Dizajn igre za učenje koncepata programiranja u razvojnom okruženju Unreal engine 4

Bileta, Toni

Master's thesis / Diplomski rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:137:567534

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Image not found or type unknown *Repository / Repozitorij:*

Digital Repository Juraj Dobrila University of Pula

zir.nsk.hr Image not found or type unknown

Sveučilište Jurja Dobrila u Puli
Fakultet informatike u Puli

Toni Bileta

Dizajn igre za učenje koncepata programiranja u razvojnom okruženju Unreal Engine 4

Diplomski rad

Pula, svibanj, 2019.

Sveučilište Jurja Dobrila u Puli
Fakultet informatike u Puli

Toni Bileta

Dizajn igre za učenje koncepata programiranja u razvojnom okruženju Unreal Engine 4

Diplomski rad

JMBAG: 0303046400, redoviti student
Studijski smjer: Informatika

Predmet: Dizajn i programiranje računalnih igara

Znanstveno područje: Područje tehničkih znanosti

Znanstveno polje: Računarstvo

Znanstvena grana: Programsko inženjerstvo

Mentor: doc. dr. sc. Tihomir Orehovački

Pula, svibanj, 2019.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani Toni Bileta, kandidat za magistra informatike, ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

_____

U Puli, _____, _____ godine

IZJAVA
o korištenju autorskog djela

Ja, Toni Bileta dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom Dizajn igre za učenje koncepata programiranja u razvojnom okruženju Unreal Engine 4 koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.
Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____ (datum)

Potpis

_____

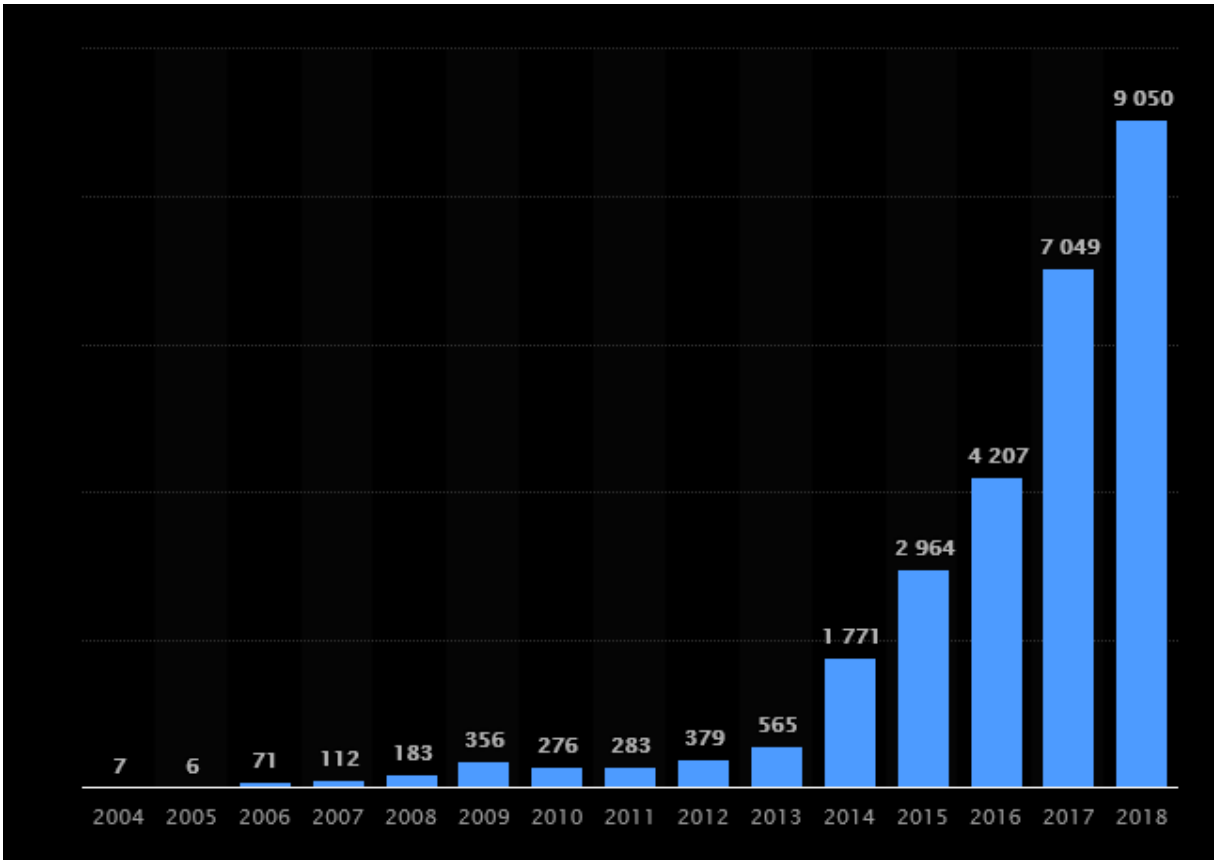
SADRŽAJ

UVOD 1
1. Analiza tržišta 4
1.1 Ciljevi i vizija 4
1.2 Konkurentska rješenja 4
1.2.1 Portal 5
1.2.2 The Talos Principle 6
1.2.3 Glitchspace 7
1.2.4 else Heart.Break() 8
1.2.5 Code Combat 9
1.2.6 Code Hunt 10
1.3 Ciljano tržište 11
1.4 Promocija i izbacivanje na tržište 15
1.5 SWOT analiza 19
1.5.1 Snage 19
1.5.2 Slabosti 20
1.5.3 Prilike 20
1.5.4 Prijetnje 20
2.Korištenje igara za podučavanje i njihov utjecaj 22
2.1 Igre u obrazovnom sustavu 22
2.2 Utjecaj igara na ljudski mozak 24
3.Razvoj igre 27
3.1 Korištene tehnologije 28
3.2 Pred-produkcija i dizajn dokument 30
3.3 Igrin svijet i priča 32
3.4 Dizajn osnovnih igrinih sustava 34
3.5 Dizajn zagonetki i proces učenja 38
3.6 Dizajn razina 42
3.7 Dizajn korisničkog sučelja 49
4.Korisnička dokumentacija 56
4.1 Pokretanje igre 56
4.1.1 Project.Start() i Project.Continue() 57
4.1.2 Project.LoadLevel() 57
4.1.3 Options.Change() 58
4.1.4 Credits.Roll() i Project.Exit() 59

4.2 Igranje igre 59
4.2.1 Interakcija sa objektima 61
4.2.2 Korištenje konzole 65
5.Testiranje i povratne informacije 70
5.1 Interno testiranje 70
5.2 Zatvoreno testiranje 70
5.3 Otvoreno testiranje 71
5.3.1 Rezultati ankete 71
Zaključak 72
Literatura 73
Popis slika 75
Sažetak 77
Ključne riječi 77
Summary 77
Key words 78

UVOD

Industrija video igara (engl. gaming industry) jedna je od najbrže rastućih industrija, svake godine izlazi sve veći broj igara, te se stalno povećava broj igrača različitih demografskih obilježja. Web stranica Statista najbolje pokazuje taj eksponencionalni rast industrije videoigara uz pomoć grafa koji pokazuje kolika količina igara je objavljena na najvećoj platformi za digitalnu distribuciju igara, Steam. Taj graf je prikazan na sljedećoj slici (slika 1).



Slika 1 – Broj igara objavljenih na Steam platformi kroz godine

Izvor: Statista (2019).

Prema ovome grafu jasno je vidljiv navedeni veliki rast broja objavljenih igara na Steam platformi kroz godine. Primjerice 2004. godine kada je lansirana Steam platforma na njoj je bilo objavljeno samo sedam videoigara. Kroz godine taj je broj rastao te je u prošloj 2018. godini na Steam platformi bilo objavljeno rekordnih 9050 video igara. Ovaj podatak pokazuje da videoigre postaju sve pristupačnije za izradu i za igranje, što će biti prikazano u kasnijem dijelu rada. Također je potrebno

spomenuti statistički podatak koji nam pokazuje koliki broj igrača koristi Steam platformu što je prikazano sljedećom slikom (slika 2)

	APPID	NAME	CURRENT	24H PEAK	ALL-TIME PEAK
+	753	Steam	14,016,517	14,855,335	18,537,490

Slika 2 – Broj istovremenih korisnika na Steam platformi
Izvor: SteamDB (2019).

Na ovoj slici je prikazan broj jedinstvenih korisnika koji su istom trenutku koristili Steam platformu. Slika je napravljena 4. travnja 2019. godine i u tom trenutku je najveći broj ljudi koji su u jednom trenutku u vremenu koristili Steam platformu bio preko 18 milijuna. Treba također napomenuti da Steam predstavlja samo jedan, iako dosta veliki, dio industrije i odnosi se samo na dio industrije vezan za igranje video igara na osobnim računalima.

Još jedan pokazatelj stalnog rasta industrije videoigara je vidljiv i u njezinoj zaradi. U 2018. godini industrija videoigara imala je zaradu od 43.8 bilijuna dolara što je povećanje od 18% u odnosu na prijašnju godinu (Shieber 2019). Primjerice za usporedbu, prošle godine filmska industrija je zaradila 41.7 bilijuna dolara dok su servisi za prenošenje (engl. streaming services) serija i filmova zabilježili 28.2 bilijuna dolara. Prema ovim podacima to stavlja industriju videoigara na sam vrh zabavne industrije (engl. entertainment industry) i čini je relevantnijom nego ikada do sada.

Sa svim ovim informacijama na umu nameće se pitanje da li se videoigre može koristiti za nešto više osim samo za zabavu i ovdje dolazimo do tematike ovoga diplomskoga rada, izrada igre korištenjem razvojnog okruženja Unreal Engine 4 koja bi korisnika učila osnovnim konceptima programiranja . Programiranje je vještina koja je uvijek tražena na tržištu rada, ali sam proces učenja programiranja nerijetko zna biti dosadna i ponekad frustrirajuća aktivnost, osim za osobe koje imaju veliku strast za programiranjem. Ideja iza ovog dipomskog rada bila je iskoristiti popularnost i pristupačnost videoigara kako bi se igrače naučilo osnovnim konceptima programiranja na interaktivan i potencijalno zabavan način. Iz te ideje proizašla je videoigra pod nazivom CodeBreaker. Kroz ovaj rad biti će objašnjena metodologija izrada videoigre koja je dizajnirana za učenje osnovnih koncepata programiranja. Biti će pojašnjeno na koji način treba pristupiti dizajniranju takve igre i koji problemi mogu

nastati prilikom njene izrade. Osim toga biti će prikazani procese izrade i dizajniranja raznih igrinih sustava i dijelova. Unutar ovoga rada također će se pričati o industriji videoigara i njenom utjecaju te demografskim skupinama igrača i njihovim značajkama. Ovaj rad sastoji se od pet glavnih poglavlja i njihove će tematike biti opisane u nastavku.

Prvo poglavlje ovoga rada baviti će se ustanovljavanjem vizije i ciljeva koji su bili iza ove igre, analizom tržišta i konkurentskih rješenja, biti će objašnjene strategije izbacivanja igre na tržište, te će naposljetku biti biti provedena SWOT analiza i detaljnije pojašnjeni svaki od njenih parametara.

Drugo poglavlje baviti će se tematikom korištenja videoigara za svrhe podučavanja. Unutar ovoga poglavlja biti će predstavljeni primjeri iz prakse za korištenje igara u edukacijskom sustavu i navedeni zaključci i rezultati koji su proizašli iz toga. Drugi dio ovoga poglavlja baviti će se istraživanjima koja su se bavila proučavanjem utjecaja videoigara na ljudski mozak.

Tematika trećega poglavlja je dizajn i izrada videoigre CodeBreaker. Unutar ovoga poglavlja biti će detaljno prikazan razvojni ciklus ove igre i pojašnjene razne dizajn filozofije. Tako će biti pojašnjeni dizajnovi i izrada raznih igrinih dijelova poput glavnih sustava, korisničkog sučelja, igrinih razina i njihovih pripadajućih zagonetki.

