

UNIVERZA V LJUBLJANI
PEDAGOŠKA FAKULTETA
Študijski program: Matematika in računalništvo

INTERAKCIJA V
IZOBRAŽEVALNIH IGRAH
DIPLOMSKO DELO

Mentor:
prof. dr. Aleš Leonardis

Kandidatka:
Vesna Jovanovič

Ljubljana, september 2011

Zahvala

Za pomoč pri izdelavi diplomske naloge se za strokovno usmerjanje in nasvete zahvaljujem mentorju prof. dr. Alešu Leonardisu.

Posebna zahvala gre Nataši Kristan, ki je bila pripravljena sodelovati z mano. Njena spodbuda in nasveti so mi bili pri izdelavi diplomskega dela v veliko pomoč.

Prav tako se zahvaljujem vsem, ki so bili pripravljene testirati izobraževalno igro, predvsem Pavletu, Kseniji ter Urbanu.

Kazalo Vsebine

POVZETEK	III
ABSTRACT	IV
1 UVOD	1
2 INTERAKCIJA IN IZOBRAŽEVALNE IGRE	3
2.1 RAZREDNA INTERAKCIJA	3
2.2 INTERAKCIJA Z RAČUNALNIKOM	4
2.3 IZOBRAŽEVALNE IGRE	6
2.4 VPLIVI IGER NA UČENJE	7
3 PREDSTAVITEV IN IZDELAVA IZOBRAŽEVALNEGA GRADIVA	9
3.1 POTEK DELA	13
3.2 OPIS IZDELKOV	15
4 UPORABA PROGRAMA BLENDER PRI IZDELAVI IGRE	21
4.1 OPIS ORODJA ZA IZDELAVO IGRE	22
4.2 PRAKTIČNI PRIMER 1	25
4.3 PRAKTIČNI PRIMER 2	35
5 OCENA USPEŠNOSTI	42
5.1 POTEK PREVERJANJA	42
5.2 REZULTATI IN ANALIZA EKSPERIMENTA	43
5.3 REZULTATI IN ANALIZA ANKETE	52
6 ZAKLJUČEK	58
7 LITERATURA	59

Kazalo tabel

TABELA 1: ŠTEVILO UČENCEV GLEDE NA SPOL IN STAROST (1. ŠOLA)	44
TABELA 2: POVPREČNO ŠTEVILO TOČK, DOSEŽENIH PRI TESTU (1. ŠOLA)	44
TABELA 3: KOLIKO TOČK SO DOBILI UČENCI (1. ŠOLA)	44
TABELA 4: ŠTEVILO UČENCEV GLEDE NA SPOL IN STAROST (2. ŠOLA)	46
TABELA 5: POVPREČNO ŠTEVILO TOČK, DOSEŽENIH PRI TESTU (2. ŠOLA)	46
TABELA 6: KOLIKO TOČK SO DOBILI UČENCI (2. ŠOLA)	46
TABELA 7: ŠTEVILO UČENCEV GLEDE NA SPOL IN STAROST (3. ŠOLA)	48
TABELA 8: POVPREČNO ŠTEVILO TOČK, DOSEŽENIH PRI TESTU (3. ŠOLA)	48
TABELA 9: KOLIKO TOČK SO DOBILI UČENCI (3. ŠOLA)	48
TABELA 10: ŠTEVILO VSEH UČENCEV PO STAROSTI IN SPOLU [7]	50
TABELA 11: KOLIKO TOČK SO DOBILI UČENCI PRI POSAMEZNEM NAČINU UČENJA [7]	50
TABELA 12: ARITMETIČNA SREDINA, MEDIANA IN STANDARDNI ODKLON [7]	51
TABELA 13: ODGOVORI NA PETO VPRAŠANJE	53
TABELA 14: ODGOVORI NA SEDMO VPRAŠANJE	54

Kazalo slik

SLIKA 1: INTERAKCIJSKO OGRODJE (SKICA MODELA) [3].....	5
SLIKA 2: NASLOVNICA [7].....	11
SLIKA 3: NAVODILA ZA UČENCA [7]	11
SLIKA 4: NAVODILA ZA IGRANJE	15
SLIKA 5: PREMİK IZ PRVEGA NA DRUGI LIST	15
SLIKA 6: IGRA SEF – PRETVARJANJE IZ DESETIŠKEGA V DVOJIŠKO ŠTEVILO	16
SLIKA 7: IGRA SEF – PRETVARJANJE IZ DVOJIŠKEGA V DESETIŠKO ŠTEVILO	16
SLIKA 8: PALICE (VAJA) - PRETVARJANJE IZ DESETIŠKEGA V DVOJIŠKO ŠTEVILO	17
SLIKA 9: PALICE (TEST) - PRETVARJANJE IZ DVOJIŠKEGA V DESETIŠKO ŠTEVILO.....	17
SLIKA 10: TEHTNICA (VAJA).....	18
SLIKA 11: TEHTNICA (TEST)	18
SLIKA 12: PRETVARJANJE IZ DESETIŠKEGA V DVOJIŠKO ŠTEVILO	19
SLIKA 13: PRETVARJANJE IZ DVOJIŠKEGA V DESETIŠKO ŠTEVILO	19
SLIKA 14: IZBIRA ŠTEVILA	20
SLIKA 15: IGRA Z BONBONI	20
SLIKA 16: LOGOTIP PROGRAMA BLENDER.....	21
SLIKA 17: BLENDER, VERZIJA 2.49B.....	22
SLIKA 18: GUMB LOGIKA NAM PRIKAŽE NASTAVITVE ZA INTERAKCIJO	22
SLIKA 19: SPREMINJANJE FIZIKALNIH LASTNOSTI.....	23
SLIKA 20: DODAJANJE SPREMENLJIVK	23
SLIKA 21: SENZORJI, NADZORNIKI IN IZVRŠEVALCI SKUPAJ Z MOŽNIMI NASTAVITVAMI	24
SLIKA 22: OBJEKTI V IGRI.....	25
SLIKA 23: NASTAVITVE ZA KAMERO.....	26
SLIKA 24: NASTAVITVE ZA GUMB IZHOD	26
SLIKA 25: NASTAVITVE ZA OBJEKT <i>BINARNO</i>	28
SLIKA 26: NAPAČEN IN PRAVILEN IZPIS NA RAČUNALU.....	28
SLIKA 27: NASTAVITVE ZA OBJEKT PLANE.080.....	29
SLIKA 28: NASTAVITVE ZA GUMB S ŠTEVILKO 5.....	30
SLIKA 29: NASTAVITVE ZA GUMB BRIŠI	31
SLIKA 30: NASTAVITVE ZA OBJEKT REZULTAT	32
SLIKA 31: NASTAVITVE ZA GUMB POTRDI	33
SLIKA 32: NASTAVITVE ZA OBJEKT OBVESTILO	34
SLIKA 33: NASTAVITVE ZA GUMB, S KATERIM PONOVRNO IGRAMO IGRO.....	34
SLIKA 34: ZAČETNI POGLED IGRE	35
SLIKA 35: FIZIKALNE NASTAVITVE	36
SLIKA 36: NASTAVITVE ZA PREMIKANJE NAPREJ	36
SLIKA 37: NASTAVITEV ZA ZAVIJANJE V LEVO.....	37
SLIKA 38: NASTAVITVE ZA KAMERO.....	37
SLIKA 39: IZBRANI OBJEKT JE V IGRI NEVIDEN	38
SLIKA 40: PRIZOR SE PONOVRNO ZAŽENE OB DOTIKU OBJEKTA Z AVTOMOBILOM	38
SLIKA 41: PRIKAZ NOVEGA PRIZORA (02-KONEC) OB DOTIKU Z AVTOMOBILOM	39
SLIKA 42: NASTAVITEV ZA DRUGI GUMB.....	39
SLIKA 43: SKRIPTA POKAZI_KAZALEC.....	40
SLIKA 44: IZVAJANJE SKRIPTE, KI KAŽE MIŠKIN KAZALEC.....	40
SLIKA 45: NASTAVITVE ZA VRTENJE AVTOMOBILA.....	41

Povzetek

V diplomskem delu se posvečamo izobraževalnim računalniškim igram in njihovi uporabi pri pouku. Zanima nas predvsem razlika v znanju, ki ga učenci pridobijo pri frontalnem pouku, in znanju, ki ga učenci pridobijo z igranjem interaktivne izobraževalne igre. Na to vprašanje smo odgovorili s pomočjo dveh eksperimentalnih skupin. V ta namen smo izdelali izobraževalno igro, ki smo jo uporabili pri eksperimentu. Ugotovili smo, da so se učenci, ki so se učili snov na frontalen način, bolje odrezali pri testu kot učenci, ki so se učili z igro. Drugi cilj diplomske naloge je, da ugotovimo, kakšen odnos imajo študentje Pedagoške fakultete (smer Matematika-računalništvo) do uporabe interaktivnih izobraževalnih gradiv pri pouku. Odgovor na to vprašanje smo dobili s pomočjo ankete. Vsi anketiranci so sprejemljivi za uporabo interaktivnih izobraževalnih iger v učnem procesu. Pripravljene so se tudi potruditi in sami izdelati izobraževalno igro. Zadnji cilj diplomskega dela je predstaviti izdelavo igre s pomočjo programa Blender in s tem naučiti učitelje (in druge), kako narediti interaktivno izobraževalno igro.

V uvodu so predstavljeni cilji in hipoteze. Drugo poglavje govori o razredni interakciji, interakciji med uporabnikom in računalnikom ter o izobraževalnih igrah. V tretjem poglavju je predstavljen potek dela ter naša izobraževalna igra. Četrto poglavje opisuje program Blender in orodje za izdelavo igre. V tem poglavju predstavimo tudi dva konkretna primera, po katerih se lahko bralec nauči izdelati interaktivno igro. V zadnjem poglavju je opisano izvajanje eksperimenta in ankete ter rezultati.

Ključne besede: interaktivne izobraževalne igre, izobraževalna računalniška igra, interakcija, razredna interakcija, 3D grafika, izdelava igre, Game Engine, Blender, programski jezik Python

Abstract

The diploma thesis is devoted to educational computer games and their use in the classroom. We want to find out the difference in knowledge that pupils gain in frontal lessons and in knowledge acquired by playing interactive educational games. This question was answered with an experiment with two experimental groups. For this experiment we created an educational game. We have found that students who were taught by frontal method, perform better in the test than students who learned on their own with the game. The second objective of the thesis is to determine what attitude the students of the Faculty of Education (studying Mathematics-Computer Science) have towards the use of interactive educational materials for teaching. The answer to this question was obtained through survey. All respondents are acceptable for using interactive educational games in the learning process. They are also willing to work hard to create their own educational game. The last goal is to present the making of games using Blender, so the teachers (and others) can learn how to make an interactive educational game.

In the introduction, we present the goals and hypotheses. The second chapter describes classroom interaction, interaction between users and computers and educational games. The third chapter presents the documentation of our work and our game. The fourth chapter describes the program Blender and its tools for making games. This chapter also presents two examples in which the reader can learn how to make an interactive game. In the last section we describe the experiment, survey implementation and results.

Keywords: interactive educational games, educational game, interaction, classroom interaction, 3D graphics, making games, Game Engine, Blender, programming language Python

1 Uvod

Informacijsko-komunikacijska tehnologija (IKT) je postala pomemben del našega vsakdana. Vedno večjo vlogo ima v poslovnem svetu, našem prostem času, kulturi in izobraževanju. S pomočjo IKT smo postali globalna družba, med seboj lahko hitro in učinkovito komuniciramo. V izobraževanju se IKT uporablja v različne namene. Omogoča učenje na daljavo, pomaga povezati šole z drugimi šolami oziroma posameznike v teh šolah med seboj. Najpogostejši način uporabe IKT v izobraževanju pa je uporaba programske opreme, predvsem urejevalnikov besedil [12]. K programski opremi štejemo tudi računalniške igre. Izobraževalne video igre pomagajo pri učenju. Še posebej so učinkovite, če so zasnovane za reševanje problemov in kadar učijo razne spretnosti, zapisane v učnem načrtu. Video igre so prejele več kritik kot druge oblike učenja. Pogosto so igram očitali, da so namenjene samo zabavi ter da povečujejo socialno odtujenost [11].

Trenutno ni nobene obsežne raziskave, ki bi dokazala, da učenje s pomočjo IKT poveča individualne učne dosežke učencev [12]. Zato smo v tej diplomski nalogi želeli ugotoviti, ali učenci bolj razumejo snov, ki je predstavljena z izobraževalno interaktivno igro, ali snov, ki se obravnava na frontalen način. Za ta namen smo ustvarili tudi 3D izobraževalno računalniško igro, ki je namenjena učencem osmih in devetih razredov osnovne šole. Izdelava take igre je zelo dolgotrajno in kompleksno delo. Zato sva se z Natašo Kristan odločili za sodelovanje in izobraževalno igro naredili skupaj. V Natašini diplomski nalogi [7] je opisana izdelava objektov ter animacija, ta diplomska naloga pa govori o izdelavi interaktivne igre.

V diplomskem delu želimo preveriti dve hipotezi.

1. hipoteza: Učenci bolj razumejo snov, če je predstavljena z interaktivno izobraževalno igro, podkrepjeno s teoretičnimi spoznanji, kot pa če je predstavljena na frontalen način.
2. hipoteza: Bodoči učitelji matematike in računalništva so sprejemljivi za uporabo didaktičnih interaktivnih iger pri pouku.

Prvo hipotezo smo preverili s pomočjo eksperimenta, v katerega so bile vključene tri osnovne šole. Učence dveh osmih in enega devetega razreda smo razdelili v dve eksperimentalni skupini. Ena skupina se je o dvojiškem številskem sestavu učila klasično (frontalno), druga pa s pomočjo računalnika in izobraževalne igre. Znanje obeh skupin smo preverili s kratkim testom. Drugo hipotezo smo preverili z anketo. Za anketni vzorec smo izbrali študente četrtega in tretjega letnika ter absolvente Pedagoške fakultete, smeri matematika-računalništvo.

S pomočjo diplomske naloge želimo doseči naslednje cilje:

- Naučiti učitelje (in druge), kako s pomočjo posebnega programa naredijo interaktivno učno gradivo.
- Ugotoviti, ali je uporaba takega gradiva pri učenju uspešnejša od klasičnega pouka.
- Ugotoviti, kakšen odnos imajo študentje Pedagoške fakultete, smeri matematika-računalništvo, do uporabe interaktivnih izobraževalnih gradiv pri pouku.

Diplomi je priložena tudi CD zgoščenka, na kateri je zbranih nekaj programov (Blender, VLC media player, Burster) ter naši izdelki (igra, Blender datoteke, video vodiči).

2 Interakcija in izobraževalne igre

interakcija -e ž (á) knjiž. sodelovanje, medsebojno vplivanje (SSKJ)

Interakcija se pojavi, ko dva ali več objektov vplivajo drug na drugega. To je lahko kakršnakoli komunikacija - pogovor med dvema ali več ljudmi, komunikacija med skupinami, organizacijami, državami. Interakcija je lahko tudi povratna informacija, ki jo dobimo od stroja, na primer računalnika [13].

Z izrazom razredna interakcija običajno označujemo medsebojno delovanje učiteljev in učencev, ki vsak s svojim vedenjem delujejo drug na drugega. Obsega tako besedno kakor nebesedno komunikacijo med učenci in učiteljem [9].

Izobraževalne igre so igre, ki so bile posebej zasnovane za poučevanje o določeni temi, za razširjanje konceptov, razumevanje zgodovinskih dogodkov in kulture ter za pomoč pri učenju določene spretnosti [11].

2.1 Razredna interakcija

Študije, ki so raziskovale interakcijo v razredu, so se pojavile že v petdesetih in šestdesetih letih. Te zgodnje raziskave so se osredotočale predvsem na interakcijo med učiteljem in razredom. Izkazalo se je, da se v razredni interakciji največkrat pojavlja vzorec pobuda - odgovor - povratna informacija. Takšno razredno interakcijo največkrat nadzoruje učitelj. Učitelj spodbudi interakcijo tako, da postavi vprašanje. Ko dobi učenčev odgovor, pa zaključi z interakcijo tako, da poda povratno informacijo o odgovoru. Spremembe v razredni interakciji so bile usmerjene predvsem v spremembo vloge posameznika pri konstruiranju znanja. Strukturirane debatne vzorce je počasi zamenjalo dinamično poučevanje in pogovori, ki so bližje vsakdanjim pogovorom. Poudarjati se je začela aktivna vloga učencev v razredni interakciji. Učitelj naj bi učence samo vodil in usmerjal. Učenje je namreč rezultat načrtnega vodenja bolj usposobljenega posameznika in sodelovanja pri aktivnostih učne skupine [8].

Raziskave so pokazale, da v državah, kjer učenci dosegajo izredno dobre rezultate pri mednarodnih testih, učitelji uporabljajo metode, s katerimi spodbujajo razredno interakcijo. Te metode povečujejo pozornost in aktivno sodelovanje učencev. Učitelj začne učno uro s predstavitvijo nove informacije, nato pa učenci sodelujejo v dialogu, povedo svoje ideje in mnenja, sprašujejo in razlagajo ostalim učenem. Razredna interakcija spodbuja učenje, saj učenci aktivno sodelujejo, izražajo svoje mnenje in dobijo povratno informacijo.

Da bi bil tak način poučevanja učinkovit, mora učitelj znati pritegniti vse učence v aktivnost s spodbujanjem in direktnimi vprašanji. Paziti mora, da zanimanje ne upade, ter se ustrezno odzivati na reakcije učencev[10].

2.2 Interakcija z računalnikom

Ljudje pridobivamo informacije iz okolja prek čutil – vida, sluha, tipa, okusa, vonja. V interakciji z računalnikom pa uporabljamo predvsem vid, sluh in tip. Računalnik nam podaja informacije s pomočjo izhodnih naprav, najpogosteje prek zaslona in zvočnikov. Računalniku pošiljamo ukaze prek vhodnih naprav, največkrat z vnašanjem besedila in kazanjem (klikanjem). Pri tem najpogosteje uporabljamo tipkovnico, miško, sledilno ploščico in ekran na dotik. Najnovejša tehnologija pa že omogoča upravljanje računalnika z očmi [3].

Da je interakcija med računalnikom in uporabnikom uspešna, mora uporabniški vmesnik dobro prevajati med uporabnikovim in računalniškim jezikom. Bolj podrobno si lahko interakcijo med uporabnikom in računalnikom pogledamo skozi različne modele interakcije, ki so jih opisali avtorji Dix, Finlay, Abowd in Beale [3]. Prvi model se imenuje Cikel izvedba - vrednotenje (*ang. Execution-evaluation cycle*), drugi pa Interakcijsko ogrodje (*ang. The interaction Framework*).

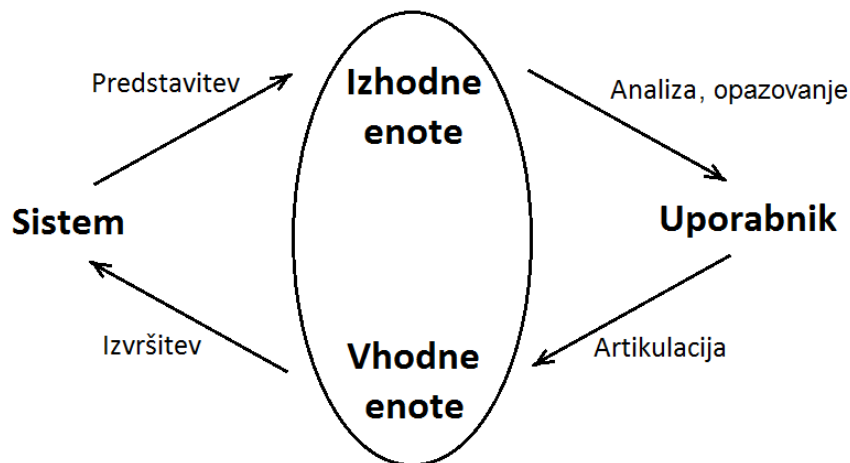
a) Cikel izvedba - vrednotenje se razdeli v dve fazi – izvajanje in vrednotenje. Ti dve fazi pa se razdelita v skupno 7 stopenj. Te stopnje so:

1. zastavljanje cilja,
2. določanje namena,
3. določanje zaporedij dejanj,
4. izvajanje akcij/dejanj,
5. zaznavanje stanja sistema,
6. interpretiranje stanja sistema,
7. evalvacija stanja sistema v povezavi s cilji in nameni.

Vse te stopnje so odvisne od uporabnika. Najprej si uporabnik zastavi cilj – kaj je potrebno narediti. Cilj je lahko nenatančen, zato se mora podrobneje določiti še z namenom in zaporedji dejanj, s katerimi bo uporabnik cilj dosegel. Po izvedbi teh dejanj, uporabnik zazna novo stanje sistema in si ga razlaga glede na svoja pričakovanja. Če smo z računalnikom dosegli cilj, je interakcija končana. V nasprotnem primeru si mora uporabnik zastaviti nov cilj in ponoviti vse faze oziroma cel cikel.

b) V modelu Interakcijskega ogrodja interakcija vsebuje štiri komponente: sistem, uporabnik, vhodne in izhodne naprave. Zadnji dve komponenti tvorita vmesnik med sistemom in uporabnikom. V tem interaktivnem ciklu so štirje koraki, vsak ustreza eni povezavi med komponentami.

