Primjena Construct 2 arhitekture u izradi računalnih igara

Šaravanja, Luka

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET

Sveučilišni studij

Primjena Construct 2 arhitekture u izradi računalnih igara

Završni rad

Luka Šaravanja

Osijek, 2017.

SADRŽAJ

1. UVOD

2D igre su jednostavne računalne igre, najčešće izrađene kao web igre. 2D igre su digitalne slike sastavljene od 2D geomtrijskih modela, tekstova i animacija. Za izradu ove igre upotrebljena je Construct 2 arhitektura. Construct 2 je *drag and drop* grafički program za izradu 2D igara. Construct 2 nam služi kako bi slikama, modelima i različitm teksturama dodali određene radnje i animacije. Kako se prolazi kroz sam proces izrade igre vidjet će se igra od svoje prve do konačne verzije. Program korišten pri izradi igre jedan je od najlakših alata jer ne zahtjeva skoro nikakvo znanje programiranja. Također sam program može naučiti korisnika osnovama programiranja i kako zapravo funkcionira izrada igre. Construct 2 ima jako dobro razvijen *event system* i samim time omogućeno je da se prenese u igru sve zamišljeno bez znanja nekog od programskih jezika. Posao izrade igre olakšava *drag and drop* izbornik jer je sve nadohvat ruke i ne mora se pretraživati više opcija. Koristeći ovaj program također se razvija logičko razmišljanje tvorca igre jer sam mora postaviti sve uvjete, objekte, akcije ili događaje.

1.1. Zadatak završnog rada

Zadatak ovog završnog rada je cijelokupan proces izrade igre, objašnjenje pojmova i dijelova programa korištenog pri izradi igre. Ovaj završni rad sastoji se od dva dijela: pismenog i praktičnog(računalnog). U pismenom dijelu rada objašnjava se korisničko sučelje Construct 2 arhitekture te pojmovi unutar sučelja i prolazi se kroz sam proces izrade igre. U praktičnom dijelu rada izrađuje se sama igra pomoću Construct 2 računalne arhitekture.

2. CONSTRUCT 2 ARHITEKTURA

U ovome poglavlju opisana je sama Construct 2 arhitektura. Detaljnije je objašnjeno od kojih se elemenata sastoji jedan projekt i tehnologija koja se koristi pri izradi igre u Construct 2 arhitekturi te na koji su način riješeni problemi vezani uz memoriju.

2.1. Struktura projekta

Svemu što se nalazi u jednome projektu pristupa se pomoću projektne trake(engl. *project bar).* Projektna traka sadrži popis svih elementa povezanih sa projektom. Struktura projekta biti će detaljnije objašnjenja u poglavlju 4. [1]

Izgled projekta(engl. *layout*)-leveli, meniji, naslovni zasloni su *layouts* odnosno scene, faze i okviri samog projekta. [1]

Lista događaja(engl. *event sheet*)-objašnjava cijelu logiku igre. U Construct 2 arhitekturi lista događaja zamjenjuje programiranje i korištenje skripti. Svaki od *layout-a* ima vlastitu listu događaja i logiku. Ako neki od *layout-a* imaju istu logiku, moguće je dijeljenje liste događaja. [1]

Tip objekta-određuje klasu objekta. Moguće je stvaranje više instanci istog objekta. Bitno je razlikovati tip objekta i instance objekta. [1]

Sistemski objekt-ugrađena funkcionalnost Construct 2 arhitekture, predstavlja jedini objekt koji prazan projekt sadrži. Pruža mogućnost pristupa *game engineu* i uslugama koje su korisne za većinu igara. [1]

Zvukovi i glazba-sve audio datoteke koje se koriste za zvučne efekte i glazbu u igri. Zvukovi se upotrebljavaju za kratke zvučne efekte, a duži zapisi se koriste kao glazba u igri. Sve glazbene datoteke su u dva formta: Ogg vorbis(.ogg) i MPEG-4AAC(.m4a). Zbog toga su podržani svi internet preglednici. [1]

Zajedničke jedinice(Common units)-u Construct 2 arhitekturi potrebno je unositi podatke poput kutova, brzine i veličine kako bi se korisnicima zajedničke jedinice olakšalo korištenje iste jedinice. Pozicije i veličine su u pixelima, kutovi u stupnjevima (0° je desno), vrijeme je u sekundama i brzina je u pixelima po sekundi. [1]

2.2. Tehnologija

Construct 2 arhitektura izrađuje HTML5 igre za širok spektar različith uređaja i operativnih sustava poput mobitela, laptopa i računala.

HTML5 je *HyperTextMarkupLanguage* ili standardni način za izradu web stranica od kada postoji internet. HTML5 najnovija je verzija HTML jezika koja se počinje upotrebljavati 2011.godine. Kod izrade web igara najnovija verzija i nije donijela puno novosti ali je poboljšala 2D/3D grafiku i efekte, brzinu i upotrebu hardware-a. Svakako najzanimljvija i najvažnija novost je <canvas> tag koji omogućuje izradu grafika pomoću skripte (najčešće Javascripta). Sve Construct2 igre koriste <canvas> tag.[2]

<canvas> tag omogućuje dva različita načina za prikaz igre na web stranici: 2D ili WebGL. WebGL je brži i ima više mogućnosti ali nije svuda dostupan te Construct 2 podržava oba načina. Ako na nekom uređaju WebGL nije dostupan arhitektura sama prebaci na 2D način prikaza igrice tako da svi mogu koristiti proizvod. <canvas> element stvara crtajuću površinu fiksne veličine za stvaranje i manipuliranje prikazanim sadržajem.

Javascript je standardni jezik za web. Osim što omogućuje dodavanje različitih interakcija, Javascript ima puno različitih biblioteka. Ovo je *garbage-collected*[1] jezik što dovodi do toga da igre ponekad zastaju. Construct 2 napravljen je tako da stvara minimalno *garbagea* reciklirajući objekte gdje god je to moguće.[3]

Offline igre-ovaj problem je u Construct 2 arhitekturi riješen pomoću HTML5 AppCache mehanizma. Osim što omugćuje neaktivnim(engl. o*ffline)* korisnicima da uživaju u proizvodima, ovaj mehanizam ubrzava rad servera jer podatci se skidaju samo jednom, a kasnije se učitavaju lokalno s diska. Aplikacije učitane pomoću AppCache mehanizma rade i ako neaktivan(engl. korisnik koristi osvježi(engl. *refresh)* tipku.[4]

[1] „Garbage-collected“-služi za oslobađanje memorije koju zauzimaju objekti koji više nisu dostupni programu.Objekt postaje garbage tek nakon što nestanu svi pozivi na njega.

2.3. Korištenje memorije

Kako bi se osiguralo da širok spektar uređaja može koristiti Construct 2 projekte vodila se briga o korištenju memorije. Korištenje memorije najbitnije je za mobilne telefone gdje je ograničena potrošnja memorije.

SLIKE-kako su slike objekata i njihove animacije najzastupljeniji dio svakog projekta i koriste najviše memorije, Construct 2 prikazuje njihovu potrošnju memorije na traci stanja(engl. *status bar).* Iako sama Construct 2 arhitektura vodi računa o veličini preuzimanja i korištenju memorije, developeri bi trebali uzeti u obzir ovaj broj kako bi napravili projekt koji će glatko raditi. Postoji nekoliko načina kojima bi sam *developer* mogao utjecati na performanse igre. To su: rekompresija slike, nedupliciranje slika i korištenje spritesheetinga. Rekompresiju slike obavlja sam program i na to ne treba obraćati veliku pozornost. *Spritesheeting* je jedan od mehanizama koji može uštediti dosta memorije. To je proces kojim se na velikoj slici napravi puno različitih animacija umjesto da svaku animaciju radi zasebno. To smanjuje broj HTTP [2] zahtjeva i veličinu preuzimanja.Construct 2 arhitektura učitava slike samo sa trenutnog *layout-a* kako se ne bi učitavao cijeli projekt u memoriju odjednom. Sve slike i animacije koje su potrebne za određeni layout se učitaju pri njegovom pokretanju te nakon završetka tog *layouta,* ako više nisu potrebne za sljedeći, brišu se iz memorije. Program je napravljen tako da prije svakog *layouta* učitava objekte u *layout view*. Ako neki objekt nije spremljen u *layout view* lista događaja ga stvori a program će ga morati učitati u trenutku stvaranja, što može dovesti do kratkih zaustavljanja igre.Jedna od bitnih stavki pri korištenju memorije su audio zapisi. U Construct 2 arhitekturi bitno je raspodijeliti audio zapise na zvukove i glazbu(duži zvučni zapisi). Kraći zvukovi su automatski dekompresirani unutar memorije kako bi mogli biti pušteni bez ikakvog prvotnog učitavanja i dekompresije, što dovodi do toga da nema kašnjenja. Kao i kod slika kompresirana veličina ne utječe na korištenje memorije nego samo na veličinu preuzimanja(engl. *download).* Za razliku od kratkih zapisa, duži glazbeni zapisi nisu učitani u memoriju već se puštaju. To znači da je u pozadini napravljen reprodukcijski međuspremnik određene dužine. Također, duži glazbeni zapisi mogu biti pušteni i prije nego su u potpunosti preuzeti.[1]

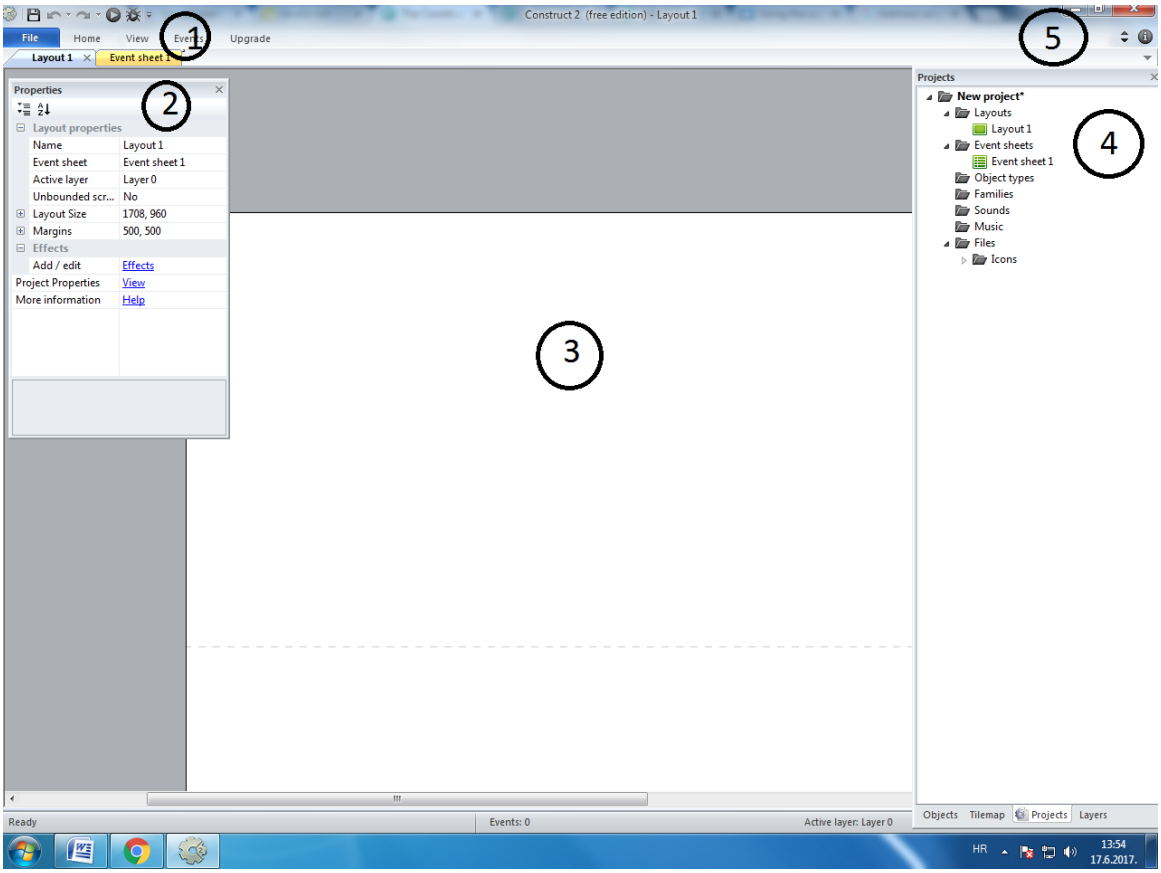
[2] HTTP je skraćenica engleskog naziva HyperText Transfer Protocol. To je protokol, odnosno skup pravila koja se koriste za prijenos hipertekstualnih dokumenata (web stranica) između dva računala.