Četvrto poglavlje sadržava korisničku dokumentaciju. Ovdje će igra biti opisana sa standardnim korisnikom na umu, dakle uz korištenje slika i što jednostavnijeg rječnika. Opisani će biti igrini osnovni sustavi i prikazan način na koji će igrač vršiti interakciju s njima i preko njih sa samom igrom. Također će biti prikazani pojedini dijelovi korisničkog sučelja i objašnjene njihove funkcionalnosti i načini korištenja.

Peto i završno poglavlje baviti će se načinom na koji je igra testirana i rezultatima tih testiranja. Biti će navedene metode koje su metode korištene za nadziranje testiranja i prikupljanje potrebnih podataka. Naposljetku će dobiveni podatci biti analizirani i na temelju tih podataka biti će izvučeni zaključci.

1. Analiza tržišta

Unutar ovoga poglavlja pričati će se općenito o igri, koje je njeno ciljano tržište, konkurentskim igrama i inspiracijama, te po čemu su te igre specifične. Naposljetku će biti provedena SWOT analiza uz pomoć koje će se ustanoviti na koji način se ova igra uspoređuje sa konkurentskim rješenjima.

1.1 Ciljevi i vizija

Osnovna vizija koja je postojala od samoga početka bila je da se izradi igra kroz čije igranje se igrača može naučiti osnovne koncepte programiranja. Kroz tu viziju jasno je vidljivo kako je cilj bio da krajnji rezultat bude izrada zanimljivog okruženja unutar kojega igrač može na zabavan način unaprijediti svoje znanje iz programiranja. Tijekom pred-produkcijskog razdoblja mnogo vremena bilo je posvećeno razmišljanju i pokušajima da se ustanovi na koji način da se navedena vizija sprovede u djelo. Prva stvar koja je bila bitna je odabir žanra za ovu igru, unutar toga procesa su postojale razne ideje koje su varirale od akcijske igre iz trećeg lica pa sve do 2D platformera u kojemu se hakira okoliš. Jedna stvar koju je bilo potrebno imati na umu prilikom odlučivanja o tipu igre koju je bilo potrebno napraviti bilo je vremensko ograničenje, samim time bilo je potrebno odlučiti se za tip igre koji je moguće u potpunosti dovršiti unutar nekoliko mjeseci, što se naposljetku ipak odužilo za još nekoliko dodatnih mjeseci. Sa svime time na umu na kraju je odlučeno kako će igra, tada zvana Operation Code, biti puzzle igra koja će se igrati iz prvog lica. Sljedeći korak koji je nakon te odluke trebalo poduzeti je istražiti tržište video igara, te pronaći slične tipove igara te ih analizirati i otkriti što one rade dobro, to jest koje su njihove jedinstvene karakteristike, i na koji način bi se naša igra mogla isticatiu odnosu na njih. Iduće poglavlje baviti će se upravo tom analizom konkurentskih rješenja i inspiracija.

1.2 Konkurentska rješenja

Sada kada je bio poznat tip igre koja će biti rađena, bilo je moguće istražiti tržište video igara i pronaći druge slične igre toga tipa. Fokus je bio na dva glavna tipa igara, a to su puzzle igre i igre koje su dizajnirane za učenje programiranja. S

obzirom da izrađivana igra objedinjuje oba žanra bilo je potrebno otkriti na kakav je način moguće napraviti zanimljivu puzzle igru koja može uspješno igrača podučiti o konceptima programiranja neovisno o njegovom prijašnjem znanju vezanom za tu tematiku. Potrebno je napomenuti da je fokus bio specifično na igrama za osobna računala, a ne na mobilne igre koje imaju najveću količinu puzzle igara jako varirajućih kvaliteta. Na kraju istraživanja izdvojene su bile sljedeće igre jer je od njih bilo moguće nešto naučiti o izradi ovakvoga tipa igara:

- Portal
- Talos Principle
- The Turing Test
- else Heart.Break()
- Glitchspace
- The Witness
- While True Learn
- CodinGame
- CodeCombat
- Code Hunt

Od ovih navedenih igara Codin Game, Code Combat i Code Hunt predstavljaju igre koje su dizajnirane za učenje programiranje dok sve ostale spadaju pod puzzle igre. U nastavku će neke od tih igara biti dodatno predstavljene i analizirane.

1.2.1 Portal

Portal je videoigra koja se igra iz prvog lica i njena cijela premisa je rješavanje niza uzastopnih zagonetki koje postepeno postaju sve teže i kompleksnije. Mehanika koja čini ovu igru jako jedinstvenom je mogućnost korištenja portala kako bi se uspješno riješilo zagonetke, ti portali su prikazani na sljedećoj slici (slika 3).



Slika 3 – Igra Portal

Izvor: Screenrant (2016).

Cilj svake od razina je uspješno riješiti dane zagonetke korištenjem navedenih portala kako bi se uspješno izašlo iz te prostorije. Svaka prostorija ima jasno vidljiv izlaz i na igraču je da otkrije kako da dođe do tog izlaza i uspješno ga otvori. Mehanika korištenja portala omogućava igraču da dođe do jako zanimljivih rješenja i nakon što uspješno riješi zagonetku, da se osjeća jako pametno. Portal je inspirirao mnogobrojne puzzle igre, pa tako i našu. U početku igrinog razvoja je cilj bio da se u njoj nalaze slične zagonetke koje bi se zasnivale na više interakcije sa okolinom, ali je s vremenom ipak stavljen puno veći fokus na samu aktivnost programiranja, pa je zbog toga stavljen puno manji fokus na klasični tip zagonetki. Bez obzira na to utjecaj Portala je još uvijek vidljiva u određenim segmentima igre. Zbog svoje jedinstvenosti Portal je stekao kultni status među igračima i povećao interes igrača za igrama ovoga tipa. Portal i njegov nastavak su se zahvaljujući tome prodali u preko 20 milijuna primjeraka (VGChartz 2019a)

1.2.2 The Talos Principle

The Talos Principle je igra napravljena od strane Hrvatskog studija Croteam. U ovoj igri igrač preuzima ulogu tek „rođenog“ androida koji se nalazi u prostoru inspiriranom antičkom Grčkom. Slično kao i u Portalu igrač mora rješavati zagonetke kako bi napredovao. Ono po čemu se ova igra razlikuje od Portala je činjenica da

igrač u jednom trenutku ima izbor između nekoliko različitih zagonetki koje može rješavati proizvoljnim redoslijedom. Nakon što igrač uspješno riješi sve zagonetke u tom prostoru otvara mu se put dalje do idućeg seta zagonetki koje se nalaze u drugačijem tematskom razdoblju, primjerice Egipatskom i srednjovjekovnom. Način na koji igrač mora rješavati zagonetke je korištenjem raznih objekata poput kutija i lasera (slika 4).

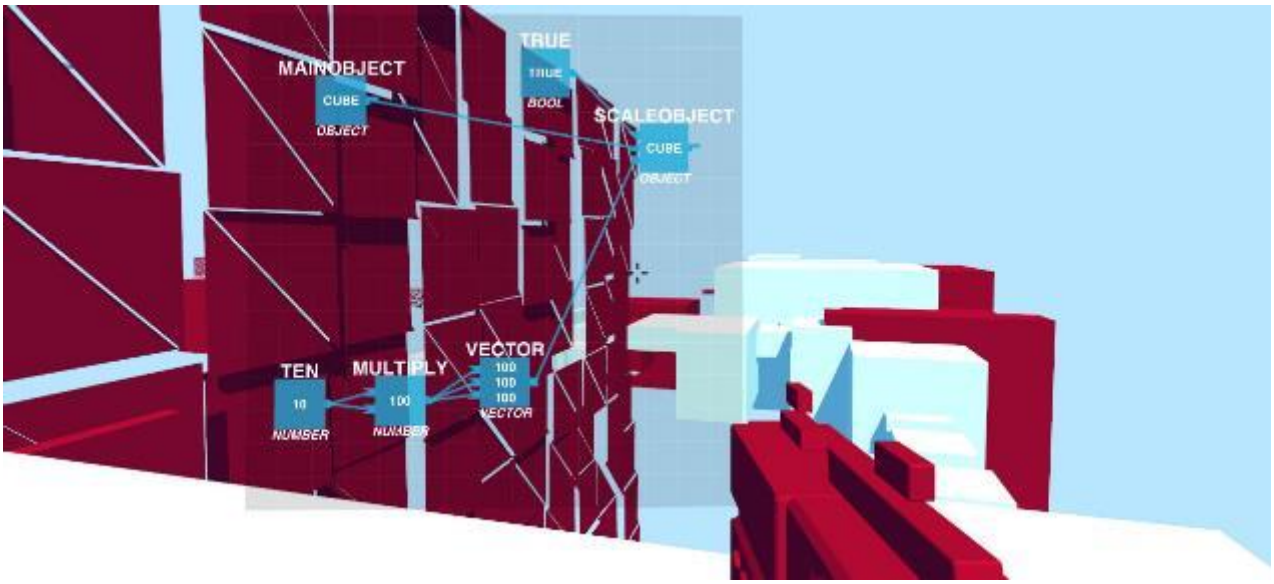


Slika 4 – Primjer zagonetke u igri The Talos Principle
Izvor: Game Reactor (2016).

Jedna od dodatnih stvari po kojoj se ova igra ističe je zbog toga što tijekom se tijekom igranja nameču razne filozofske teme vezane za smisao života i što to doista znači biti živ i imati slobodnu volju. Kombinacija svih tih elemenata omogućila je ovoj igri da se proda u preko milijun primjeraka (VGChartz 2019b).

1.2.3 Glitchspace

Glitchspace je još jedna igra koja se igra iz prvoga lica gdje je cilj rješavanje zagonetki kako bi se napredovalo kroz igrine razine. Razlika između ove igre i onih prije navedenih je taj da se u ovoj igri koristi vizualno programiranje kako bi se rješavale zagonetke (slika 5).



Slika 5 – Rješavanje zagonetke u igri Glitchspace

Izvor: Gamers Nexus (2014).

Kao što je vidljivo sa slike rješavanje zagonetki se ovdje vrši korištenjem vizualnog programiranja kako bi se manipuliralo objektima iz okoline. U ovoj igri naglasak nije na klasičnom načinu programiranja već na logičkom programiranju. Recenzije za ovu igru često napominju kako je veoma zanimljiva, vizualno i mehanički, ali da ima veliki problem u činjenici da loše podučava o svojim mehanikama pa igrači nerijetko ostaju frustrirani jer ne znaju kako riješiti neku zagonetku.

1.2.4 else Heart.Break()

Ova igra je jako unikatna iz dva razloga: ima jako specifičan vizualni stil i ima otvoreni svijet s kojime igrač može vršiti interakciju na jako velik broj načina i većinom je ograničen samo svojom maštom i trenutnom razinom znanja. U else Heart.Break() igrač vrši interakciju sa okolinom koristeći kod pod nazivom Språk koji je inspiriranim programerskim jezikom Basic (slika 6).



Slika 6 – Kodiranje u igri else Heart.Break()

Izvor: GameSideStory (2015).

Kako bi uspješno mogao igrati ovu igru od igrača se ne očekuje nikakvo prijašnje znanje programiranja već su mu svi važni koncepti objašnjeni tijekom igranja. Kako igrač postepeno uči nove stvari tako otkriva sve veći broj načina na koji može vršiti interakciju sa okolinom. Primjerice ukoliko se igračevom liku počne spavati on može hakirati čašu vode i modificirati njezina svojstva tako da mu u potpunosti napuni energiju. Igra omogućava ogroman broj mogućnosti za interakciju sa svijetom što također znači da će svaki igrač moći na svoj vlastiti način rješavati zagonetke i probleme. Recenzije i igrači su nahvalili ovu igru zbog njene jedinstvenosti i pametnog načina na koji igrače uči programiranju, prodajni rezultati nažalost nisu poznati.

1.2.5 Code Combat

CodeCombat je projekt otvorenog koda (engl. open source) čiji je cilj na što jednostavniji i zanimljiviji način omogućiti zainteresiranim igračima da nauče programirati. CodeCombat se u potpunosti kontrolira uz pomoć pisanja koda, kao što je vidljivo na sljedećoj slici (slika 7)



Slika 7 – Situacija iz igre CodeCombat

Izvor: Google Sites (2019).