1. Artikulacija: Uporabnik prične interakcijo s formuliranjem cilja in dejanjem, s katerim bo dosegel cilj. Edini način, da uporabnik upravlja z računalnikom, je prek vhodnih naprav.
2. Izvršitev: Ukazi, ki jih uporabnik sporoča računalniku, se prevedejo v računalniku razumljiv jezik. Sistem nato te ukaze izvede.
3. Predstavitev: Po izvajanju ukazov je sistem v drugačnem stanju, ki ga mora prikazati uporabniku prek izhodnih naprav.
4. Opazovanje, analiza: Uporabnik mora dojeti in ovrednotiti novo stanje sistema glede na zastavljeni cilj. S tem korakom se cikel tudi konča [3].



Slika 1: Interakcijsko ogrodje (skica modela) [3]

2.3 Izobraževalne igre

Danes živimo v informacijski dobi, kjer nas tehnologija spremlja na vsakem koraku, zato jo vse pogosteje srečujemo tudi na področju izobraževanja. Ena izmed razlik med industrijsko dobo in današnjo digitalno dobo je zagotovo uporaba računalniških iger in simulacij v izobraževanju. Igre in simulacije vsebujejo model sistema, kjer učenci lahko opazujejo posledice svojih dejanj. Simulacija predstavlja vsak poskus posnemanja realnega ali namišljenega okolja ali sistema. Igre pa različni avtorji definirajo različno. Definirajo jih kot organizirano igranje, kot aktivnost, pri kateri igralec sledi predpisanim pravilom in si prizadeva doseči zastavljen cilj ali pa kot tekmovalno dejavnost, ki je kreativna in v njej uživamo, je omejena s pravili in zahteva določene spretnosti. Poznamo veliko vrst iger, mi pa se bomo osredotočili na izobraževalne igre [5].

Ker je eden izmed ciljev diplomske naloge tudi naučiti bralca izdelati izobraževalno igro, si najprej pogledajmo, katerih devet elementov mora po Gagneju vsebovati vsaka dobra igra.

1. Pridobivanje pozornosti (sprejem)

Tu govorimo predvsem o tem, kako pritegniti igralca, še preden začne igrati. Na to moramo biti pozorni predvsem pri predstavitvi igre. To lahko naredimo tako, da na začetku prikažemo elemente igre ali ustvarimo uvodni videoposnetek.

2. Obveščanje učencev o cilju (pričakovanja)

Ponavadi je razlaga cilja del neke zgodbe, ki se odvija v ozadju in opisuje pogoje za dosego cilja. Tudi to lahko predstavimo, preden učenec prične igrati igro z reklamo, napovednikom ali v uvodnem samem igri.

3. Spodbujanje priklica predhodnega znanja (priklic)

Četudi v igri ni izrecno navedeno, se danes vsi igralci zavedajo, da igra vsebuje več stopenj oziroma ravni. Nova raven naj se povezuje s prejšnjimi, gradi naj na znanju, ki ga je učenec pridobil v prejšnjih ravneh. Na začetku nove stopnje igre lahko opišemo, kaj se pričakuje od učenca, da že zna.

4. Predstavitev dražljajev (selektivno zaznavanje)

Ta vidik igre govori o spodbujanju in izzivih. Ključni element je, da mora biti vidik predstavljen tako, da obdrži učenca v igri. Če učenec ne ve točno, kaj mora v dani situaciji narediti, bo kmalu postal razočaran in nehal igrati. Kadar učenec v igri ne ve, kako naprej, naj mu igra ponudi namig ali sporočilo, ki ga še enkrat opomni o cilju.

5. Zagotavljanje učnega usmerjanja (semantično kodiranje)

Igra mora biti samostojna. Igralec ne uporablja priročnikov, niti nima vedno ob sebi moderatorja, ki bi mu povedal, kako igrati. Napotki, kako se igra igra, morajo biti v igri sami.

6. Sproženje izvršitve (odzivanje)

To je bistvena sestavina interaktivnosti, brez tega igre ni. Fizični vmesniki za igranje iger pri vseh igrah podobni. Kako pa se dejansko igra igra, je odvisno od posamezne igre.

7. Zagotavljanje povratnih informacij (okrepitev)

To je eden glavnih elementov igre. Brez pravočasne povratne informacije igralec ne more vedeti, ali napreduje. Povratno informacijo lahko podamo na več načinov; z rezultati, meniji, zvočnimi signali in verbalnimi povratnimi informacijami.

8. Ocenjevanje uspešnosti (pridobivanje)

Ker je vsaka igra neke vrste tekmovanje, je doseganje cilja razlog za igranje. Potovanje do cilja je pomembno, a vseeno je rezultat tisti, ki je na koncu najpomembnejši.

9. Ohranjanje in prenašanje (posplošitev)

Napredovanje po ravneh v igri zahteva, da učenci uporabljajo zanje in spretnosti iz prejšnjih ravni. Če so spretnosti, ki so zahtevane za doseg cilja neke ravni, popolnoma drugačne od znanja in spretnosti iz prejšnjih ravni, igra ni dobra [5].

2.4 Vplivi iger na učenje

Nekatere igre so nezahtevne, saj za doseg cilja zahtevajo samo preproste spretnosti (priklic besednih ali vizualnih elementov). Pri takih igrah lahko cilj dosežemo tudi z ugibanjem. Igre, pri katerih je pomembna hitrost, zmanjšujejo možnost, da se igralec ustavi in kritično razmisli. Pri igranju zabavnih iger igralec ne nameni veliko časa razmišljanju in refleksiji [5].

Po drugi strani pa s pomočjo iger igralci razvijajo vizualne sposobnosti, kot sta prostorska predstava in vizualna pozornost. Drugi pozitivni učinki iger na učenje so tudi kritično razmišljanje in reševanje problemov, oblikovanje smiselnih sklepov, različne strategije raziskovanja in opazovanje. Otroci, ki igrajo veliko iger, lahko obdelujejo veliko informacij naenkrat ter uporabljajo alternativne poti, da pridejo do informacij. Ko se soočijo z novo situacijo, se ne ustavijo, ampak se učijo s poizkušanjem in napakami. Rajši pridejo do rešitve sami, kot pa da bi o

njej poslušali ali brali. Aktivno sodelovanje igralca ter interaktivnost iger zagotavljata učinkovito učenje. Igre zagotavljajo veliko povratnih informacij, kar je ključnega pomena za učenje [5].

Igre motivirajo učence, da prevzamejo odgovornost za svoje učenje. Kako močno so igralci motivirani, lahko preverimo s štirimi kriteriji. Dobra igra mora igralcu ponuditi pravo mero izziva, fantazije, radovednosti in nadzora. Igra, ki vsebuje vsa ta štiri merila, bo dobro motivirala učence vseh različnih učnih stilov. Izzivi v igri naredijo igro zanimivo, poleg tega pa izzivi z različno težavnostjo učenca še bolj pritegnejo. Fantazija z namišljenimi konteksti v igri poveča navdušenje, medtem ko radovednost ponuja zanimive in nove kontekste, ki spodbujajo učence, da raziskujejo neznano. Nadzor v igri pa daje učencem občutek samoodločanja [5].

V splošnem ljudje uživajo v igrah, kadar:

- zmorejo doseči cilj, vendar ne preveč zlahka,
 - je naloga za dosego cilja 'poštena', vsi igralci imajo enake možnosti za zmago,
 - tveganje za neuspeh ni preveliko, a je vseeno prisotno,
 - je dovolj pozitivne povratne informacije (nagrade za dosežek), ki se mora pojaviti takoj, ali pa je prikazana pot do cilja in napredovanje,
 - je prisotna tudi negativna povratna informacija,
 - obstaja nekaj naključnih elementov (zmanjša se občutek krivde pri neuspehu)
- [5].

3 Predstavitev izobraževalnega gradiva

S kolegico Natašo Kristan smo izdelali izobraževalno igro s pomočjo programskega orodja Blender. Cilj igre je, da se učenec spozna z dvojiškim številskim sistemom ter da zna pretvarjati števila iz desetiškega sistema v dvojiški sistem in obratno. Igra je namenjena učencem osmih in devetih razredov osnovne šole. Lahko pa jo igra vsak, ki se želi spoznati z dvojiškim številskim sistemom, in vsak, ki želi svoje znanje o tej temi utrditi. Uporabljajo jo lahko tudi učitelji pri kakšnem krožku ali dodatni uri, saj snovi ni več v učnem načrtu. Za lažjo predstavo zgradbe in poteka igre sledi najprej diagram igre z opisom nato pa še bolj podroben diagram.

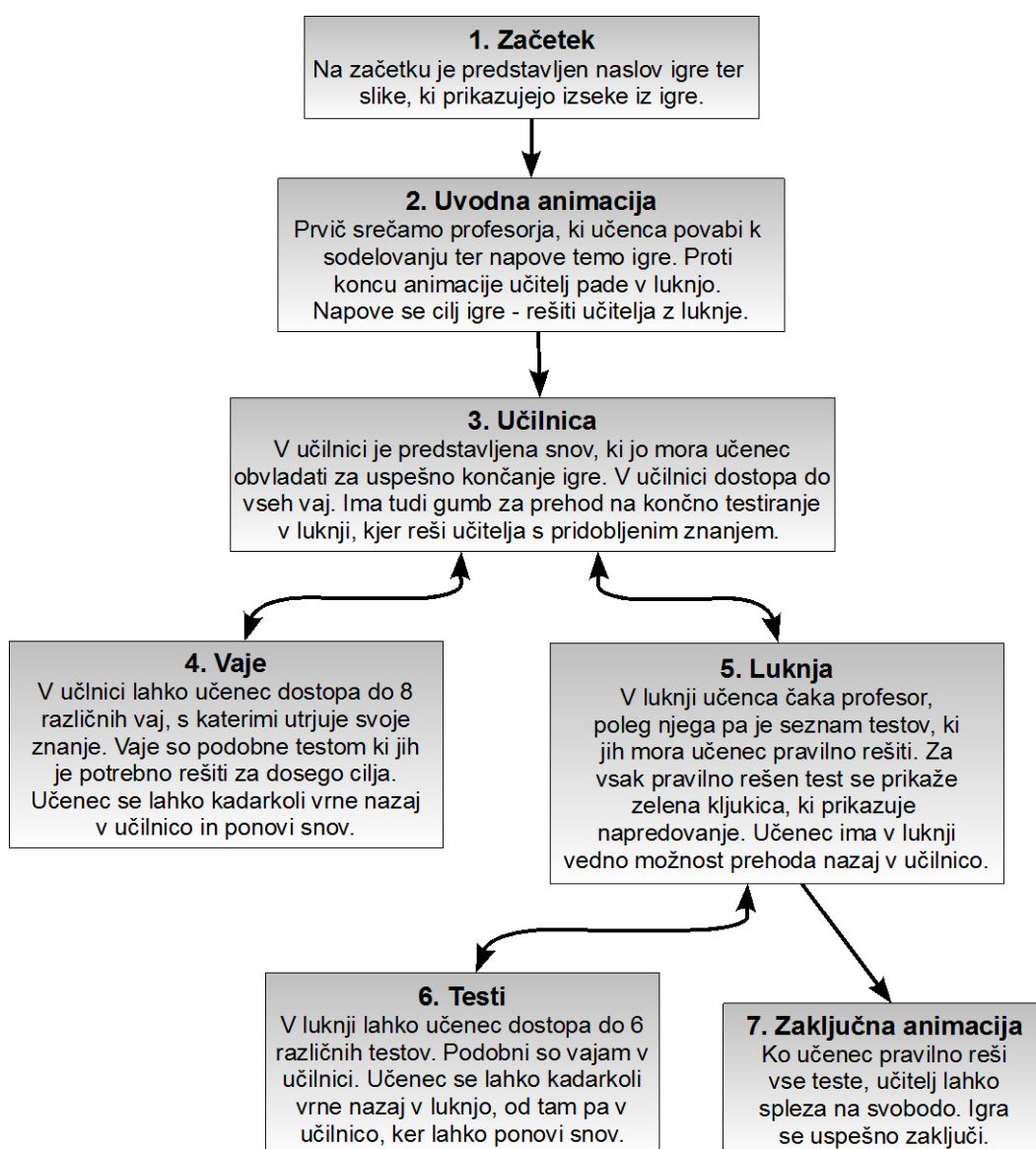


Diagram 1: Diagram igre - puščice prikazujejo možen prehod med različnimi deli igre.

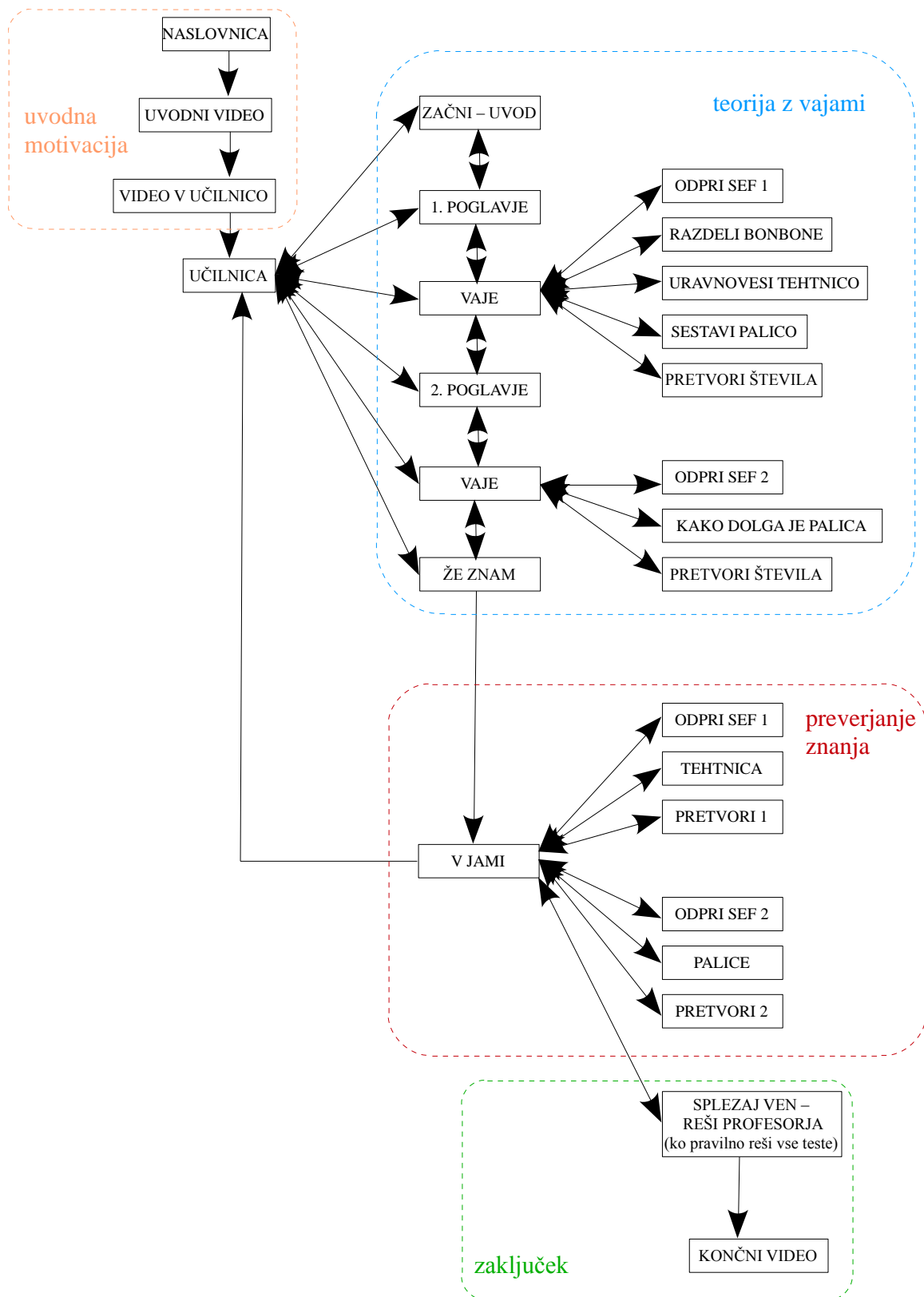


Diagram 2: Diagram igre [7]

Na začetku učenec zagleda uvodno animacijo, kjer ga učitelj pozdravi in povabi, naj mu sledi. Še preden pa učitelj odpre vrata učilnice, pade v luknjo in tam ostane ujet. Učenčeva naloga je, da reši učitelja iz luknje s svojim znanjem o dvojiškem številskem sistemu. Igra nato učenca popelje v učilnico, kjer si na tabli lahko prebere teorijo in vadi za testiranje. V učilnici lahko prosto preskakuje med poglavji in vajami. Ko je učenec prepričan, da zna že dovolj, ga igra pripelje nazaj do profesorja, ki je ujet v luknji. Tu mora rešiti šest kratkih iger oziroma testov, ki so zelo podobni vajam. Če pri testiranju ugotovi, da še ne zna dovolj, ima vedno možnost, da se vrne v učilnico in ponovi snov. Za vsak pravilno rešen test dobi učitelj kos vrvi, ki mu pomaga do svobode. Ko učenec pravilno reši vse teste, sledi zaključna animacija – učitelj po vrvi spleza na svobodo.

Bolj podrobno si pogledjmo igro glede na devet Gagnejevih elementov, opisanih v poglavju 2.3.

1. Pridobivanje pozornosti

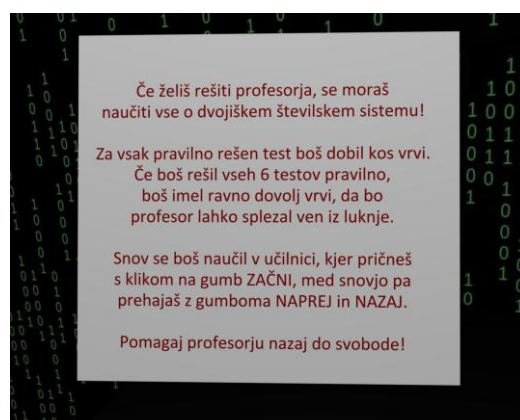
Preden učenec začne igrati, zagleda naslovnico igre. Na njej je prikazana glavna oseba v igri ter nekaj slik, ki prikazujejo prizore iz igre.



Slika 2: Naslovnica [7]

2. Obveščanje učencev o cilju

Zgodba, ki se odvija v ozadju igre, je prikazana v začetni animaciji. Na koncu animacije si učenec lahko prebere, kaj je cilj igre in kako bo prišel do cilja.



Slika 3: Navodila za učenca [7]

3. Spodbujanje priklica predhodnega znanja

Prva raven v igri je učilnica, kjer se učenec nauči snov. Druga raven z vajami zahteva, da uporabi znanje, ki ga je pridobil v učilnici. Tretja raven pa je reševanje testov. Testi niso enaki vajam, so pa zelo podobni, zato lahko učenec zopet uporabi znanje iz predhodnih ravni.

4. Predstavitev dražljajev

Učenca motiviramo že z začetno animacijo in ciljem. Med samo igro mu za utrjevanje znanja ponujamo veliko izbiro iger. Vsaka igra je videti drugače, cilj pa je enak – pretvoriti število iz dvojiškega v desetiški sistem in obratno. Navodila za igranje posamezne igre so prikazana pred začetkom vsake igre. Če učenec ne zna rešiti določenega testa, se lahko kadarkoli vrne nazaj k reševanju vaj in v učilnico, kjer lahko ponovi snov. Pri reševanju vaj učenca za vsak pravilno rešen primer pohvalimo in spodbudimo, naj reši še kateri primer.

5. Zagotavljanje učnega usmerjanja

Na začetku igra napoti učenca v učilnico. Snov je razdeljena po poglavjih, tako da lahko učenec enostavno sledi snovi. Igra omogoča tudi preskoke na katerikoli del snovi, saj predpostavljamo, da se učenec želi vrniti v učilnico in ponoviti poljuben del snovi. Kako pravilno rešiti vajo oziroma test, je napisano v navodilih pred začetkom igre.

6. Sproženje izvršitve

Pri igri učenec za izvajanje ukazov uporablja miško. Na mestih, kjer lahko učenec klikne in pričakuje določeno reakcijo, se miškin kazalec spremeni v simbol roke.

7. Zagotavljanje povratnih informacij

Pri reševanju vaj in testov se ob vsakem napačnem odgovoru zasliši poseben zvok. Prav tako pa se na ekranu izpiše, da je odgovor napačen. Pri testiranju se je za vsak primer možno zmotiti trikrat, nato pa učenca igra avtomatsko premakne k učitelju v luknjo. Za vsak pravilen odgovor se prav tako zasliši zvok in na zaslonu se izpišejo spodbudne besede, ki učencu povedo, da je primer pravilno rešil in da lahko nadaljuje z igro.

