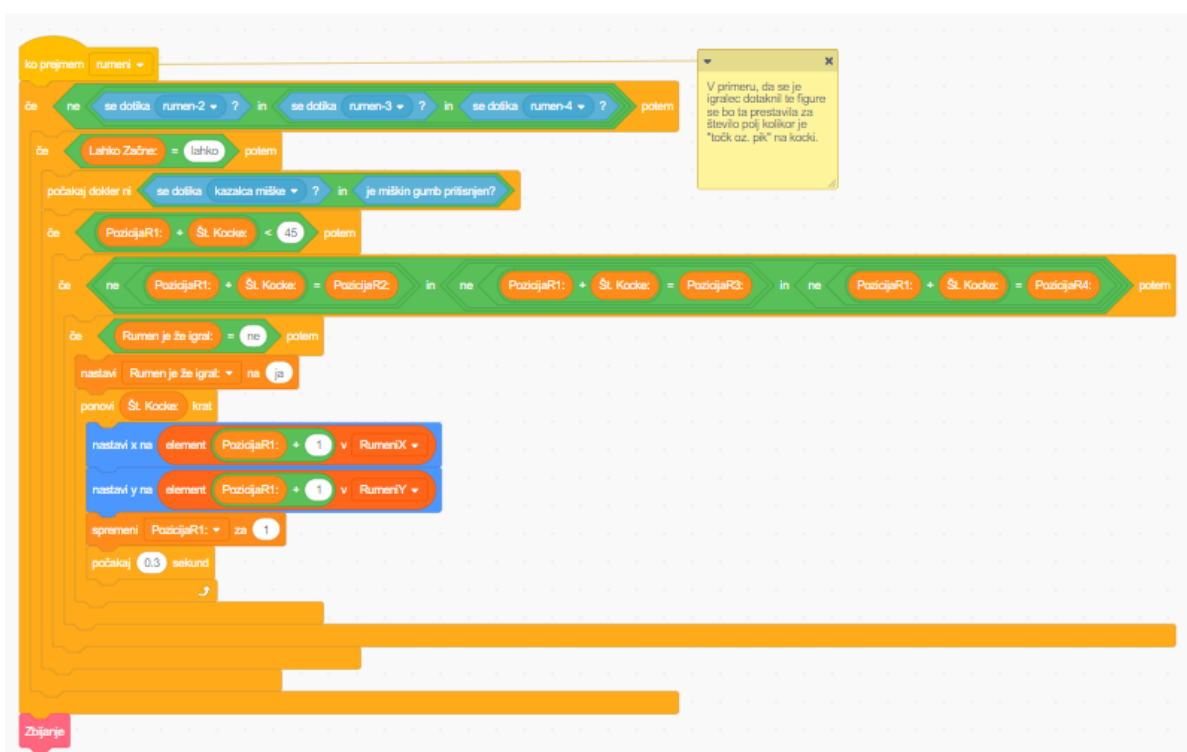
Slika 18: Program za postavitev figure glede na koordinate x in y

V prvem delu programa koda poskrbi, da se pozicija figure resetira na koordinate doma. Ta del programa je pomemben za vizualno postavitev figure.



Slika 19: Premik figure na začetno polje

V zadnjem delu prve kode figure sem umestil kodo, ki preverja, ali je bila figura pritisnjena in je številka na kocki enaka 6. V tem primeru se premakne na začetno polje.



Slika 20: Preverjanje pogojev za premikanje figure

V drugem delu program za figuro preveri, če se dotika kakšne druge figure, ki je iste barve. V tem primeru bi se premaknila le figura na najvišjem sloju oz. tista, ki je bližje domu.

Nato čaka na ponovni dotik, ki sproži reakcijo, da figura preveri, če bo njena končna destinacija manjša od 44 (odvisno od figure).