U ovoj igri igrač kontrolira svoga lika tako da mu daje naredbe preko koda. Na početku svake razine igraču je rečeno koje uvjete mora zadovoljiti kako bi uspješno završio razinu. Na prvih nekoliko razina ti uvjeti budu da igrač mora dovesti svoga lika do kraja razine koristeći funkcije za kretanje, te ti uvjeti postepeno postaju sve kompleksniji. CodeCombat igračima nudi na izbor pet različitih programerskih jezika za učenje a to su: Python, JavaScript, HTML5, CSS3 i jQuery. Od 2013. godine kada je igra izdana prikupila je preko 5 milijuna igrača, prevedena je na preko 50 jezika i igrana u preko 190 različitih zemalja (CodeCombat 2019).

1.2.6 Code Hunt

Code Hunt je igra koja se može igrati u internet pregledniku i dizajnirana je od strane Microsoftovog istražiteljskog tima kako bi ga svi igrači sa interesom za programiranje mogli na zabavan način naučiti. Code Hunt funckionira tako da igraču kao zagonetku prezentira neki testni slučaj i igračev zadatak je da napiše verziju koda koji uspješno riješava taj testni slučaj (slika 8).



Slika 8 – Primjer Code Hunt zagonetke
Izvor: Microsoft Tech Community (2015).

S vremenom Code Hunt platforma se počela koristiti za održavanje raznih natjecanja za kodiranje. Zahvaljujući činjenici da je Code Hunt igra ispitivači su lako mogli vidjeti razne metrike poput prosjeka rješenih zagonetki, prosječnog broja pokušaja prije nego što je igrač došao do rješenja itd. Te metrike im omogućavaju da lako izoliraju ponajbolje kodere u tim natjecanjima. Zbog svoje specijalizirane prirode ova igra je prikupila oko 350 tisuća igrača što je manje nego ostale ovdje navedene igre.

1.3 Ciljano tržište

Sljedeći korak koji je potrebno provesti je demografska analiza tržišta kako bi se otkrilo koji tip igrača igra koje igre, te koje su njihove navike i motivacije. Osim toga potrebno je pogledati koji tip igara je najpopularniji na tržištu, za te potrebe pronađeni su podatci od deset najprodavanijih igara za osobna računala. Sljedeća slika prikazuje upravo taj popis (slika 9).

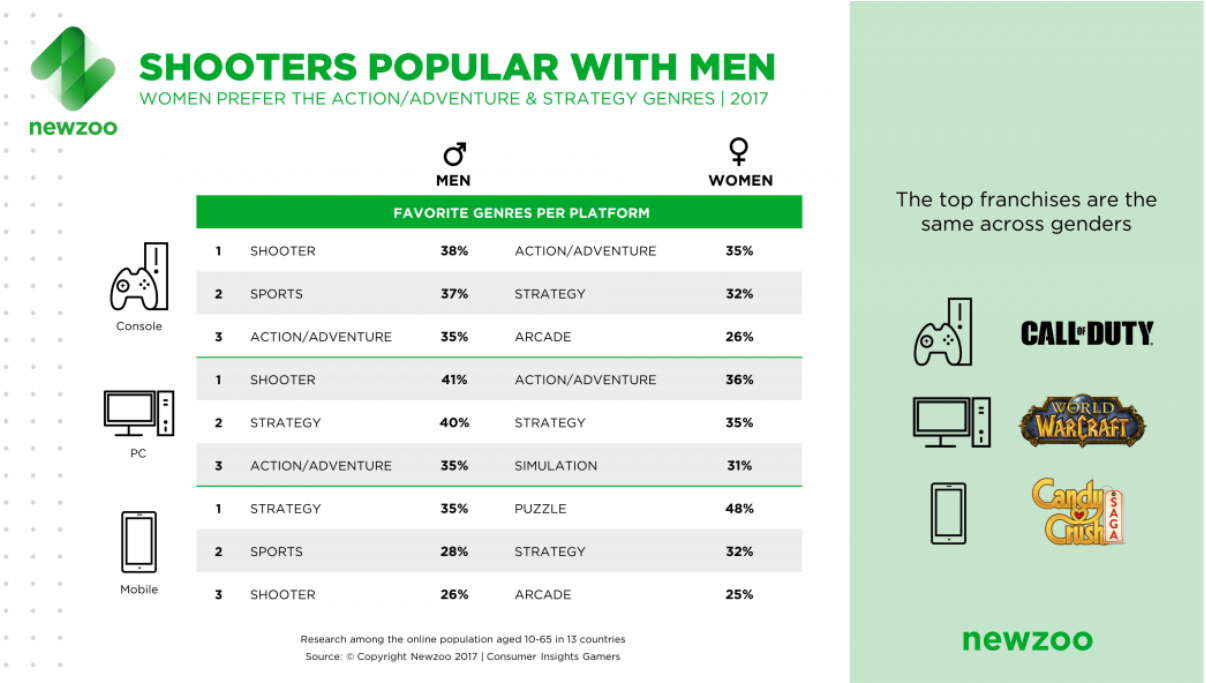
Top premium PC games by revenue, 2017

Rank	Title	Publisher	Revenue
1	*PlayerUnknown's Battlegrounds*	Bluehole	$714M
2	*Overwatch*	Activision Blizzard	$382M
3	*Counter-Strike: Global Offensive*	Valve Corporation	$341M
4	*Destiny 2*	Activision Blizzard	$218M
5	*Grand Theft Auto V*	Rockstar Games	$118M
6	*Battlefield 1*	Electronic Arts	$113M
7	*Minecraft*	Microsoft Studios	$92M
8	*Guild Wars 2*	NCsoft	$87M
9	*Divinity: Original Sin 2*	Larian Studios	$85M
10	*Tom Clancy's Rainbow Six: Siege*	Ubisoft	$67M

SUPERDATA

Slika 9 – 10 najprodavanijih igara za osobna računala u 2017. godini
Izvor: Super Data Research (2018).

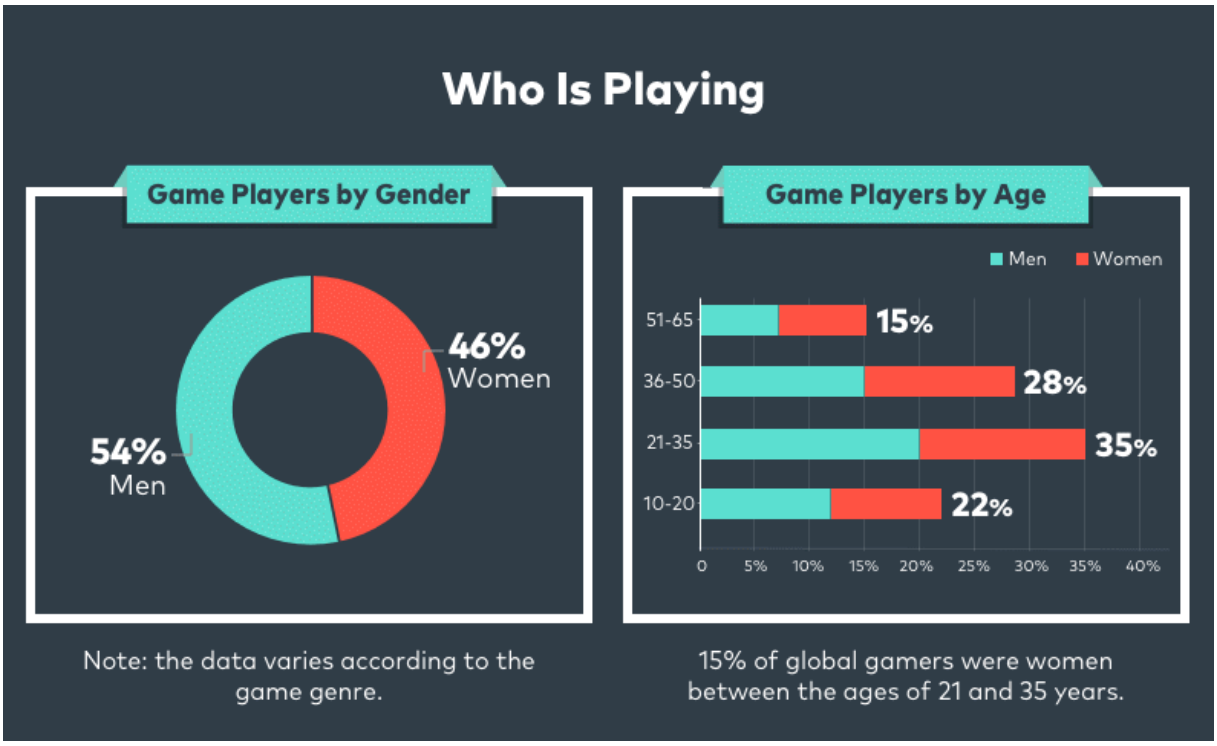
Kao što je vidljivo na slici najpopularnije su akcijske igre od kojih velika većina njih ima mrežnu komponentu, to jest omogućava igranje sa drugim igračima. Dakle iz toga se nameće zaključak kako puzzle igre za osobna računala ne spadaju u kategoriju najpopularnijih igara. Sljedeće pitanje koje se nameče je koliko su zapravo puzzle igre popularne među igračima. Web stranica NewZoo je 2017. godine provela istraživanje na temu upravo toga koji je tip igara najpopularniji na kojoj platformi među igračima muškoga i ženskoga spola zasebno, rezultati toga istraživanja nalaze se na sljedećoj slici (slika 10).



Slika 10 – Popularnost različitih žanrova po platformama

Izvor: Osborne (2017).

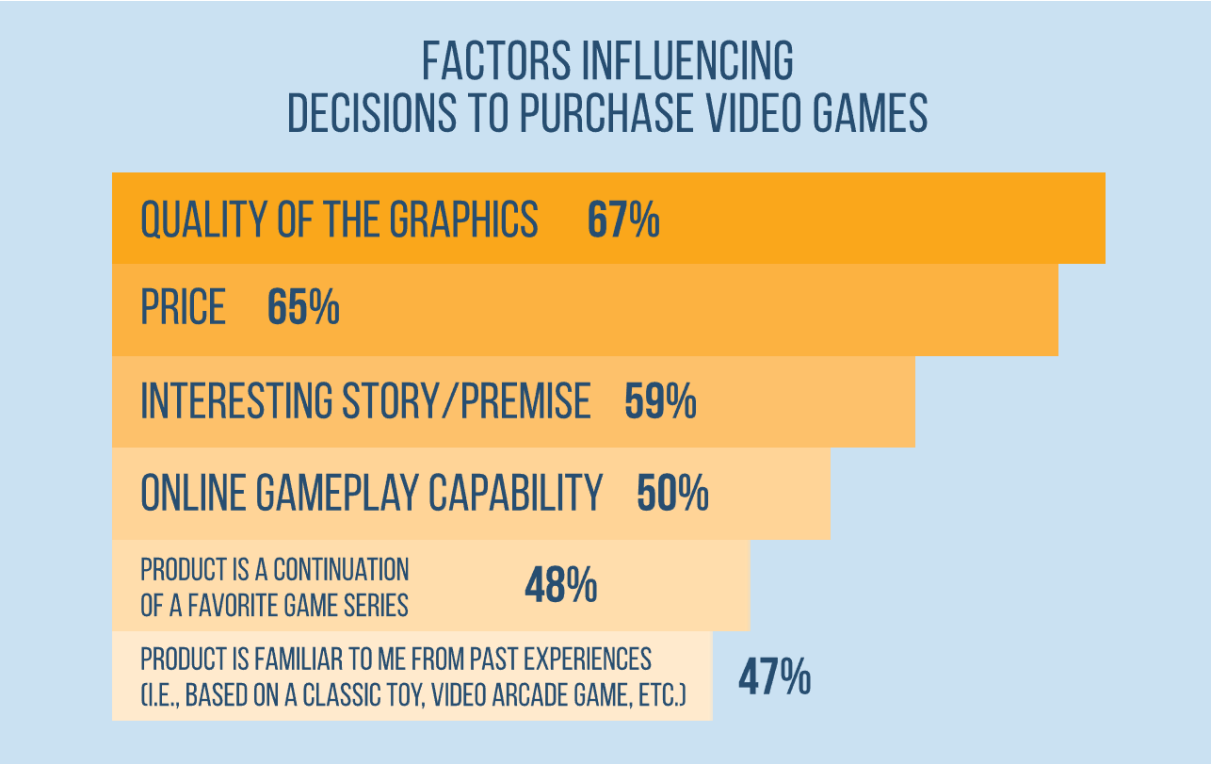
Istraživanje je pokazalo da su među muškarcima najpopularniji žanrovi pucačine, sportske igre i strategije, neovisno o platformi na kojoj te igre igraju. Među ženama je pak ta situacija nešto drugačija, one naime na osobnim računalima i konzolama preferiraju akcijske avanture i strategije. Što se tiče mobilnih igara ovdje žene najviše preferiraju igrati puzzle igre. Dakle na temelju ovih informacija najviše bi se isplatilo raditi puzzle igru za mobilnu platformu sa ženama kao ciljanom publikom. S obzirom da je CodeBreakers igra koja je razvijena specifično za osobna računala ovi podatci nam govore da puzzle igre na toj platformi imaju manju publiku. Na tematiku demografija koje igraju igre i njihovih obilježja proveden je jako velik broj istraživanja. Na temelju tih istraživanja prosječna dob igrača je 35 godina (Lofgren 2017), također istraživanje provedeno 2018. godine od strane Filmora tima pokazalo je da osobe između 36 i 50 godina predstavljaju 28% sveukupnih igrača (slika 11).



