

Grafičko programiranje igara kao metoda za brže usvajanje temeljnih koncepata programiranja

Primjer dobre prakse

Opis/Motivacija

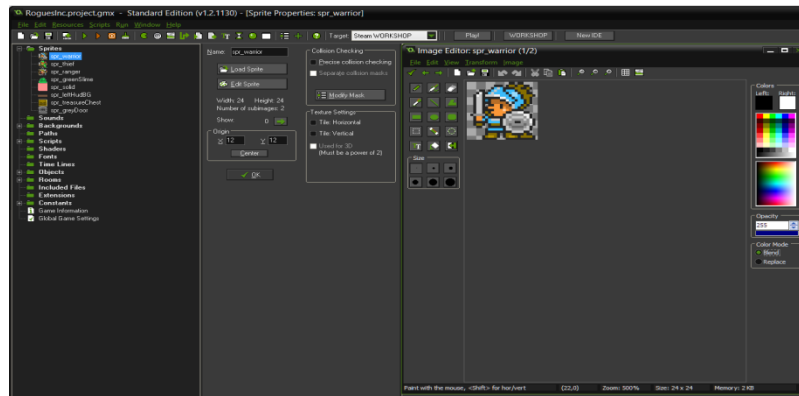
Živimo u vremenu u kojem su djeca izložena neprestanoj interakciji s modernih tehnologija, koje omogućavaju udaljenu komunikaciju, pretraživanje raznih sadržaja i igranje igara. Sam koncept tekstualnog programiranja takvoj generaciji koja je zbog učestale izloženosti tehnologijama, razvila vrlo osebuje vizualne sposobnosti, postaje demotivirajuć i odbojan. Učenici bi trebali naučiti logički i apstraktno razmišljati to jest naučiti stvarati apstraktne modele rješenja, a tek onda te modele realizirati u programski kod. Povećati motivaciju znači nastavni sadržaj prilagoditi očekivanjima učenika. Ukoliko učenici imaju vrlo razvijene vizualne sposobnosti, potrebno je koristiti rješenja i metode koje zadovoljavaju to područje. Uključiti grafičko programiranje prije tekstualnog programiranja, znači naučiti ih razmišljati. Programiranje treba promatrati kao alat pomoću kojeg se razvija apstraktno mišljenje. Upravo to rješenje je prikazano u ovom radu.

Aplikacije za programiranje igara

Aplikacije za programiranje igara namijenjene su stvaranju raznih scenarija računalne igre u dvodimenzionalnoj ili trodimenzionalnoj grafici. Postupak programiranja, ne sastoji se od tekstualnog upisivanja linija koda u programski editor, već grafičkim načinom pisanja programa. Programira se tehnikom pritisni-povuci (*drag and drop*), odnosno slaganjem grafičkih ikona. Grafički način pisanja programa isključuje sintaktičke i leksičke pogreške koje se javljaju pri tekstualnom unosu što olakšava smišljanje rješenja i izradu programa. Aplikacije za programiranje igara, sadržavaju identične programske konstrukte poput većine programskih jezika. Svaki programski konstrukt poput: uvjetovanja, grananja, petlja, funkcija, izveden je u obliku grafičke ikone s istom svrhom i funkcijom kao i u standardnom programiranju u jezicima Python, Logo, C++ ili Basic.

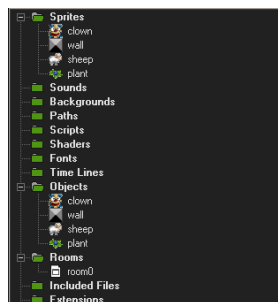
GameMaker-Studio 1.4

GameMaker-Studio 1.4 je komercijalna aplikacija za razvoj dvodimenzionalnih i trodimenzionalnih računalnih igara. Podržana je kao desktop verzija za Windows i Mac OS operacijske sustave te za sustav Android. Dostupna je kao besplatno download rješenje na stranici <http://www.yoyogames.com/studio/download> za probno korištenje u trajanju od trideset (30) dana. U probnoj inačici, korisniku ove aplikacije, dostupni su neograničeni resursi za kreiranje igara poput raznovrsnih gotovih oblika i pozadina. Osim probne verzije, moguće je kupiti licenciranu verziju *Multiplatform development* za 149,99\$ dolara. Tom verzijom se kupuju i prava na dodatne nadogradnje u obliku dodataka ili *plug-ins*. Drugu licenciranu verziju *Complete Studio package* za profesionalni rad, moguće je nabaviti za 799.99\$ dolara[6]. U tu se verziju se ubraja mogućnost distribucije igara, te korištenje dodatnih resursa za korištenje na mobilnim aplikacijama poput aplikacije Android.