8. Ocenjevanje uspešnosti

Več testov učenec pravilno reši, bolj je uspešen v igri. Za doseg cilja pa mora pravilo rešiti vseh šest testov. Če učenci igrajo igro hkrati, na primer pri učni uri, lahko med seboj tekmujejo v hitrosti.

9. Ohranjanje in prenašanje

Spretnosti in znanje, ki jih učenec pridobi na različnih ravneh igre, so si podobni. Možnost, da bi nekdo uspešno končal igro z ugibanjem, je zelo majhna. Učenec mora uporabiti znanje iz učilnice. To znanje nadgradi s pomočjo vaj. Vaje sicer niso nujno potrebne za doseg cilja, so pa učencu v zelo veliko pomoč.

Igralci bodo v igranju igre uživali, saj ustreza vsem kriterijem (razen zadnjemu), ki smo jih navedli v poglavju 2.4. Cilj je lahko doseči, a ne brez truda. Ker se pri testiranju učenec lahko zmoti samo trikrat, je možnost doseganja cilja z ugibanjem precej majhna. Vsi igralci imajo enako izhodišče. Tisti, ki se prvič srečajo z dvojiškim številskim sistemom, se lahko o tej temi sami naučijo v učilnici. Igralec, ki pa že nekaj ve o tej temi, lahko teorijo tudi preskoči. Če učenec razume, kaj so dvojiška števila in zna pretvarjati med dvojiškim in desetiškim sistemom, bo po vsej verjetnosti pravilno rešil vaje in teste. Vseeno pa obstaja majhno tveganje za napačne odgovore, saj se vsak človek kdaj zmoti. Pozitivna in negativna povratna informacija se pojavita ob pravem času, takoj ko učenec poda odgovor.

Igro si lahko ogledate na CD zgoščenki, ki je priložena diplomski nalogi. Za izvajanje igre potrebujete operacijski sistem Windows (XP, Vista ali 7) in vsaj 1 GB pomnilnika. Okrnjena verzija igre pa se nahaja na spletni strani <http://hrast.pef.uni-lj.si/~drbinarko/>. Za igranje igre na spletu pa potrebujete vtičnik Burster. Dobite ga na spletni strani <http://geta3d.com> za operacijski sistem Windows (za brskalnike Mozilla Firefox, Internet Explorer, Google Chrome ter Opera) in Linux (za brskalnike Mozilla Firefox, Opera in Google Chrome).

3.1 Potek dela

V tem razdelku so predstavljeni koraki pri izdelavi izobraževalne interaktivne igre [7]. Razdelitev dela je razvidna iz priimkov, ki so zapisani v oklepajih.

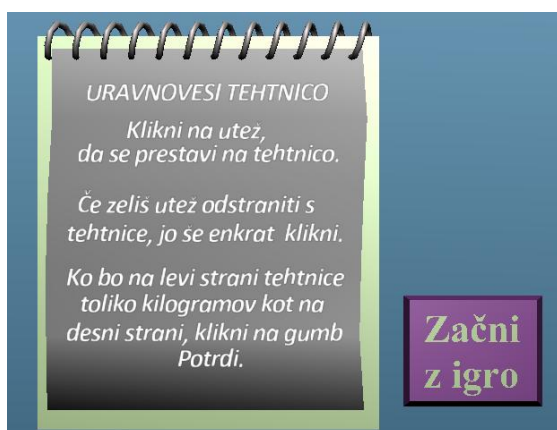
1. Izbira teme in zbiranje idej.
2. Učenje programa Blender.
3. Razdelitev dela in izdelava izdelkov.
 - uvodni motivacijski video (Kristan):
 - objekt profesorja (telo, obleka, lasje, teksture, kosti, animacija),
 - snemanje začetnega govora, ki ga dodamo profesorju,
 - prizor hodnika (teksture sten, rože, vrata),
 - prizor jame (teksture ozadja, navodila).

- video ob prihodu v učilnico (Kristan):
 - prizor učilnice (mize, stoli, kateder, rože, omare, tabla, stene, teksture, gumbi, animacija).
 - učilnica:
 - objekti gumbov, puščici naprej in nazaj (Kristan),
 - tabla z vso teorijo (Kristan),
 - vaje ob zaključku vsakega teoretičnega sklopa (Jovanovič),
 - vaje po prvem sklopu: Odpri sef 1, Razdeli bonbone, Uravnovesi tehtnico, Sestavi palico, Pretvori števila,
 - vaje po drugem sklopu: Odpri sef 2, Koliko je dolga palica, Pretvori števila,
 - izris vseh objektov, animacija, interakcija.
 - jama s testi:
 - animacija profesorja, ki čaka (Kristan),
 - gumbi s testi, dodajanje vrvi za pravilno rešen test, vsi testi v celoti (Jovanovič),
 - testi: Odpri sef 1, Tehtnica, Pretvori 1, Odpri sef 2, Palice, Pretvori 2 (Jovanovič).
 - končni video (Kristan):
 - animacija profesorja, ki pleza na svobodo,
 - okolica (vrv, trava),
 - dodajanje zvoka.
4. Testiranje testov in popravljanje napak (Jovanovič).
 5. Povezava vseh datotek in videov v eno samo datoteko (Jovanovič).
 6. Testiranje igre in popravljanje napak (Kristan, Jovanovič).
 7. Naslovnica igre (Kristan).
 8. Test za učence (Kristan).
 9. Testiranje učencev na osnovnih šolah (Kristan, Jovanovič).
 10. Analiza testov (Kristan, Jovanovič).
 11. Anketna vprašanja in osnutek spletne strani (Kristan, Jovanovič).
 12. Implementacija spletne strani na šolskem strežniku (Kristan).
 13. Spletna anketa (Jovanovič).
 14. Posodobitev igre za spletno verzijo (Jovanovič).
 15. Nalaganje igre in videov na spletni strežnik (Jovanovič).
 16. Analiza anket iz spletne strani (Kristan, Jovanovič) [7].

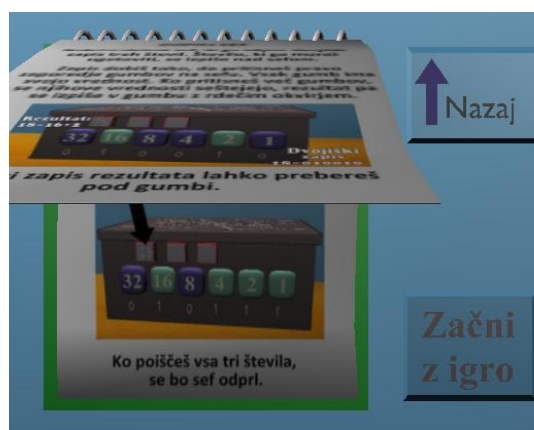
3.2 Opis izdelkov

Naredili smo pet osnovnih kratkih iger, kjer se učenec uči pretvarjati iz desetiškega v dvojiško število in obratno. Pri vseh igrah učenec za interakcijo z računalnikom uporablja miško. Te igre smo morali tudi spremeniti, saj so igre namenjene tako vajam kot testiranju in pretvarjanju iz enega sistema v drugega. Težavnost iger se stopnjuje, najprej učenec sestavlja števila do 63, nato pa do 255. Na začetku vsake igre je navodilo za igranje. Pri igrah za testiranje se štejejo tudi napake, tako da je možnost ugibanja čim manjša. Pri vsakem pravilnem oziroma napačnem odgovoru igralca opozori tudi zvok. Zvočne posnetke smo dobili na spletni strani www.soundjay.com. Na koncu je iz petih osnovnih iger nastalo vsega skupaj 14 iger, 8 za vajo in 6 za testiranje. Sledijo opisi teh petih osnovnih iger. Ne bomo pa posebej poudarjali, da se ob vsakem odgovoru (pravilen ali nepravilen) zasliši določen zvok.

Preden začnemo, si oglejmo še, kako so predstavljena navodila za igranje. Pri vseh igrah so predstavljena na enak način. Na listu papirja no napisana kratka navodila. Učenec pa začne igrati igro, ko klikne na gumb v desnem spodnjem kotu. V primerih, ko je navodilo dolgo za dva lista, lahko učenec med njima prehaja z gumboma na desni strani.



Slika 4: Navodila za igranje



Slika 5: Premik iz prvega na drugi list

- Sef

Ta igra ima 4 različice – dve za pretvarjanje iz desetiškega v dvojiško število (eno za vajo, eno za test) ter dve za pretvarjanje iz dvojiškega v desetiško število. Cilj igre je ugotoviti kombinacijo treh števil, ki odprejo sef. Vsak gumb na sefu ima svojo vrednost (1, 2, 4, 8, 16, ali 32). Da se sef odpre, mora učenec pretvoriti tri naključna števila. Vsako od teh števil odklene eno ključavnico. Ključavnica je predstavljena kot kvadrat nad gumbi. V vsakem trenutku pa lahko učenec igro

zapusti s klikom na gumb v desnem zgornjem kotu. Iz spodnjih slik se zelo dobro vidi razlika med vajo in testom. Vaja ima za pomoč na gumbih napisane vrednosti, medtem ko so na gumbih pri testu te vrednosti izbrisane.



Slika 6: Igra sef – pretvarjanje iz desetiškega v dvojiško število



Slika 7: Igra sef – pretvarjanje iz dvojiškega v desetiško število

Pri igri, kjer mora učenec pretvoriti število iz desetiškega v dvojiško, se najprej nad sefom izpiše naključno število, ki odpre eno ključavnico. To število moramo sestaviti s pomočjo gumbov na sefu. Učenec mora z miško klikati po gumbih. Ko učenec klikne na gumb, se izvrši veliko akcij. Pod gumbom se spremeni vrednost v nič ali v ena, gumb pa spremeni barvo in pozicijo. Če je gumb aktiviran, kar pomeni, da je pod njim število ena, se vrednost tega gumba prišteje k vrednosti v kvadratu nad gumbi. Zaporedje ničel in enic pod gumbi predstavlja dvojiški zapis števila, ki je trenutno zapisan v kvadratu. Ko je število v kvadratu enako številu nad sefom, mora učenec klikniti na kvadrat, da izbiro potrdi in lahko nadaljuje z naslednjim številom. Če se števili ne ujemata, se na sefu izpiše beseda Nepravilno.

Pri pretvarjanju iz dvojiškega v desetiško število se nad sefom izpiše dvojiški zapis števila, ki odpre sef. Gumbi na sefu so aktivirani tako, da predstavljajo to naključno dvojiško število. Uporabnik tokrat ne more klikati na gumbe, pač pa lahko klika na štiri puščice nad sefom, s katerimi sestavi iskano število. Ob klikanju na puščice se spreminja število pod oziroma nad puščico in hkrati tudi število v kvadratu nad gumbi (ključavnica). Učenec lahko klikne tudi na gumb z napisom Potrdi. Ta gumb je namenjen preverjanju. Ob kliku na ta gumb se preveri, ali je dvojiški zapis nad sefom res enak številu, ki ga je s pomočjo puščic sestavil učenec. Če se števili ujemata, učenec lahko nadaljuje s pretvarjanjem novega števila. V nasprotnem primeru pa se na sefu izpiše opozorilo, da je učenčev odgovor nepravilen.

Ko se sef končno odpre, učenec v njem najde zlatnike.

- Palice

Ta igra ima 3 različice – dve za vajo in eno za testiranje. Pri vaji za pretvarjanje iz desetiškega v dvojiško število lahko učenec sestavi palico dolgo največ 63 enot (na voljo so palice dolžine 1, 2, 4, 8, 16 in 32 enot). Pri vaji in testiranju za pretvarjanje dvojiškega števila v desetiško, pa je stvar malce otežena. Na voljo sta dve palici več (za 64 in 128 enot). Zopet lahko učenec igro kadarkoli zapusti s klikom na gumb v desnem zgornjem kotu.



Slika 8: Palice (vaja) - pretvarjanje iz desetiškega v dvojiško število



Slika 9: Palice (test) - pretvarjanje iz dvojiškega v desetiško število

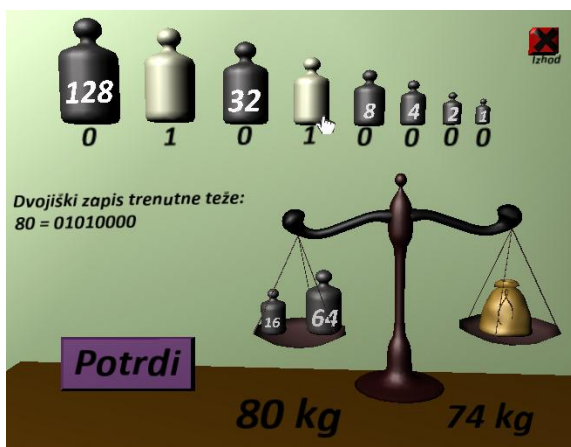
Pri pretvarjanju števila iz desetiškega v dvojiški sistem se na začetku pojavi obvestilo. To obvestilo učencu pove, kako dolgo palico mora sestaviti. Po treh sekundah se obvestilo samodejno prestavi na dno ekrana, tako da ima učenec navodilo vedno pred seboj. Poleg tega je na vrhu ekrana ravnilo s puščico, ki učenca opozarja, kako dolgo palico mora sestaviti. Palico sestavi s pomočjo manjših palic različnih dolžin tako, da klika nanje. Ob kliku na palico se spet izvede veliko akcij. Število pred palico se spremeni iz nič v ena in obratno – iz ena v nič. Če je palica izbrana, postane zamegljena. Dolžine vseh izbranih palic se seštevajo. Seštevek se izpisuje v številu nad palicami. Najdebelejša palica pod ravnalom grafično prikazuje trenutno dolžino sestavljene palice. Njena dolžina se spreminja z vsakim klikom. Dvojiški zapis trenutne velikosti palice se izpisuje levo od pravokotnika s palicami. Ko učenec klikne na gumb Potrdi, se preveri, ali je palica res dolga toliko, kot je igra na začetku zahtevala. V vsakem primeru se na ekranu izpiše, ali je odgovor pravilen ali napačen. Pri pravilnem odgovoru učencu sporočilo ponudi možnost ponovnega igranja ali pa izhod iz igre.

Drugi dve različici igre sta namenjeni pretvarjanju iz dvojiškega v desetiško število. Ena je namenjena vaji, druga pa testiranju. Pri igri se na vrhu izpiše dvojiški zapis, ki predstavlja dolžino palice. Manjše palice v pravokotniku, pred katerimi je zapisano število ena, skupaj sestavljajo dolžino palice, ki jo iščemo.

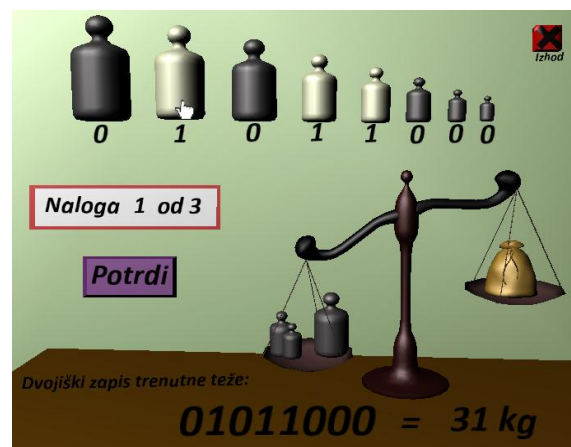
Učenec mora to število poiskati in ga sestaviti s pomočjo puščic poleg gumba Potrdi. Ob kliku na puščico se spreminja samo število pod puščico oziroma nad njo. S klikom na gumb Potrdi se preveri, ali je učenčev odgovor pravilen ali ne. Pri napačnem odgovoru se pod gumbom Potrdi izpiše beseda Nepravilno. Pri pravilnem odgovoru pa se pojavi sporočilo, ki učenca spodbuja, naj reši še kateri primer (vaja), oziroma ga pohvali in prikaže se naslednji primer (test). Pri testu mora učenec pravilno pretvoriti tri števila.

- Tehtnica

Ta igra ima dve različici, obe za pretvarjanje iz desetiškega v dvojiško število. Ena različica je za vajo, druga za testiranje. Učenec ima na voljo tehtnico in uteži, s katerimi lahko sestavi števila do 255 (utež za 1, 2, 4, 16, 32, 64 in 128 kilogramov). Na desni strani tehtnice je že vreča z naključno težo. Učenec pa mora postaviti uteži na levo stran tehtnice tako, da bo tehtnica uravnotežena.



Slika 10: Tehtnica (vaja)



Slika 11: Tehtnica (test)

Razliki med vajo in testom sta dve. Prva je ta, da uteži pri testu nimajo napisanih vrednosti. Drugo razliko pa lahko opazimo pri izpisovanju trenutne teže na levi strani tehtnice. Pri vaji je teža zapisana v desetiškem številu, pri testu pa v dvojiškem. Učenec lahko klika na uteži, gumb za izhod ter na gumb Potrdi. Najprej si pogledjmo, kaj vse se zgodi, ko učenec klikne na neko utež. Utež na vrhu se zamegli, na levi strani tehtnice pa se pojavi ta utež. Vrednost vseh izbranih uteži se sešteva pod levo stranjo tehtnice. Odvisno od razmerja med levo in desno stranjo tehtnice se spreminja tudi nagib tehtnice. Če je na levi strani večja teža kot na desni, se tehtnica prevesi na levo stran, tako kot je prikazano na sliki 11. S klikom na gumb Potrdi, se preveri, ali je tehtnica uravnotežena. Če je leva stran enaka desni, se učencu prikaže sporočilo, ki ga pohvali. Pri vaji ga tudi spodbudi, da reši še kateri primer, pri testu pa ga igra preusmeri na naslednji

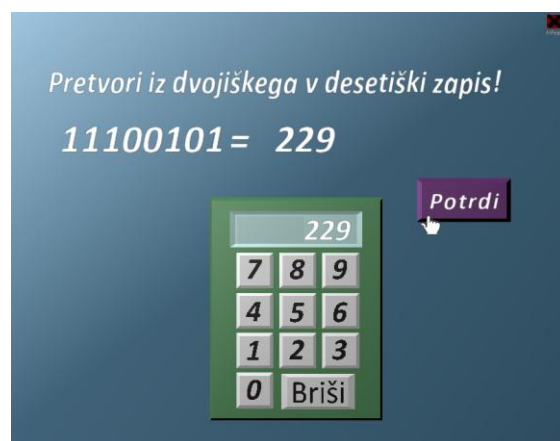
primer. Tudi tokrat pri napačnem odgovoru učenec dobi takojšnjo povratno informacijo prek sporočila. Pri vaji lahko naredi neomejeno število napak, pri testu pa ga tretji napačen odgovor opozori, naj ponovi snov, in igra s tehtnico se konča.

- Pretvarjanje

Pri tej igri gre za klasično pretvarjanje. Učenec ima zapisano desetiško število in mora ugotoviti, kakšen je njen zapis v dvojiškem sistemu, in obratno. Ta igra ima 4 različice. Pri vsaki mora učenec pravilno pretvoriti tri naključna števila od 0 do 255. Pri vseh različicah igre se na vrhu ekrana izpiše navodilo in naključno število, ki ga je potrebno pretvoriti. Pretvorjeno število učenec vnese s pomočjo računalu. Učenec lahko klika na gumbе na računalu, na gumb za izhod ter na gumb Potrди. Ko učenec klikne na število na računalu, se ta zapiše na zadnje mesto v pravokotnik nad številkami. Če je pravokotnik že zapolnjen, se prvo število v pravokotniku izbriše. Gumb Briši na računalu briše iz desne strani, torej pobriše zadnje število, ki ga je učenec vnesel. Ob kliku na gumb Potrди se izvede enaka operacija kot v prejšnjih primerih – učenec dobi povratno informacijo o pravilnosti odgovora.