3. CONSTRUCT 2-KORISNIČKO SUČELJE

U ovome poglavlju prikazano je i objašnjeno korisničko sučelje programa i njegovih najvažnijih karakteristika. [1]

3.1 Osnovni dijelovi korisničkog sučelja

Na slici 3.1.1 vidi se kako izgleda korisničko sučelje Constrcut 2 arhitekture praznog projekta.



Slika 3.1.1-pogled na korisničko sučelje:

1-*File* izboornik i *Ribbon* kartice
2.Traka za svojstva
3.Izgled *layout-a*
4.Projektna traka
5.Tipke na desnom vrhu

3.2. Ribbon traka

Slika 3.2.1-prikaz *ribbon* trake

Ribbon traka pruža pristup glavnim značajkama programa. Nalazi se na samom vrhu korisničkog sučelja i sastoji se od nekoliko podkartica. *Ribbon* menu po pravilu je skriven sve dok ne kliknemo na njega. *Ribbon* traka dobila je ime po engleskoj riječi ribbon što označava duguljastu, usku traku.

Home kartica

Slika 3.2.2-prikaz Home kartice na ribbon meniju

Home kartica, kao i kod većine drugih programa, pruža osnovne mogućnosti: kopiraj(engl. *copy)*, zalijepi(engl. *paste)*, izreži(engl. *cut)* ,*undo, redo*. Izaberi(engl. *select)*, izaberi sve(engl. *select all)* i izbaeri ništ(engl. *select none*) utječu na ono što se odabere (listu događaja ili izgled levela(layout)). Tipka postavki(engl. *configurations button*) omogućuje postavljanje različitih verzija igre bez potrebe snimanja više projekata. Aktivne postavke(engl. *active configuration)* pokazuje koja se verzija trenutno koristi ili uređuje. *Run layout* tipka omogućuje pokretanje igre i da se vidi sve što se do tada izradilo.

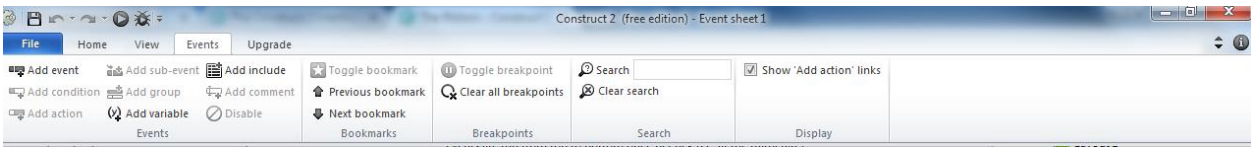
View kartica

Slika 3.2.3-prikaz view kartice na ribbon meniju

View kartica omogućuje odabir ili skrivanje određenih alatnih traka. Zoom opcija omogućuje zoomiranje liste događaja ili izgled projekta.

Pomoću *grid* opcija možemo pojedine objekte, na izgledu projekta, stavljati u zamišljenu rešetku(grid). Zaslonski dio(engl. *display)* odnosi se samo na izgled projekta. Može se koristiti kako bi prikazali neaktivne slojeve(engl. *layer)*, tj. one koji se trenutno ne uređuju i način na koji ih prikazuje- ispod trenutnog sloja djelomično prozirne. Djelomično prozirni l slojevisu blokirani i ne možemo odabrati objekte na njima. Ova opcija može biti vrlo korisna pri uređivanju pojedinog sloja.

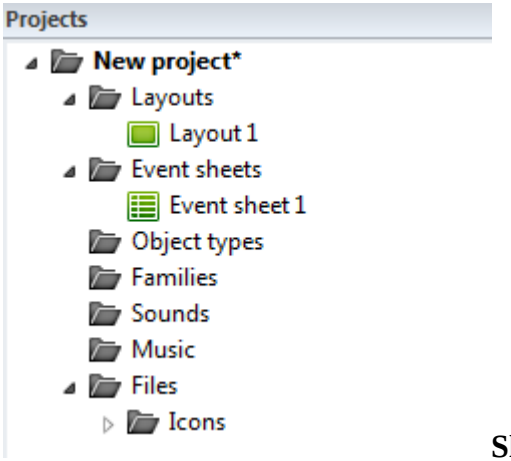
Kartica događaja(event tab)



Slika 3.2.4-prikaz kartice događaja na ribbon meniju

Kartica događaja vrijedi samo za listu događaja(engl. *event sheet*). Pomoću tipke dodaj mogu se dodati novi događaji, poddogađaji, uvjeti i varijable. Pomoću tipke *disable* može se neke događaje ili uvjete uključiti ili isključiti, što je vrlo korisno kod testiranja projekta jer se ne mora ponovno pisati. Također jedna od vrlo korisnih opcija je pretraga(engl. *search)*, pomoću nje se može pronaći događaj, uvjet i varijabla na listi događaja pomoću određene riječi.

3.3. Projektna traka

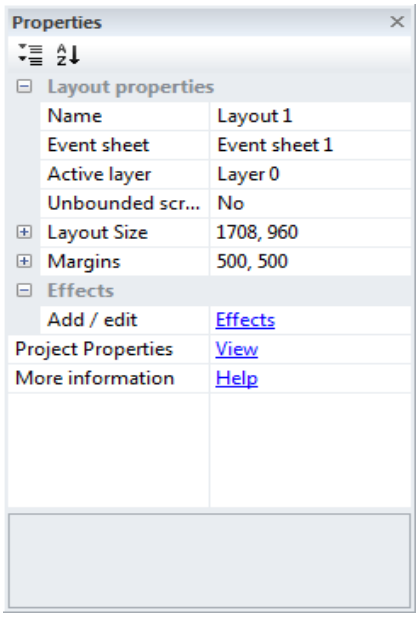


Slika 3.3.1-prikaz projektne trake

Projektna traka možda je i najvažniji dio samog projekta jer na njoj se vidi sve što se nalazi u jednom projektu. Ukoliko se radi u licenciranoj verziji projekta moguće je dodavati podmape po vlastitoj želji te na taj način organizirati projekt. Na projektnoj traci najbolje se vidi koje prednosti

donosi *drag and drop* opcija jer se može lagano podesiti po vlastitoj želji. Pomoću desnog klika na neki od dijelova projekta pristupa se postavkama ovisno o tome što smo kliknuli. Također služi za dodavanje vanjskih datoteka u projekt.

3.4. Traka za svojstva(engl. *properties bar*)



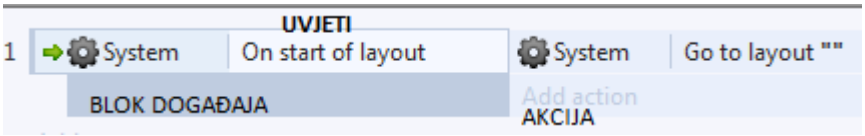
Slika 3.4.1:prikaz trake svojstava

Uz projektnu traku ova traka sigurno je jedan od najvažnijih dijelova programa, pomoću nje se određuju postavke svega što je odabrano. Nemoguće je nabrojati sve te postavke jer svaki dio projekta ima drugačije postavke. Svojstva su raspoređena po određenim kategorijama, poput veličine, teksta i efekata. Kada je odabran određeni element projekta, poput nekog objekta u traci za svojstva, otvorit će se postavke vezane samo za taj element. Postavke mogu imati: projekti, slojevi, izgled projekta, objekti, instance objekta i animacije. Kako bi se olakšalo korisnicima pri odabiru određenog svojstva, na dnu trake vidi se objašnjenje tog svojstva odnosno za što služi.

3.5. Lista događaja(engl. *event sheet*)

Slika 3.5.1-prikaz liste događaja

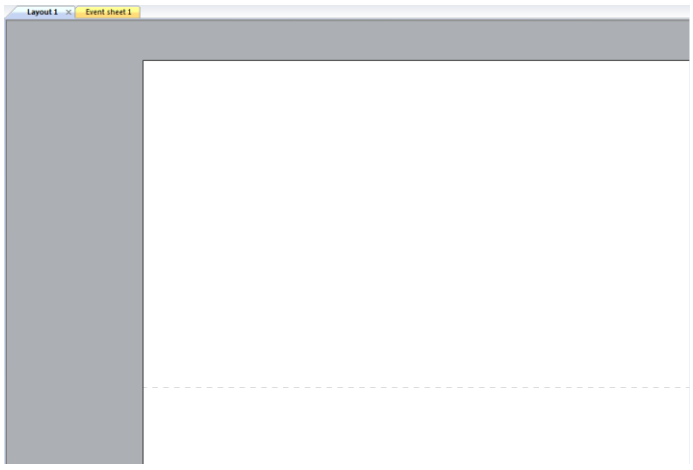
Ovaj dio programa služi za dodavanje događaja i logike u projekt. Iako je ovaj dio programerski, Construct 2 napravljen je na način da omogući svima bez većeg znanja programiranja da naprave projekt.



Slika 3.5.2-prikaz tri glavna dijela liste događaja

Na slici 3.5.2 vidi se da je lista događaja sastavljena od tri glavna dijela: blok događaja, uvjeta i akcija. Blok događaja služi za odabir pojedinog događaja. Dodavanjem više događaja dodat će se i više blokova događaja. Kao što je rečeno prije Construct 2 je *drag and drop* program, i na listi događaja može se na taj način pomicati događaje,uvjete ili akcije. Moguće je organizirati događaje u pojedine grupe događaja.