Slika 1: GameMaker-Studio 1.4. glavno sučelje

U GameMakeru-Studio 1.4. postoje sve programske strukture koje se uobičajeno koriste u programiranju: uvjetna naredba, *while*, *repeat*, *do i for* petlje, funkcije i nizovi. Programiranje grafičkim ikonama, sastoji se od kreiranja prvotnih objekata koji će poslužiti kao glavni likovi u igrici. Takve objekte nazivamo *Sprites*. To su grafičke ikone ili slike koje predstavljaju objekte. Možemo ih sami nacrtati ili ih preuzeti iz neke baze sličica. Uz njih, potrebno je kreirati i sobu *Rooms* to jest predložak u kojemu će se odvijati radnja. Kao i kod *Sprites* objekata i ovdje je moguće izraditi vlastiti predložak ili koristiti neki iz baze sličica.



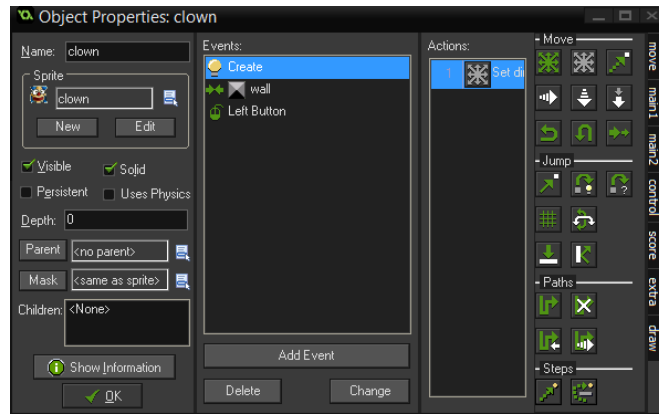
Slika 2: Izbornik za kreiranje osnovnih objekata

Svi objekti, podvrgnuti su opređenim gibanjima koje nazivamo akcijama ili *Actions*. Akcije se mogu odvijati nad događajima podijeljene u nekoliko grupa. Postoje akcije gibanja i/ili skakanja *Move*, akcije za rad s objektima *Objects*, akcije za mjerenje vremena *Timing*, akcije za rad s uvjetima i upitima *Questions*, akcije za različite načine bodovanja *Score*, akcije crtanja *Drawing* i druge.



Slika 3: Izbornik s kcijma za objekte

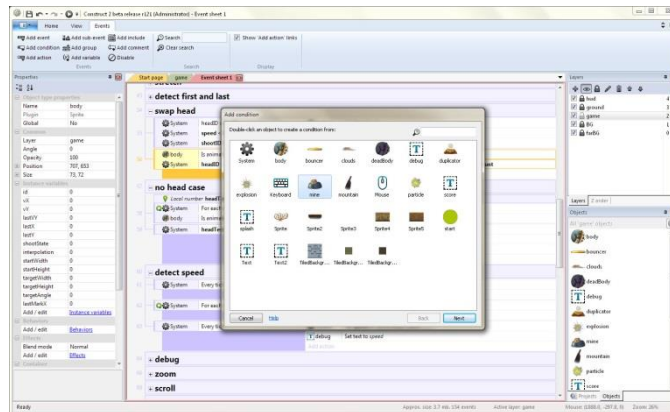
Nakon prethodno uređenih koraka program je moguće testirati naredbom pokreni ili *Run*. Igra će se pokrenuti otvaranjem sobe *Rooms* u kojoj će se nalaziti svi odabrani objekti s dodijeljenim funkcijama.