Slika 11 – Demografija igrača prema spolu i prema godinama

Izvor: Filmora (2019).

Prema ovoj infografici vidljivo je da je broj muškaraca i žena koji igraju videoigre podjednak, i da je najveći broj igrača u dobi između 21 i 35 godina. Još jedno istraživanje pokazalo je da osobe između 15 i 25 godina preferiraju akcijske i kompetitivne igre dok kako postaju stariji sve veći interes imaju prema strateškim i općenito igrama koje zahtjevaju razmišljanje (Yee 2016). Još jedna potencijalno zanimljiva stvar je vidjeti do kojeg igrinog segmenta je prosječnom igraču najviše stalo. Upravo su to The Entertainment Software Association istražili u svome istraživanju provedenom 2016. godine, i rezultat toga istraživanja prikazan je na sljedećoj slici (slika 12).



Slika 12 – Faktori koji utječu na odluku da igrač kupi neku igru

Izvor: Strickland (2017).

Rezultati tog istraživanja pokazali su da je prosječnom igraču najvažniji grafički izgled igre, pa tek onda cjena i igrina potencijalno zanimljiva priča.

Na temelju svih ovih podataka i istraživanja nameće se zaključak da bi idealna tržišna demografija za igru CodeBreaker bile osobe muškog i ženskog spola starije od 25 godina. Stvarna situacija je ipak nešto drugačija, osnovno ciljano tržište za ovu igru predstavljaju osobe raznih dobi počevši od srednjoškoloca i studenata pa nadalje, jedino što je potrebno je interes za učenjem programiranja. Još jedna bitna stvar koju bi trebalo imati na umu je da bi igra trebala imati zanimljiv vizualni stil kako bi privukla igrače, također treba imati na umu da su puzzle igre najuspješnije na mobilnim uređajima dok na osobnim računalima imaju mnogo manju publiku osim ako nisu rađene od strane velikih studija i/ili imaju jedinstvene mehanike; kao prije navedeni Portal.

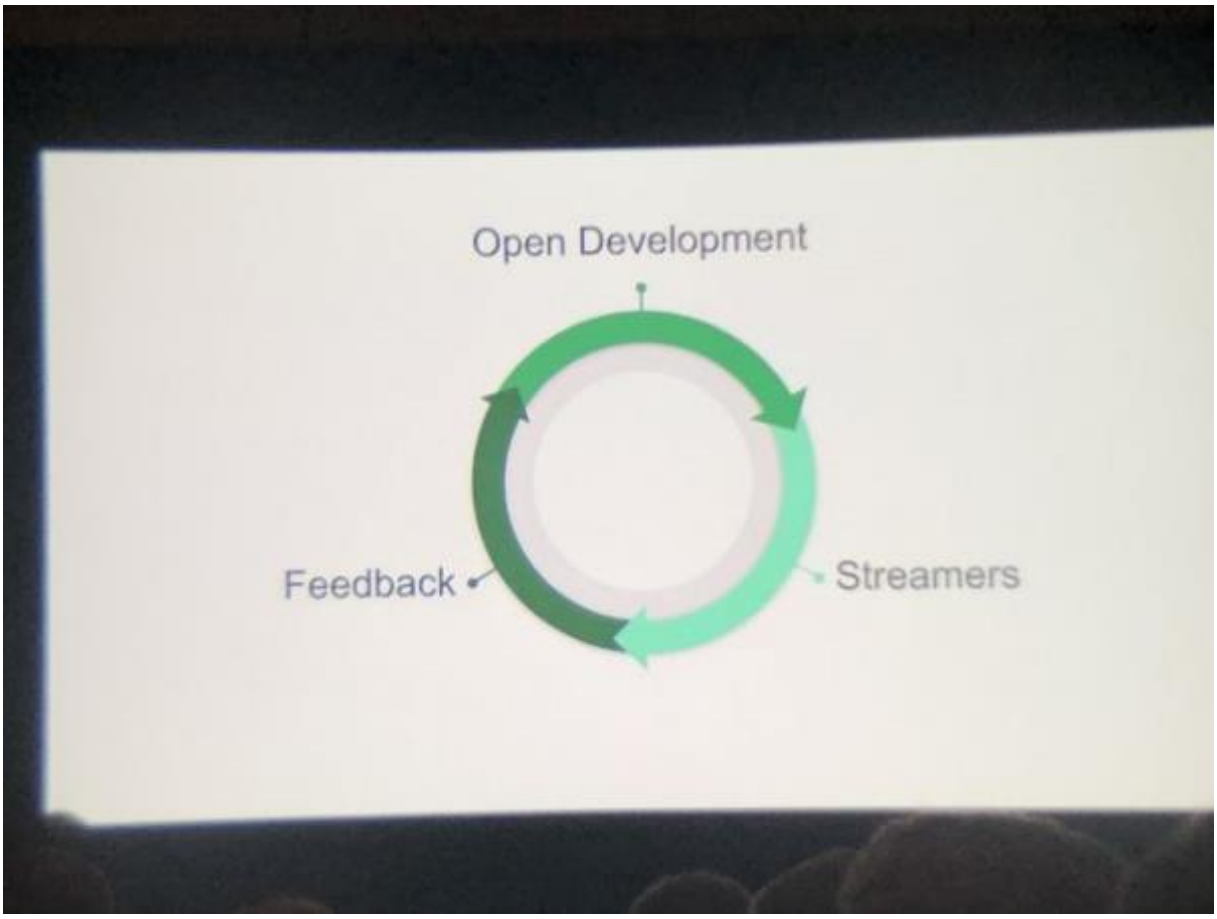
1.4 Promocija i izbacivanje na tržište

Sada kada je provedena analiza tržišta i sakupljene potrebne informacije o ciljanoj demografskoj skupini igrača potrebno je osmisliti marketinšku strategiju i

odlučiti na kojim platformama za digitalnu distribuciju videoigara bi igra bila objavljena. Ovdje pri donošenju odluka također kao faktor treba uzeti činjencu da se ovdje radi o malome timu koji na raspolaganju nema veliku količinu resursa.

Prvi dio promocije igre bi se tako vršio preko stranica specijaliziranih za vijesti o videoigrama. Jedna od tih stranica bila bi HCL (Hrvatska Cyber Liga), to je najveća i najpopularnija domaća stranica za pokrivanje ovakvih sadržaja, te bi vjerojatno bili jako voljni poduprijeti Hrvatsku igru tako da napišu članak o njoj na svojoj stranici. Druga glavna stranica bila bi PCGamer, stranica specijalizirana za prenošenje vijesti o igrama za osobna računala. Drugi način promocije igre bilo bi objavljivanjem informacija o igri na raznim forumima koji su specijalizirani za igre ovoga tipa, na primjer Puzzle Games Reddit, ovakav oblik promocije je besplatan i može potencijalno privući veliki broj zainteresiranih igrača.

Drugi dio promocije igre bi se vršio preko popularnih streamera (streamer - osoba koja uživo prenosi svoje igranje neke igre), te preko raznih Youtube kanala. Važna stvar ovdje je da se odabere Youtube kanale i streamere koji se ili specijaliziraju za ovakav tip igara ili imaju interes za igranjem ovakvih igara. Ukoliko se odabere pogrešna osoba za takvu promociju vjerojatno igra ne bi bila prikazana u najboljem svijetlu. Još jedna korist od korištenja streamera za promociju je činjenica da se može jednostavno dobiti povratna informacija na taj način. Naime na ovogodišnjem GDC-u (Game Developers Conference), najvećoj konferenciji za developere videoigara, Jonas Boetel je održao govor o razvoju igre na kojoj je radio pod nazivom Subnautica (Gamasutra 2019). Unutar toga govora prikazao je prezentacijski slide sa iduće slike (slika 13).

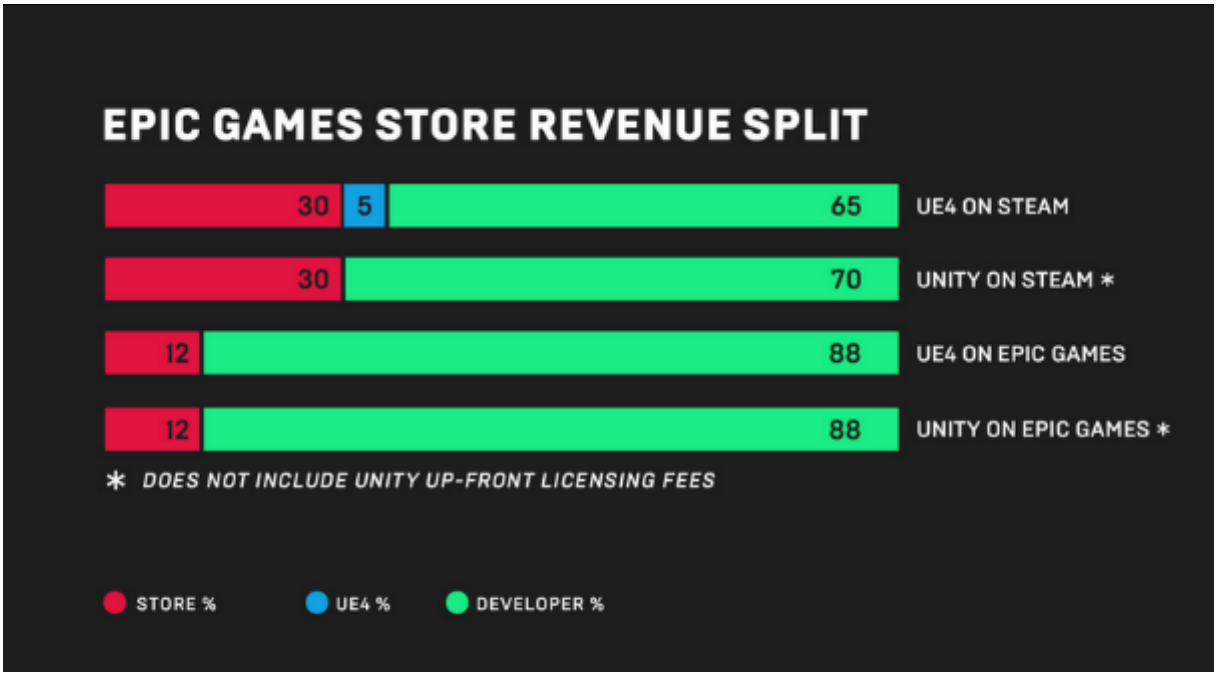


Slika 12 – Utjecaj streamera na razvoj igre
Izvor: Gamasutra (2019).

Gornji diagram prikazuje na kakav način streameri mogu pozitivno utjecati na igrin razvoj. Streameri krozi prenošenje svog igranja igre mogu developerima biti izvor velikog broja informacija, primjerice koliko je igru zabavno igrati, koji su problematični dijelovi igre, koje sustave je potrebno doraditi i tako dalje. Na temelju svih tih dobivenih inormacija developeri mogu napraviti prioritetnu listu poboljšanja i na temelju svih tih povratnih informacija nastaviti rad na igri.