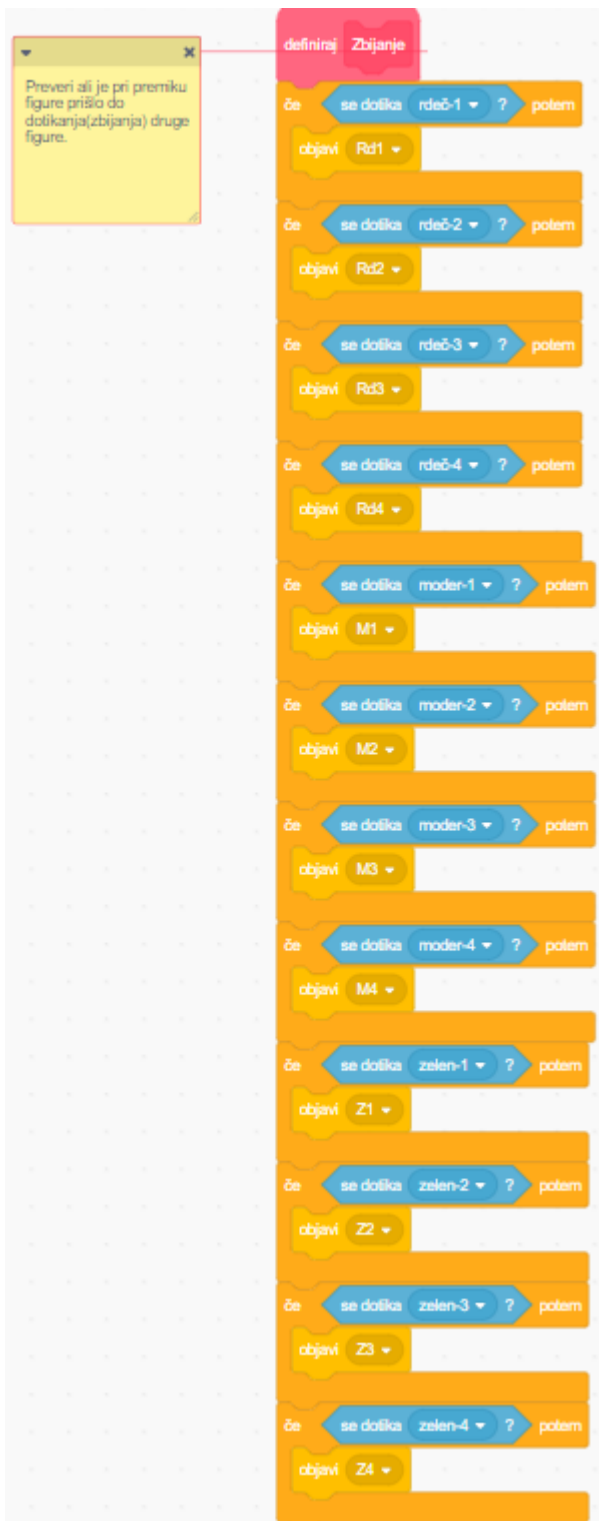
Nato se izvede premikanje ustrezne figure. Figure sem moral povezati s poljem (potjo) tako, da so vedele, kje in kako naj se premikajo. Problem sem za začetek rešil tako, da sem vsaki poziciji določil

ime (npr. gor=30, dol=-30, levo=-30, desno=30), a se je pojavila napaka, če je igralec figuro premaknil namesto kliknil, saj je figura nadaljevala s pozicije, na kateri je ostala, in ne s pozicije, na kateri je bila v osnovi.

Problema nisem mogel rešiti kar nekaj časa, dokler se nisem spomnil, da bi lahko namesto besed uporabil kar koordinate. Tako bi figura, tudi če je bila premaknjena, nadaljevala s pozicije, na kateri je bila. Hkrati pa bi ob premiku zgledala veliko bolj vizualno privlačna. Seveda je pri zapisu koordinat prišlo tudi do manjših težav, saj so morale biti vse koordinate natančno zapisane, sicer se je figura premikala po svoje. Po pregledu sem ugotovil, da bi zadevo lahko še poenostavil in bi vse figure iste barve imele isto tabelo s koordinatami, a je s to odločitvijo prišel nov problem. Ko je figura zaključila z igro in prišla v »dom«, so se figure začele nalagati ena na drugo, kar je povzročilo, da se igra ne zaključi, ko so vse figure ena za drugo, temveč bi bile vse štiri naložene ena na drugo, kar pa bi preskočilo naš pogoj ali pa bi se igra prehitro končala.

Reševanje tega problema mi je vzelo nekaj časa, saj sem se zavedal, da bi do te težave lahko prišlo tudi med igro. To na splošno ne bi bil problem, a sem hotel igro obdržati čim bolj realistično. Problem sem rešil tako, da sem v zapis figure vstavil blok, ki bi ob vsakem premiku pogledal lokacijo drugih figur. Če bi bila pozicija figur ista kakor pozicija figure, ki bi se je igralec dotaknil, potem se ta figura ne bi prestavila.