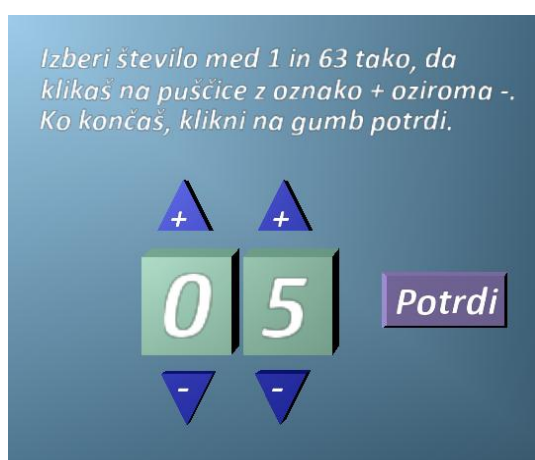
Slika 12: Pretvarjanje iz desetiškega v dvojiško število



Slika 13: Pretvarjanje iz dvojiškega v desetiško število

- Bonboniera

Ta igra ima samo eno različico, ki je namenjena vaji. Učenec si izbere poljubno število bonbonov med 0 in 63. Te bonbone nato poizkuša razdeliti v škatle (velikosti za 1, 2, 4, 8, 16 ali 32 bonbonov) tako, da so polne. Število bonbonov tokrat ni naključno, učenec si ga pred začetkom igre izbere sam. Na vrhu ekrana je izpisano navodilo (Slika 14). S klikanjem na štiri puščice z oznakami + in - učenec določi število. Izbrano število ne more biti večje od števila 63, prav tako pa ne more biti enako nič. K klikom na gumb Potrdi se igra prestavi na novo prizorišče (Slika 15). Na vrhu ekrana je izpisano število bonbonov, ki jih mora učenec še razdeliti v škatle. Škatla se napolni z bonboni, če je na razpolago dovolj bonbonov, torej če je število na vrhu ekrana enako ali večje od števila bonbonov, ki jih lahko sprejme določena škatla. Škatlo učenec napolni oziroma izprazni tako, da klikne na napis pod škatlo. Če škatle ni mogoče napolniti, se to tudi izpiše na ekranu. Poleg tega se zasliši tudi zvok, škatla pa zasveti v rdeči barvi. Učenec lahko klikne tudi na gumb za izhod ter na gumb Končaj. V primeru, da učenec klikne na gumb Končaj in še ni razdelil vseh bonbonov v škatle, ga igra na to tudi opozori. Če pa je pravilno razdelil vse bonbone, ga igra s sporočilom pohvali ter ponudi dve možnosti –ponovno igranje ali izhod iz igre.



Slika 14: Izbira števila



Slika 15: igra z bonboni

4 Uporaba programa Blender pri izdelavi igre

„Blender je odprtokodna programska oprema za 3D modeliranje, animacije, izrisovanje, postprodukcijo, interaktivno ustvarjanje in predvajanje. Služi kot zamenjava za komercialne rešitve, kot so Maya, Softimage in Cinema 4D.“ [1]

Tehnologija hitro napreduje, tudi Blender se razvija. Trenutno (20. 8. 2011) je najnovejša verzija 2.59, ki je na voljo za operacijski sistem Windows, Mac OS X, Linux in FreeBSD. Blender pri izdelavi iger omogoča pisanje in uporabo skript, ki jih lahko uporabniki napišemo v programskem jeziku Python [2]. Program, skupaj z vsemi navodili za namestitev, lahko dobite na spletni strani www.blender.org, kjer si lahko začetniki zelo dobro pomagajo tudi z različnimi primeri in navodili za uporabo.



Slika 16: Logotip programa Blender
(Vir: <http://www.blender.org/blenderorg/blender-foundation/logo/>)

3D grafika je zapleteno področje. Vsa programska oprema, ki se ukvarja s 3D grafiko ima ogromno gumbov, nastavitev, možnosti in edinstven način dela. Med vsemi programi za delo s 3D grafiko, velja Blender za enega izmed težje razumljivih programov. Običajno ga ne priporočajo začetnikom [6].

V diplomski nalogi ne bomo opisovali osnov, kot so premikanje v 3D prostoru, uporaba bližnjic, izdelava preprostih objektov, itd. Osredotočili se bomo na interakcijo in izdelavo interaktivnih aplikacij. Preden se lotite izdelovanja interaktivne aplikacije, si pogledjte osnove dela s programom Blender v diplomski nalogi kolegice Nataše Kristan [7]. Igro smo izdelali v verziji 2.49b. Vse slike in navodila v tej diplomi se torej nanašajo na starejšo verzijo 2.49b, ki pa je podobna verziji 2.59.

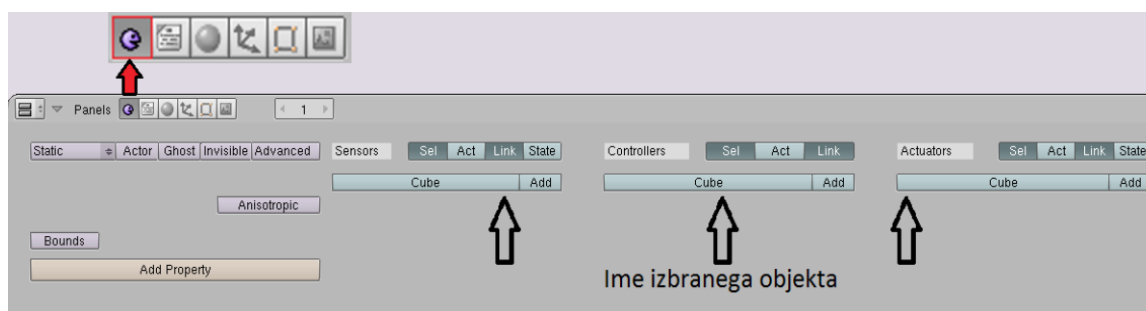


Slika 17: Blender, verzija 2.49b

4.1 Opis orodja za izdelavo igre

V spodnjih poglavjih sta opisana dva primera, s katerima bomo predstavili, kako s programom Blender izdelamo igro. Prvi primer opisuje nastavitve za igro, ki je opisana v poglavju 3.2. Predstavljenih je tudi nekaj skript, napisanih v programskem jeziku Python, ki so uporabljene v igri. V drugem primeru je predstavljena izdelava 'dirkalne' igre. Zadnji primer smo dodali zato, ker lahko z njim predstavimo še nekaj dodatnih uporabnih funkcij. Najprej se bomo spoznali z orodjem, s katerim izdelamo igro.

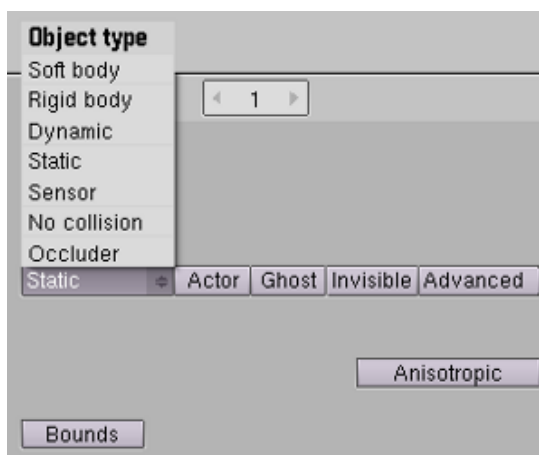
Blender ima vgrajeno močno programsko orodje, imenovano *Game Engine*, ki omogoča izdelavo interaktivnih 3D aplikacij. Namenjen je predvsem razvijanju iger. Vse, kar potrebujemo za izdelavo igre, se skriva pod gumbom *Logika*, ki je na sliki 4 obkrožen z rdečo.



Slika 18: Gumb Logika nam prikaže nastavitve za interakcijo

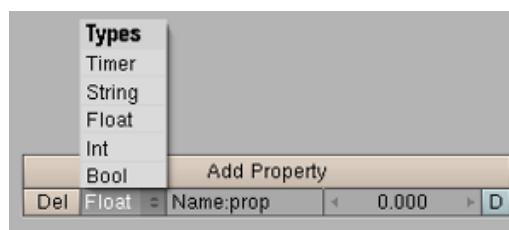
Kar se pokaže pod gumbom *Logika*, je odvisno od tega, kateri objekt imamo trenutno izbran (objekti, za katere še nismo nič nastavljali, bodo seveda brez nastavitvev). Ime izbranega objekta se izpiše v treh svetlomodrih pravokotnikih. Za vsak objekt lahko tudi spreminjamo lastnosti in nastavitve.

a) Fizikalne lastnosti določajo, kako se objekt obnaša v igri (kako nanj vpliva gravitacija, kaj se zgodi ob stiku z ostalimi objekti, itd.). Nahajajo se v spodnjem levem delu zaslona.



Slika 19: Spreminjanje fizikalnih lastnosti

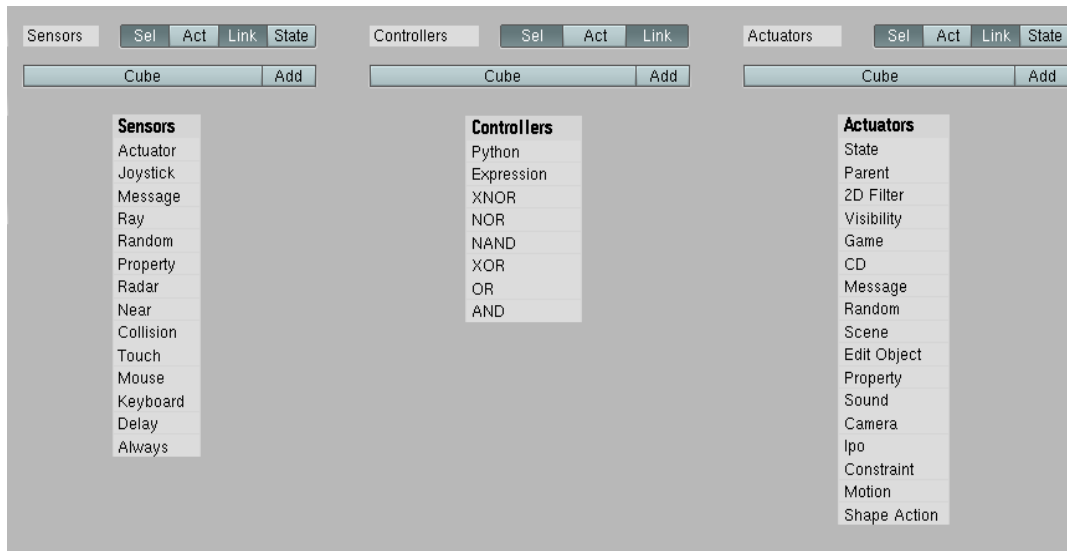
b) Vsakemu objektu lahko dodamo več spremenljivk z gumbom *Add Property*, ki se nahaja pod fizikalnimi lastnostmi. Spremenljivko, ki jo dodamo, lahko izbrisemo (*Del*), ji določimo tip (*Timer*, *String*, *Float*, *Int*, *Bool*), spremenimo ime in določimo začetno vrednost. Z aktiviranjem gumba *D* se med izvajanjem igre izpisujejo vrednosti spremenljivke.



Slika 20: Dodajanje spremenljivk

c) Desno od Fizikalnih lastnosti so trije logični deli: senzorji (*ang. Sensors*), nadzorniki (*ang. Controllers*) in izvrševalci (*ang. Actuators*). S senzorji se pričnejo vse akcije. Ko se senzor sproži (pritisnemo gumb na tipkovnici, kliknemo, poteče določen čas, itd.), pošlje signal do nadzornika. Če je senzor aktiven, pošlje pozitiven signal, drugače pa negativni. Nadzornikova naloga je, da preverja in kombinira te signale, da se lahko sproži ustrezna reakcija. Dejanje, ki se izvede, ko so senzorji aktivirani, je zapisano v izvrševalcu (objekt se premakne, predvaja se zvok, itd.). Če želimo dodati senzor, nadzornika ali izvrševalca, moramo pod to rubriko poleg imena objekta klikniti na gumb *Add*.

Čisto vseh možnosti, ki jih lahko nastavimo v senzorjih, nadzornikih in izvrševalcih, ne bomo podrobneje opisovali. Nekaj jih bomo spoznali sproti v naslednjih poglavjih s konkretnimi primeri.



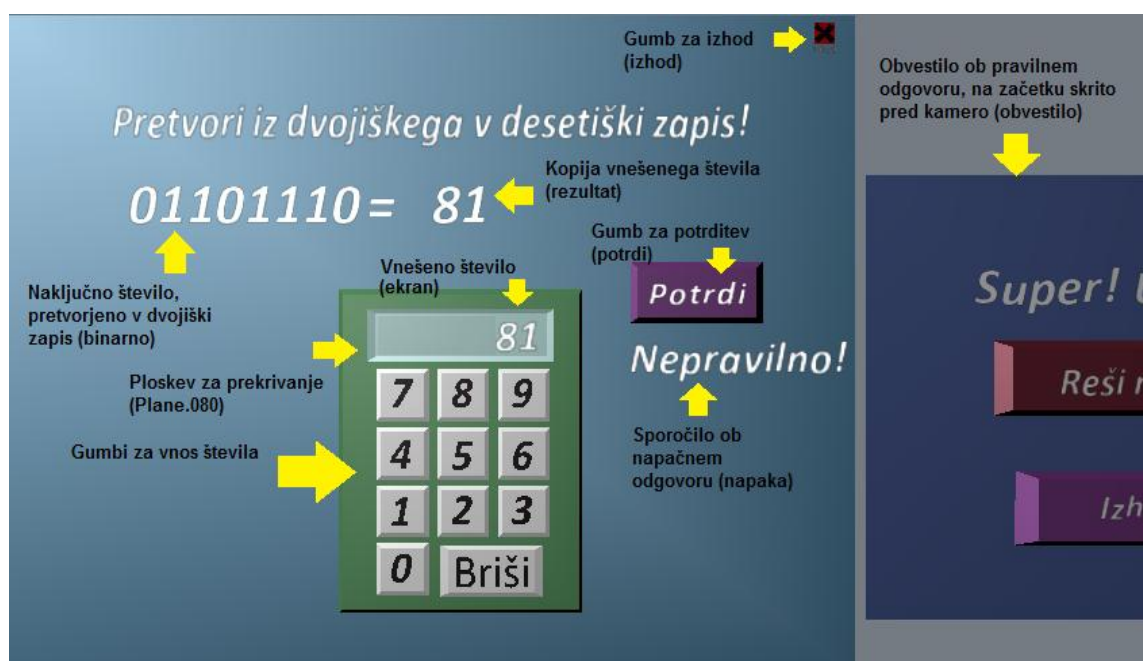
Slika 21: Senzorji, nadzorniki in izvrševalci skupaj z možnimi nastavitvami

Če želimo igro zagnati kar v programu Blender, moramo imeti miškin kazalec postavljen v delu okna, kjer je prizor, in pritisniti tipko *P* na tipkovnici. Za izhod iz igre pa moramo pritisniti tipko *ESC* [4].

4.2 Praktični primer 1

V tem poglavju bomo opisali konkreten primer uporabe orodja Game Engine. Za predstavitev smo si izbrali igro, ki smo jo v poglavju 3.2 poimenovali Pretvarjanje. Izbrali smo različico, narejeno za pretvarjanje iz dvojiškega v desetiško število, namenjeno vaji. Igro bomo obravnavali kot samostojno, kar pomeni, da se ob kliku na gumb izhod igra zaključi. V primeru, ko je igra povezana v celoto, se ob kliku na izhod prikaže učilnica. Dodana programska koda je napisana v programskem jeziku Python, komentarji v kodi pa so označeni z znakom #.

Najprej si pogledjmo, katere objekte bomo potrebovali. Na sliki kaže nanje rumena puščica, ime posameznega objekta pa je zapisano v oklepaju.

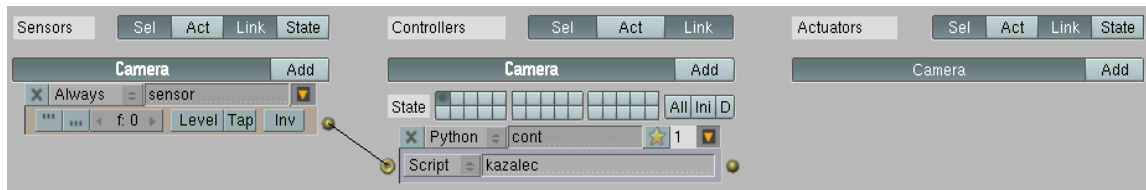


Slika 22: Objekti v igri

Da bomo miškin kazalec lahko videli v igri, moramo napisati skripto. Odpremo Blenderjev urejevalnik besedil (pogled *Text Editor*) in ustvarimo nov prazen dokument. Preimenujemo ga v *kazalec*. Vanj vpišemo naslednji dve vrstici:

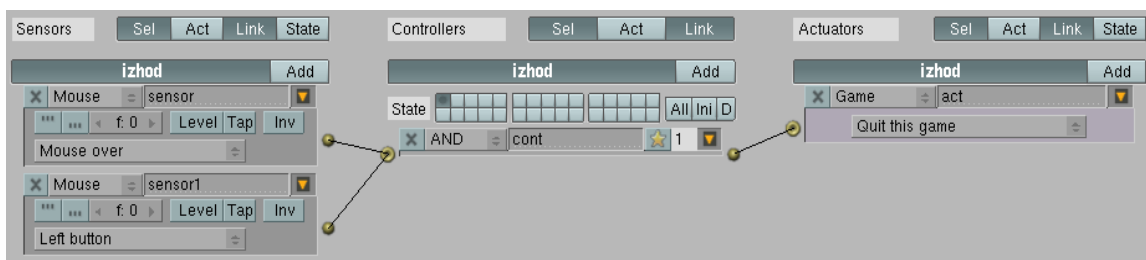
```
import Rasterizer
Rasterizer.showMouse(1)
```

Nato izberemo katerikoli objekt, ki je v igri vedno navzoč, na primer kamero. Dodamo senzor in nadzornika. Povežemo rumeno piko na desni strani senzorja z rumenim obročem na levi strani nadzornika. Senzorja ne spreminjamo, spremenimo samo nadzornika. Možnost *And* spremenimo v *Python*. V prazno polje poleg besede *Script* vpišemo ime naše skripte (*kazalec*).



Slika 23: Nastavitve za kamero

Sedaj lahko izberemo gumb za izhod. Dodamo dva senzorja, enega nadzornika in enega izvrševalca. Oba senzorja povežemo z nadzornikom, nadzornika pa z izvrševalcem. Obema senzorjema spremenimo možnost *Always* v možnost *Mouse*. Pri prvem spremenimo možnost *Left button* v *Mouse over*. Izvrševalcu spremenimo možnost *Motion* v *Game*, nato pa izberemo še možnost *Quit this game*.



Slika 24: Nastavitve za gumb Izhod

Sedaj se lahko posvetimo glavnemu delu igre. Začnimo z objektom *binarno*, ki prikazuje binarni zapis naključnega števila, ki ga želimo pretvoriti v desetiško število. Objekt *binarno* ima spremenljivko tipa *string* (niz) z imenom *Text*, v katerega se shrani dvojiški zapis. Zopet si bomo morali pomagati s skripto. V urejevalniku besedil ustvarimo nov dokument in ga poimenujemo v *random*. Sledi komentirana koda, ki naključno število pretvori v zaporedje ničel in enic ter to zaporedje shrani v spremenljivko *Text*.

```

#Uvozimo modul Mathutils, ki omogoča uporabo matematičnih funkcij.
import Mathutils

#Definiramo funkcijo, s katero dobimo naključno število.
def stevilo():
    #Naključno število med 0 in 255 shranimo v globalno spremenljivko stevilo
    (globalno spremenljivko definiramo tako, da pred njo dodamo 'GameLogic.').
    GameLogic.stevilo=int(Mathutils.Rand(0.0, 1)*255)
    #Če je število enako 0, 1, 2, 4, 8, 16 ali 32 (sicer bi bila naloga prelahka),
    poiščemo novo število.
    if (GameLogic.stevilo<=2) or (GameLogic.stevilo==4) or GameLogic.stevilo==8)
    or (GameLogic.stevilo==16) or (GameLogic.stevilo==32):
        stevilo()

#Kličemo funkcijo za naključno število.
stevilo()

#Dostopamo do nadzornika, ki izvaja skripto, ter ga shranimo v spremenljivko cont.
cont = GameLogic.getCurrentController()
#Prek tega nadzornika lahko dostopamo do objekta binarno in ga shranimo v
spremenljivko own.
own = cont.owner
#Spremenljivka objekta binarno z imenom Text postane prazen string.
own['Text']="

#Naključno število shranimo v novo spremenljivko.
n=GameLogic.stevilo
#Naključno število pretvorimo v dvojiški zapis - niz shranimo v spremenljivko Text v
objektu binarno.
while n > 0:
    own['Text'] = str(n % 2) + own['Text']
    n = n >> 1

#Pred dvojiški zapis postavimo toliko ničel, da bo v zapisu 8 znakov.
for i in range(len(own['Text']), 8):
    own['Text']="0"+own['Text']

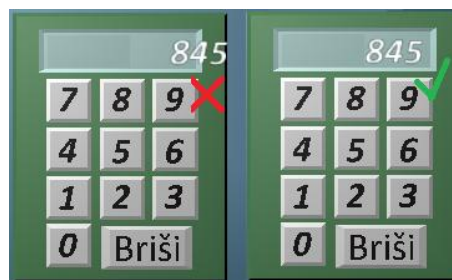
```

Objektu *binarno* dodamo senzor in nadzornika ter ju povežemo. V senzorju nastavimo možnost *Delay*. V nadzorniku pa spremenimo možnost *And* v *Python*. V prazno polje poleg besede *Script* vpišemo ime naše skripte (*random*).



Slika 25: Nastavitve za objekt *binarno*

Preden si pogledamo nastavitve za gumbe na računalu, moramo razložiti še, kako se spreminja število (objekt z imenom *ekran*) na računalu in kako je povezano s ploskvijo za prekrivanje (*Plane.080*). Vsakemu nizu se znaki dodajajo na koncu, z desne strani. Pri računalu pa moramo ta niz še premakniti v levo, sicer število pade iz okvirja (slika 26).