Zadnja stvar koja preostaje je odabir platforme za distribuciju igre. Ovdje se nameću dvije glavne platforme, a to su Steam i Epic Games Store. Kao što je prijašnji dio rada prikazao Steam predstavlja najveću digitalnu platformu za distribuciju video igara sa jako velikim brojem korisnika. Dakle Steam definitivno predstavlja platformu na kojoj bi se isplatilo objaviti igru, ovdje postoji samo jedan problem a to je problem vidljivosti igre. Kao što slika u prijašnjem dijelu rada (slika 1) prikazuje, na Steam platformi se iz godine u godinu objavljuje sve veći broj igara, što znači da manji razvojni timovi imaju problema sa isticanjem svoje igre u velikom moru

ostalih igara, dakle ovdje bi marketinška strategija imala veliki utjecaj. Idući mogućnost je Epic Games Store platforma. Epic Games Store je otvoren u prosincu 2018. godine i zbog te činjenice trenutno ima mali broj igara u svojoj ponudi. Za razliku od Steama, Epic Games Store ne dopušta da bilo kakve igre budu objavljene na njihovoj trgovini već moraju biti određene kvalitete kako bi ih dopustili. Osim manje konkurencije Epic Games Store ima još jednu važnu prednost u odnosu na Steam, a to je postotak koji uzimaju od prodaje svake igrine kopije što je prikazano sljedećom slikom (slika 13).



Slika 13 – Udio prodaje koji Epic Games i Steam dobivaju
Izvor: Epic Games (2019).

Developeri koji objavljuju svoju igru na Steam platformi daju 30% svoje zarade Steam-u, a ukoliko je njihova igra rađena u Unreal Engine razvojnom okruženju daju još dodatnih 5% zarade Epic Gamesu koji su kreatori toga razvojnog okruženja. Ukoliko se developeri odluče objaviti svoju igru na Epic Games Store-u tada trgovini daju samo 12% svoje zarade što je jako dobro za manje timove sa malom količinom dostupnih resursa. Na temelju svih ovih informacija dolazi se do zaključka kako bi bilo najpametnije pokušati objaviti igru na obije platforme neovisno o postotku koji uzimaju od prodaje igre. Razlog tome je činjenica kako ne postoji garancija da bi Epic

Games smatrao da je CodeBreaker dovoljno kvalitetna igra da se stavi na njihovu trgovinu.

1.5 SWOT analiza

Nakon što je provedena analiza tržišta, napravljena marketinška strategija i donesena odluka o platformama na kojima će igra biti objavljena, potrebno je provesti SWOT analizu. SWOT analiza omogućava da se procjeni kako se igra Codebreaker uspoređuje sa konkurentima i njihovim rješenjima, da se otkriju njene prednosti i nedostatci, te da se otkrije kako u budućnosti poboljšati igrui. SWOT tablica prikazana je na sljedećoj slici (slika 14).

CodeBreaker SWOT analiza	
Strenghts	**Weaknesses**
- Jasna vizija - Upoznatost sa ciljanom publikom - Relativno originalan tip igre - Daje mogućnost učenja na interaktivan način - Velik baza znanja programerskih koncepata	- Mali tim - Nedostatak iskustva - Publika su specifičan tip ljudi - strpljivi i sa željom da nešto nauče - Nedostatak kreativnosti u zagonetkama
Opportunities	**Threats**
- Prebacivanje igre na mobilnu platformu - Dodavanje novih sadržaja i koncepata - Prerada nekih zagonetki kako bi ih se učinilo zanimljivijima	- Kompanije sa više iskustva i boljim resursima - Potencijalno bolje suparnička rješenja - Igra spada u manje popularan žanr igara - Loše odrađen marketing bi imao jako negativan utjecaj na vidljivost igre

Slika 14 – SWOT analiza igre CodeBreaker

Izvor: Vlastita izrada

1.5.1 Snage

Glavna snaga iza ove igre je činjenica da postoji jasna vizija oko toga kako bi na kraju ona trebala izgledati i samim time je jasno kakav tip igrača predstavlja igrinu ciljanu publiku. Igra sama po sebi je relativno originalna, relativno zbog toga što kao što je navedeno u dijelu koji se bavio sličnim igrama vidljivo je da postoje igre slične tematike, ali ni jedna nije identična ovoj. Razlog tome je činjenica kako ova igra predstavlja puzzle igru koja se igra iz prvog lica u kombinaciji sa pisanjem pravoga

koda a ne nekom aproksimacijom kodiranja. Na kraju zadnja snaga ove igre je činjenica da u sebi sadrži veliku bazu znanja programerskih koncepata i to omogućava korisnicima da na interaktivan način uče programerske koncepte. Takav način učenja bi potencijalno mogao biti zanimljiviji i zabavniji od onog standardnog, gledanje videa, čitanje prezentacija i programerskih foruma i tako dalje.

1.5.2 Slabosti

Glavnu slabost ovdje predstavlja činjenica da na ovoj igri rade samo dvije osobe koje nemaju prijašnjega iskustva sa izradom igara ovoga tipa, dakle igre bazirane na zagonetkama. Zagonetke i programerski aspekt igre je također čine poprilično kompleksnom za izradu, te navedene slabosti također ovdje igraju ulogu. Sljedeća slabost je činjenica da je ovo igra za jako specifičan tip igrača, a to su osobe koje žele naučiti programirati i imaju strpljenja za proučavanje svih potrebnih materijala kako bi mogli uspješno rješavati zagonetke. Zadnja slabost također proizlazi iz nedostatka iskustva, a to je nedostatak kreativnosti u zagonetkama što bi moglo dovesti do toga da igrači postepeno počnu gubiti interes za igranjem.

1.5.3 Prilike

Kao što je navedeno u ranijem dijelu ovoga poglavlja najveći interes za puzzle igrama postoji na mobilnoj platformi. Samim time ta informacija pruža priliku da se omogući igranje ove igre na mobilnim uređajima, što bi potencijalno poboljšalo igrinu vidljivost i popularnost. Iduća prilika postoji u mogućnosti da se nadodaju novi programerski koncepti i razine, te naposljetku da se iskoristi znanje stečeno prilikom izrade ove igre kako bi se preradile neke od postojećih zagonetki da budu zanimljivije i manje repetitivne.

1.5.4 Prijetnje

Činjenica da je ova igra rađena od strane malog tima sa limitiranim resursima dovodi do prijetnje od boljih suparničkih rješenja. Ukoliko neka veća kompanija sa većim resursima na raspolaganju odluči napraviti igru sličnoga tipa ona će potencijalno izgledati bolje samo zbog činjenice da na njoj radi veći broj ljudi koji na raspolaganju imaju više resursa. Još jedna prednost koju takvi timovi imaju je bolja vidljivost za njihovu igru, bilo zbog činjenica da je ta kompanija već poznata u

industriji videoigara ili zbog činjenice da imaju više resursa na raspolaganju za kvalitetniju marketinšku kampanju. Još jednu prijetnju predstavlja činjenica da je ovo igra napravljena za jako specifičnu publiku što znači ukoliko ta publika ne bude zadovoljna igra neće biti uspješna. Zadnja velika prijetnja koja ovdje postoji je loša marketinška kampanja. Zbog činjenice da je ovo igra rađena od strane malog nepoznatog tima potrebno je što kvalitetnije odraditi marketinšku kampanju i povećati igrinu vidljivost. Neovisno o tome koliko je igra potencijalno zanimljiva i kvalitetna, nitko je neće odigrati ukoliko ne znaju za njezino postojanje.

2.Korištenje igara za podučavanje i njihov utjecaj

U današnje doba velika količina ljudi ima pristup nekom oblikom tehnologije, bilo nekom novijem pametnom telefonu, osbnom računalu ili laptopu. Također kao što su prijašnje statistike pokazala sve veći broj ljudi raznih dobnih skupina vrijeme provode igrajući videoigre. Zbog toga se nameče pitanje da li bi se videoigre mogle koristiti i za edukaciju, a ne samo za zabavu. Ovo poglavlje baviti će se upravom tematikom korištenja videoigara za potrebe podučavanja, biti će opisani primjeri iz prakse, te studije provedene na tu tematiku i njihovi zaključci.

2.1 Igre u obrazovnom sustavu

Tehnologija je kroz godine počela imati sve veći utjecaj u obrazovnim sustavima diljem svijeta. Taj utjecaj je vidljiv kroz korištenje tehnologije poput tableta i raznih edukativnih aplikacija. Kako bi se učenicima proces učenja napravio zanimljivijim počelo se razmatrati korištenje video igara za potrebe edukacije i počela su se provoditi istraživanja na tu temu. Jedno od takvih istraživanja bilo je provedeno 2018. godine u Dag Hammarskjold srednjoj školi u Wallingford, Connecticutu. Učenici te škole su sudjelovali u edukativnoj aktivnosti unutar koje su imali pristup igri dizajniranoj za podučavanje. Na prvi pogled način igranja te igre djeluje u potpunosti standardno, igrači ispunjavaju izazove i na taj način dobivaju bodove koji im omogućavaju da kupuju nadogradnje ili novu odjeću za svoje likove. Za razliku od mnogih drugih igara ovdje se igrači ne natječu međusobno, već se te igre koriste od strane učitelja kako bi učenicima prezentirali razne edukacijske teme. Učenici na raspolaganju imaju nekoliko različitih razina za svaku od tih tema, te njihovim uspješnim ispunjavanjem dobivaju bodove. Kada je jedan od učenika upitan za svoje mišljenje o korištenju igara za edukaciju rekao je kako je učenje na ovakav način zabavno i da jako uživa u tome. Jedna od učiteljica u toj školi, Gianna Gurga, je već 2017. godine počela koristiti igru za edukaciju pod imenom Classcraft. Nakon uvođenja te igre u svojoj učionici primjetila je napredak u znanju i motivaciji svojih učenika, taj napredak bio je očit i ostalim učiteljima pa su neki od njih također počeli

koristiti Classcraft u svojim učionicama. Način na koji Classcraft funkcionira je jednostavan, igrači odgovaraju na pitanja te za svaki točan odgovor dobivaju bodove. Dobivene bodove mogu koristiti na više načina, mogu kupovati nove predmete za svoj lik unutar igre, koristiti bodove kako bi se natjecali protiv drugih razreda ili jednostavno zamijeniti bodove za mogućnost da slušaju muziku u razredu. Također ukoliko se učenik loše ponaša postoji i mogućnost oduzimanja bodova (Lynn 2019). Dakle na primjeru ove škole čini se da korištenje videoigara u učionici ima pozitivan utjecaj na učenike, ali potrebno je promotriti još nekoliko istraživanja kako bi se došlo do jasnijeg zaključka.

Jedno od tih istraživanja predstavlja rad „The Effect of using educational games on the students' achievment in english language for the primary stage“ kojeg je napisala Mani Moayad Mubaslat. Tematika ovoga rada bilo je otkriti da li je korištenje edukacijskih videoigara za učenje nekog novog jezika efektivinije od standardnih metoda učenja. Unutar ovoga rada autor navodi kako je učenje nekog novog jezika teško i kako uključuje jako puno truda kroz veći period vremena i zbog toga želi istražiti da li je učenje moguće na neki zabavniji način. Istraživanje je provedeno na sljedeći način: korištene su tri nasumično odabrane grupe ispitanika koji su bili sa četvrte, sedme i desete godine. Potom su ispitanici podjeljeni u dvije skupine, grupu A koja je predstavljala eksperimentalnu grupu koja je učila engleski uz pomoću igara i grupu B koja je predstavljala kontrolnu grupu koja je učila engleski koristeći standardne metode učenja. Kako bi se obradili statistički podatci i kako bi se moglo doći do potrebnih zaključaka korištena je jednosmjerna analiza varijanci (ANOVA), a kako bi se našla relacija između edukacijskih igara i učenja korištena je Pearsonova korelacijska analiza. Rezultat ovog istraživanja pokazao je da je eksperimentalna grupa imala daleko bolje rezultate od kontrolne grupe, dakle zaključak koji se nameče je kako videoigre imaju jako pozitivan efekt kod učenja engleskog jezika (Mubaslat 2012).