3.6. Izgled projekta(engl. *layout view*)



Slika 3.6.1-prikaz izgleda praznog projekt

Može se reći da je izgled projekta zapravo grafički dizajn igre. Iscrtkani dio na slici 3.6.1 predstavlja veličinu prozora igre. Sve što se napravi u izgledu projekta radi se pomoću *drag and drop* - od dodavanja objekata do dodavanja običnog teksta. Da bi se počelo uređivati liste događaja povezane s layoutom dovoljno je kliknuti desnim klikom na layout i izabrati edit event sheet. Jedna od zanimljivih stavki layout view-a je prikaz odabranih efekata bez pokretanja samog layouta (za tu opciju mora biti uključen WebGL).

4. ELEMENTI PROJEKTA

Riječ projekt u Construct 2 arhitekturi označava kompletnu igru ili aplikaciju. Projekt sadrži sve, od levela do glazbenih zapisa. Sve što se nalazi u jednom projektu vidi se u projektnoj traci opisanoj u poglavlju 3.3..

4.1 Izgled projekta(engl. l*ayout*)

Ovaj dio projekta odnosi se na izgled level-a, naslovnog menija i game over menija. *Layout* može se u ovome slučaju prevesti kao faza, soba ili level. Sve što ste mijenja u ovom dijelu projekta radi se u prethodno opisanom izgledu levela(engl. *layout view).* Svaki *layout* ima svoju listu događaja koja određuje funkcije tj.kako pojedini *layout* radi. Svaki layout se sastoji od više slojeva ovisno o potrebama. Osim liste događaja, *layouti* imaju mogućnost primjene efekata. Svaki *layout* ima nekoliko karakteristika koje ga opisuju. To su ime,lista događaja, aktivni sloj, veličina i margine.

4.2 Slojevi(engl. l*ayers*)

Sloj je dio izgleda projekta. Slojeve se može zamisliti kao prozirne papire na kojima su dodani različiti objekti. Služe za prikazivanje razlitičih objekata, jedni ispred ili iza drugih, u pozadini. Slojevi se mogu grupno ili pojedino rotirati i prikazivati te tako tvoriti posebne efekte. Najvažnija svrha slojeva je prikazivanje nepomičnih dijelova levela poput *HUD-a*, *foregrounda* i *backgrounda*. Postoje takozvani globalni slojevi koji rješavaju problem ako više *layouta* ima isti sadržaj na istom sloju. Ako se uključi postavka za globalni sloj za pojedini sloj, svaki sloj s istim imenom u projektu će poprimiti postavke tog globalnog. Promjena neke postavke u globalnom sloju utječe na sve slojeve istog imena. Sloj ima nekoliko stvari po kojima ga se razlikuje a to su: ime, providnost, rate, global, itd.

4.3 Objekti

Objekti su sve što igrač koristi pri igranju. Objekti su najvažniji dio samog projekta koji odrađuju sve zadaće. Pri dodavanju novog objekta u projekt prvo treba odlučiti kojeg će *plugina* (sprite ili nešto drugo) biti objekt. Kada se odabere plugin potrebno je odabrati tip objekta. Nakon toga može se napraviti više instanci pojedinog tipa. Bitno je razlikovati tip objekta i instancu objekta. Plugin predstavlja što će točno biti pojedini objekt(*sprite*,zvuk,tekst).

Tip objekta predstavlja klasu objekta što znači da postoji više različitih klasa *plugina* sprite ili nekog drugog. Svaki pojedini tip može imati svoja svojstva. Određeno ponašanje dodaje se određenoj klasi objekta. To znači da svi objekti tipa, tj. sve instance objekta, *„enemy“* imaju isto ponašanje.Spremnici (engl. *container*) pružaju mogućnost stvaranja složenih objekata (objekti stvoreni od više objekata). To znači da je moguće povezati određenu grupu instanci nekog objekta da tvori nešto novo.

4.4 Događaji(engl. *events*)

Događaji su način na koji projekt funkcionira ili prikaz logike projekta. Construct 2 arhitektura riješila je problem logike na vrlo jednostavan način. Umjesto kompliciranih skripti ili programskih jezika dodani su takozvani logički blokovi. Logički blokovi su događaji. Jedna od dodatnih prednosti samih događaja je filtriranje instanci objekta pod određenim uvjetima te izvršavanje akcija na tim instancama. Svaki logički blok sastoji se od uvjeta, akcija i pod-događaja. Nakon što pojedini događaj završi, izvršava se slijedeći koji ponovno provodi filtriranje neovisno o prethodnom. Ako postoje pod-događaji oni se nastavljaju izvršavati tamo gdje je stao njihov glavni događaj. Kao i kod programskih jezika postoje petlje. Petlje u Construct 2 arhitekturi označavaju izvrašvanje pojedinih događaja ili akcija više puta.

5. IZRADA IGRE

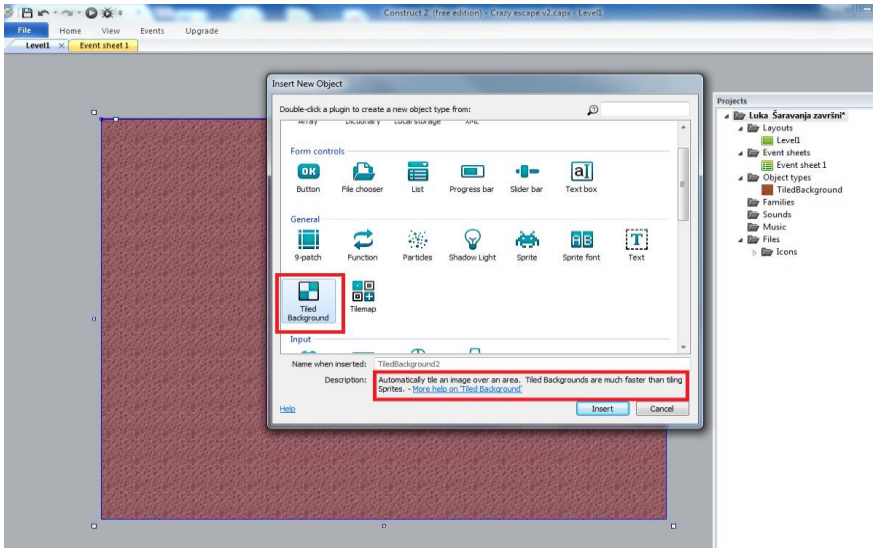
U 5.poglavlju prolazi se izradu igre od njene prve do finalne verzije. Prije nego što se krenulo u izradu igre u arhitekturi trebalo je smisliti koncept same igre i sve njene dijelove. Sama igra smišljena je na način da se igrač (engl. *player*) nalazi u nekoj vrsti labirinta te mora pronaći ključ kako bi uspješno završio igru. Zadatak igraču otežava ograničen vidokrug jer vidi samo dio levela koji je osvijetljen. Igrač posjeduje baterije i šibice koje mu osvjetljavaju dio levela. Baterija koja se troši, nestaje nakon nekog vremena, ima veći vidokrug od šibica i igrač prvo koristi bateriju pa šibice. Nakon nekog vremena u levelu se počinju stvarati i „neprijatelji“ koji otežavaju pronalazak ključa te mogu dovesti do kraja levela (engl. *game over*) ukoliko ih igrač ne izbjegava.

5.1 Pozadina i izgled levela

Pozadina tj.background levela napravljena je na taj način da je prije samog umetanja u level u programu za izradu slika napravljen uzorak koji će poslužiti kao pozadina(SL.5.1).

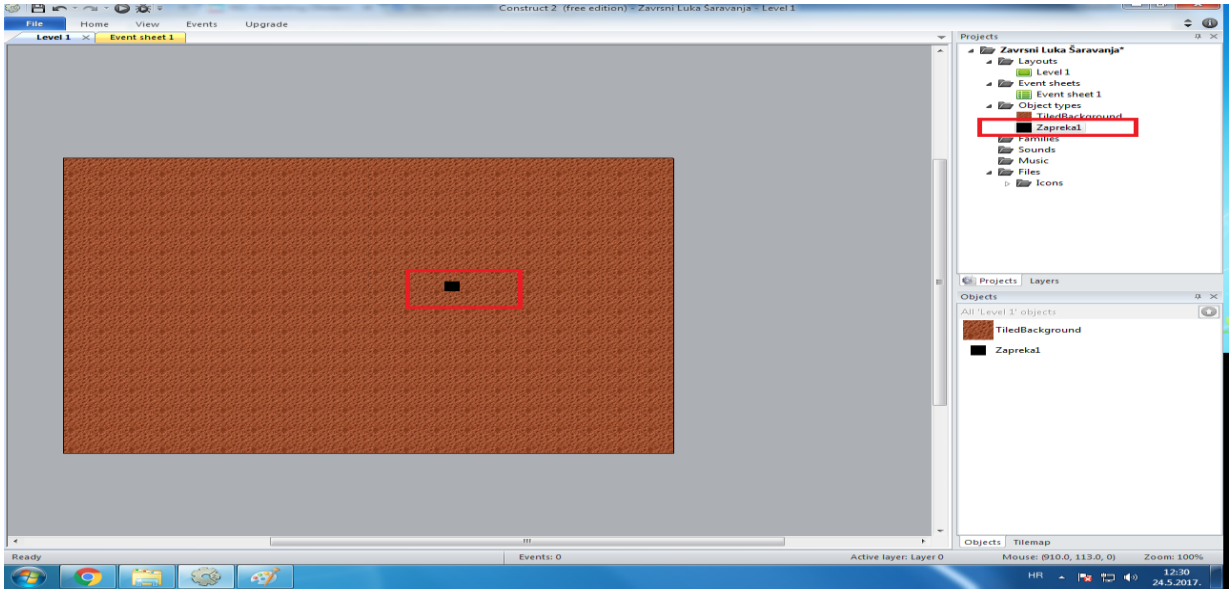
 **Slika 5.1.1**

Nakon što je uzorak napravljen pomoću opcije *TiledBackground*, raširen je po cijelom prozoru. *TiledBackground* služi kako bi se pravilno rasporedio uzorak po cijeloj površini što je bolje, jer se brže učitava nego da se sliku raširi kao objekt.