Še vedno je ostajal en problem – ko bi figura prišla v »dom«, bi vse figure šle na koordinate polja 44, zato bi bila enkrat na prvem mestu figura »rumen1«, drugič pa figura »rumen2«. S tem problemom se je program še bolj zapletel. Za pomoč sem naredil repliko programa in preizkusil različne načine, dokler nisem prišel do rešitve, ki sem jo uporabil v končni verziji. Med kodo sem umestil pogoj, da preden se figura premakne, pogleda, ali je položaj manjši od 44, 43, 42 ... Tako sem preprečil naključen zaključek figur, saj bi se igra vedno končala po vrsti (R1, R2, R3, R4).



Slika 21: Pogoji, kdaj zbijamo figuro

Na koncu sem definirjal tudi zbijanje figur različnih barv. V primeru, da figura pride na isto polje, kot je figura druge barve, jo program postavi nazaj v hiško.

5 Zapis dnevnika in uvedba sprememb

Vse spremembe svojih rešitev sem skrbno beležil v inženirskem dnevnik. Pri tem sem lahko sledil, na katerem koraku razvoja rešitve sem, kaj je potrebno spremeniti in kako sem težavo rešil.

Začetek	Datum	Problem	Opis problema
#0	September 2022	Definicija problema	Odločiti se za tip podlage za izdelavo igre človek-ne-jezi-se, najti podlago in začeti programirati.

#	Datum spremembe	Vrsta spremembe	Opis problema	Rešitev problema
#1	November 2022	Določanje poti	Povezovanje figure z njeno potjo	Imenovanje poti (gor, dol ...) in določanje spremembe koordinat.
#2	10. 11. 2022	Dotikanje figur 1.0	Neželeno dotikanje figur	Dodajanje bloka (pogoj) »if not touching«.
#3	13. 11. 2022	Dotikanje figur 1.1	Neželeno dotikanje figur ob začetku	Dodajanje pogoja, kida preveri, ali se bo ob izhodu iz hiške dotaknila katerih od soigralcev.
#4	14. 11. 2022	Določanje poti	Povezovanje figure s koordinatami	Namesto besed sem uporabil koordinate.
#5	29. 12. 2022	Konec Poti	Zaključevanje poti	Dodal sem blok, ki figuri pove, postavitve figur v domu. Pri tem se pot figur konča na 44. polju.
#6	3. 1. 2023	Dotikanje figur 2.0	Dotikanje figur	Dodal sem blok, da v primeru, da pride do stika se bo premaknila samo zgornja figura.
#7	6. 1. 2023	Menjava barv	Spreminjanje barv	S podaljšane verzije sem skrčil na manjšo in prvič uporabil blok če oz. sicer.
#8	13. 1. 2023	Prehitro premikanje	Premikanje je hitro	V kodo sem dodal, da figura počaka 0.3 sekunde, predno se premakne.
#9	18. 1. 2023	Določanje poti	Napačni koordinati	Figure so imele napačne koordinate in sem jih moral ročno popraviti.
#10	19. 1. 2023	Določanje poti	Premaknjeni koordinati	Figure so bile zamaknjene in jih je bilo potrebno premakniti za -4 v Y-u.
#11	3. 2. 2023	Odstranjevanje koordinat	Pomota v koordinatah	Pri prenavljanju kode je prišlo do pozabe odstranjevanja stare kode.
#12	4. 2. 2023	Zadnji popravki	Manjši popravki pred zaključitvijo	Pri programu je bilo še nekaj manjših napak pri koordinatah, ki sem jih hitro popravil.

Tabela 1: Dnevnik programiranja – sledenje spremembam

5.1 Ime projekta

Ime svojega projekta sem poimenoval Računalničar ne jezi se. Ime je sestavljeno iz osebe, ki programira (računalničar), in igre, ki sem jo naredil (človek ne jezi se). Tako sem s premetanko dobil ime Računalničar ne jezi se, kjer vsaka beseda poimenuje eno izmed dejavnosti, ki sem jo potreboval. Tako moje ime predstavlja **Računalničar** – vztrajnost, **Ne** – ne prenehaj, **Jezi** – izzivi problemov, **Se** – vse ti bo uspelo, če vztrajaš!

6 Sklepne ugotovitve

V sklepnih ugotovitvah bomo potrdili ali ovrgli hipoteze, ki smo si jih zastavili na začetku raziskovalne naloge.

H 1: Namizna igra je lahko narejena v računalniškem programu.