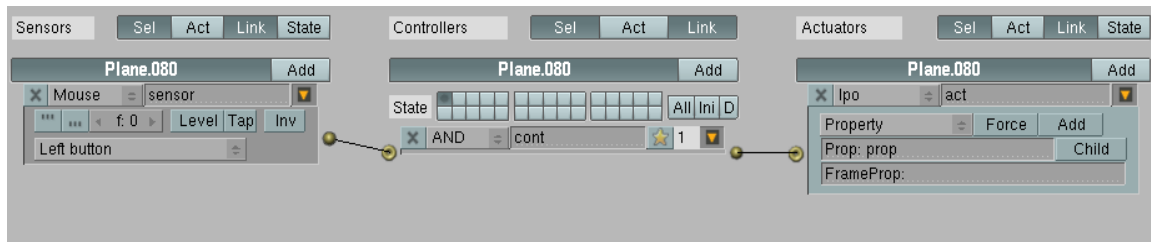


Slika 26: Napačen in pravilen izpis na računalu

Objekt z imenom *ekran* ima spremenljivko

Text, kamor se zapisuje število, ki ga vnese učenec. Na začetku smo tej spremenljivki nastavili vrednost sedmih ničel in jo postavili na skrajni levi del pravokotnika. Ob vsakem kliku na številko se odreže prvi znak v spremenljivki *Text*, novi znak pa se doda na koncu niza. Objekt *Plane.080* pa prekriva del števil, ki se ne smejo videti. Na začetku prekriva vse številke v objektu *ekran*. Ko učenec enkrat klikne na številko, ploskev pokriva le še prvih 6 števil in tako dalje. Koliko števil naj pokriva, ima objekt *Plane.080* zapisano v spremenljivki z imenom *prop*. Prekrivanje je izvedeno s pomočjo animacije. Na prvih osmih okvirjih (*ang. Frame*) objekta *Plane.080* smo spremenili velikost ploskve. V prvem okvirju je ploskev velika toliko, da pokrije vsa števila. Na zadnjem, osmem, pa je tako majhna, da se je skoraj ne vidi in ne pokrije nobene številke. Ob vsakem kliku miške, se animacija ploskve prestavi v okvir, ki je zapisan v spremenljivki *prop*. Nastavitve za objekt *Plane.080* so sledeče:

Dodali smo senzor, ki se sproži ob levem kliku miške. Senzor smo povezali z nadzornikom, tega pa z izvrševalcem. Možnost *Motion* v izvrševalcu smo zamenjali z možnostjo *Ipo*, možnost *Play* pa smo zamenjali s *Property*. V polje *Prop* vpišemo ime spremenljivke (*prop*), ki določa, na kateri okvir naj se prestavi animacija.



Slika 27: Nastavitve za objekt Plane.080

Vsakemu gumbu na računalu dodamo dva senzorja in enega nadzornika. Oba senzorja povežemo z nadzornikom in jima možnost *Always* spremenimo v možnost *Mouse*. Pri prvem senzorju spremenimo možnost *Left button* v *Mouse over*. Pri nadzorniku možnost *And* zamenjamo z možnostjo *Python*. Ob kliku na gumb s številko se bo izvedla drugačna skripta kot ob kliku na gumb *Briši*. Poglejmo si najprej gumbe s številkami. Vsak gumb ima spremenljivko z imenom *prop*, ki je tipa *Integer* in je enaka vrednosti, napisani na gumbu. Skripto, ki se izvede ob kliku na gumb, smo poimenovali *izpis*. Tokrat si bolj podrobno pogledjmo tudi imena senzorjev, ki se nahajata desno od možnosti *Mouse*, saj jih bomo potrebovali pri pisanju skripte. Imena senzorjev sta *sensor* in *sensor1*.

```
#Dostopamo do nadzornika, ki izvaja skripto, ter ga shranimo v spremenljivko cont.
cont = GameLogic.getCurrentController()
#Prek nadzornika dostopamo do objekta ekran in ga shranimo v spremenljivko own.
own=cont.owner
#Prek nadzornika dostopamo do senzorjev. Shranimo oba, vsakega v svojo
spremenljivko.
mouseOverSensor = cont.sensors["sensor"]
leftButtonSensor = cont.sensors["sensor1"]

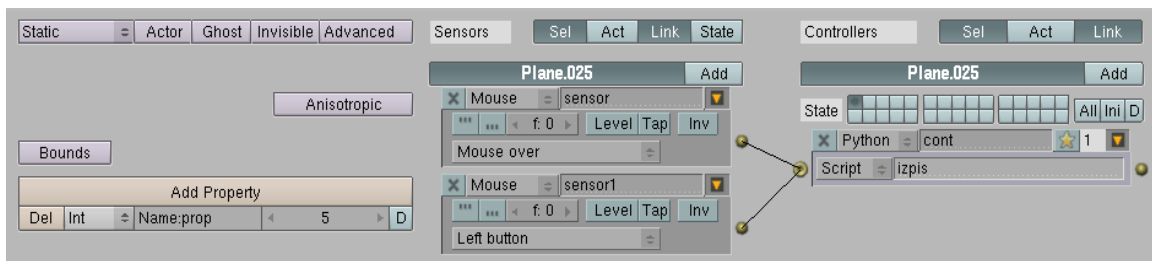
#Preverimo, ali sta oba senzorja pozitivna; če nista, se ne zgodi nič.
if (leftButtonSensor.positive and mouseOverSensor.positive):
    #Prek modula GameLogic dostopamo do trenutnega prizora.
    scena = GameLogic.getCurrentScene()
    #Prek prizora pa lahko dostopamo do kateregakoli objekta na tej sceni, v našem
    primeru Plane.080.
    obj = scena.objects["OBPlane.080"]
    #Če je spremenljivka prop v objektu Plane.080 manjša od 8, potem se
    spremenljivka poveča in objekt pokriva manj številčk.
    if obj['prop']<8:
        obj['prop']+=1
```

```
#Dostopamo še do objekta ekran.
obj = scena.objects["OBekran"]
#Spremenljivki prvi znak odstranimo in na koncu dodamo še vrednost, ki jo je
kliknil učenec (je zapisana za vsak gumb posebej v spremenljivki prop, ki jo
moramo najprej še spremeniti v niz).
obj['Text'] = obj['Text'][1:]+str(own['prop'])
```

Kasneje bomo videli, da moramo tej kodi dodati še dve vrstici. Ob kliku na gumb *Preveri* se pri napačnem odgovoru izpiše beseda *Neppravilno* v objektu *napaka*. Ko učenec poskuša napako popraviti in klikne na druge gumbe na računalu, se mora beseda *Neppravilno* odstraniti. Zato bomo v zgornji skripti izpis dodali še naslednji vrstici, s katerima bomo spremenljivki *Text* v objektu *napaka*, ki izpisuje tekst, nastavili vrednost praznega niza. Vrstici dodamo na koncu zanke *if*.

```
obj = scena.objects["OBnapaka"]
obj["Text"]=""
```

Spodnja slika prikazuje nastavitve za gumb s številko 5. Zaradi boljše vidljivosti ni prikazan del z izvrševalcem. Vsi gumbi s številkami imajo enake nastavitve.



Slika 28: Nastavitve za gumb s številko 5

Edini gumb na računalu, ki deluje drugače od ostalih, je gumb *Briši*. Nastavitve so podobne kot pri ostalih gumbih, izvede se le druga skripta. Ob kliku na gumb *Briši* se namreč zadnje število v spremenljivki *Text*, ki pripada objektu *ekran*, izbriše. Na začetku niza pa se doda poljuben znak. Poleg tega mora ploskev *Plane.080* pokriti eno številko več. Skripto smo poimenovali *brisi*.

```

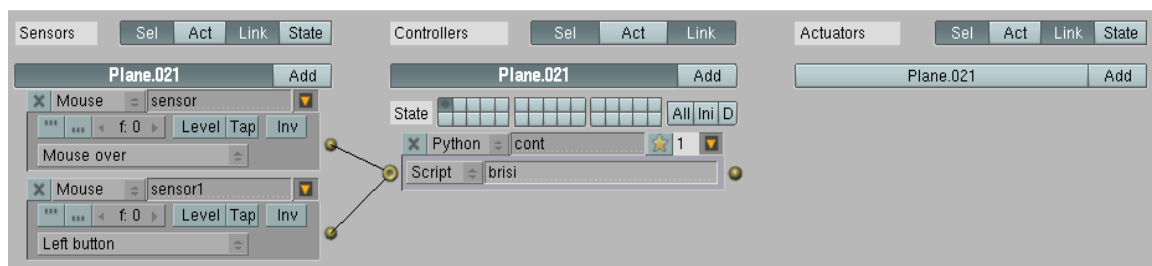
#Dostopamo do nadzornika, ki izvaja skripto, ter ga shranimo v spremenljivko cont.
cont = GameLogic.getCurrentController()
#Prek nadzornika dostopamo tudi do senzorjev. Shranimo oba, vsakega v svojo
spremenljivko.
mouseOverSensor = cont.sensors["sensor"]
leftButtonSensor = cont.sensors["sensor1"]

#Preverimo, ali sta oba senzorja pozitivna; če nista, se ne zgodi nič.
if (leftButtonSensor.positive and mouseOverSensor.positive):
    #Prek modula GameLogic dostopamo do trenutnega prizora.
    scena = GameLogic.getCurrentScene()
    #Prek prizora pa lahko dostopamo do objekta Plane.080
    obj = scena.objects["OBPlane.080"]
    #Če je spremenljivka večja od 1, potem se spremenljivka zmanjša in objekt
    pokriva več števil.
    if obj['prop']>1:
        obj['prop']-=1
    #Dostopamo še do objekta ekran.
    obj = scena.objects["OBekran"]
    #Vzamemo prvih 6 znakov spremenljivke Text (zadnjega, sedmega, s tem
    pobrišemo). Da pa bo niz imel zopet 7 znakov, na začetku niza dodamo
    poljuben znak (v tem primeru znak 0).
    obj['Text'] = '0'+obj['Text'][:6]

    #Spremenljivki Text v objektu napaka nastavimo vrednost praznega niza (glej
    razlago prejšnje kode).

    obj = scena.objects["OBnapaka"]
    obj['Text']="

```



Slika 29: Nastavitve za gumb Briši

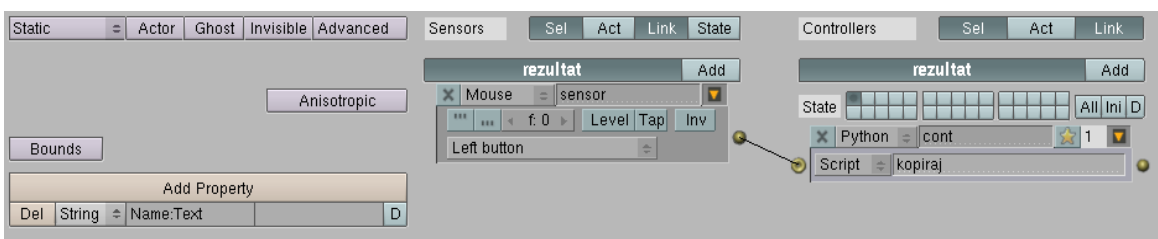
Objekt *rezultat* ima samo eno funkcijo. Število na računalu, ki ga učenec sestavi, še enkrat izpiše. Ima spremenljivko *Text*, v katero se shrani niz števil, ki ga je učenec vnesel. Objektu *rezultat* dodamo en senzor in ga povežemo z enim nadzornikom. Senzorju nastavimo možnost *Mouse*, nadzorniku pa možnost *Python*. Napisali bomo še eno skripto z imenom *kopiraj*, v kateri bomo iz objekta *ekran* skopirali števila, ki jih je vnesel učenec, ter jih shranili v objekt *rezultat* v spremenljivko *Text*.

```

#Dostopamo do nadzornika, ki izvaja skripto, ter ga shranimo v spremenljivko cont.
cont = GameLogic.getCurrentController()
#Preko nadzornika dostopamo do objekta rezultat, in ga shranimo v spremenljivko
own.
own=cont.owner

#Preko modula GameLogic dostopamo do trenutnega prizora.
scena = GameLogic.getCurrentScene()
#Preko prizora pa lahko dostopamo do objekta Plane.080 in ekran.
plane = scena.objects["OBPlane.080"]
ekran = scena.objects["OBekran"]
#Iz objekta ekran skopiramo znake. Koliko znakov bomo skopirali, pa je zapisano v
spremenljivki objekta Plane.080.
own['Text']=ekran['Text'][:8-plane['prop']:]

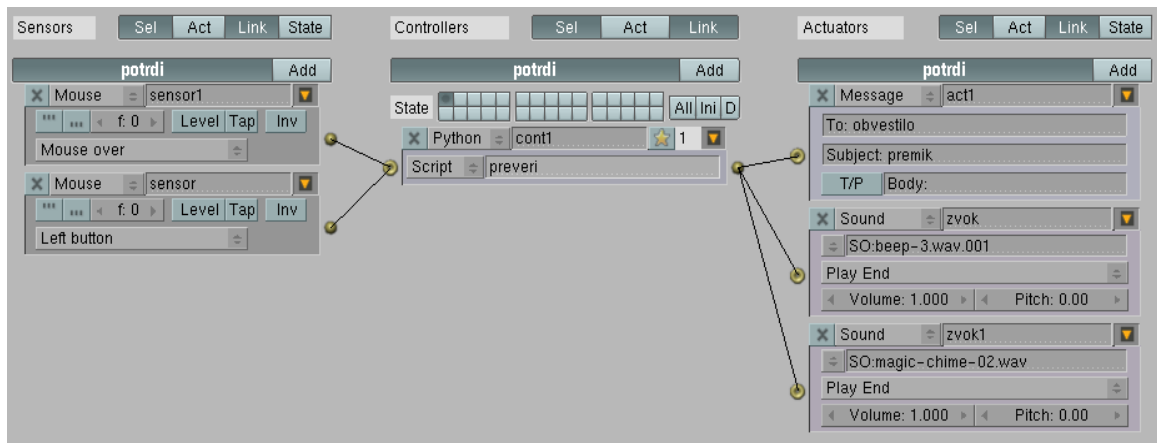
```



Slika 30: Nastavitve za objekt rezultat

Manjka nam še najbolj pomemben del – preveriti, ali je rezultat pravilen ali ne, in se na to ustrezno odzvati. Nastavljali bomo nastavitve za gumb *Potrdi*. Dodali bomo dva senzorja, enega nadzornika in tri izvrševalce. Pri obeh senzorjih bomo nastavili možnost *Mouse*, pri prvem spremenimo tudi možnost *Left button* v *Mouse over*. Nadzorniku zopet nastavimo možnost *Python*, ime skripte, ki jo bomo za ta namen napisali, pa je *preveri*.

Potrebujemo tri izvrševalce, dva za predvajanje zvoka in enega za pošiljanje sporočila. Sporočilo bomo poslali objektu z imenom *obvestilo*, ki se bo ob pravilnem odgovoru prikazal na ekranu. Prvemu izvrševalcu nastavimo možnost *Message*. V polje *To*, vpišemo ime objekta, kateremu želimo poslati sporočilo, torej *obvestilo*. Izpolniti moramo še polje *Subject*, kamor vpišemo poljubno zadevo sporočila. Mi smo vpisali besedo *premik*. Pri ostalih dveh izvrševalcih nastavimo možnost *Sound*, ter za lažje prepoznavanje nastavimo tudi imena (desno od možnosti *Sound*) izvrševalcev (*zvok* in *zvok1*). V polju *Sound* lahko z izbirnikom izberemo zvok, ki ga želimo predvajati. Seveda pa moramo ta zvok prej dodati preko gumba *Scene* (bližnjica *F10*). Ko zvok izberemo, možnost *Play Stop* spremenimo v *Play End*. Sedaj pa se lahko lotimo pisanja skripte.



Slika 31: Nastavitve za gumb potrdi

```

#Dostopamo do nadzornika, ki izvaja skripto, ter ga shranimo v spremenljivko cont.
cont = GameLogic.getCurrentController()
#Prek nadzornika dostopamo do objekta rezultat, in ga shranimo v spremenljivko own.
own=cont.owner
#Prek nadzornika dostopamo tudi do senzorjev. Shranimo oba, vsakega v svojo
spremenljivko.
leftButtonSensor = cont.sensors["sensor"]
mouseOverSensor = cont.sensors["sensor1"]

#Preverimo, ali sta oba senzorja pozitivna; če nista, se ne zgodi nič.
if (leftButtonSensor.positive and mouseOverSensor.positive):
    #Prek modula GameLogic dostopamo do trenutnega prizora.
    scena = GameLogic.getCurrentScene()
    #Prek prizora pa lahko dostopamo do objekta rezultat.
    obj = scena.objects["OBrezultat"]
    #Če je rezultat enak številu, ki smo ga v skripti random shranili v spremenljivko
    GameLogic.stevilo, potem aktiviramo izvrševalca z imenom act1 in zvok1.
    if (obj['Text']==str(GameLogic.stevilo)):
        #Izvrševalec z imenom act1 pošlje sporočilo objektu obvestilo.
        GameLogic.addActiveActuator("act1",1)
        #Izvrševalec z imenom zvok1 predvaja zvok.
        GameLogic.addActiveActuator("zvok1",1)
    else:
        #Če je odgovor napačen, se v objektu napaka izpiše niz 'Nepravilno!' in
        aktiviramo izvrševalca z imenom zvok, ki predvaja zvok.
        obj = scena.objects["OBnapaka"]
        obj["Text"]="Nepravilno!"
        GameLogic.addActiveActuator("zvok",1)

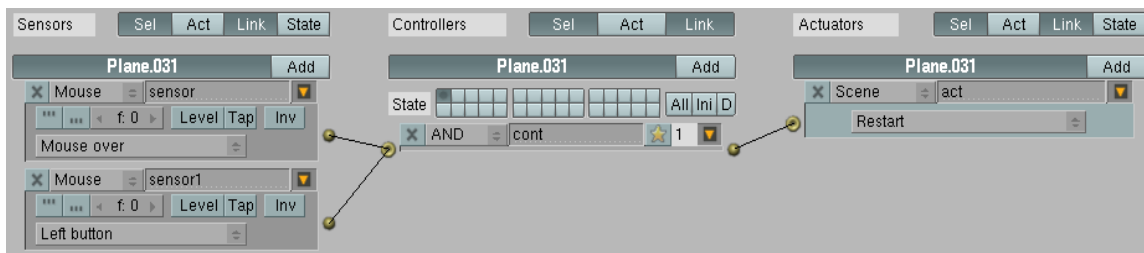
```

Objekt *obvestilo* se prikaže s pomočjo animacije. V prvem okvirju je postavljen na desni strani, skrit pred kamero. V petem okvirju pa smo ga postavili točno pred kamero, tako da prekrije cel prizor. Objektu dodamo senzor, nadzornika in izvrševalca. Možnost sensorja nastavimo na *Message*. V polje *Subject* vpišemo enako zadevo, kot smo jo vpisali v izvrševalcu objekta *potrdi*, torej *premik*. Izvrševalcu spremenimo možnost *Motion* v *Ipo* ter določimo začetni in končni okvir animacije. Spremeniti moramo samo končni okvir v polju *End* iz vrednosti 1 v 5.



Slika 32: Nastavitve za objekt obvestilo

Za konec sta nam ostala še gumba, ki se pojavita hkrati z obvestilom. S klikom na prvi gumb učenec ponovno igra igro, s klikom na drugi gumb pa se ta konča. Nastavitve za drugi gumb so enake kot za gumb *izhod*, zato si jih pogledjmo le za prvi gumb. Dodamo dva sensorja, enega nadzornika ter enega izvrševalca. Ne pozabimo jih med seboj povezati. Obema sensorjema spet nastavimo možnost *Mouse*, ter prvemu sensorju spremenimo možnost *Left button* v *Mouse over*. Pri izvrševalcu izberemo možnost *Scene* ter pustimo možnost *Restart*.



Slika 33: Nastavitve za gumb, s katerim ponovno igramo igro

4.3 Praktični primer 2

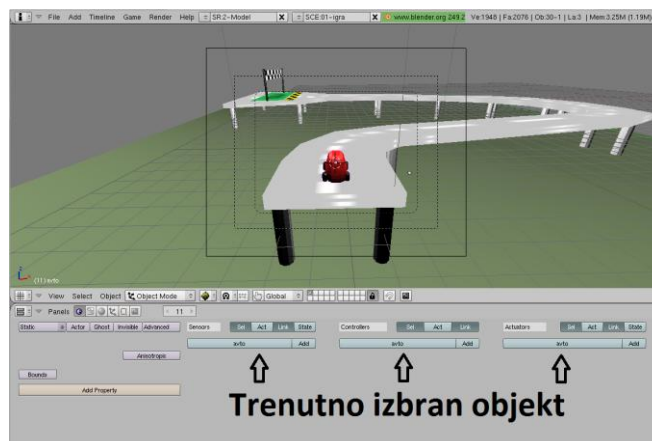
V tem poglavju bomo opisali izdelavo še ene igre. Naučili se bomo nadzirati objekt s tipkovnico, dodajati nove prizore, spoznali bomo nekaj novih možnosti pri izvrševalcu ter pomen kamere. Ponovili bomo, kako nastavimo vidljivost miškekega kazalca, kako ponovno zaženemo igro ter kako jo zapremo. Na priloženi CD zgoščenki lahko dobimo tudi Blender datoteko z že izdelanimi objekti, s pomočjo katere lahko sledimo navodilom in izdelamo igro.