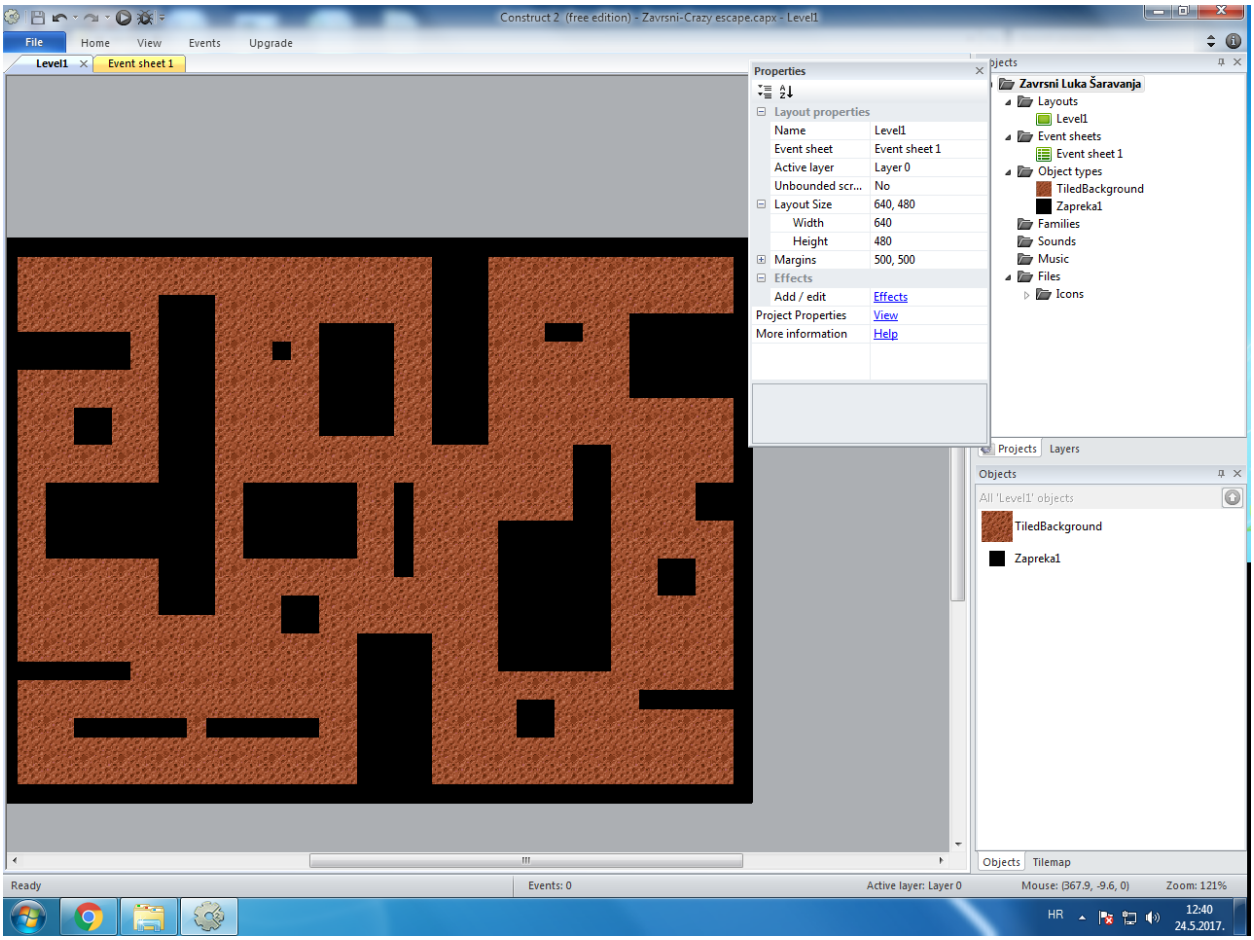
Slika 5.1.2-dodavanje pozadine level-a

Nakon što je dodana pozadina, na isti način izrađen je uzorak za zapreke koje će biti raširene po cijelom levelu (slici 5.1.2). Na isti način kao i uzorak za pozadinu, uzorak za zapreke izrađen je u programu za obradu slika.



Slika 5.1.3-dodavanje uzorka zapreke

Prvo što je napravljeno pomoću uzorka zapreke je zid koji će okružiti level tj.ograničiti kretanje igrača. Na isti način, kao zid koji okružuje level u *layout viewu*, koji je ponuđen u samoj Construct 2 arhitekturi, nacrtan je „labirint“ pomoću uzorka. U *layout viewu* pomoću *drag and drop* funkcija može se nacrtati željeni izgled levela. Jedan uzorak veličine je 16x16pixela kako bi se lakše nacrtao labirint.

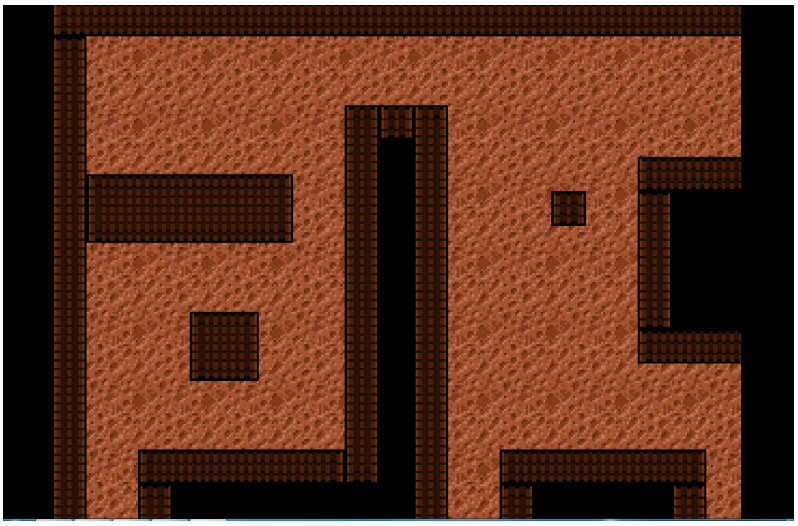


Slika 5.1.4-izgled levela(„labirint“) nakon dodavanja zapreka

Kako bi igra,tj. sam level izgledao vizualno atraktivnije, na sve zapreke i zidove dodan je još jedan uzorak prethodno napravljen u programu za obradu slika. Uzorak predstavlja ogradu na zaprekama.

Slika5.1.5-uzorak za ogradu

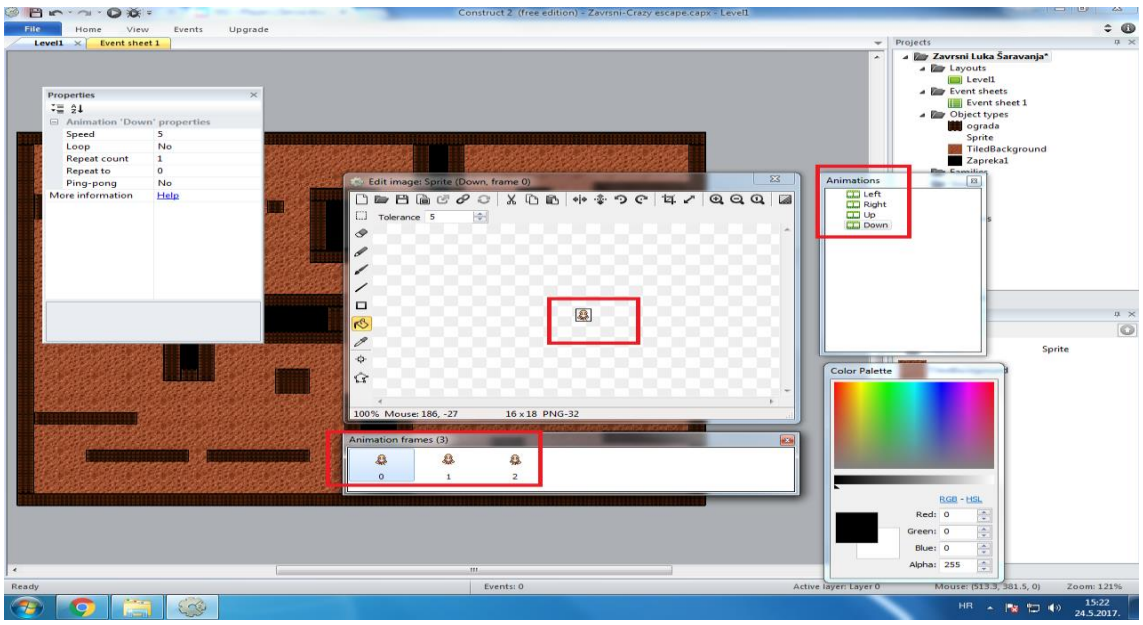
Nakon umetanja svih potrebnih uzoraka i uređivanja samog levela izrađena je i prva verzija same igre.



Slika 5.1.6-prva verzija igre sa uređenim izgledom levela

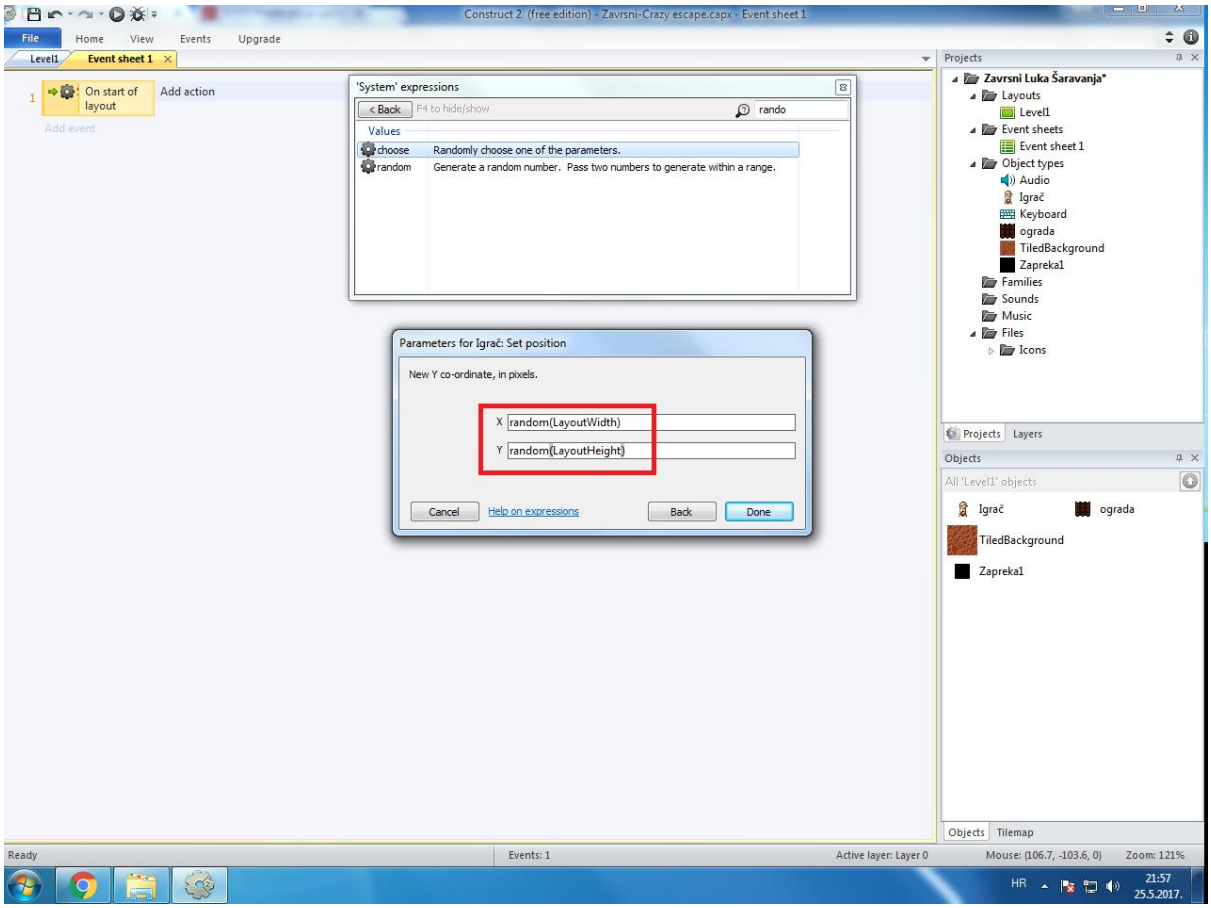
5.2 Igrač

Kao pozadina i zapreke, izgled samog igrača napravljen je prije u programu za obradu slika. Radi korištenja animacije za kretanje igrača, igrač je umetnut kao Sprite objekt. Igrač mora imati 4 različite animacije: kretanje za naprijed, nazad, lijevo i desno. Izgled animacija napravljen je prije u programu za obradu slika te pomoću „import sprite strip“ naredbe, koja je jedna od mogućnosti Sprite objekata. Napravljena je animacija tako da se igrač kreće 5 pixela po animaciji (Slika 5.2.1). Import sprite strip pruža mogućnost dodavanja više slika istog objekta koji se tada pretvaraju u animaciju.