Slika 4: Dodavanje aktivnosti objektima

Construct 2

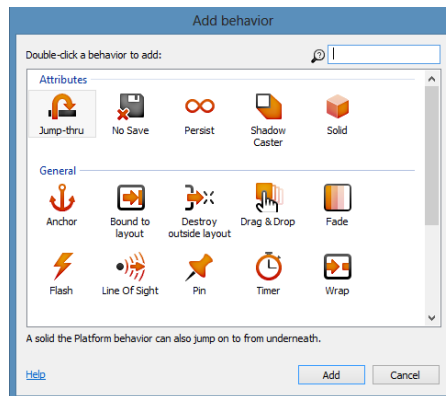
Construct 2 je vrlo jednostavna i ne zahtjevna komercijalna aplikacija za kreiranje i dizajniranje dvodimenzionalnih igara. Ne zahtijeva klasično kodiranje, nego kodiranje putem odabira grafičkih ikona [7]. Ikone se razvrstavaju tehnikom pritisni–povuci (*drag and drop*). Dostupan je kao besplatna download verzija sa stranice <https://www.scirra.com/construct2> u probnom roku korištenja od trideset dana (30). Dostupan je u više komercijalnih inačica. Ukoliko se koristi za osobnu upotrebu, cijena iznosi 129.99\$ dolara. Za poslovnu upotrebu, cijena iznosi 429.99\$ dolara. Naravno u cijenu su ubrojene i nadogradnje te mogućnost za distribuciju igara[8]. Kompatibilan je za rad u više operacijskih sustava: Windows, Mac OS.



Slika 5: Construct 2 glavno sučelje

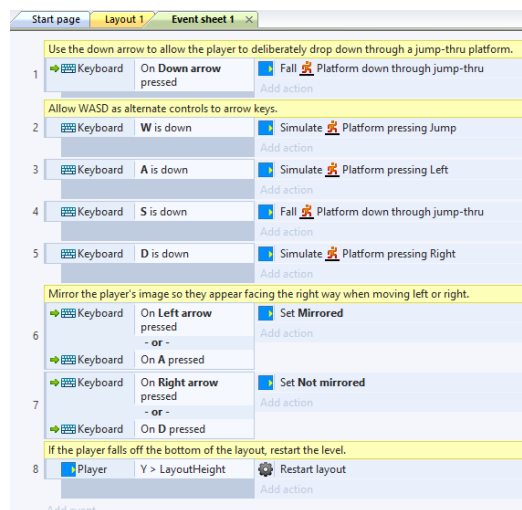
Sučelje *Construct 2* aplikacije, je vrlo intuitivno i jednostavno za korištenje. Sastoji se od nekoliko dijelova. S lijeve strane, nalazi se *Properties* sa svim podacim o pojedinom objektu ili pozadini. Podaci su vezani uz veličinu, posebne efekte, animacije i slično. Odmah do njega, nalazi se tab za dizajniranje objekata i pozadine. Objekti se mogu vlastoručno nacrtati i dizajnirati, ali se mogu preuzeti u obliku slike. Isto vrijedi za pozadinu. Do taba za kreiranje osnovnih objekata i pozadine, nalazi se takozvani *Event Sheet* koji služi kao editor

za programiranje igre. S krajnje desne strane, nalaze se slojevi za rad. Na svaki sloj, moguće je uključiti pojedinačne objekte ili stvoriti novu pozadinu.



Slika 6: Dodavanje osnovnih animacija

Postupak programiranja započinje odabirom ključnih objekata, dizajnom isith te dodjeljivanjem pojedinačnih akcija. Isto vrijedi i za pozadinu. Svaki element igre, moguće je smjestiti na vlastiti sloj čime se olakšava rad.



Slika 7: Određivanje odvijanja akcija

Nakon dizajniranja i animiranja objekata, potrebno je u *Event Sheet* programski odrediti redosljed odvijanja akcija. Kada je to određeno, pozivanjem naredbe *Run Layouti* koaj se nalazi an samom vrhu sučelja aplikacije, pokreće se kreirana igra.