- **Opis igre**

Na začetku igralec s klikom na ustrezen gumb prične igro. V igri bo s tipkovnico upravljal avtomobil, ki se vozi po cesti. Če avtomobil pade s ceste, se igra začne od začetka. Ko pride na cilj, ima igralec izbiro – ali bo končal z igro ali pa jo bo ponovno igral.

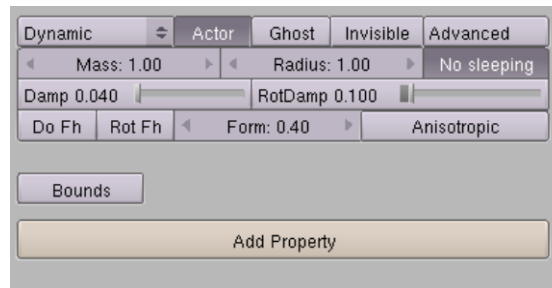
- **Upravljanje avtomobila s tipkovnico**

Začeli bomo s spodnjo sliko. Vsi objekti so že postavljeni. Najprej moramo nastaviti, kako bomo upravljali avtomobil. Izbran je že gumb *Logika*, prav tako pa objekt avtomobila. Ime objekta smo preimenovali v *avto*, kar se vidi tudi v svetlomodrih pravokotnikih.



Slika 34: Začetni pogled igre

Najprej moramo avtomobilu določiti fizikalne lastnosti, da bo lahko padel s ceste. V prvem meniju namesto *Static* izberemo *Dynamic* in vklopimo gumba *Actor* in *No sleeping*. Avtomobil ne bo imel nobene spremenljivke, zato bo levi del videti tako, kot prikazuje slika 36.



Slika 35: Fizikalne nastavitve

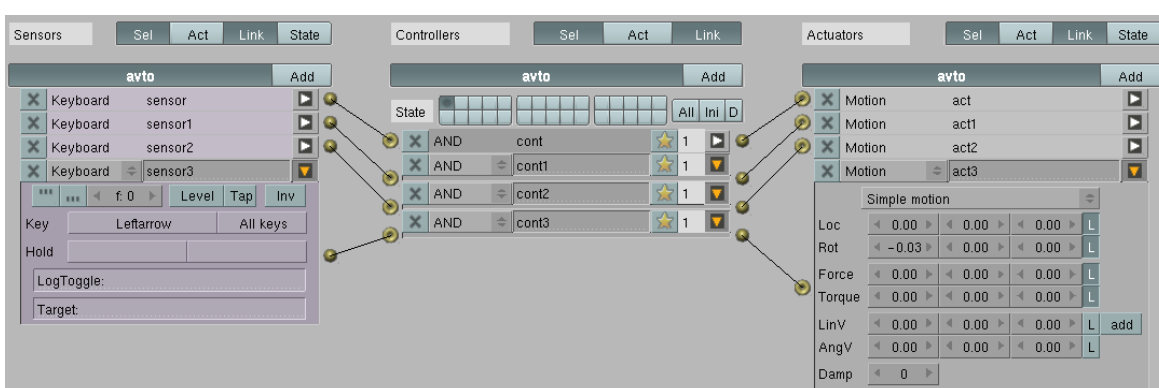
Najprej bomo določili, naj se avtomobil ob pritisku na ustrezno tipko premika naprej. Dodamo senzor s klikom na gumb *Add* pod besedo *Sensor*. V meniju namesto *Always* izberemo *Keyboard* (tipkovnica). Kliknemo v polje poleg besede *Key*, izpiše se *Press a Key*. Na tipkovnici pritisnemo tisto tipko, s katero se bo avtomobil premikal naprej. Mi bomo izbrali smerno tipko navzgor. Dodamo še nadzornika, ki ga ne spreminjamo, ter izvrševalca. Pri izvrševalcu pustimo možnost *Motion*. V vrstici *Loc* bomo določili, kako naj se spremeni lokacija objekta. Če spremenimo prvo vrednost, se bo lokacija spremenila glede na koordinato x, druga vrednost spremeni koordinato y, tretja pa koordinato z. V desnem spodnjem kotu lahko opazimo majhen koordinatni sistem, ki nam pove, da se bo avto premikal v smeri y. Torej bomo srednjo možnost spremenili na vrednost 0.10. Vse tri dele moramo sedaj tudi povezati. Povezati moramo rumeno piko na desni strani senzorja s piko na levi strani nadzornika, ter piko na desni strani nadzornika s piko na levi strani izvrševalca. Miškin kazalec postavimo na prizor, pritisnemo tipko P na tipkovnici in preverimo, ali se avtomobil ob pritisku na smerno tipko naprej res premika naprej.



Slika 36: Nastavitve za premikanje naprej

Povsem enako storimo za premikanje nazaj, nastavimo samo drugo tipko. Namesto premikanja po osi y za 0.10, nastavimo negativno vrednost -0.10. Ne pozabimo na povezave med senzorjem, nadzornikom in izvrševalcem. Opomba: polje, kjer spreminjamo senzorce, nadzornike in izvrševalce lahko tudi skrijemo s klikom na rumeni trikotnik pod gumbom *Add*.

Tudi primer za zavijanje avtomobila je precej podoben. Dodamo tretji senzor. Nastavimo tipko za desno obračanje. Razlika je, da v izvrševalcu ne bomo spreminjali lokacije, pač pa rotacijo. Avto se bo obračal v smeri osi x , zato bomo prvo vrednost v vrstici *Rot* spremenili v 0.03. Enako moramo nastaviti še za obračanje v levo, vrednost v vrstici *Rot* bo negativna (-0.03).

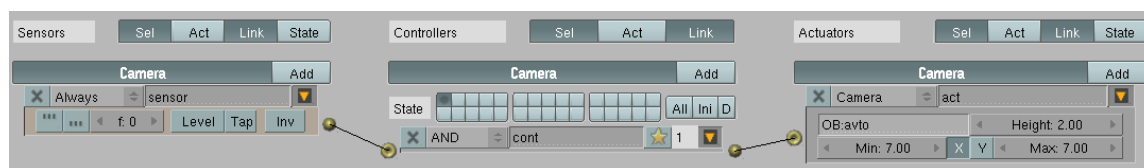


Slika 37: Nastavitve za zavijanje v levo

• Delo s kamero

Objekt kamera definira lokacijo in perspektivo, iz katere vidimo prizor [6]. V igri lahko kamero premikamo z njenimi lastnostmi (tako kot smo premikali avtomobil), lahko pa kamera sledi poljubnemu objektu in se premika skupaj z njim. Mi bomo opisali to drugo možnost.

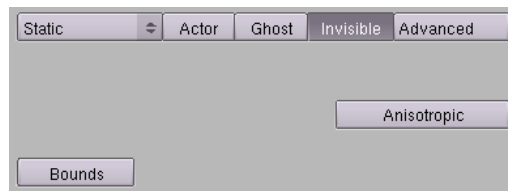
Najprej označimo kamero. Dodamo ji vse tri logične dele in jih med seboj povežemo. Tokrat bomo spremenili samo nastavitve za izvrševalca. Možnost *Motion* bomo spremenili v *Camera*. V polje *OB* pa moramo vpisati ime objekta, ki naj mu kamera sledi - *avto*. Nastavimo še višino (*Height*) na 2 ter minimalno in maksimalno razdaljo na 7. Ob zagonu igre kamera zdaj sledi avtomobilu.



Slika 38: Nastavitve za kamero

- **Določanje akcije ob dotiku dveh objektov**

Zelena ploskev pod mostom bo poskrbela, da se takrat, ko avtomobil pade z mostu spet začne od začetka. Najprej jo označimo in nastavimo to ploskev za nevidno. Pod fizikalnimi lastnostmi vklopimo gumb *Invisible*.



Slika 39: Izbrani objekt je v igri neviden

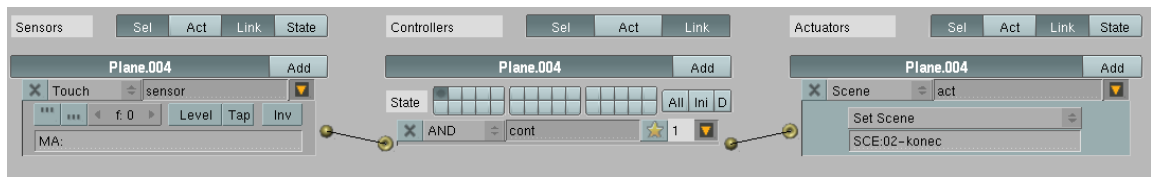
Dodamo še vse tri logične dele in jih povežemo. Senzor se mora aktivirati, ko se avto dotakne ploskve, zato bomo izbrali možnost *Touch*. Pri izvrševalcu pa bomo izbrali možnost *Scene*. Možnost *Restart* je že pravilno nastavljena, saj želimo ta prizor ponovno zagnati. Rezultat lahko sami preverimo tako, da v igri avto zapeljemo s ceste.



Slika 40: Prizor se ponovno zažene ob dotiku objekta z avtomobilom

- **Preklapljanje med prizori**

Preden nadaljujemo, si pogledjmo, katere prizore imamo v tej datoteki. V zgornjem delu ekrana, poleg logotipa programa, nam v polju piše ime trenutnega prizora. Ime smo mu dali *01-igra*. Poleg tega prizora imamo še prizor *00-start* in *02-konec*. Ko avtomobil pripelje do cilja, želimo, da se pojavi prizor *02-konec*. Tu bo lahko uporabnik izbral, ali želi ponovno igrati ali pa želi igro končati. Na prizoru *01-igra* izberimo ploskev, na kateri piše *CILJ*. Dodamo in povežemo vse tri logične dele, senzor nastavimo na *Touch*, izvrševalca pa na *Scene*. Namesto *Restart* izberemo *Set Scene* in vpišemo ime prizora, za katerega želimo da se prikaže – torej *02-konec*.

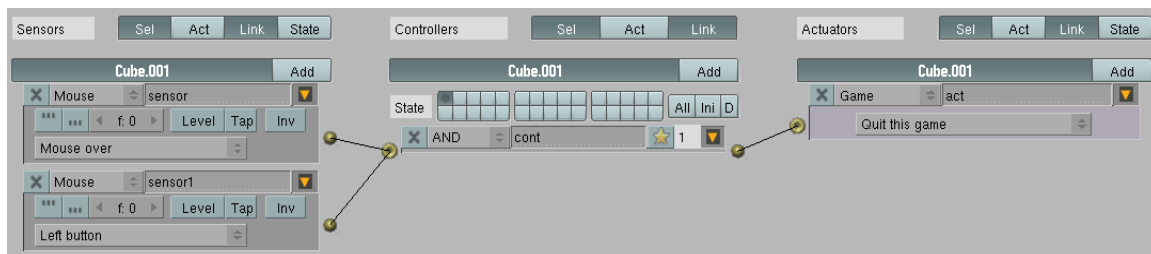


Slika 41: Prikaz novega prizora (02-konec) ob dotiku z avtomobilom

- **Ponovni zagon igre, izhod iz igre ter prikaz miškega kazalca (ponovitev)**

Najprej se postavimo na prizor *02-konec*. V prizoru sta dva gumba, s katerima bo uporabnik upravljal igro. S prvim gumbom bo ponovno igral igro, z drugim pa bo igro zaprl. Začnimo s prvim gumbom. Tokrat bomo dodali dva senzorja in oba povezali z enim nadzornikom. Tega pa bomo povezali z enim izvrševalcem. Pri obeh senzorjih nastavimo možnost *Mouse*. Pri enem možnost *Left button* spremenimo v *Mouse over*. Kar bomo nastavili v izvrševalcu, se bo torej izvedlo, ko bo uporabnik kliknil na gumb. V izvrševalcu izberemo možnost *Scene* in še možnost *Set Scene*. V polje *SCE* vpišemo ime prizora z igro, torej *01-igra*.

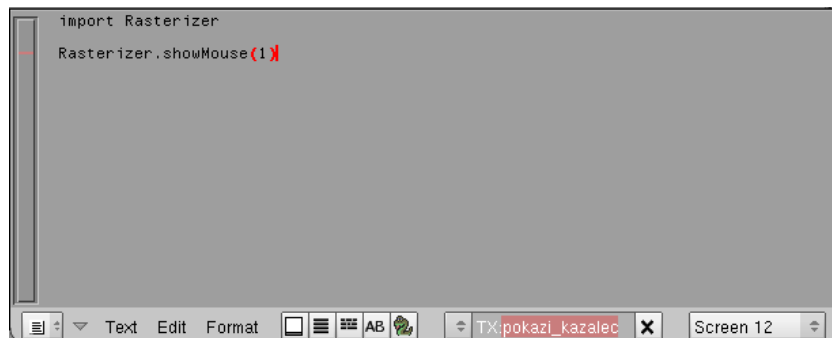
Drugi gumb nastavimo enako, le da pri izvrševalcu izberemo možnost *Game* in nato še *Quit this game*.



Slika 42: Nastavitve za drugi gumb

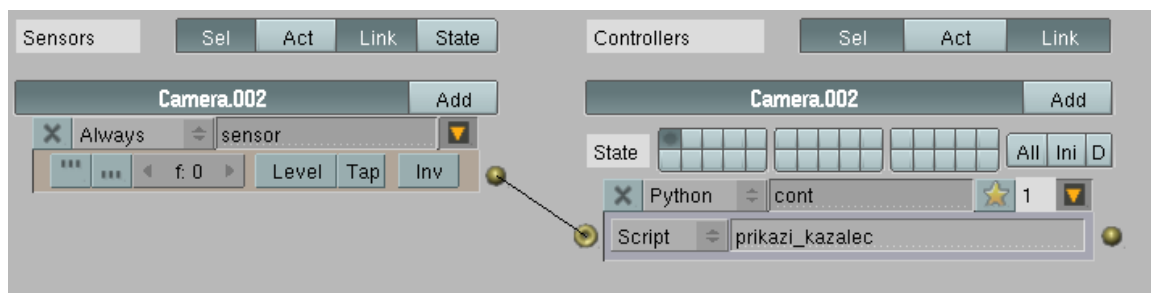
Ko na prizoru *02-konec* pritisnemo tipko *P* za igranje, opazimo, da nikjer ni miškega kazalca. Naučiti se moramo še, kako prikazati kazalec. V tem primeru bomo uporabili kratko skripto. Najprej moramo izbrati neki objekt, ki je vedno na prizoru – na primer kamera. Dodamo samo senzor in nadzornika. Senzor pustimo tako, kot je, pri nadzorniku pa namesto *AND* izberemo možnost *Python*. Glavno okno razdelimo na dva dela in v enem izberemo pogled *Text Editor*. Ustvarimo nov tekstovni dokument s kombinacijo tipk *Alt+N* ter ga preimenujemo iz *Text* v *prikazi_kazalec* (v polju *TX*). V besedilo je potrebno napisati naslednji dve vrstici:

```
import Rasterizer
Rasterizer.showMouse(1)
```



Slika 43: Skripta pokazi_kazalec

Sedaj moramo v nadzorniku v prazno polje vpisati ime skripte – *prikazi_kazalec*. Ko zaženemo igro, se mora na prizoru *02-konec* pokazati miškin kazalec. Ko kliknemo na ustrezen gumb, se igra zapre oziroma začne od začetka.



Slika 44: Izvajanje skripte, ki kaže miškin kazalec

Če ste poizkusili pripeljati avto do konca in ste na končnem prizoru izbrali ponovno igranje, ste opazili, da je miškin kazalec prikazan tudi med vožnjo avtomobila, kar je zelo moteče. Z manjšim dodatkom bomo popravili prikaz kazalca. Ustvarimo nov tekstovni dokument, ga preimenujemo v *skrij_kazalec*, ter vanj vpišemo vrstici:

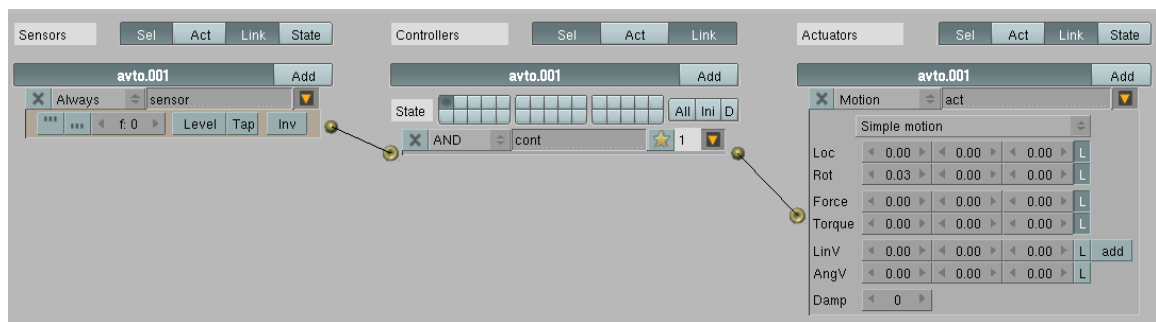
```
import Rasterizer
Rasterizer.showMouse(0)
```

Na prizoru *01-igra* izberemo katerikoli objekt. Priporočamo, da zaradi boljše preglednosti izberemo neki nov objekt, na primer cesto. Enako kot prej, dodamo senzor in nadzornika. V nadzorniku spremenimo *AND* v *Python* in v polje *Script* vpišemo *skrij_kazalec*. Še enkrat poizkusimo igro in kazalca se med vožnjo ne vidi več.

- **Prikaz kazalca, spreminjanje prizora in animacija avtomobila**

Za lepši videz igre bomo dodali še začetni prizor z enim gumbom in vrtečim se avtomobilom. Postavimo se na prizor *00-start*. Na tem prizoru bo potrebno klikniti na gumb *Začni*, zato je potrebno spet prikazati miškin kazalec. Tokrat nam ne bo potrebno pisati nove skripte, lahko uporabimo kar skripto *prikazi_kazalec*. Kazalec prikažemo na enak način kot v prejšnjem razdelku. Tudi nastavitve za gumb *Začni* že poznamo (identično kot smo to nastavili pri prvem gumbu na prizoru *02-konec*). Preostane nam samo še animacija avtomobila.

Označimo avtomobil in dodamo vse tri logične dele ter jih povežemo med seboj. Spremenili bomo samo vrednost pri izvrševalcu, in sicer v vrstici *Rot*. Prvo vrednost pri koordinati x bomo spremenili v 0.03.



Slika 45: Nastavitve za vrtenje avtomobila

Naša igra je sedaj dokončana. Če želimo svojo igro komu pokazati, priporočamo, da vse slike in zvoke (če jih ni veliko) shranimo skupaj v Blender datoteko. To storimo tako, da na vrhu ekrana izberemo meni *File*, nato pa *External Data* in *Pack into .blend file*.

Igro shranimo v izvršljivo datoteko tako, da v meniju *File* izberemo možnost *Save Game As Runtime*, nato pa izberemo mapo, v katero želimo igro shraniti. V tej mapi potrebuje igra za izvajanje še določene datoteke s končnico *.dll*, ki jih najdete v mapi, kamor ste shranili program Blender.

Spletne strani, s katerimi smo si pomagali pri izdelavi primera:

<http://www.youtube.com/watch?v=zNnq9kWIIQc&feature=related>

<http://www.youtube.com/watch?v=HjtNL6u5tjA&NR=1>

http://wiki.blender.org/index.php/Doc:Tutorials/Game_Engine/BSoD

5 Ocena uspešnosti

V diplomski nalogi želimo preveriti dve hipotezi.

- 1. hipoteza:** Učenci bolj razumejo snov, če je predstavljena z interaktivno izobraževalno igro, podkrepljeno s teoretičnimi spoznanji, kot pa če je predstavljena na frontalen način.
- 2. hipoteza:** Bodoči učitelji matematike in računalništva so sprejemljivi za uporabo didaktičnih interaktivnih iger pri pouku.

Našo igro smo preverili v praksi. Zanimalo nas je, ali je uporaba igre pri pouku uspešnejša od klasičnega poučevanja. Na vprašanje smo skušali odgovoriti s pomočjo eksperimenta. Drugo hipotezo pa smo preverjali z anketo. Želeli smo ugotoviti, ali bodoči učitelji matematike in računalništva sprejemajo uporabo vizualno zanimivih in interaktivnih izobraževalnih gradiv. Poleg preverjanja druge hipoteze smo želeli ugotoviti tudi, za kakšen namen bi učitelji uporabljali interaktivne izobraževalne igre ter kakšna se jim je zdela naša igra. Rezultati sledijo v nadaljevanju.