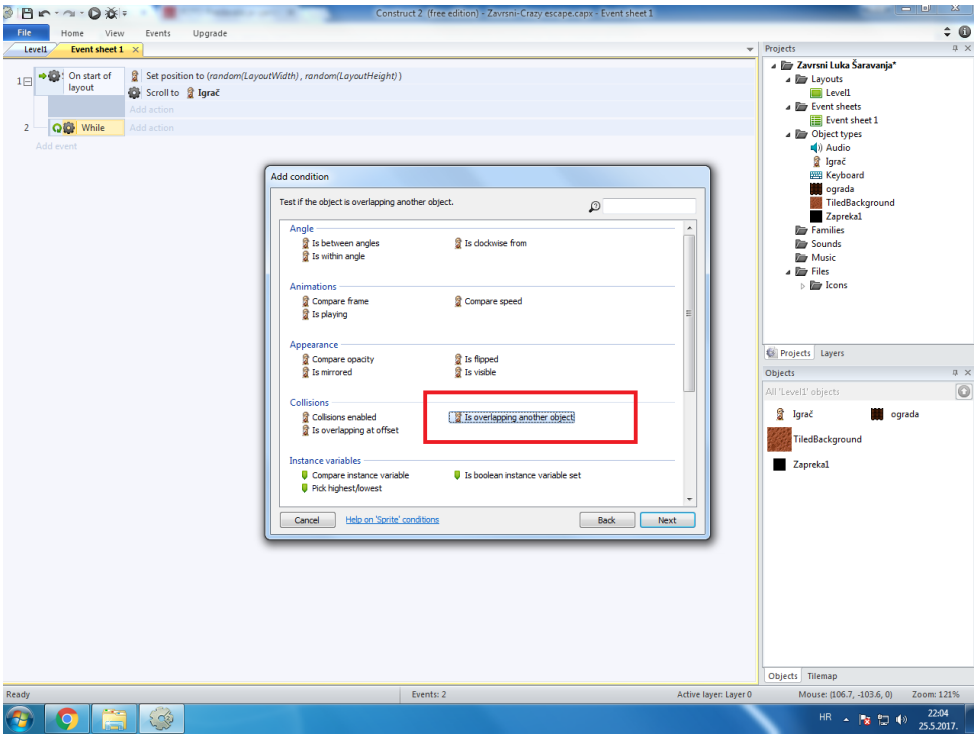


Slika 5.2.1

Nakon što je dodan igrač kao objekt u level pomoću liste događaja, treba se napraviti da na početku *level layouta* igra stvori i postavi igrača na slučajnu poziciju koja nije na ogradi ili zidu kako bi se igrač mogao nastaviti kretati. Construct 2 arhitektura nudi mogućnost da se akcija izvrši na početku određenog layouta. Napravljeno je da na početku level *layout-a* igrač bude postavljen tj.stvoren na slučajnoj poziciji (slika 5.2.2) a da to nije ograda ili zid(slika 5.2.3).

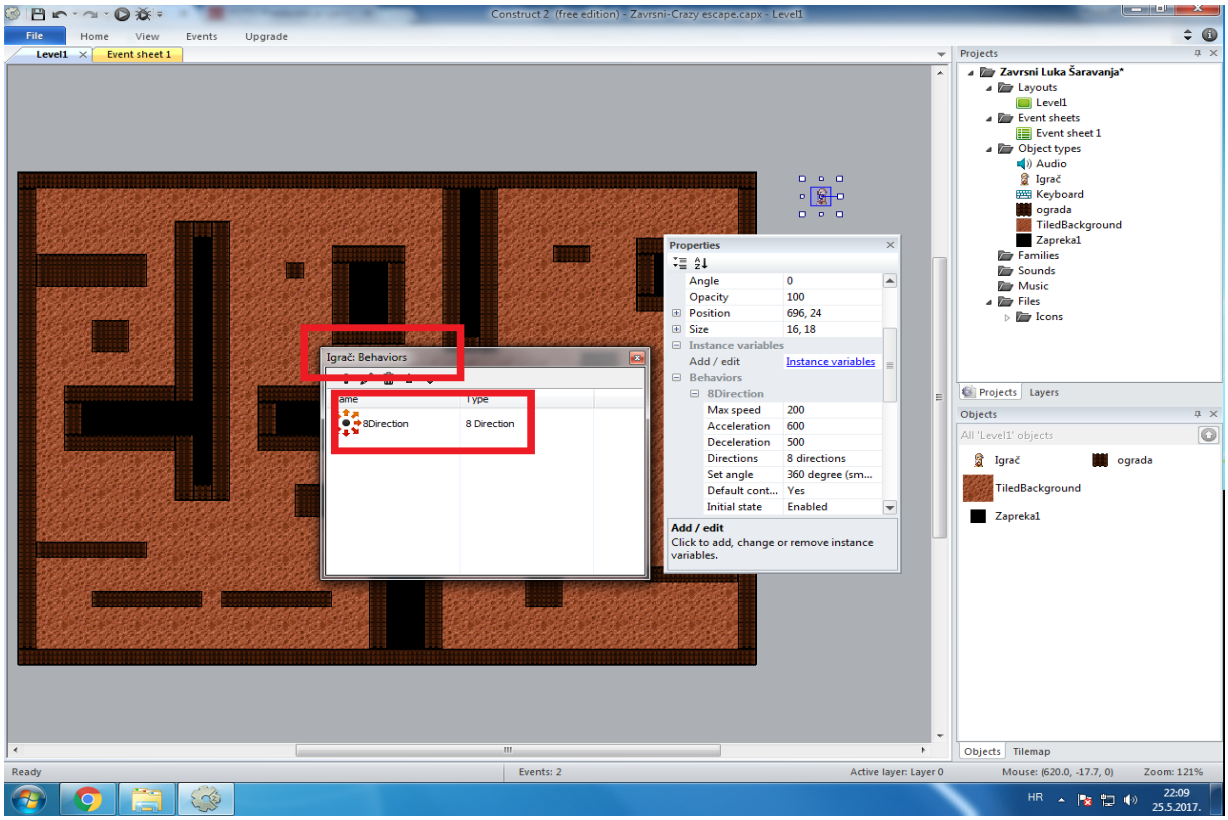


Slika 5.2.2-stvaranje igrača na slučajnoj poziciji na početku levela

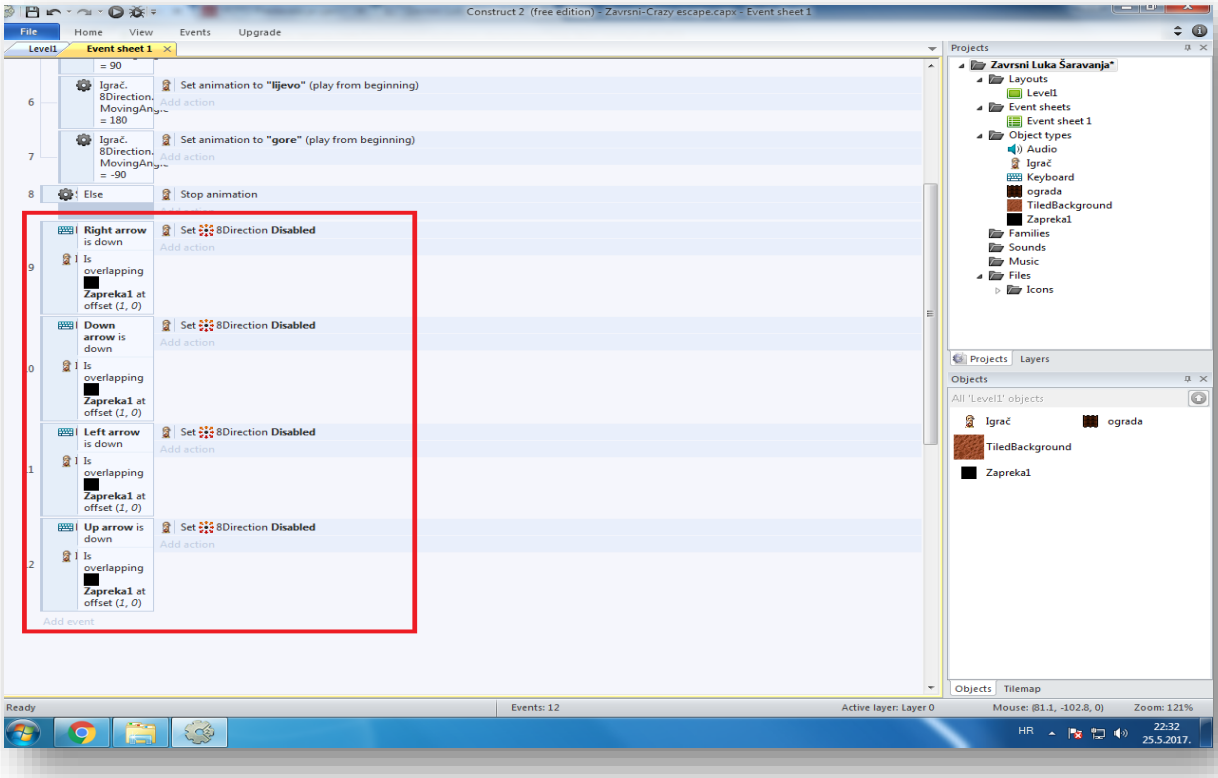


Slika 5.2.3

Nakon što se igrač stvori u levelu potrebno mu je dodati opciju da se može kretati. Postoji već ugrađena opcija dodavanja određenih ponašanja (engl. *behaviors*) pojedinim objektima. Jedno od osnovnih ponašanja je kretanje igrača u 8 smjerova (Slika 5.2.4). U ovom projektu je odlučeno da će se igrač moći kretati samo u 4 osnovna smjera (naprijed,nazad,lijevo i desno). Svakom od četiri smjera kretanja pridružena je prethodno umetnuta animacija za kretanje u tome smjeru Moralo se voditi brigu o tome da se igrač ne može kretati po zaprekama ili zidovima. To je napravljeno na način da igraču u dodiru sa zaprekom ili zidom kretanje u 8 smjerova (8 direction) onemogući (slika 5.2.5).