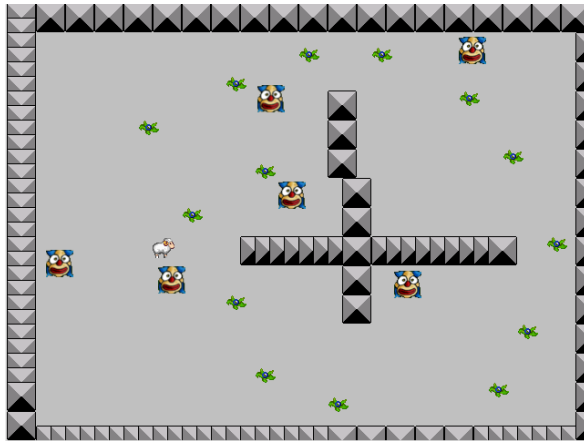
Provedba i rezultati

Metode i materijali

Istraživanje je provedeno na satu informatike nad nekolicinom učenika Druge gimnazije Varaždin. Na izbornom satu informatike, odabrane su dvije grupe učenika prvih razreda. Grupe su se sastojale od učenika s osrednjim znanjem i vještinama programiranja. Svaka grupa učenika je radila u jednoj aplikaciji. Predviđeno vrijeme provođenja istraživanja je bilo dvanaest (12) sati. Od dvanaest sati, dva sata su utrošena na upoznavanje i učenje osnovnih funkcionalnosti rada s aplikacijama, a drugih deset sati za izrađivanje jednostavne dvodimenzionalne igre. Odluka o izradi dvodimenzionalne igre, potekla je iz ograničenja aplikacije Construct 2. Naime ta aplikacija nema mogućnosti kreiranja 3D igara. Svaka grupa učenika je dobila isti zadatak. Željela se utvrditi razina motivacije učenika te zainteresiranost za programiranje kroz koncepte programiranja računalnih igara. Temu i scenarij igre su određivali sami učenici u grupi. Osnovni kriterij izrade igara kojeg su se učenici u obje grupe morali čvrsto držati je bio korištenje programskih konstrukata. Učenici su u obje grupe za izradu igre, trebali koristiti uvjetne naredbe (za ispitivanje uvjeta i postavljanje pitanja) i ponavljanje naredbi (petlje).

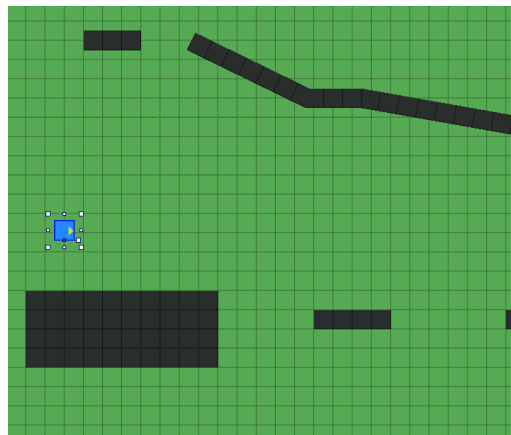
Rezultati

Nakon dvanaest sati rada, uz minimalne intervencije nastavnika, učenici su uspjeli na vrijeme završiti obje igre. U igrama je uvažen osnovni kriterij određen u metodama i materijalima. Dvije grupe, izradile su dva različita scenarija svojih igara. Učenici koji su igru programirali u *Construct 2* bili su gotovi gotovi puno ranije od učenika koji su programirali u *GameMaker-Studio 1.4*. Razlog je bio sljedeći. Aplikacija *GameMaker-Studio 1.4* ima dosta kompleksno sučelje. To znači da se svaka aktivnost mora odrađivati u novome nepovezanome dijaloškom okviru što djeluje dosta zbunjujuće. Ukoliko je za svaki objekt otvoreno više dijaloških okvira, može doći do konfuzije i zatvoriti se pogrešni dijaloški okvir. Kod *Construct 2* aplikacije, učenici nisu imali takvih problema jer nije bilo mnogo dijaloških okvira koji bi ometali u radu. Bez obzira na to učenici su pokazali i dizajnerske vještine i vrlo promišljen scenarij.



Slika 8: Primjer igre izrađene u aplikaciji GameMaker-Studio 1.4.

Scenarij igre izrađene u *GameMaker-Studio 1.4.* temelji se na glavnom objektu (ovci) koja jede travu (nepomični objekti) i te koji u koliziji sa samopokretnim objektima koji mijenjaju smjer svakih pola sekunde biva vraćen na početak igre.