5.1 Potek preverjanja

V mesecu maju in v začetku junija smo obiskali tri osnovne šole, OŠ Rodica, OŠ Staneta Žagarja Kranj ter OŠ Preserje pri Radomljah. V eksperiment smo vključili učence osmih in devetih razredov. Učitelji so učence razdelili v dve heterogeni skupini. Vsaka od naju je prevzela eno skupino. Ena skupina se je učila snov o dvojiških številskih sistemih preko predavanja, druga pa samostojno, s pomočjo naše igre. Igro smo namestili na šolske računalnike, seveda z vnaprejšnjim dogovorom s profesorji. Obe skupini sta se učili istočasno, eno šolsko uro. Na koncu ure smo učencem razdelili še kratek test, s katerim smo preverili pridobljeno zanje.

V mesecu juliju pa smo študentom Pedagoške fakultete, smeri Matematika-računalništvo, po internetu poslali okrnjeno verzijo igre ter anketo. Anketiranci so bili študentje tretjih in četrth letnikov ter absolventi.

5.2 Rezultati in analiza eksperimenta

Spodnji rezultati so opisani po posameznih šolah. Najprej so predstavljeni naši rezultati, zaradi boljše primerjave pa so povzeti tudi rezultati kolegice Nataše, ki je izvajala drugo polovico eksperimenta. Na koncu sledijo še skupne ugotovitve.

Test, ki so ga reševali učenci:

DVOJIŠKI ŠTEVILSKI SISTEM

Spol:

Starost:

1. Kaj je dvojiški številski sistem? Katera števila so v dvojiškem številskem sistemu?
2. Zapiši 2 različni dvojiški števili.
3. Pretvori iz desetiškega v dvojiški številski sistem
 $21_{(10)} =$
 $34_{(10)} =$
4. Pretvori iz dvojiškega v desetiški številski sistem
 $010011_{(2)} =$
 $101001_{(2)} =$
5. Oglej si Mihovo vabilo na rojstni dan. Kdaj bo praznoval rojstni dan?

Vabim te, da se udeležiš mojega rojstnega dne, ki bo

DAN: 011101

MESEC: 001011

URA: 010000

S seboj prinesi dobro voljo!

a) OŠ Rodica, Kettejeva 13, 1230 Domžale

Ravnateljica: Milena Vidovič, profesor: Martin Mah

Na tej šoli smo eksperiment izvedli 23. maja. V skupinah so bili učenci osmih razredov, ki so obiskovali predmet Računalniška omrežja. Sama sem prevzela skupino, ki se je učila z igro, Nataša pa je drugo skupino učila frontalno.

V naši skupini je bilo 8 učencev, od tega 6 učencev in 2 učenki. Med reševanjem igre noben učenec ni prebral navodil oziroma teorije v igri. Takoj so se lotili reševanja testov. Kmalu pa so ugotovili, da ne znajo rešiti niti ene naloge. Verjetno jih je to demotiviralo, saj očitno niso bili pripravljeni bolj potruditi. Po neuspešnem reševanju testov, so se učenci snov začeli učiti s pomočjo vaj. Nekaterim je šlo kar dobro od rok. Proti koncu ure so nekateri videli, da jim ne bo uspelo rešiti vsega in so se samo še zabavali. En učenec in ena učenka sta igro rešila čisto do konca, eni učenki pa je zmanjkalo časa. Skupina je potrdila trditev, ki smo jo omenili v teoriji – učenci rajši pridejo do rešitve sami, kot pa da bi o njej poslušali ali brali.

Rezultati pisnega testa so sledeči:

Spol \ starost	13	14	15
Moški	2	3	1
Ženski	0	2	0

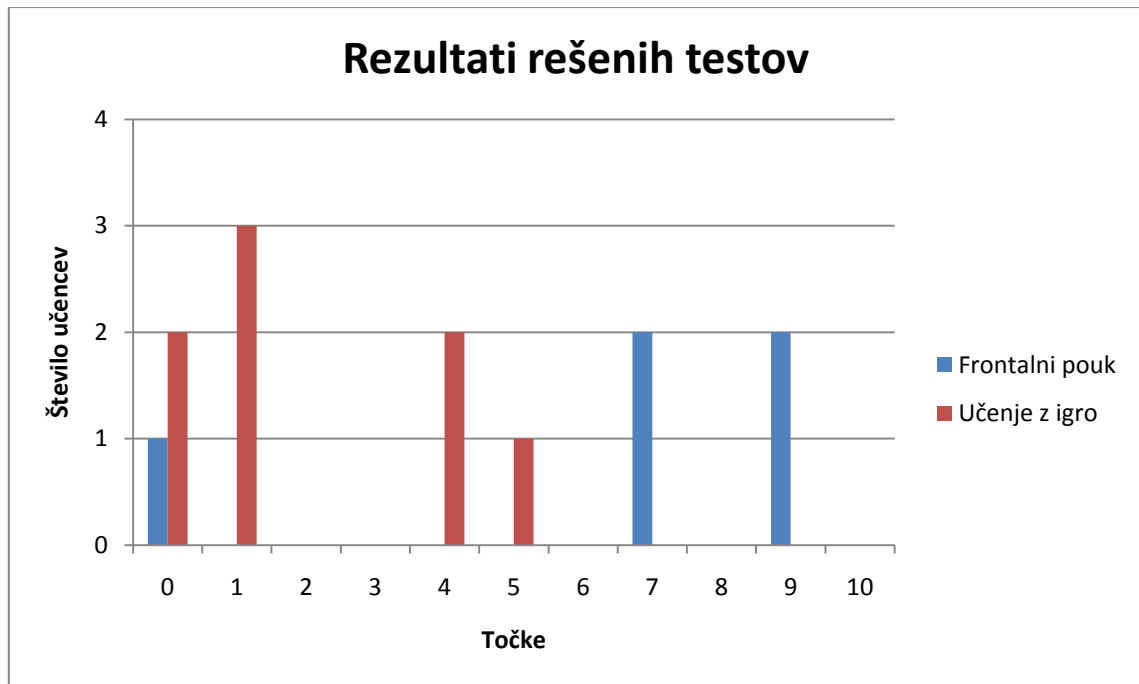
Tabela 1: Število učencev glede na spol in starost (1. šola)

Naloga	1.	2.	3.	4.	5.	Skupaj
Število možnih točk	2	1	2	2	3	10
Povprečno število točk	0	0	0	0	1,5	2

Tabela 2: Povprečno število točk, doseženih pri testu (1. šola)

Dosežene točke	0	1	2	3	4	5	6	7	8	9	10
Število učencev, ki so dosegli zgornje točke	2	3	0	0	2	1	0	0	0	0	0

Tabela 3: Koliko točk so dobili učenci (1. šola)



Graf 1: Rezultati rešenih testov (1. šola) [7]

Ko so učenci reševali pisni test, je učenec, ki je sicer rešil igro do konca, pustil test napol prazen, saj je predčasno zapustil razred. Rezultati testa so bili slabi. Nihče ni pravilno rešil prve naloge, vprašanje je bilo teoretično, nihče pa si ni prebral teorije. Tudi druge naloge ni nihče rešil pravilno, kar pomeni, da sploh niso razumeli, kaj je to dvojiško število. Četrto nalogo sta rešila dva učenca, eden je zbral obe točki, drugi pa eno. Pri zadnji nalogi so dosegli boljše rezultate, trije so pravilno rešili vse primere, trije pa en primer. Povprečno število točk, doseženih na testu, je 2 (20%). Največje doseženo število točk je bilo 5 (50%), dobil pa jih je učenec. Sledita pa mu učenka in učenec s štirimi točkami (40%). Vsi ostali so dobili eno ali nič točk.

Primerjava z učenci, ki so se učili z igro:

Učenci so dosegli bistveno boljše rezultate, a bilo jih je samo pet, štirje učenci in ena učenka. Razen ene učenke (ki je imela težave z razumevanjem slovenskega jezika), so vsi rešili test s 70% ali 90%. Iz rezultatov je razvidno, da so učenci razumeli pomen dvojiških števil in so se naučili pretvarjati iz dvojiškega v desetiški sistem ter obratno. V tem primeru se je klasični način izobraževanja izkazal kot uspešnejši. Ena izmed razlag za take rezultate je, da so igro učenci vzeli preveč za zabavo. Druga razlaga bi bila, da se učenci še ne znajo učiti samostojno, da niso navajeni drugih oblik učenja, ampak so še vedno vajeni frontalnega pouka [7].

b) OŠ Staneta Žagarja Kranj, Cesta 1. maja 10a, 4000 Kranj**Ravnateljica: Fani Bevk, profesorici: Breda Markelj, Alenka Glastovec**

V četrtek, 2. 6. 2011, smo obiskali drugo osnovno šolo. Prevezli smo dva osma razreda in nadomeščali njihovo uro matematike. Učenci obeh razredov so se razdelili na polovico, tako da smo dobili heterogeni skupini. Pri matematiki so namreč učenci razdeljeni po nivojih, v najinem primeru prvi in drugi nivo. Tokrat sem izvedla klasično učno uro, Nataša pa je spremljala dogajanje v računalniški učilnici.

Poučevali smo 8 učencev in 7 učenk, skupno 15 učencev. Učna ura je potekala brez težav. Učenci so sodelovali in razumeli snov. Snov smo obdelali do konca in rešili kar nekaj primerov.

Rezultati pisnega testa so sledeči:

Spol \ starost	13	14	15
Moški	5	3	0
Ženski	3	3	1

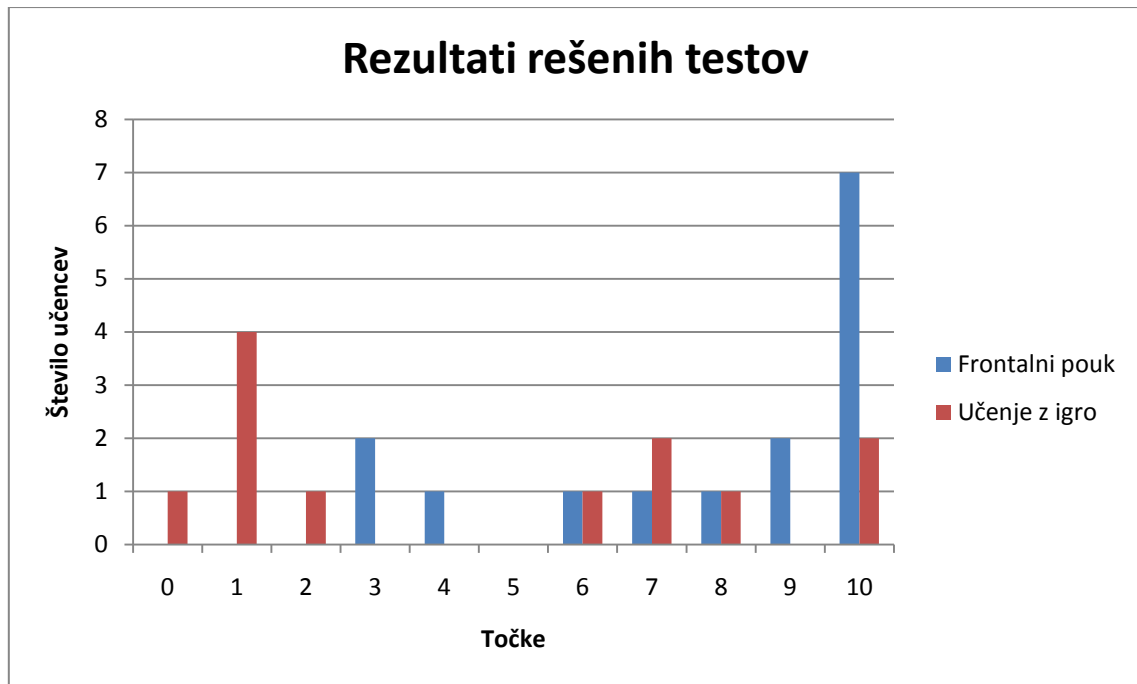
Tabela 4: Število učencev glede na spol in starost (2. šola)

Naloga	1.	2.	3.	4.	5.	Skupaj
Število možnih točk	2	1	2	2	3	10
Povprečno število točk	1,53	0,93	1,67	1,46	2,33	7,93

Tabela 5: Povprečno število točk, doseženih pri testu (2. šola)

Dosežene točke	0	1	2	3	4	5	6	7	8	9	10
Število učencev, ki so dosegli zgornje točke	0	0	0	2	1	0	1	1	1	2	7

Tabela 6: Koliko točk so dobili učenci (2. šola)



Graf 2: Rezultati rešenih testov (2. šola) [7]

Učenci so zelo dobro rešili test. Nekdo je napisal v dvojiškem zapisu celo svojo starost. Vse kaže, da so učenci razumeli snov in se nauči pretvarjati iz dvojiškega v desetiško število in obratno. Cilj učne ure je bil torej dosežen. Povprečno število točk je 7,9 oziroma 79%. Razen dveh učenk in enega učenca so vsi pravilno rešili več kot polovico testa. Kar sedem učencev je dobilo vse možne točke. Bilo je tudi nekaj manjših napak pri seštevanju. V povprečju so naloge boljše reševali fantje (86,25%) kot punce (71,43%).

Primerjava z učenci, ki so se učili z igro:

Druga skupina se je na testu slabše odrezala. Sodelovalo je 12 učencev, 8 učencev in 4 učenke. Povprečno število doseženih točk je 4,5. Rezultati so sicer zelo različni, razlike v točkah so velike. Šest učencev je doseglo manj kot 20%, pet pa več kot ali enako 70%. Tudi v tej skupini so fantje boljše rešili teste. So pa rezultati te skupine veliko boljši kot rezultati učencev iz OŠ Rodica, ki so reševali isto igro. Tudi tokrat se je klasični način izobraževanja izkazal kot uspešnejši [7].

c) OŠ Preserje pri Radomljah, Pelechova 83, 1235 Radomlje
Ravnateljica: Ana Nuša Kern, profesorica: Polona Oletić

Zadnjo šolo smo obiskali 13. 6. 2011. Tokrat smo testirali učence devetih razredov pri predmetu matematika. Na tej šoli so učenci razdeljeni v heterogene skupine. Ponovno sem skupino poučevala klasično, Nataša pa je opazovala dogajanje v računalniški učilnici. Moja skupina je štela 13 učencev, 8 fantov in 5 deklet. Ura jr potekala brez težav. Učenci so lepo sodelovali. Snov smo predelali zelo hitro, tako da nam je na koncu ostalo veliko časa za reševanje primerov. Verjetno smo bili tako hitri, ker so bili učenci starejši.

Rezultati pisnega testa so sledeči:

Spol \ starost	14	15
Moški	5	3
Ženski	1	4

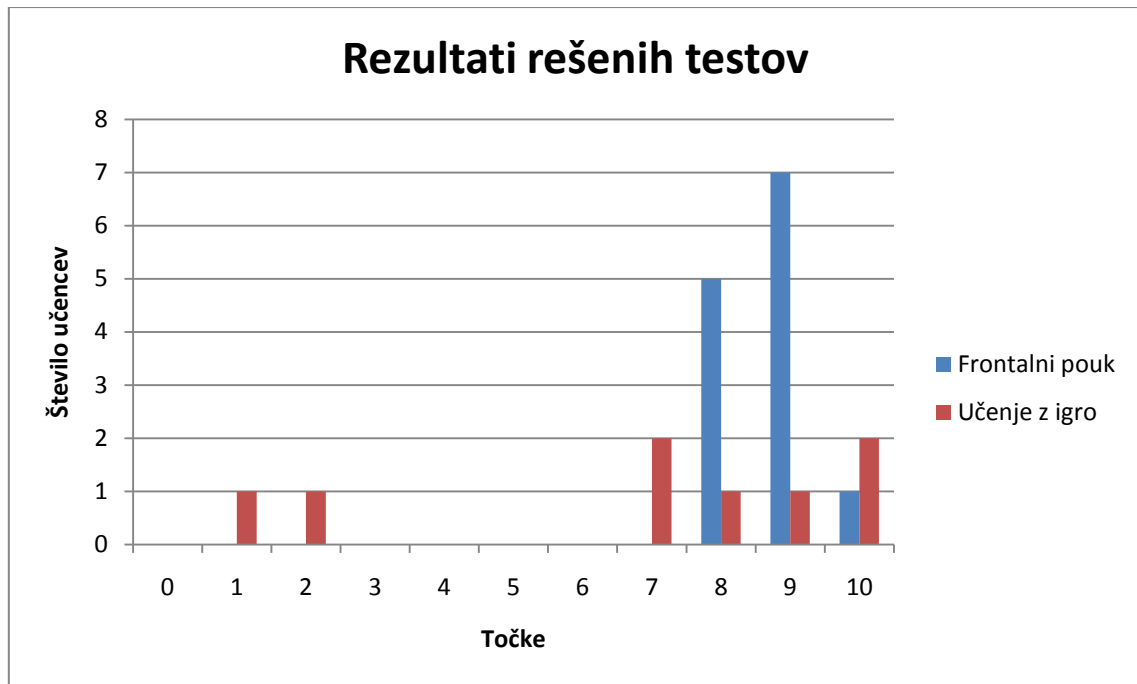
Tabela 7: Število učencev glede na spol in starost (3. šola)

Naloga	1.	2.	3.	4.	5.	Skupaj
Število možnih točk	2	1	2	2	3	10
Povprečno število točk	1	0,92	1,77	1,92	3	8,69

Tabela 8: Povprečno število točk, doseženih pri testu (3. šola)

Dosežene točke	0	1	2	3	4	5	6	7	8	9	10
Število učencev, ki do dosegli zgornje točke	0	0	0	0	0	0	0	0	5	7	1

Tabela 9: Koliko točk so dobili učenci (3. šola)



Graf 3: Rezultati rešenih testov (3. šola) [7]

Že med učno uro je kazalo, da učenci razumejo snov, kar se je pokazalo tudi pri testih. Rezultati so bili odlični, največ težav je učencem delalo prvo vprašanje, ki je bilo teoretično. Ostale napake so se pojavljale predvsem zaradi površnosti (narobe prepisana števila). Nihče ni dosegel manj kot 8 točk oziroma 80%. Povprečno število točk je 8,69 oziroma 87%. Tako fantje kot dekleta so se odlično odrezali. K temu je gotovo pripomoglo tudi to, da so to bili učenci devetih razredov.

Primerjava z učenci, ki so se učili z igro:

Tudi tokrat so bili učenci, ki so se učili z igro, slabši. V primerjavi z ostalimi šolami, pa so tokrat učenci prebrali teorijo zapisano v igri. Sodelovalo je 8 učencev, 2 fanta in 6 deklet. Razpon med rezultati je velik – dva učenca sta dosegla manj kot oziroma enako 20%, vsi ostali pa več kot oziroma enako 70%. Povprečno število doseženih točk je 6,8 oziroma 68% [7].

d) Skupna primerjava vseh učencev

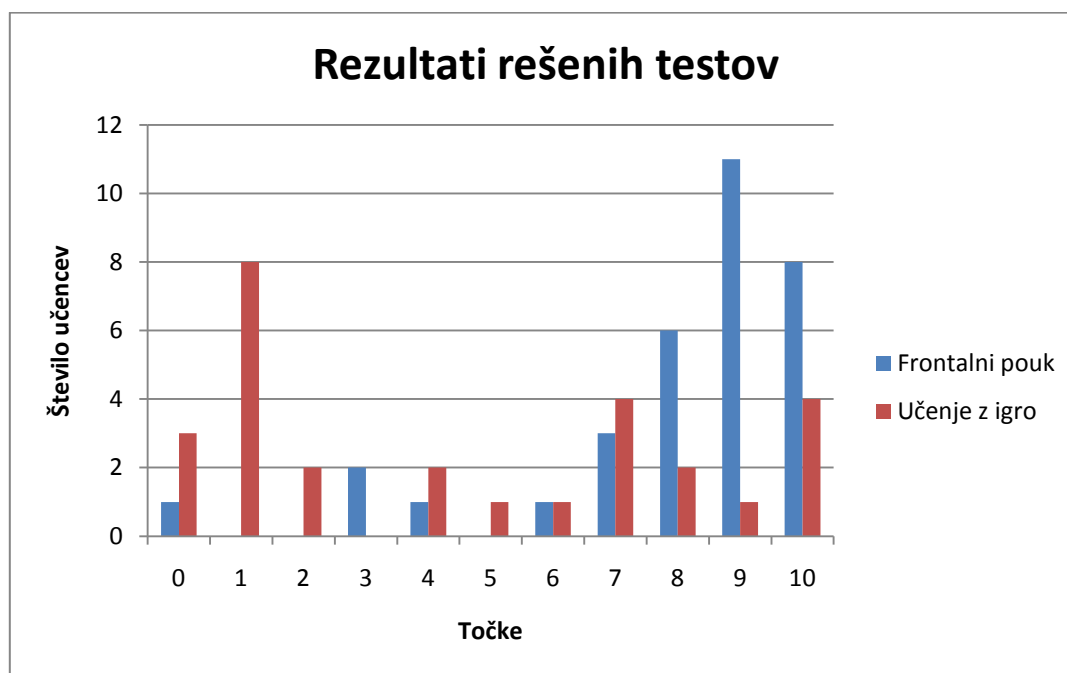
Skupaj smo testirali 61 učencev starih od 13 do 16 let. Od tega je bilo 36 (59%) fantov in 25 (41%) deklet. 28 (46%) učencev se je učilo samostojno, z izobraževalno igro, ostalih 33 (54%) učencev pa se je učilo frontalno.