Slika 5.2.4

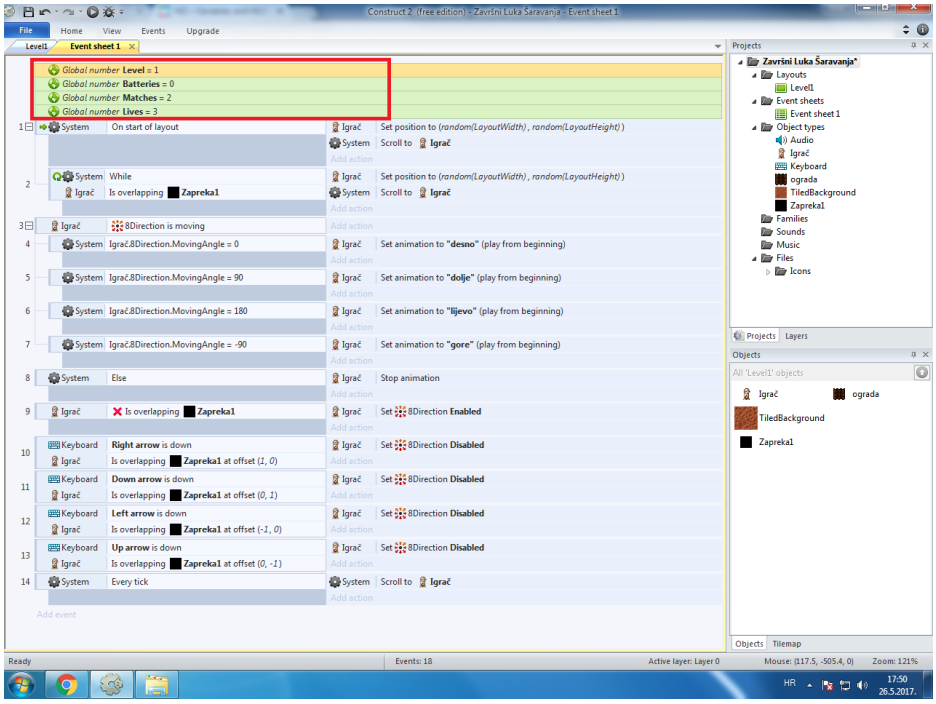


Slika 5.2.5

Zadnja stvar koja se dodala projektu je da kamera prati igrača. To znači da se na početku levela igrač stvori na slučajnoj poziciji koja nije zapreka ili zid te se kamera fokusira na njega i prati ga kroz njegovo kretanje. Opcija *scroll to object* omogućuje fokusiranje na određeni objekt, u ovome slučaju to je igrač.

5.3 Varijable i HUD

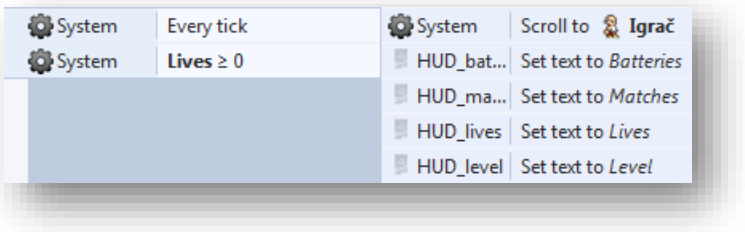
Riječ «varijabla» dolazi iz engleske riječi «variable» što znači «promjenjiva vrijednost». To može biti broj, riječ ili nešto drugo[5]. U ovome projektu postoje dijelovi koji se prate i mijenjaju u vremenu. To su broj levela, baterija, šibica ili života koji igrač posjeduje .Postavljene su kao globalne varijable (slika 5.31).



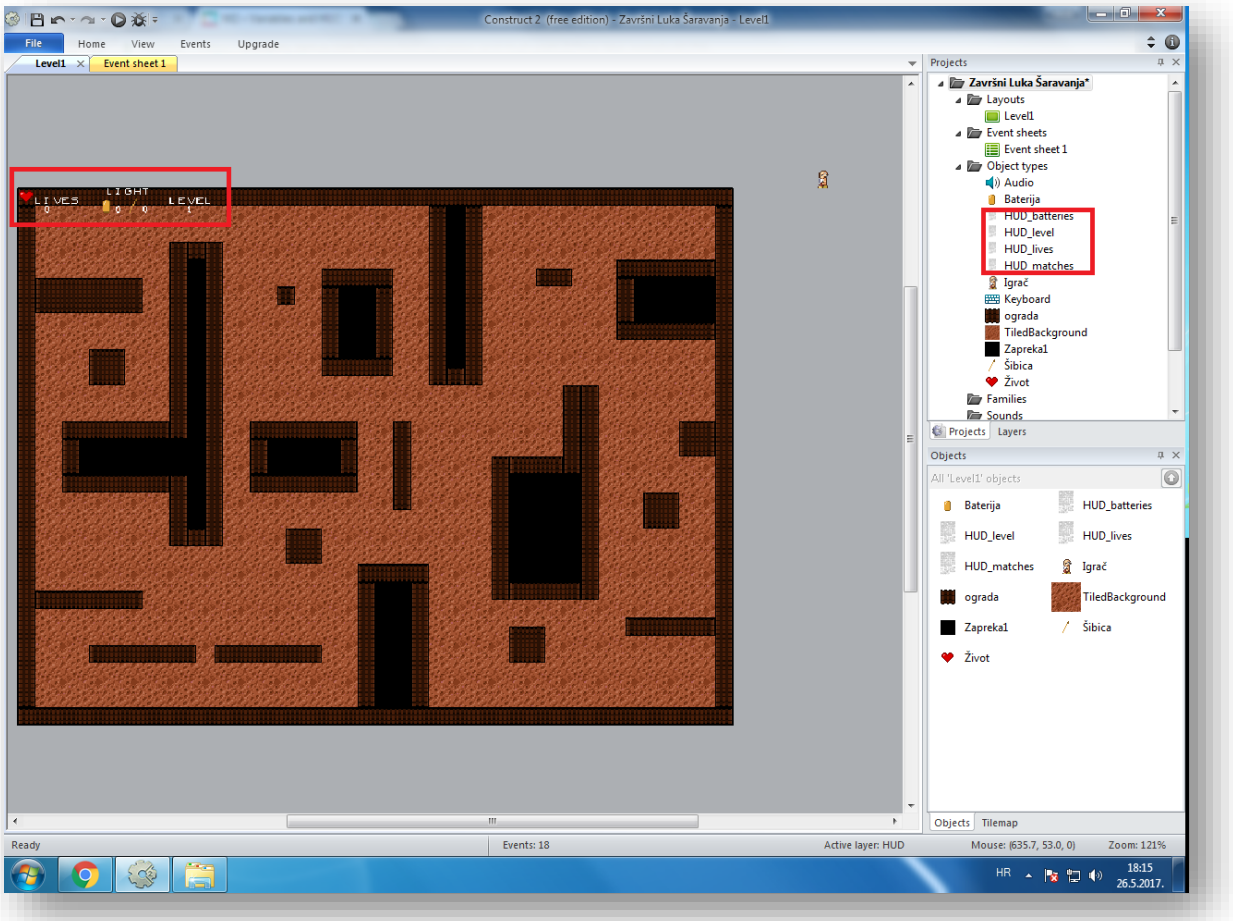
Slika 5.3.1-globalne varijable

HUD (*heads up display*) predstavlja pogled na igračev napredak, stvari koje posjeduje i levele.Većinom je napravljen od tekstualno-brojevnog objekta povezanog sa nekom varijablom. U ovome projektu HUD je upotrebljen kako bi igrač mogao pratiti koliko ima baterija, šibica ili kako bi mogao vidjeti na kojem je levelu (slika 6.3).

Za HUD je potrebno napraviti poseban sloj, kako se ne bi pomjerao ovisno o levelu nego da bi bio stalno na istoj poziciji. Prethodno je odlučeno kakav font treba biti za HUD te na kojoj poziciji će biti prikazan. HUD se povezuje sa globalnim varijablama kako bi se ovisno o njihovim promjenama i sam mijenjao (slika 6.2).