Iduće bitno istraživanje provedeno je od strane GlassLab-a koji je osnovan uz veliku pomoć Bill & Melinda Gates udruge. GlassLab dizajnira i implementira videoigre koje unutar sebe imaju materijale za učenje i alate za praćenje napretka učenika. Ideja iza ovoga je jednostavna, djeca vole igre pa samim time korištenje igara u kontekstu učionice bi im trebalo učenje učiniti zanimljivijim. Jedan od zaključaka do kojega su došli je da kod učenika koji je po svojem znanju na medijanskoj vrijednosti postojao napredak od čak 12% ukoliko bi koristio neku od

njihovih igara za učenje. Ključna stvar kod ovakvoga pristupa je činjenica da je igranje korisno zbog toga što simulira iskustva iz stvarnoga života u sigurnom socijalnom okruženju unutar kojega učenik može osjećati još veće zadovoljstvo kada nešto uspješno nauči. Korištenje simulacija za edukaciju je imalo još veći utjecaj na poboljšanje kod studenata. Simulacije su napravljene na način da veći naglasak bude stavljen na implementaciju edukacijskih sustava a manje na igrine sustave, primjerice animirana lekcija iz anatomije. Korištenje simulacija dovelo je do napretka od čak 25% kod ispitanika (Shapiro 2015).

U američkim gradovima New York i Chicago 2010. godine uveden je program Quest to Learn. Quest to Learn je program unutar kojega je nekoliko škola u tim gradovima za edukaciju primarno koristilo sustave za edukaciju temeljene na igrama. Jedan od primjera korištenja igre za edukaciju u tome sustavu je za učenje biologije u devetom razredu. Ovdje učenici kroz godinu igraju ulogu radnika u izmišljenoj bio-tech kompaniji i njihov je posao da kloniraju dinosaure i stvaraju stabilne ekosustave za njih. Na ovaj način studenti uče o genetici, biologiji i ekologiji (Barseghian 2013). Ovisno o predmetu koji uče, učenici uzimaju uloge dizajnera, znanstvenika, doktora, detektiva i ostalih uloga koje imaju smisla u kontekstu toga predmeta. Quest to Learn se pokazao poprilično uspješnim, naime u prve četiri godine učenici iz New Yorka koji su učili unutar toga sustava imali su bolje rezultate iz engleskog jezika, matematike i umjetnosti u odnosu na ostale škole iz New Yorka. Jedina iznimka je bila 2010. godina kada su imali lošije rezultate iz matematike u odnosu na ostale škole. Nakon ovako dobrih rezultata nameće se pitanje zašto više škola nije počelo uvoditi sustave bazirane na igrama, taj razlog je jednostavan – novac. Za implementaciju ovakvih tehnoloških naprednijih ssustava potrebno je imati pristup velikim resursima, a mnoge škole ili nemaju takav novac ili ga ne žele trošiti za ovakve potencijalno eksperimentalne sustave.

2.2 Utjecaj igara na ljudski mozak

Nakon što je promotreno na kakav način edukacijske igre utječu na učenike i njihovu sposobnost učenja, također treba promotriti na kakav način igranje igara utječe na ljudski mozak. Ovdje će naglasak biti na puzzle igrama zbog toga što CodeBreaker predstavlja upravo takav tip igre.

2014. godine na Nanyang Technological University (NTU) provedena je studija unutar koje je testirano na kakav je način svakodnevno igranje određenih igara utjecalo na mozgove ispitanika. Ova studija provedena je od strane asistenta profesora Michael D. Pattersona i njegovog studenta Adam Oei-a. Njih dvojica su za testiranje koristili četiri različita tipa igre na mobilnoj platformi zbog toga što je njihovo prijašnje istraživanje pokazalo da različiti tipovi igara imaju različite utjecaje na ljudski mozak. Korištene igre su bile pucačina Modern Combat, arkadna igra Fruit Ninja, strateška igra StarFront Collision i puzzle igra Cut the Rope. Za testiranje su odabrani NTU studenti koji su imali malo iskustva sa igranjem videoigara. Odabrani studenti su igrali odabranu igru sat vremena po pet dana u tjednu na svojim mobilnim uređajima. Razdoblje ovoga testiranja bilo je četiri tjedna što dolazi do sveukupno 20 sati igranja. Istraživanje je pokazalo kako su učenici koji su igrali puzzle igru Cut the Rope pokazali značajno unaprijeđenje kod rješavanja zadataka izvršnih funkcija (engl. executive function tasks), dok učenici koji su igrali ostale igre nisu pokazali nikakvo značajno unaprijeđenje svojih kognitivnih funkcija. Izvršne funkcije mozga su bitne za donošenje odluka u svakodnevnom životu kada se dogode neke nagle promjene unutar tog svakodnevnog okruženja, kao primjerice odluka da li će se osoba zaustaviti ili nastaviti voziti kada semafor neočekivano pokaže žuto svijetlo. Način na koji su studenti testirani je sljedeći: koliko brzo igrač može promijeniti zadatak i početi izvoditi novi, koliko se brzo igrač može prilagoditi na novu situaciju umjesto da se oslanja na dosadašnju strategiju i koliko se dobro mogu fokusirati na neku informaciju dok blokiraju razne distrakcije. Razlog zašto je igra Cut the Rope imala tako pozitivan utjecaj je vjerojatno zbog činjenice da kako se napreduje kroz igru, prijašnje strategije prestaju biti efektivne i zbog toga igrač mora pronaći nove strategije za rješavanje tih zadataka. Nakon 20 sati igranja igre Cut the Rope studenti su uspjevali prebacivati između raznih zadataka i do 33% brže, prilagođavati se novim situacijama 30% brže i bili su sposobni blokirati razne distrakcije i fokusirati se na potrebnu informaciju čak 60% brže. Dakle zaključak ove studije bio je da ukoliko netko želi poboljšati svoje mentalne sposobnosti trebao bi svakodnevno bar sat vremena igrati neku puzzle igru koja je slična kao Cut the Rope (Science Daily 2014).

Još jedna zanimljiva studija provedena je 2013. godine u njemačkim Max Planch Institute for Human Development i Charité University of Medicine St. Hedwig-Krankenhaus. Tematika ovoga istraživanja bilo je otkriti da li igranje videoigara utječe na dijelove mozga zadužene sa orijentaciju u prostoru, formiranje pamćenja,

strateško planiranje i unaprijeđenje motornih funkcija. Voditeljica ove studije, Simone Kühn, kaže kako je ova studija pokazala da postoji direktna veza između igranja videoigara i volumetričkog povećanja mozga. Ova činjenica je dokazala kako se specifični dijelovi mozga mogu istrenirati korištenjem videoigara i to znači kako videoigre mogu biti terapeutski korisne za pacijente koji pate od mentalnih poremećaja zbog kojih su im određeni dijelovi mozga smanjeni. Neki od tih poremećaja su shizofrenija, post traumatski stres sindrom i bolesti koje degeneriraju mozak poput Alzheimerove bolesti (Alonso 2016).

Znanstvenici na Oxford fakultetu su također dokazali kako poznata videoigra Tetris smanjuje utjecaj post traumatskog stres sindroma. Način na koji su to dokazali je bio da su uzeli 40 ispitanika i pokazali su im cijeli niz slika jako uznemirujućih tematika. Pola sata nakon što su im pokazali te slike, polovici ispitanika dano je da igraju Tetris dok ostalima nije dano da rade išta. Ispitanici koji su igrali tetris patili su od puno manjeg broja epizoda prisjećanja tih slika (engl. flashbacks), a upravo je neželjeno prisječanje uznemirujućih događaja jedna od glavnih karakteristika post traumatskog stres sindroma.

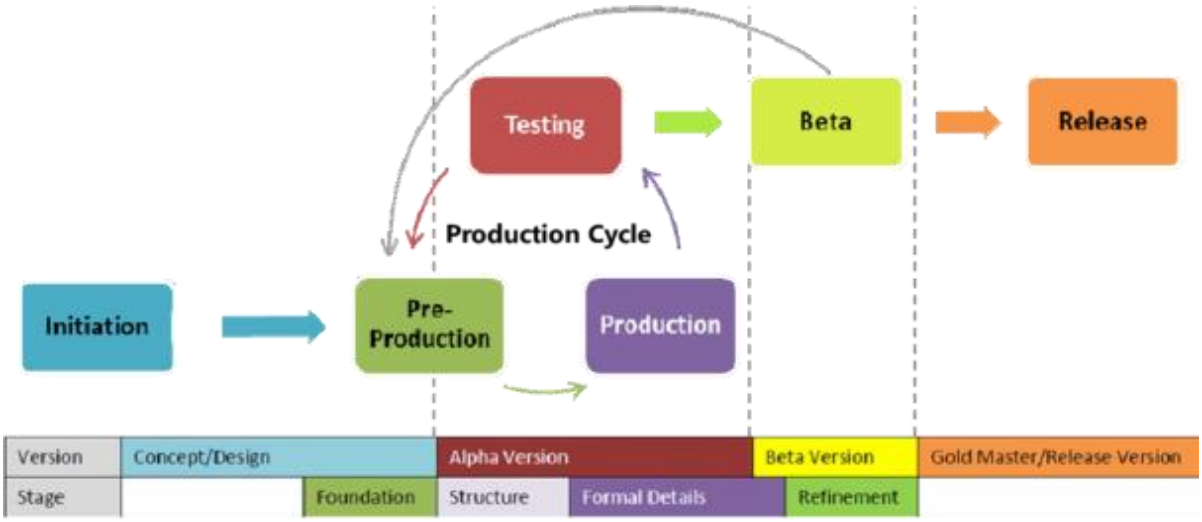
Još jedan tip istraživanja koji je provođen od strane različitih istraživačkih timova bavio se tematikom da li igranje puzzle videoigara čini igrače pametnijima. Studija provedena 2012. godine od strane dvojice španjolskih profesora psihologije analizirala je puzzle igru Professor Layton. Cilj njihovog istraživanja bio je da pokušaju otkriti da li ovakav tip video igara može povećati količinu sive tvari u mozgu, to jest da li čini igrače pametnijima. Za potrebe istraživanja oni su surađivali sa volonterima koji su znali kako funkcionira konzola za igranje te igre, ali nikada nisu igrali tu igru. Nakon mjesec dana testiranja i sveukupnog igranja oko 16 sati, znanstvenici su primjetili da je mozak ispitanika postao snažniji što ih je dovelo do zaključka kako videoigre mogu biti od velike pomoći u procesu učenja.

Kroz godine provedeno je još mnogo istraživanja koja su se bavila utjecajem igara na mozak igrača i neki od zaključaka tih istraživanja bila su da djeca koja igraju videoigre pokazuju veću kreativnost kod pisanja i crtanja, puzzle igre poboljšavaju vizualno i prostorno procesiranje, vizualnu percepciju, topografsku orijentaciju, planiranje i mogućnosti prosuđivanja. Sve u svemu ova istraživanja su pokazala da videoigre, specifično puzzle igre, imaju jako pozitivan utjecaj na ljudski mozak i da predstavljaju jako dobar način za jačanje raznih funkcija ljudskoga mozga.

3.Razvoj igre

Ovo poglavlje baviti će se razvojnim ciklusom igre, sa naglaskom na dizajn. Biti će prikazane metodologije razvoja i dizajniranja pojedinih igrinih dijelova, navedeni korišteni alati i tehnologije te objašnjeno zašto su određene stvari dizajnirane na određeni način. Također će biti pričano o tome što čini dobru puzzle igru, te na kakav način je potrebno pristupiti dizajniranju i izradi takve igre, pogotovo kada je poanta te igre da igrača nešto nauči.

Kao i svaki drugi softverski proizvod videoigre također imaju specifični razvojni ciklus, generalizirani prikaz razvojnog ciklusa videoigara prikazan je na sljedećoj slici (slika 15).