Spol \ starost	13	14	15	16	Skupaj
Moški	10	20	5	1	36
Ženski	5	12	8	0	25
Skupaj	15	32	13	1	61

Tabela 10: Število vseh učencev po starosti in spolu [7]

Dosežene točke	0	1	2	3	4	5	6	7	8	9	10
Št. učencev (Frontalno)	1	0	0	2	1	0	1	3	6	11	8
Št. učencev (Igra)	3	8	2	0	2	1	1	4	2	1	4

Tabela 11: Koliko točk so dobili učenci pri posameznem načinu učenja [7]



Graf 4: Rezultati testov glede na način izvedbe učne ure [7]

	Aritmetična sredina	Mediana	Standardni odklon
Frontalni pouk	8,00	9	2,33
Učenje z igro	4,43	4	3,61

Tabela 12: Aritmetična sredina, mediana in standardni odklon [7]

Hipoteza: Učenci bolj razumejo snov, če je predstavljena z interaktivno izobraževalno igro, podkrepljeno s teoretičnimi spoznanji, kot pa če je predstavljena na frontalen način.

Hipoteze ne moremo potrditi ali ovreči, saj sta bili eksperimentalni skupini majhni. Iz rezultatov je razvidno, da so učenci, ki so se učili s pomočjo igre, na testu dosegli slabše rezultate. Razlika v povprečnem številu točk je velika. Učenci, ki so se učili frontalno, so v povprečju dobili 8 točk, učenci, ki so se učili z igro pa 4,43 točk. Potrebno je poudariti tudi standardni odklon, ki je pri učencih, ki so igrali igro, kar 3,61.

Rezultate si lahko razlagamo tako, da so učenci igro vzeli bolj za zabavo, ali pa se ne znajo sami učiti in reševati problemov. Učenci, ki so se učili z igro so bili bolj prepuščeni sami sebi in očitno je, da potrebujejo nekoga, ki jih v procesu učenja vedno usmerja. Verjetno je tako zaradi navade, saj v šolah še vedno prevladuje frontalni način poučevanja.

Kljub vsemu pa so v peščici osnovnošolcev tudi učenci, ki izstopajo. Nekatere je igra zanimala in so na testu dosegli zelo dobre rezultate. Ob začetku izvajanja tega eksperimenta nismo vedeli, ali bo ena šolska ura dovolj za reševanje igrice. Za nekatere učence je to kar držalo, za večino pa niti ne. Precej učencev je v igri doseglo cilj in so rešili profesorja. Drugi, ki jih igra ni zanimala, ne bi prišli do cilja niti, če bi imeli na razpolago dve šolski uri.

Na koncu menimo, da ima uporaba igre v izobraževanju nekatere prednosti. Ni ravno primerna za pridobivanje novega znanja, bi pa bila dobrodošla pri uri utrjevanja. S tem bi mogoče bolj motivirali tudi učenci, ki se ne znajo učiti sami.

5.3 Rezultati in analiza ankete

V tem razdelku so najprej predstavljena vprašanja ankete, nato pa še odgovori in analiza odgovorov, ki sva jo napisali skupaj z Natašo [7].

ANKETA	
1. Spol	<ul style="list-style-type: none">• Ženski• Moški
2. V kateri letnik si trenutno vpisan/a?	<ul style="list-style-type: none">• 3.• 4.• Absolvent
3. Ali ti je bila igra všeč?	<ul style="list-style-type: none">• Da• Ne• Drugo
4. Ali se ti zdi uporaba igre primerna v izobraževalnem procesu?	<ul style="list-style-type: none">• Da• Ne• Drugo
5. Ali bi sam/a naredil/a podobno izobraževalno igro?	<ul style="list-style-type: none">• Da• Ne• Drugo
6. Ali bi uporabil/a igro pri učni uri?	<ul style="list-style-type: none">• Da• Ne• Drugo
7. Kako bi jo uporabil/a?	<ul style="list-style-type: none">• Za vajo (utrjevanje)• Za samostojno učenje (učenec se uči sam)• Ne bi je uporabil• Drugo
8. Ali misliš, da bi se učenci bolje/več naučili z igro, kot s frontalno razlago?	<ul style="list-style-type: none">• Da• Ne• Drugo
9. Ali te zanima, kakšne so še ostale vaje in testi v igri?	<ul style="list-style-type: none">• Da• Ne• Drugo
10. Napiši še svoje mnenje o igri: pohvale, pripombe,... [7]	

Prvi dve vprašanji ankete se navezujeta na spol in starost študentov. Anketo je rešilo 23 študentov, od tega 5 moških in 18 žensk. Šest jih je bilo v času anketiranja vpisanih v 3. letnik, pet v 4. letnik, absolventov pa je bilo 12. V nadaljevanju sledi opis odgovorov na posamezna vprašanja.

3. vprašanje: Ali ti je bila igra všeč?

Vsem anketirancem, razen enemu, je bila igra všeč. Ker je to vprašanje subjektivno, saj osebno poznamo večino anketirancev, smo pri zadnjem vprašanju anketirancem dali možnost, da napišejo svoje mnenje o igri.

4. vprašanje: Ali se ti zdi uporaba igre primerna v izobraževalnem procesu?

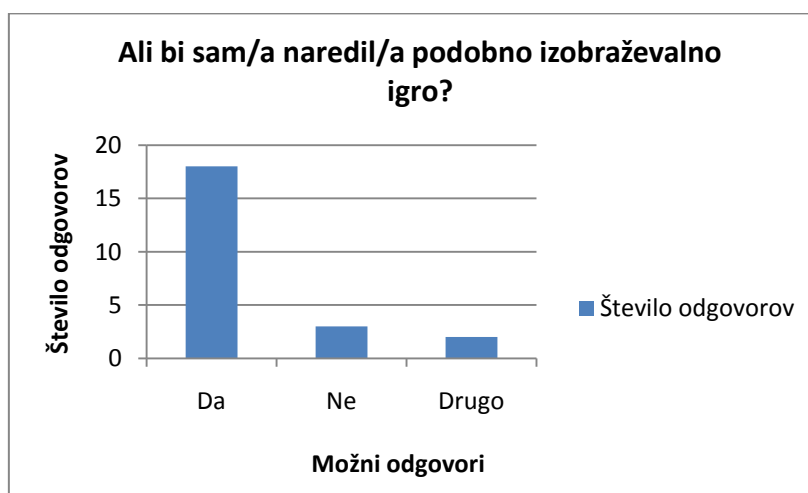
Čisto vsi anketiranci so na vprašanje odgovorili pritrdilno. Iz tega lahko sklepamo, da študentje pedagoške fakultete že razmišljajo o novih metodah in oblikah dela.

5. vprašanje: Ali bi sam/a naredil/a podobno izobraževalno igro?

Ker smo anketirali bodoče učitelje matematike in računalništva, nas je zanimalo, ali bi se sami lotili izdelave podobnih izdelkov. Kar 18 (78%) anketirancev bi se lotilo izdelave igre, medtem ko se trije (13%) ne bi. Dva (9%) pa bi izdelala igro le, če bi imela dovolj časa.

Odgovor	Število odgovorov	Število odgovorov [%]
Da	18	78
Ne	3	13
Drugo	2	9

Tabela 13: Odgovori na peto vprašanje



Graf 5: Odgovori na peto vprašanje

6. vprašanje: Ali bi uporabil/a igro pri učni uri?

Pri četrtem vprašanju smo ugotovili, da se anketirancem zdi uporaba interaktivnih izobraževalnih gradiv pri učni uri primerna. Ker takšna učna ura zahteva veliko priprav, smo s šestim vprašanjem želeli preveriti, koliko anketirancev bi osebno uporabilo izobraževalno igro pri učni uri. Vsi anketiranci so se pripravljani dodatno potruditi pri pripravi na takšno uro, saj so vsi na to vprašanje odgovorili pritrdilno.

7. vprašanje: Kako bi jo uporabil/a?

Več kot polovica anketirancev (64%) bi igro uporabila pri utrjevanju (za vajo), 27% za samostojno učenje, 9% pa se jih je odločilo za kombinacijo obojega. Teh 9% predstavljata dva študenta, ki sta sama v odgovoru napisala to možnost, saj je primarno ni bilo med možnimi odgovori.

Odgovor	Število odgovorov	Število odgovorov [%]
Za vajo (utrjevanje)	14	64
Za samostojno učenje (učenec se uči sam)	6	27
Ne bi je uporabil	0	0
Drugo	2	9

Tabela 14: Odgovori na sedmo vprašanje



Graf 6: Odgovori na sedmo vprašanje

Odgovora pod možnostjo Drugo:

- Tako za vajo, kot za samostojno učenje.
- Pri bolj nadarjenih učencih bi jo uporabila kot pripomoček za samostojno učenje, pri ostalih pa bolj za vajo oz. utrjevanje ter kasnejšo osvežitev snovi.

8. vprašanje: Ali misliš, da bi se učenci bolje/več naučili z igro, kot s frontalno razlago?

Mnenja anketirancev so različna. 14 (61%) jih meni, da trditev drži, 4 (17%) pa menijo nasprotno. Ostalih 5 (22%) anketirancev je napisalo svoje dogovore takole:

- Mislim, da je kombinacija obeh pristopov (frontalna razlaga in učenje z igro) uspešna kombinacija za doseganje učnih ciljev. Pomembno je, da se igra vključi v frontalno razlago, saj je potrebno najprej povedati, zakaj želimo, da jo učenci igrajo; po koncu pa je obvezno narediti evalvacijo same igre in rezultatov, saj s tem igri damo smisel. Torej odgovor: kombinacija obeh pristopov.
- Nekateri zagotovo.
- Najprej potrebujejo frontalno razlago, potem šele pride na vrsto igra. Samo z igro se ne bi dobro naučili, ker se nekaterim ne da brati vsega, kar piše.
- Odvisno od snovi.
- Sem bolj za to, da je igra dopolnilo.

Ti odgovori kažejo na to, da na vprašanje ni tako enostavno odgovoriti le z da ali ne. Koliko se učenci naučijo, je odvisno od učencev samih in od učne snovi. Trije anketiranci so mnenja, da je igra uspešna v kombinaciji s frontalno razlago.

9. vprašanje: Ali te zanima, kakšne so še ostale vaje in testi v igri?

Vse anketirance, razen enega, zanimajo še ostale vaje in testi v igri. Sklepava, da je anketiranec, ki je na to vprašanje odgovoril negativno, tudi edini, ki je pri tretjem vprašanju odgovoril, da mu igra ni všeč.

10. vprašanje: Napiši še svoje mnenje o igri: pohvale, pripombe,...

To je edino opisno vprašanje, ki pa ni bilo obvezno. Nanj je odgovorilo 19 anketirancev. Njihovi odgovori, ki smo jih slovnično popravili za namen diplome, sledijo v nadaljevanju:

- Odlična ideja in izpeljava igre. Veliko vloženega truda. Učencem bi zagotovo bilo zelo všeč reševati takšne naloge in se učiti na takšen način. Skratka pohvalno za avtorici!
- Zelo zanimiva igra, s katero se lahko učenec nauči dvojiškega številskega sistema. Predvsem moram pohvaliti igro z utežmi, saj zelo inovativno

prikaže prehod iz desetiškega v dvojiški sistem in obratno. Glede vsebinskega dela igre nimam pripomb, le pohvalo. Priporočal bi le nadgradnjo grafičnega vmesnika oz. grafičnega dela računalniške igre, saj so barve malenkost pretemne in rahlo demotivirajoče (čeprav je vsebina odlična). Mogoče bi uporabil malo svetlejše in bolj nežne barve, ki se ponavadi uporabljajo za didaktične namene. Če bi ocenil, bi igri dal oceno 4+.

- Fino! Sem se nasmejala pri uvodnem videu. Pri tem bi navodila na koncu malo podaljšala... Pri končnem videu pa ni zvoka (verjetno bi moral biti, ker profesor na koncu nekaj govori). Drugače pa super. Le tako naprej, če že niste na koncu!
- Zelo zanimivo predstavljena tema. Nazorno in z različnimi primeri. Začetna razlaga v učilnici (primeri s tekočinami) se mi zdi še posebej dobra. Pripombe: počasno delovanje in dokaj slaba odzivnost. Nekoliko je nerodno, ker je ravno tako velika igra, da moram malo premikati drsnik - res pa je, da je moj ekran 15,4 inch. Grafična dodelava ne bi škodila. Profesor je videti malo sporen. Aha pa pri testu, v drugem delu, kjer sestavljaš palico, bi tudi lahko umaknile podatek koliko je katera enota, ker potem je preveč enostavno, samo sešteješ-ni potrebno vedeti, katera vrednost je na katerem mestu (tako kot sta to naredili v prvem delu, ko na gumbih in utežeh ni vrednosti).
- Igra je zanimiva, liki zanimivi. Tudi vaje za utrjevanje so zelo zanimive. Dobro je tudi to, da je vmes malo humorja- profesor pade v jamo.
- Igra mi je všeč, edino pri testu s tehtnico, ko sem kliknila na utež 32 ali 16, se utež na tehtnici ni pokazala, drugače je pa super igra. Pri uvodnem delu se na koncu prehitro konča, nisem uspela vse prebrati. Igrica je zabavna in poučna.
- Igra je zanimiva. Pri testu postane zahtevnejša, kar je zelo dobro, je pa še vedno dovolj podobna prejšnji, da učenec uvidi, da je znanje še nepopolno, in ne misli, da je to le zaradi druge igre. Edino pri testu nekaj nagaja miška, če je možen popravek...
- Igra je zelo dobra in poučna.
- Takšne igre so gotovo koristne, predvsem kot popestritev in dopolnilo pouku.
- Odlična igra!!!!
- Super igra.
- Igra zelo zanimivo in jasno predstavi pretvarjanje. Učenci bi imeli občutek, da igrajo igrice obenem pa bi se nekaj naučili. Igra je res dobra. Motilo me je le to, da včasih igra ni na celem zaslonu, prav tako pa mi na zadnji prosojnici razlage ni bilo jasno, na kateri spodnji gumb naj kliknem.

- Superca sta naredili. Čestitke!
- Všeč mi je grafika, mislim da bi pritegnila učence. Hkrati so mi izredno zanimive igre, saj se prek prektičnih nalog naučijo nove snovi.
- Potrditev klikanja na gumb mi je delovalo zelo počasi oz. sem kliknila 3x, da mi je zaznalo en klik - morda je težava le v mojem brskalniku, a tudi ob ponovnem zagonu ni bilo bolje. Čestitke za idejo in izvedbo!
- Igra odpri sef se mi zdi zelo zanimiva in zelo primerna za utrjevanje snovi.
- Igrica je super zasnovana, je dobra...
- Zanimivo, lušno.
- Igrica je zelo dobro sestavljena, saj preverja znanje in razumevanje dvojiškega števila, pretvarjanje iz desetiškega v dvojiško in obratno. Tole igrico bom najverjetneje sam uporabil, ko bom učitelj, saj je izredno dobro narejena in zelo primerna tudi za samostojno učenje.

Veliko anketirancev nam je odgovorilo pozitivno, nekateri pa so podali nekaj kritik. Kritike so se nanašale na počasno odzivnost programa. Ker je igra namenjena igranju z diska in ne prek spleta, pri tej zadevi ne moremo ničesar izboljšati. Igro smo dali na splet zato, da smo anketirancem olajšali reševanje igre (jim ni bilo potrebno igre in vseh dodatkov naložiti na računalnik). Tako imajo tudi vsi ostali zainteresirani hiter vpogled v igro. Dva anketiranca sta opozorila, da se je navodilo ob koncu prve animacije prehitro končalo. V igri je to že rešeno z gumbom, s katerim učenec sam določi čas prikaza navodila. V video posnetku nismo mogli realizirati interaktivnega gumba, zato pa v predvajanju posnetka obstaja možnost pavze. Mnenja glede grafike so različna, nekaterim je všeč, nekaterim pa ni [7].

Hipoteza: Bodoči učitelji matematike in računalništva so sprejemljivi za uporabo didaktičnih interaktivnih iger pri pouku.

Drugo hipotezo ne moremo z zagotovostjo potrditi, saj smo anketirali premalo študentov (23). Sklepamo lahko, da so bodoči učitelji sprejemljivi za uporabo didaktičnih interaktivnih iger pri pouku. To so pokazali odgovori na četrto in šesto vprašanje. Vsi anketiranci (100 %) so sprejemljivi za uporabo interaktivnih izobraževalnih iger v učnem procesu. Večina anketirancev bi izobraževalno igro uporabila za vajo, saj menijo, da se učenci veliko naučijo z igro. Pripravljene so se tudi potruditi in sami izdelati izobraževalno igro. Anketni vzorec je bil majhen, zato ne moremo zagotovo trditi, da naše ugotovitve veljajo za vse bodoče učitelje matematike in računalništva [7].

6 Zaključek

V diplomski nalogi je opisana razredna interakcija ter interakcija med uporabnikom in računalnikom. Opisana sta dva modela interakcije med uporabnikom in računalnikom, predstavljeni pa so tudi elementi, ki jih mora vsebovati dobra igra. V skladu s teorijo smo poskušali izdelati dobro igro. Veliko pozornosti v delu smo namenili opisu igre, saj je njena izdelava zahtevala veliko časa in znanja. S predstavitvijo, kaj vse lahko naredimo s programom Blender, smo poskušali bralca navdušiti, da se še sam loti izdelave kakšne igre, saj jih v izobraževanju primanjkuje.

Rezultati eksperimenta so pokazali, da so učenci, ki so se novo snov učili sami s pomočjo igre, teste rešili slabše kot učenci, ki so snov predelali z učiteljem. Kljub temu smo pri nekaterih učencih, ki so igrali igro, opazili tudi zelo dobre dosežke. Prišli smo do sklepa, da igre ne bi uporabili za samostojno učenje nove snovi. Uporabili bi jo za vajo in utrjevanje snovi.

Bodoči učitelji matematike in računalništva so sprejemljivi za uporabo didaktičnih interaktivnih iger. Upamo le, da bodo za uporabo interaktivnih gradiv v šolah navdušili tudi ostale učitelje ter tudi sami izdelali kakšno zanimivo in inovativno učno gradivo.

Iz anketnih odgovorov smo dobili tudi nekaj idej, kako našo igro še izboljšati. Ena izmed možnosti je uporaba drugačnih barv, druga pa je grafična dodelava. Igro bi lahko tudi nadgradili. Lahko bi jo uporabili za učenje drugih številskih sistemov. Lahko pa izdelamo nove igre, pri katerih obdržimo le glavno zgodbo naše igre (profesor naleti na težavo).

7 Literatura

1. Blender - COKS – Center odprte kode Slovenije. URL: <http://www.coks.si/index.php5/Blender> (Citirano 18. 5. 2011).
2. blender.org – Development. URL: <http://www.blender.org/development/> (Citirano 20. 8. 2011).
3. Dix, A., Finlay, J., Abowd, G., & Beale, R. (2004). *Human-computer interaction* (Third Edition). Harlow: Pearson Education Limited.
4. Doc:Manual/Game Engine - Blender Wiki. URL: http://wiki.blender.org/index.php/Doc:Manual/Game_Engine (Citirano 14. 8. 2011).
5. Gibson, D., Aldrich, C. & Prensky, M. (2007). *Games and simulations in online learning: research and development frameworks*. London: Information Science Publishing.
6. Gumster, J. (2009). *Blender For Dummies*. Hoboken: Wiley Publishing.
7. Kristan, N. (v tiskanju). *3D vizualizacija v izobraževanju*. Diplomsko delo, Ljubljana: Univerza v Ljubljani, Pedagoška fakulteta.
8. Kumpulainen, K., Wray, D. (2002). *Classroom interaction and social learning: from theory to practice*. London: RoutledgeFalmer.
9. Razdevšek-Pučko, C. (1993). *Razredna interakcija: študijsko gradivo za pedagoško psihologijo*. Ljubljana: Univerza v Ljubljani, Pedagoška fakulteta.
10. Westwood, P. (2008). *What Teachers Need to Know About Teaching Methods*. Camberwell, Vic.: ACER Press.
11. Wikipedia contributors. *Educational game*. URL: http://en.wikipedia.org/wiki/Educational_game (Citirano 19. 8. 2011).
12. Wikipedia contributors. *Information and communications technology*. URL: http://en.wikipedia.org/wiki/Information_and_communications_technology (Citirano 15. 8. 2011).
13. Wikipedia contributors. *Interaction*. URL: <http://en.wikipedia.org/wiki/Interaction> (Citirano 8. 8. 2011).