Slika 9: Primjer igre izrađene u aplikaciji Construct 2

Scenarij igre u *Construct 2* temelji se na glavnom objektu pokretanom od strane igrača, koji se suočava s uspješnim savladavanjem prepreka. Prepreke su raznovrsne od ravnih staza, stepenica, do vrlo strmih objekata. Ukoliko glavni objekt ne savlada prepreku, vraća se na početak igre.

Već na samom početku rada u ovom istraživanju prilikom uvođenja u rad ovih aplikacija, učenici su pokazali velik interes za izradu igara pomoću grafičkog programiranja. Iako se radilo o učenicima s osrednjim znanjem i vještinama programiranja, napravili su vrlo uspješne i zabavne igre savladavajući pritom osnovne programske konstrukte: varijable, petlje i uvejtne naredbe. Izrađene igre su rezultat logičkog promišljanja. Sa sigurnošću se može reći da je motivacija bila na razini. Prilikom izrade zadataka, bile su potrebne minimalne intervencije nastavnika. U povremenim intervencijama, učenici bi eventualno tražili savjete oko dizajna pojedinih objekata. Učenci koji su u radu na aplikaciji *Construct 2*, ranije završili

svoju igru, bili su motivirani za izradu dodatka svoje igre. No izvedba zamišljeneog scenarija je bila dosta kompleksna i zahtjevla bi dodatnih pet sati rada. Sa sigurnošću se može reći da je programiranje igara u obje aplikacije jako zainteresiralo i motiviralo učenike.

Zaključak

Učenje programiranja grafičkim programiranjem igara, pokazalo je vrlo dobre rezultate. Postignuta je zadovoljavajuća razina motivacije i interes za programiranje. Učenici osrednjih razina programiranja pa i oni koje nikad programiranje kao proces nije zanimalo, pokazali su veliki interes za rad u ova dva programa. Učenici su proniknuli u značenje i funkciju programskih konstrukata programske petlje i uvjetne naredbe kroz kreiranje igra, bez mukotrpnog i „suhoparnog“ i vrlo demotivirajućeg tekstualnog pisanja programskog koda. Praktičnim radom i paralelnom vizualnom intervencijom s aplikacijama, stvorili su logičko i računalno razmišljanje te tako proširili svoja znanja. Osim toga učenici su se okušali i u vještinama timskog rada. Učenici su osim znanja iz područja logike i programiranja koristili i proširivali svoja znanja iz područja dizajna. Što se tiče samih aplikacija, pokazale su se izrazito korisnim. Iako je *GameMaker-Studio 1.4.* pokazao neke manjkavosti tijekom rada, to nije uvelike utjecalo na kreiranje konačnog rješenja i uspješnosti rješavanja zadatka. U svrhu popularizacije programiranja, bilo bi potrebno uvesti više ovakvih aplikacija.

Korištena literatura

1. Bubica, N. Mladenović, M. Boljat, I. Programiranje kao alat za razvoj apstraktnog mišljenja [pristupano 03.01.2019.]
Dostupno na
<https://bib.irb.hr/datoteka/702093.Programiranje_kao_alat_za_razvoj_apstraktnog_miljenja-CUC-zbornik.pdf >
2. Resnick, M., i dr. Scratch: Programming for all. Communications of the ACM. 2009
3. Mladenović, M. Učenje i poučavanje programiranja temeljeno na igrama [pristupano 03.01.2019.] Dostupno na <http://www.pmfst.unist.hr/wp-content/uploads/2014/06/MonikaMladenovic_seminar.pdf >
4. Kirriemuir, J. McFarlane, A. Literature Review in Games and Learning, Futurelab, Bristol, 2004.
5. Deljac, S. Učenje programiranja programiranjem robota i igara [pristupano 03.01.2019.] Dostupno na< <https://pogledkrozprozor.wordpress.com/2009/03/31/ucenje-programiranja-programiranjem-robota-i-igara/> >
6. Službene stranice aplikacije GameMaker-Studio 1.4. [pristupano 03.01.2019.] Dostupno na< <http://gamedevelopment.tutsplus.com/articles/how-to-learn-gamemaker-studio--gamedev-10911> >
7. Službene stranice aplikacije Construct 2 [pristupano 03.01.2019.] Dostupno na <<https://www.scirra.com/construct2> >
8. Službene stranice aplikacije Construct 2 [pristupano 03.01.2019.] Dostupno na <<https://www.scirra.com/store/construct-2> >