Slika 15 – Razvojni ciklus videoigara

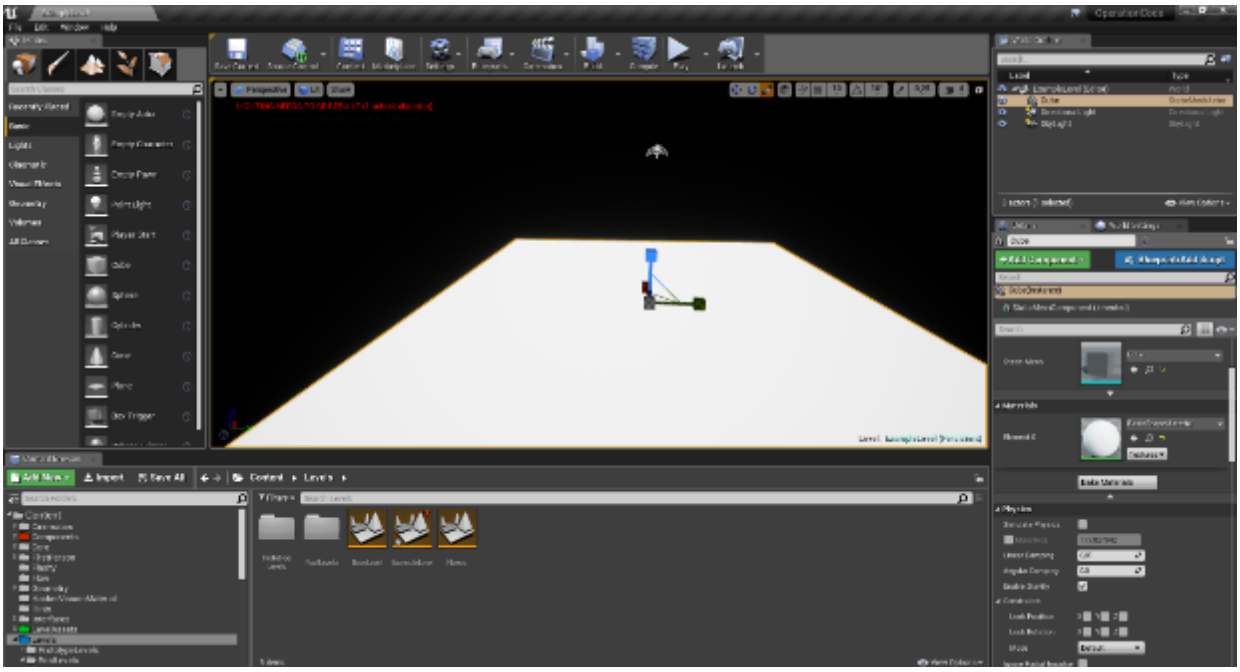
Izvor: Researchgate (2013).

Dakle razvojni ciklus se sastoji od nekoliko glavnih aktivnosti, a to su pred-produkcija, produkcija i testiranje, te četiri glavna stanja igre: prototipna verzija, alpha verzija, beta verzija i završna verzija. Na početku razvoja igra postoji samo kao ideja,

ta ideja se tijekom pred-produkcije pokušava pretvoriti u prototipnu verziju, te kada su developeri zadovoljni sa prototipom, igra ulazi u pravu produkciju. Kada tijekom produkcije igra dođe do igrivog stanja i svaki put kada se nadoda neki novi sustav igra se testira i na temelju rezultata testiranja se produkcija nastavlja. U tom trenutku igra se nalazi u alpha verziji. Kada su developeri zadovoljni sa rezultatima alpha testiranja i kada su svi veliki bugovi popravljene igra ulazi u beta verziju, se daje većem broju ljudi na tesitranje. Kada igra dobije pozitivne rezulate beta testiranja ona ulazi u završnu zlatnu verziju koja je spremna za izbacivanje na tržište.

3.1 Korištene tehnologije

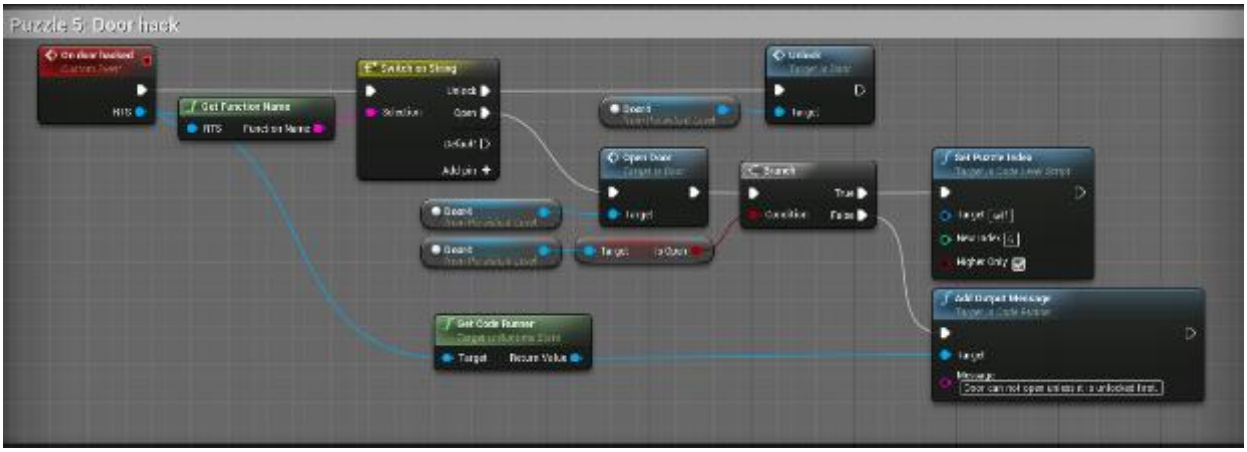
Prije nego što se krene sa objašnjavanjem procesa dizajniranja i izrade videoigre potrebno je spomenuti koje su tehnologije korištene pri njenoj izradi. Za dizajn raznih koncepata vezanih za razvoj korisničkog sučelja korišten je Adobe Photoshop, za vođenje dokumentacije korišten je Microsoft Word, Github za kontrolu verzija (engl. version control), te naposljetku od najveće važnosti bio je Unreal Engine 4 u kombinaciji sa Microsoftovim Visual Studiom. Unreal Engine 4 (UE4) je razvojno okruženje za izradu videoigara, drugi naziv bi bio igrin pokretač (engl. game engine). UE4 je editor koji u sebi sadrži cijeli set alata koji su potrebni developerima za izradu videoigara poput alata za izradu igrinih razina, animacija, vizualnih efekata, snimanje cutscena te mnoge druge, izgled osnovnog UE4 editora je prikazan na sljedećoj slici (slika 16).



Slika 16 – Izgled osnovnog Unreal editor prozora

Izvor: Vlastita izrada

Jedna od najbitnijih stvari koje UE4 nudi developerima je njihov Blueprints sustav. Blueprints je sustav koji omogućava vizualno skriptiranje i razvijen je od strane Epic Gamesa, kreatora ovog razvojnog okruženja. Ovaj sustav omogućava da se kod piše na vizualan način, uz pomoć tako zvanih node-ova koje je moguće povlačiti i spuštati na željenom dijelu radne površine (engl. drag and drop), te direktno mjenjati njihova svojstva. Ti node-ovi predstavljaju individualne dijelove koda te ih je potrebno međusobno povezivati kako bi se stvorilo funkcionalan kod. Blueprints sustav je prikazan na sljedećoj slici (slika 17).



Slika 17 – Izgled Blueprint sustava
Izvor: Vlastita izrada

Blueprints sustav u pozadini koristi programski jezik C++, pa ukoliko korisnik ima potrebu da piše svoj vlastiti kod ima tu mogućnost. Za potrebe ovoga projekta je korišten blueprint sustav u kombinaciji sa ručno pisanim C++ kodom za čije je pisanje korišten Microsoft Visual Studio. Detaljnije informacije o ovome može se pronaći u radu kolege A. Drozine koji se bavi upravo implementacijom razih sustava i objašnjava kako kod tih sustava funkcionira (Drozina 2019).

3.2 Pred-produkcija i dizajn dokument

Tijekom pred-produkcijskog razdoblja potrebno je osmisliti ideju za igru, njene mehanike, priču, te sve ostale bitne dijelove. Kada je započeta pred-produkcija bili su poznati samo vizija za igru i cilj koji je željeno postići ovom igrom. Unutar ovoga razdoblja igrini dizajneri (engl. game designers) imaju jako važnu ulogu. Tijekom pred-produkcije dizajneri moraju osmisliti igrine koncepte, sustave, likove, priču i detaljno objasniti na kakav način će se ona igrati. Dizajneri blisko surađuju sa programerima i umjetnicima koji rade na igri kako bi zajednički otkrili koji bi bio najbolji smjer za igru sa vizualne i sistemske strane, to jest koje od ideja je moguće izvesti u praksi a koje nije. Prema Schell igrini dizajneri trebaju imati jako veliki niz različitih sposobnosti i neke od njih su menadžment, psihologija, kreativno pisanje, animacija, kinematografija, arhitektura i mnoge druge. Razlog tome je što je dizajnerov glavni posao da kreira igrino iskustvo koje će igrač proživljavati prilikom igranja (Schell 2008). U ovome procesu veliku ulogu ima igrin dizajn dokument. Dizajn dokument predstavlja bibliju za sve uključene u razvoj igre jer su unutar njega definirane sve bitne stvari vezane za njen razvoj i jednom kada su sve specifikacije definirane razvojni tim ih mora pratiti kako bi vizija za igru na kojoj rade bila ispunjena. Dizajn dokument je također „živi“ dokument koji konstantno evoluira što znači da se u njega nerijetko nadodaju razni konceptni crteži i specifikacije za nove sustave koji komplementiraju postojećima. Dizajn dokumenti obično u sebi sadrže tekstualne opise, slike, konceptne crteže, dijagrame i razne ostale medije koji su potrebni da tim ima jasnu viziju o tome kakvu igru stvaraju. Ti dokumenti su većinom samo za internu upotrebu i samim time su klasificirani kao tajna, što znači da su jako rijetko dani u javnost za sve zainteresirane da ih pročitaju. Dizajn dokument nema

nikakav određeni standard vezano za to kako ga treba pisati i u kojemu ga programu treba izrađivati, ta odluka ovisi o razvojnom timu. Za potrebe CodeBreaker projekta izrađen je Microsoft Word dokument u kojemu su se nalazili svi potrebni opisi, dijagrami i koncepti, jedna stranica CodeBreaker, koji se u početku zvao Operation Code, dizajn dokumenta prikazana je na sljedećoj slici (slika 18).



OPERATION CODE

Design document

Possible names: Project CodeBreaker,Code.break

What is it

-puzzle game that teaches you basics of programming

Goals, vision, scope

-game should teach the players the basics of programming in an interactive way wich should be more interesting than the usual teaching methods

-the goal is that the player with no previous programming knowledge should be able to go through the game and in the end have a grasp on its basic concepts

-scope – game should have at 30+ minutes worth of content

Concepts that should be taught

-variables and how variable types work

-functions

-classess

-inheritance

-polymorphism

-encapsulation

-loops

-conditions

-arrays

-algorithms

-interface

-templates*

-complexities*

Slika 18 – Jedna od stranica CodeBreaker dizajn dokumenta

Izvor: Vlastita izrada

Prikazana stranica predstavlja prvu stranicu dokumenta i na njoj se nalaze predložena imena za završnu verziju igre, kratko objašnjenje kakav je to tip igre,

navedeni su ciljevi, vizija i opseg projekta. Zadnja navedena stvar su koncepti koji bi bili predstavljeni igračima, koncepti označeni sa zvjezdicom su bili opcionalni jer tada nije bilo poznato da li će ih biti moguće izvesti sa sistemske strane.