V času mojega raziskovanja sem uspel spisati program, ki mi omogoča igranje namizne igre človek ne jezi se. Pri tem sem opravil veliko testiranja, ampak ker je igra precej kompleksna, bi se še vedno lahko našla kakšna napaka, za katero verjamem, da bi uspel najti rešitev. Tako lahko hipotezo potrdim.

H 2: Namizno igro je mogoče grafično narediti v poenostavljenem računalniškem okolju, kot je Scratch.

V času mojega raziskovanja je bila ideja, da celotno igro spišem v programskem okolju Scratch. Pri tem sem želel uporabiti znano okolje, ki mi že ponuja grafični vmesnik. Razlika med programiranjem in grafičnim delovanjem je bila ogromna. Veliko lažje bi bilo program spisati le v številkah in ga predstaviti v tabeli. Pri samem programiranju sem ugotovil, da je bilo najtežje povezati podatke, ki sem jih imel za figuro, in jih grafično predstaviti. Ker sem to uspel rešiti z uporabo koordinatnega sistema in pametno postavitvijo ozadja, lahko hipotezo potrdim.

H 3: Računalniško igro bo lahko igralo poljubno število igralcev.

Smisel računalniških iger na spletu je ravno njihova fleksibilnost. Včasih lahko posamezno igro igramo na daljavo ali pa je naš nasprotnik preprosto računalnik. Ker Scratch okolje ne ponuja možnosti dvojne uporabe programa na takšen način, igre ne moremo igrati z več igralci. Prav tako na začetku nisem upošteval, da bi se izbralo število igralcev, saj takšna izbira celotno kodo zelo zakomplicira. Igra se lahko igra vedno med 4 igralci. Zadnjo hipotezo moram tako ovreči.

7 Zaključek

Z raziskovalno nalogo sem ugotovil, da je programiranje dokaj težko, a ne nemogoče. Pri programiranju je zelo pomembno analitično razmišljanje, podobno kot pri šahu. Problem je potrebno zaznati in predvideti, nato pa najti pot do rešitve. Vsak problem se da razdrobiti na manjše probleme in jih rešiti. Pri tem je pomembno, da se pri programiranju poiščejo podobni vzorci, kar olajša programiranje. Vseskozi pa je potrebno izbirati in se odločati, kako bomo posamezni program zapisali.

Spoznal sem, da je pametneje vnaprej razmisliti, kot pa iskati napake za nazaj in jih popravljati. Ugotovil sem, da je pomembna vztrajnost ne glede na vse izzive in probleme, ki jih programiranje prinese. Skozi izdelavo raziskovalne naloge sem spoznal različne vidike programiranja in uporabe Scratcha. Podrobno sem se seznanil s programskim okoljem in ugotovil, da ponuja veliko več, kot smo se učili v šoli. Spoznal sem nove načine razmišljanja in programiranja, kar mi bo v pomoč v prihodnosti.

Ugotovil sem, da pridobljeno znanje nikoli ni dovolj in da ga je potrebno vedno nadgrajevati. Velikokrat je pomembno to, da se naučimo pravilno razmišljati in opredeliti posamezne probleme. Ugotovil sem, da je računalništvo edini predmet, kjer so napake del učenja. Pri iskanju napak sem velikokrat rešil več napak oz. problemov, predvsem pa sem ves čas izpopolnjeval svoje znanje.

Zagotovo lahko zapišem, da se med raziskovalno nalogo nisem preveč jezil, ampak precej zabaval, kot se to spodobi za vse namizne ali računalniške igre.

8 Viri in literatura

Scratch (2023). Pridobljeno s 10. 11. 2022 s <https://scratch.mit.edu>.

Rodič, R. (2023). Računalničar-ne-jezi-se. Dostopno na spletnem naslovu: <https://scratch.mit.edu/projects/782539734/>.

Človek ne jezi se (Ludo game). Dostopno na spletnem naslovu: <https://www.mb.sik.si/clovek-ne-jezi-se.html>.

Kernc, A. (2022). Scratch za učenje programiranja: diplomska naloga. Pridobljeno 10. 10. 2022 s <http://revis.openscience.si/lzpisGradiva.php?id=8817>.

Bruning, K. (2022). Zgodovina družabnih iger: kaj je bila prva/najstarejša družbena igra? Pridobljeno 10. 1. 2023 s <https://gamecows.com/sl/zgodovina-dru%C5%BEdabnih-iger/>.