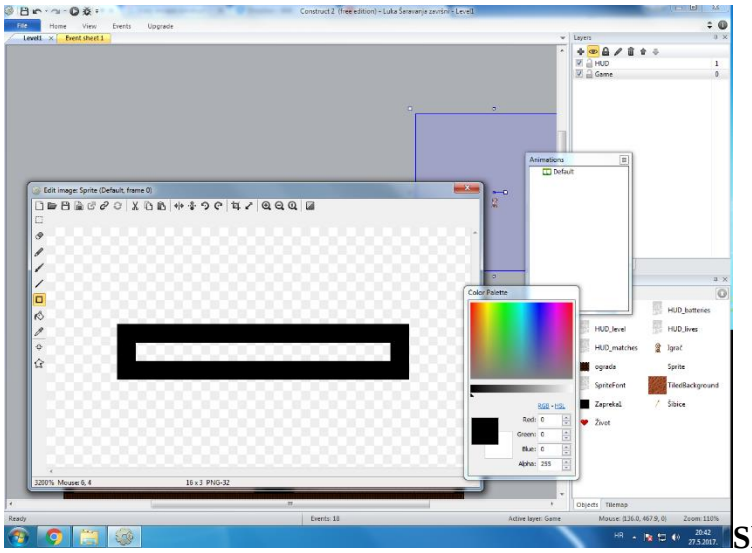
Slika 5.3.2



Slika 5.3.-HUD

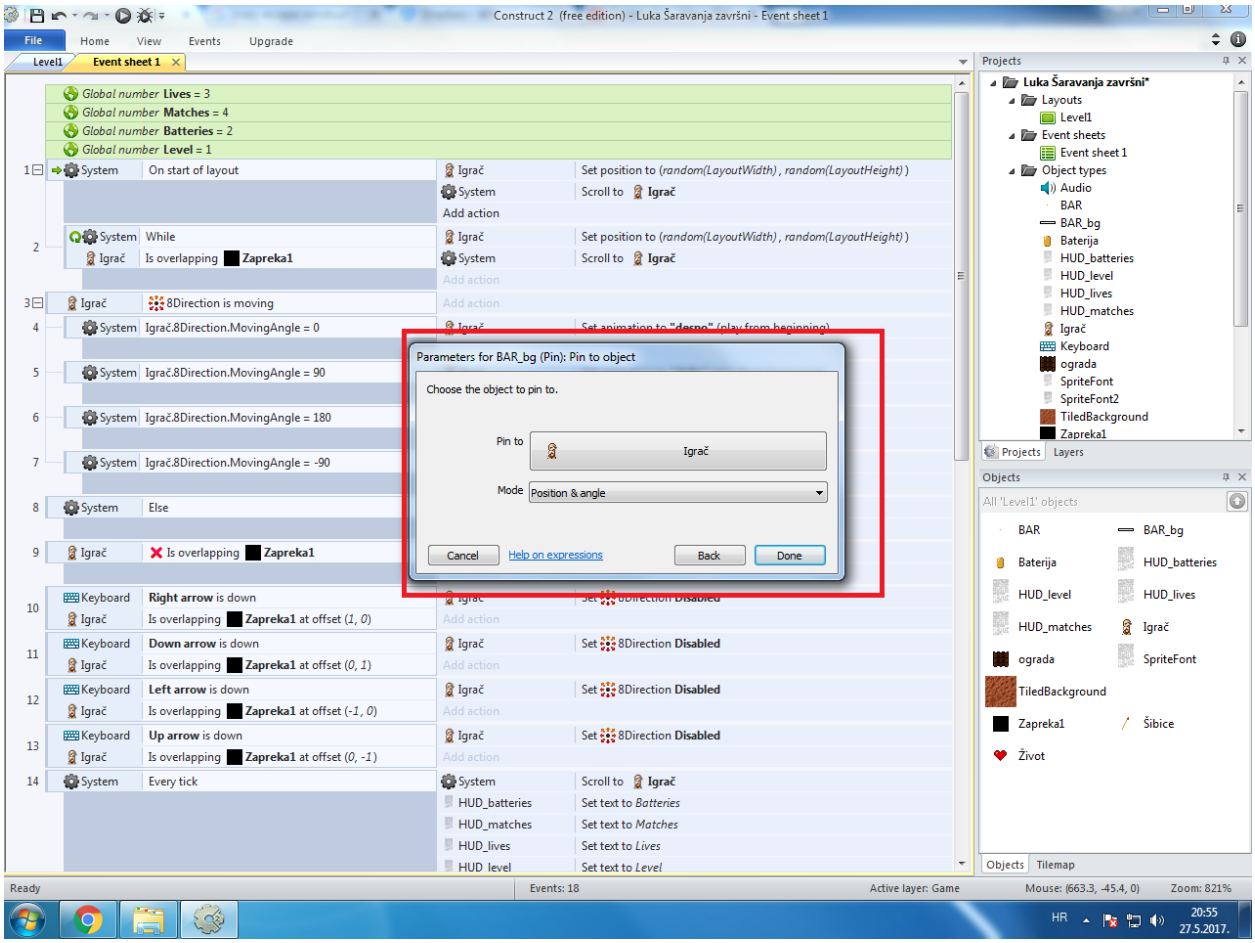
5.4 Traka napretka

Kako bi igrač tijekom igre znao koliko mu se troši baterija ili šibica bilo je potrebno dodati traku napretka(engl. *progress bar)* koja će pratiti potrošnju baterije. Zamišljeno je da traka bude iznad igrača u igri te da ga prati. Nacrtan je izgled trake napretka u uređivaču slika (slika 6.4).



Slika 5.4.1

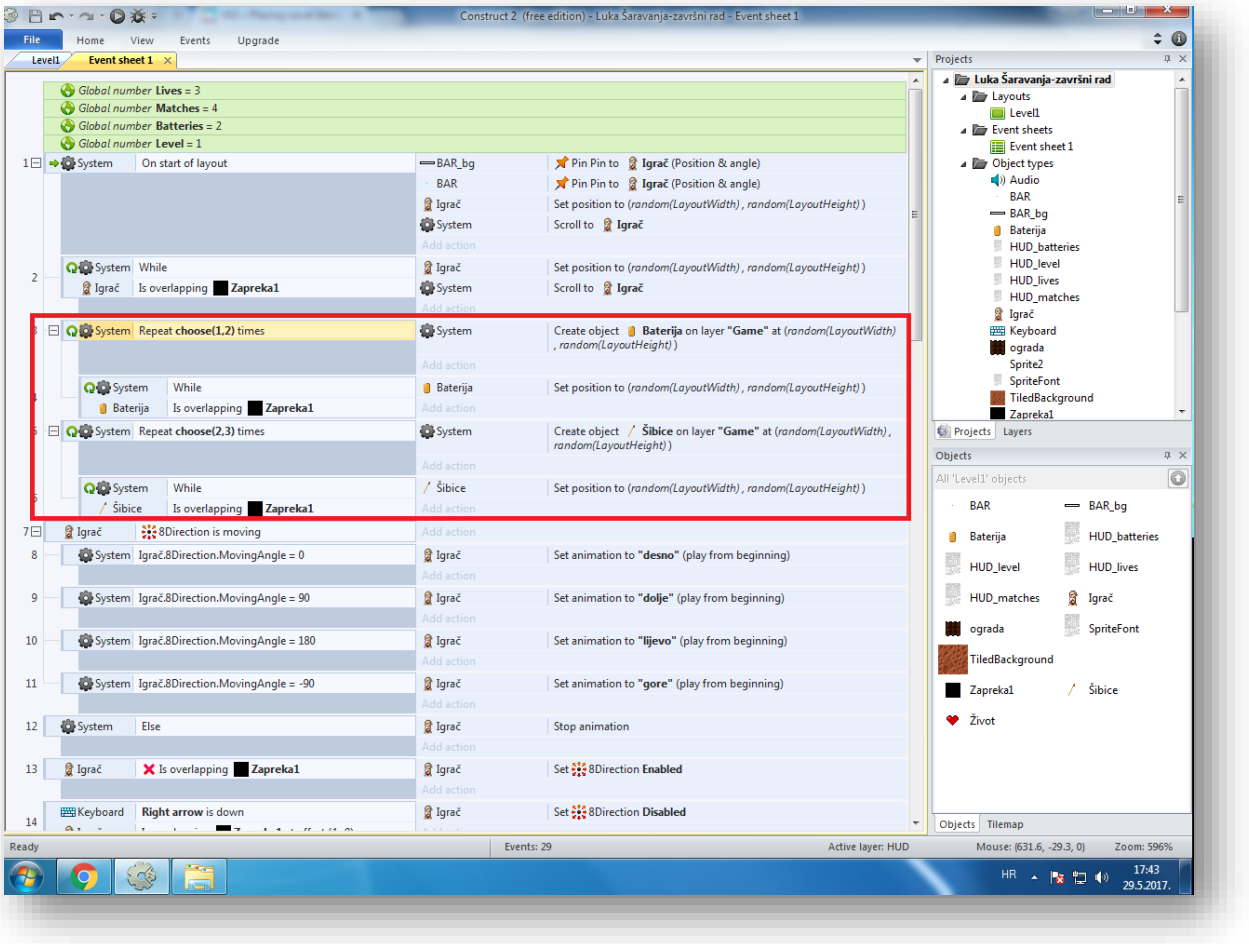
Moralo se paziti da traka bude iznad igrača i da ga prati sve dok hoda. Igrača će traka napretka pratiti pomoću opcije *pin to object* (slika 6.5). Postavljeno je da se zelena površina počne kretati od lijevo prema desno tako da je postavljen origin position odakle da kreće. Bar će se širiti određenom brzinom za baterije i određenom za šibice. Morala se voditi briga o tome ako su baterije<0 da se počinje korisiti šibica i da se drugačije kreće traka napretka. Kako traka pokazuje koliko se troši baterija ili šibica povezani su traka napretka i globalne varijable kako bi se smanjivale ovisno o potrošnji.



Slika 5.4.2

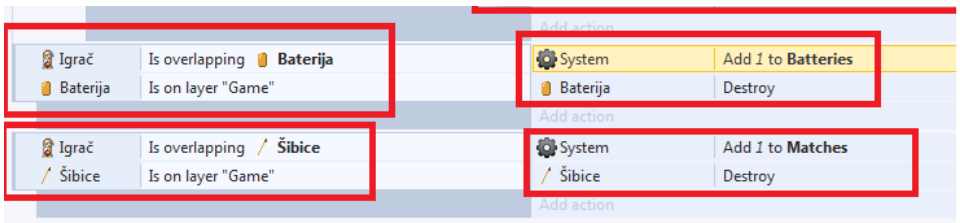
5.5 Stvaranje i prikupljanje stvari

U projektu postoje različite stvari(engl. *item*) koje igrač može skupiti (šibice,baterije,ključ) i potrebno ih je stvoriti na početku levela. Construct 2 događaji dopuštaju da se neka akcija događa svakih nekoliko sekundi. Jedna od stvari na koju je trebalo paziti je da stvoreni objekti ne budu na zaprekama ili zidovima kako bi ih igrač mogao pokupiti. Bitno je odrediti da se ti objekti stvaraju na sloju igre (slika 6.6).



Slika 5.5.1

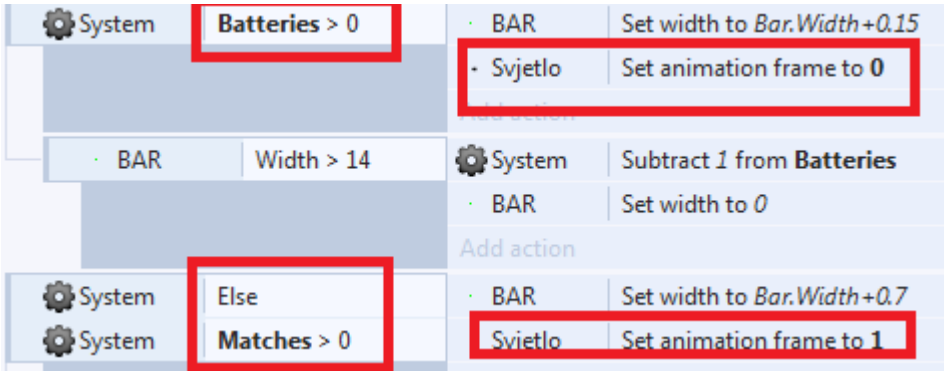
Nakon što su stvari stvorene bitno ih je postaviti tako da ih igrač može skupiti. To je odrađeno pomoću is overlapping. Kada igrač prijeđe preko baterije,šibice ili ključa to će se dodati globalnim varijablama a ujedno će ta instanca nestati sa levela (**s**lika 5.5.2).



Slika 5.5.2

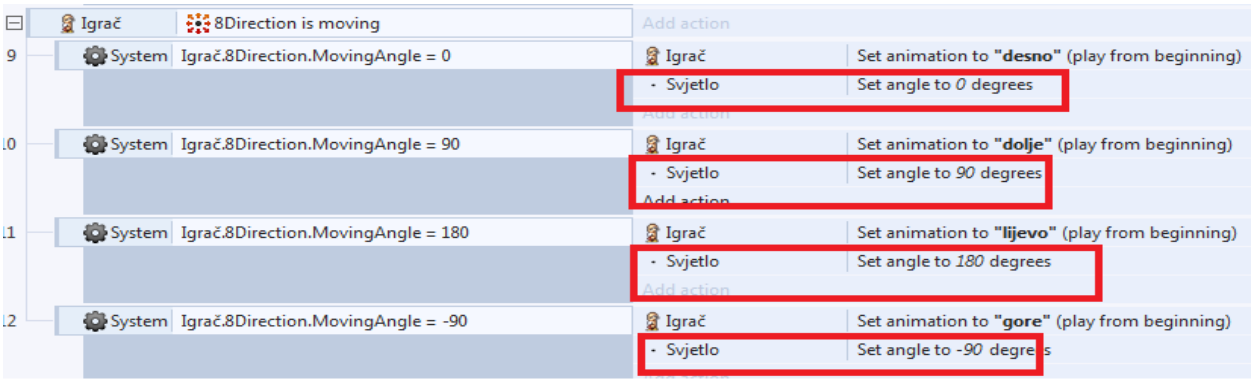
5.6 Svjetlo

Igrač ima ograničen vidokrug koji mu otežava prolazak kroz igru. Prethodno je u programu za obradu slika nacrtano kako izgleda vidokrug za baterije, a kako za šibice. Dodana je animacija sa tri *frame-a* gdje svaki od *frameova* predstavlja jedan vidokrug. Ovisno o tome što se koristi animacija će pratiti igrača(Sl. 5.6.1).



Slika 5.6.1

Kako bi animacija za svjetlo pratila igrača u njegovim pokretima dodani su novi uvjeti u *8 direction* (Sl. 5.6.2).

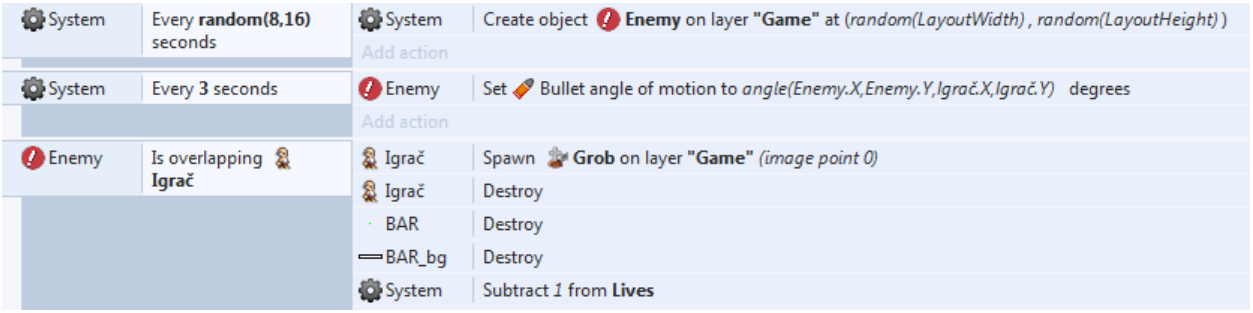


Slika 5.6.2

Jedna od prepreka je što se ostali objekti ne vide kada igrač ima ograničen vidokrug. To je rješeno na način da se tama doda na sloj tek nakon što su objekti šibica, baterija i ključa već dodani na sloj.

5.7 Neprijatelji(engl. *enemies*)

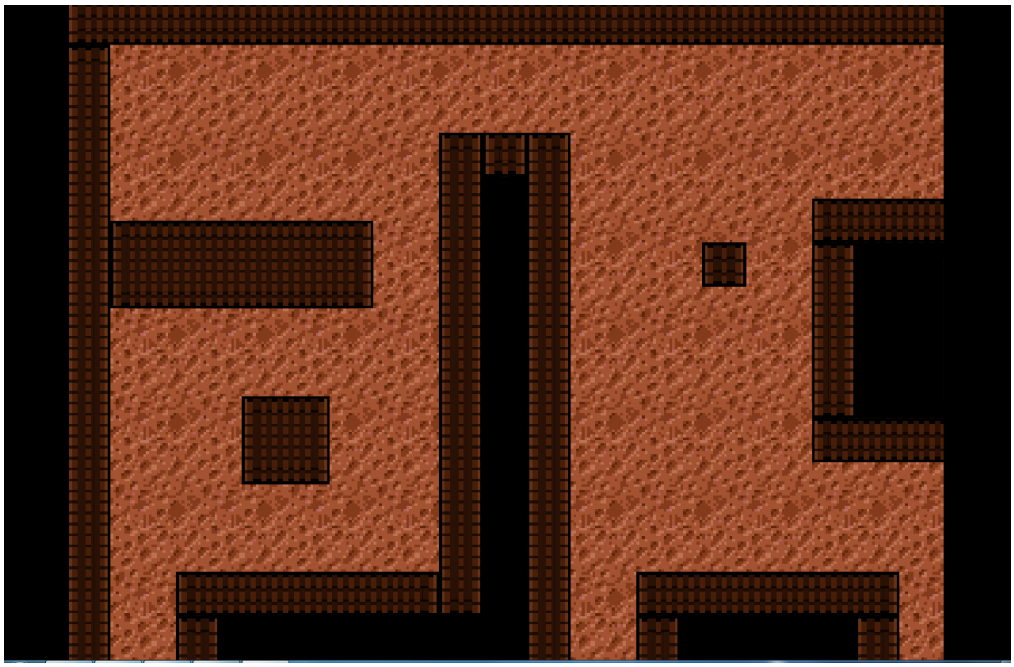
Osim smanjenog vidokruga tu je još jedna prepreka koja igraču otežava igru. To su neprijatelji . U igru su dodane, prethodno napravljene, grafike za njih. Stvaraju se svakih nekoliko sekundi kako ne bi od početka igre otežavali igraču. Jedan od problema je kako usmjeriti neprijatelje prema igraču. To je učinjeno pomoću *bullet behaviora* (omogućuje pojedinim objektima kretanje prema drugim objektima). Bitno je postaviti dobre kuteve prilaska neprijatelja prema igraču kako bi sve izgledalo što prirodnije. Kako bi igra bila zanimljiva igračima, umetnuto je stvaranje objekta koji podsjeća na grob na mjestu igrača ukoliko neprijatelji oduzmu sve živote igraču.



Slika 5.7.1-stvaranje i kretanje neprijatelja te „groba“

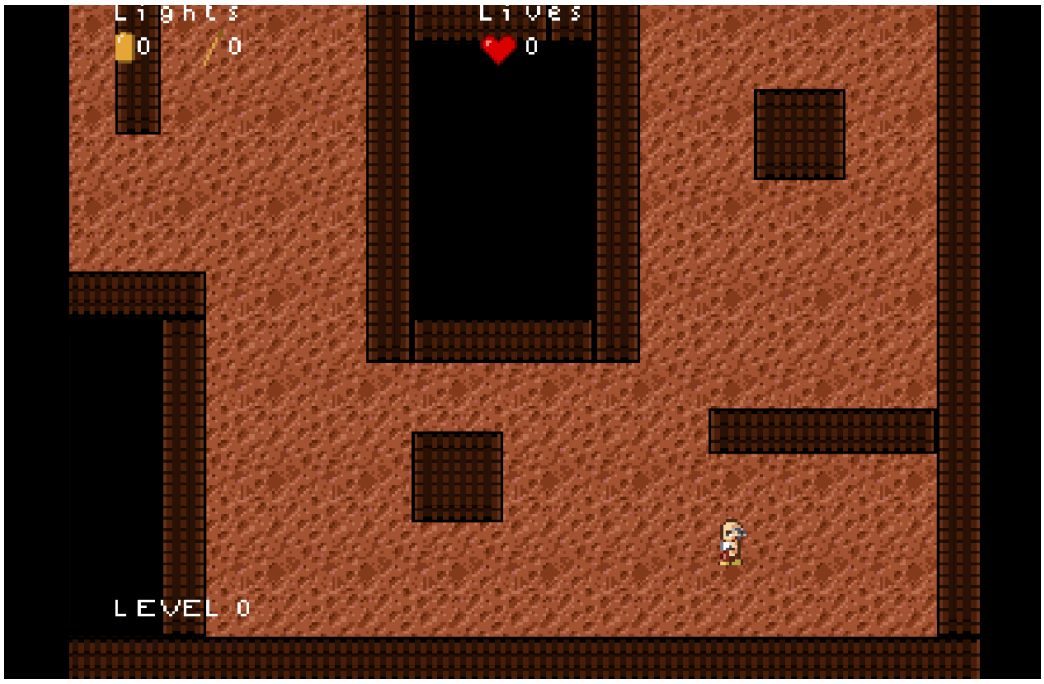
5.8 Verzije igre

Prva verzija igre je nastala nakon dodavanja zapreka i zidova:



Slika 5.8.1

Druga verzija igre je nastala nakon dodavanja igrača te HUD-ova :



Slika 5.8.2-stvaranje i kretanje neprijatelja te „groba“

Treća i finalna verzija je nastala nakon dodavanja progress bara i smanjenog vidokruga:

Slika 5.8.3-stvaranje i kretanje neprijatelja te „groba“

6. Zaključak

2D igre su najraširenije web igre. Kod njihove izrade potrebno je smisliti nešto što će igračima biti zanimljivo. Najteži dio posla je ideja igre. Kada je igra smišljena treba sve prebaciti u jedan Construct 2 projekt . Construct 2 arhitektura vrlo temeljito podijeli projekt na više dijelova kako bi korisnicima olakšala njeno korištenje. Igra je na kraju točno onakva kakva je bila i zamišljena jer pomoću Construct 2 arhitekture moguće je logiku same igre vrlo detaljno razraditi. U poglavljima 3 i 4 detaljno je opisan način na koji nastaje projekt u Construct 2 arhitekturi. U 5 poglavlju prolazi se kroz izradu same igre gdje se točno može vidjeti na koje sve načine Construct 2 arhitektura rješava određene probleme.

.

7. Literatura

[1] https://www.scirra.com/manual/1/construct -2

[2] HTML5-https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5

[3] https://www.codecademy.com/learn/javascript

[4]https://developer.mozilla.org/en-US/docs/Web/HTML/Using_the_application_cache

[5]http://www.tutorijali.net/teorija-programiranja/varijable

8. Sažetak

Zadatak završnog rada izrada je projekta u Construct 2 arhitekturi. Prilikom izrade projekta najteži dio je kako smisliti projekt koji bi mogao biti zanimljiv igračima. Construct 2 arhitektura uvelike je olakšala izradu zamišljenog projekta jer je vrlo jednostavna za korištenje. Bez ikakvog prethodnog znanja ili prakse u izradi 2D igara na kraju je uspješno projekt proveden u djelo. Construct 2 arhitektura služi kako bi naučila i uvela ljude u svijet programiranja na jednostavan i zabavan način.

Ključne riječi:2d igre,Construct 2 arhitektura,projekt,programiranje

Key words:2D games,Construct 2 architecture,project,programming

Summary:

Application of Construct 2 architecture in computer game design

Task of this final work is production of project in Construct 2 architecture. When designing the project the hardest part was to inevnt project that could be interesting to other players. Construct 2 architecture has greatly facilitate the design of the project because it is very easy to use. Without any previous knowledge or practice in creating 2D games in the end project was sucessfully carried out .Construct 2 architecture is here to help and teach people to enter world of programming in very easy and fun way.

9. Životopis

Luka Šaravanja rođen je 21.studenoga 1994 u Osijeku. Nakon završene Osnovne škole Svete Ane u Osijeku upisao je Matematičko-prirodoslovnu gimnaziju u Osijeku. Nakon završetka srednje škole upisuje se na preddiplomski studij Elektrotehnike na Fakultetu elektrotehnike, računalstva i informacijskih tehnologija u Osijeku. Za svoje područje izobrazbe odabrao je telekomunikacije.