3.3 Igrin svijet i priča

Jedna od prvih stvari koje je trebalo definirati bila je tematika igrinog svijeta, to jest gdje će se radnja igre odvijati i kakav bi vizualni stil trebala igra imati. Za ovu tematiku postojale su tri glavne ideje. Prva za svijet bila je da se igrina radnja odvija unutar simulacije koja se raspada te kako igrač prolazi kroz igru tako svijet dolazi u sve bolje ili lošije stanje ovisno o tome u kojem smjeru bi bilo odlučeno odvesti priču. Iduća ideja bilo je da igra bude smještena unutar nekakvoga laboratorija za testiranje koji bi mogao biti kombinacija interijernih i eksterijernih razina. Zadnja od ideja je bio jednostavni virtualni svijet koji bi u sebi sadržavao okoliše različitih tematika. Završna verzija igrinog svijeta predstavljala je hibrid svih ovih ideja. Završna verzija igrinog svijeta predstavlja simulaciju koja se sastoji od nekoliko različitih tematskih okruženja poput laboratorija za testiranje, parka pored jezera, razine sa otočićima i ruševinama. Prva polovica igre sastoji se od razina koje se nalaze u eksterijeru dok u drugoj polovici se sve razine nalaze u interijeru. Neka od tih okruženja pate od korupcije podataka (engl. data corruption), pa zbog toga okruženja pod tim utjecajem imaju „pogrešne“ teksture i fizika na njima ne funkcionira ispravno. Kako igrač napreduje kroz igru tako okruženja postaju sve normalnija. Primjer okoliša pod utjecajem korupcije u završnoj verziji igre pokazan je na sljedećoj slici (slika 19).



Slika 19 – Primjer igrinog svijeta pod utjecajem korupcije

Izvor: Vlastita izrada

Sada kada su bile poznate ideje vezane za izgled i tematiku igrinog svijeta bilo je potrebno osmisliti igrinu priču i potencijalne likove koji će se pojavljivati unutar igre. Postojale su dvije glavne ideje za priču, prva ideja bila je da je igračev lik poslan u igrin svijet od strane neke organizacije kako bi otkrio zašto se simulacija raspada, te da je pokuša popraviti. Ta ideja je naposljetku bila odbačena, te je osmišljena druga ideja koja je naposljetku postala igrina glavna priča.

Igrač se probudi unutar igrinog svijeta bez ikakvih sjećanja i znanja o tome što se događa u igrinom svijetu. Kako igrač prolazi kroz igru rješavajući zagonetke i učeći nove koncepte tako postepeno otkriva što se događa čitajući razne tekstove koji se nalaze unutar igre, te promatrajući kako se igrin svijet mijenja tijekom njegovog putovanja. Dakle igrin svijet je simulacija koja je napravljena od strane tvrtke Advanced Security Solutions kako bi koristeći testne avatare, koji izgledaju kao android, njihovi programeri mogli razvijati i testirati razne sigurnosne mjere. Nakon nekoga vremena simulacija se počela kvariti te je napuštena i igračev lik se tada probudi u tom pokvarenom uništenom svijetu. S obzirom da je to svijet dizajniran za testiranje svaka razina predstavlja neko testno područje čiji objekti imaju izloženi kod s kojim igrač mora vršiti interakciju kako bi mogao napredovati unutar tih razina.

Što se tiče likova postojalo je nekoliko ideja za likove koji bi se pojavljivali unutar igre poput AI pomagaća koji bi pričao s igračem i pomogao mu pri rješavanju zagonetki, zatim je postojala ideja za lika koji bi pričao s igračem i davao mu naredbe

o tome što mora raditi. Postojala je ideja o dodavanju lika koji bi igraču služio kao učitelj koji bi ga učio nove programerske koncepte i pomagao mu tijekom njegova putovanja igrinim svijetom. Jedina ideja koja je od samoga početka ostala ista je vezana za glavnog lika, naime igračev lik u igri je android koji se koristio za testiranje koji dobije mogućnost da hakira okoliš tako što na početku igre skine potrebne protokole za to. Na kraju je donesena odluka kako unutar igre neće postojati nikakvih drugih likova osim igrača zbog toga što je odlučeno kako će fokus biti na izradi igrinih sustava i zagonetki umjesto na razvoj likova i pisanje njihovih dijaloga. Kako bi se kompenziralo za nedostatak likova unutar igre je postavljen cijeli niz tekstova koje igrač može čitati. Ti tekstovi daju dodatne informacije o igrinom svijetu i referenciraju osobe koje su radile na pojedinim testnim područjima, te razne događaje koji su popratili izradu tih područja. Primjer takvoga teksta prikazan je na sljedećoj slici (slika 20).



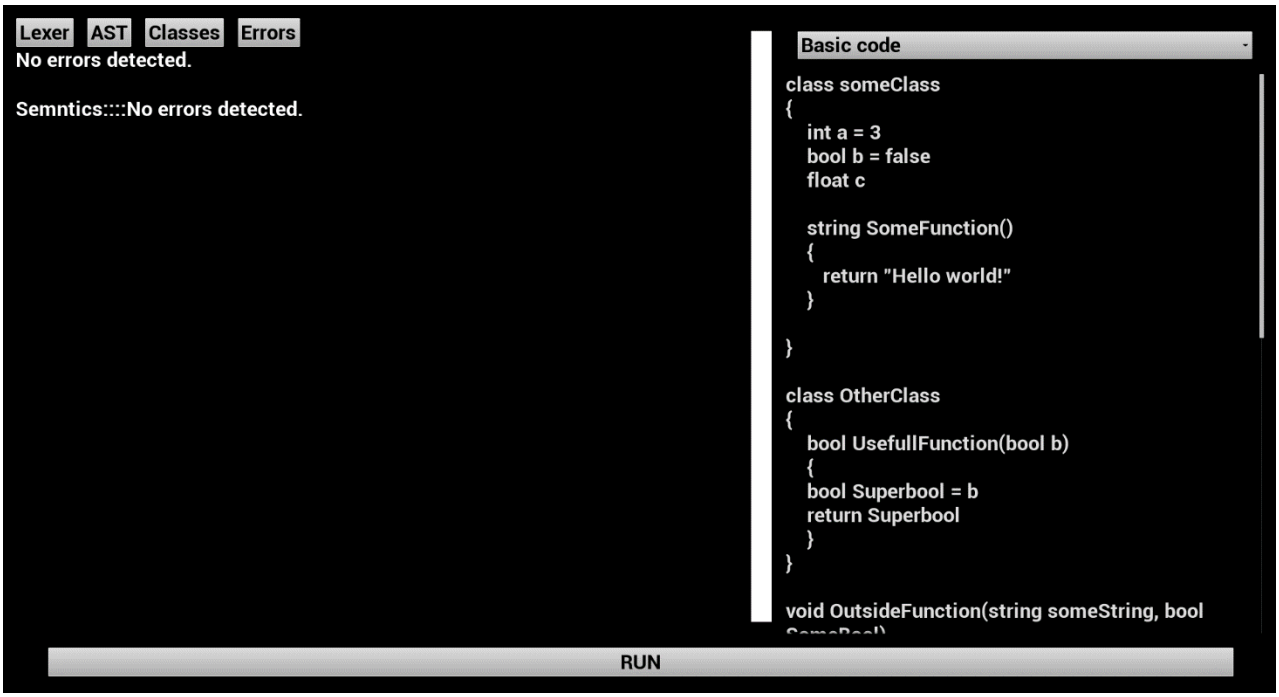
Slika 20 – Primjer teksta sa pozadinskom pričom

Izvor: Vlastita izrada

3.4 Dizajn osnovnih igrinih sustava

Velika većina igara se vrti oko svojih sustava i loše dizajniran sustav može imati značajan utjecaj na igrinu završnu kvalitetu. Zbog te činjenice igrini developeri moraju jasno definirati na kakav način žele da igrači igraju njihovu igru, to jest potrebno je definirati igrinu petlju igranja (engl. gameplay loop) koja definira koji niz aktivnosti će igrač ponavljati tijekom igranja. Glavna stvar koja je bila poznata za CodeBreaker je da će igrači moći hakirati razne objekte iz njihovog okruženja kako bi mogli uspješno rješavati zagonetke. Sa time na umu bilo je potrebno razviti sustav hakiranja i pisanja koda. Za potrebe ove igre razvijen je programski jezik koji predstavlja hibrid programskih jezika C++, C# i Python. Sustav hakiranja omogućava igraču da pritiskom na tipku za hakiranje unutar određenog objekta upiše svoj vlasiti

kod, taj kod je potom preveden u strojni jezik i pokrenut. Ukoliko je kod neispravan korisnik dobiva poruku da u njegovom kodu postoji greška, a ukoliko je kod ispravan i ispunjava zadane uvjete za rješavanje zagonetke, igrač otvara daljnji put kroz razinu. Rana verzija igrinog sustava hakiranja prikazana je na sljedećoj slici (slika 21).

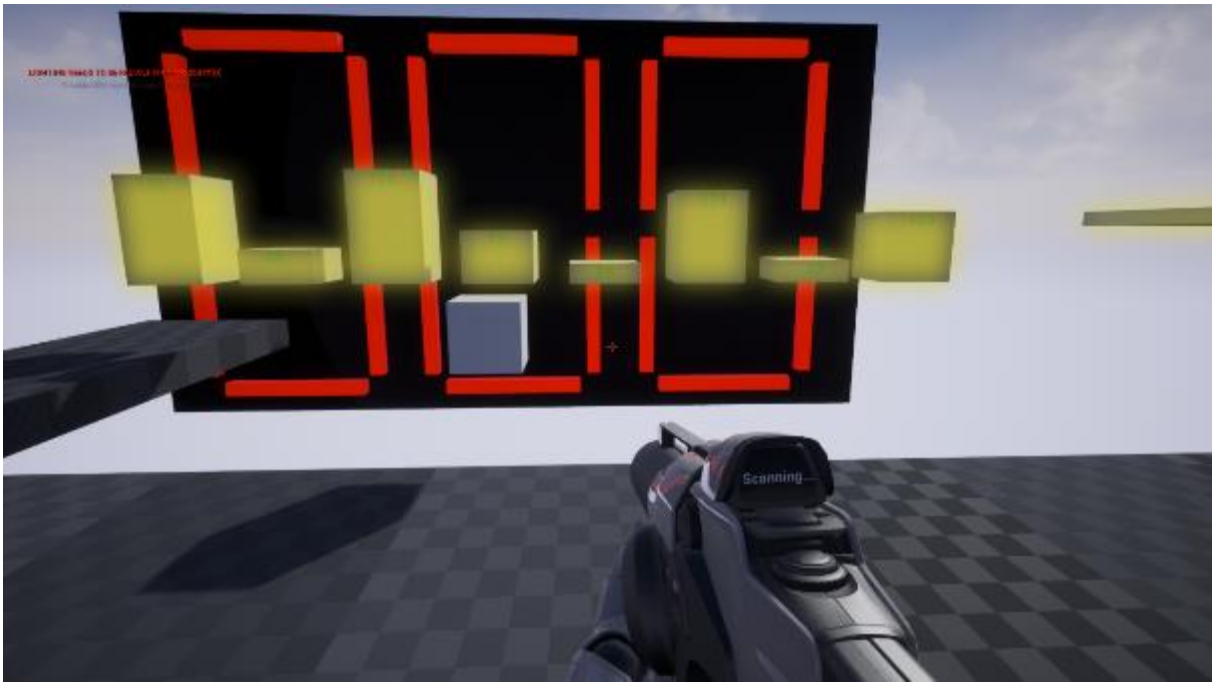


Slika 21 - Prva verzija sustava za pisanje koda

Izvor: Vlastita izrada

Sljedeći problem koji je trebalo riješiti je bilo kako igraču dati do znanja sa kojim objektima može vršiti interakciju i hakirati ih, a sa kojima ne može. Ovdje je postojalo nekoliko ideja, prva takva ideja bila je da igrač ima pristup popisu svih objekata unutar razine koje može hakirati. Kod te ideje problem je bio u činjenici da igrač nije mogao znati točna imena svih objekata u svom okruženju, pa bi zbog toga morao nagađati kako se određeni objekt zove. Rješenje koje je bilo osmišljeno za taj problem bio bi sustav koji bi igraču prikazao imena svih objekata iznad samih objekata kada bi igrač pritisnuo tipku za aktivaciju toga sustava. Na kraju to rješenje nije bilo iskorišteno zbog činjenice da bi tekstovi potencijalno pokrivali veliki dio ekrana i blokirali igračev pogled što bi moglo biti iritantno. Rješenje svih tih problem postao je sustav koji je nazvan HackerVision. Igrač bi pritiskom odgovarajuće tipke aktivirao HackerVision koji bi mu tada osvijetlio na ekranu sve objekte s kojima može

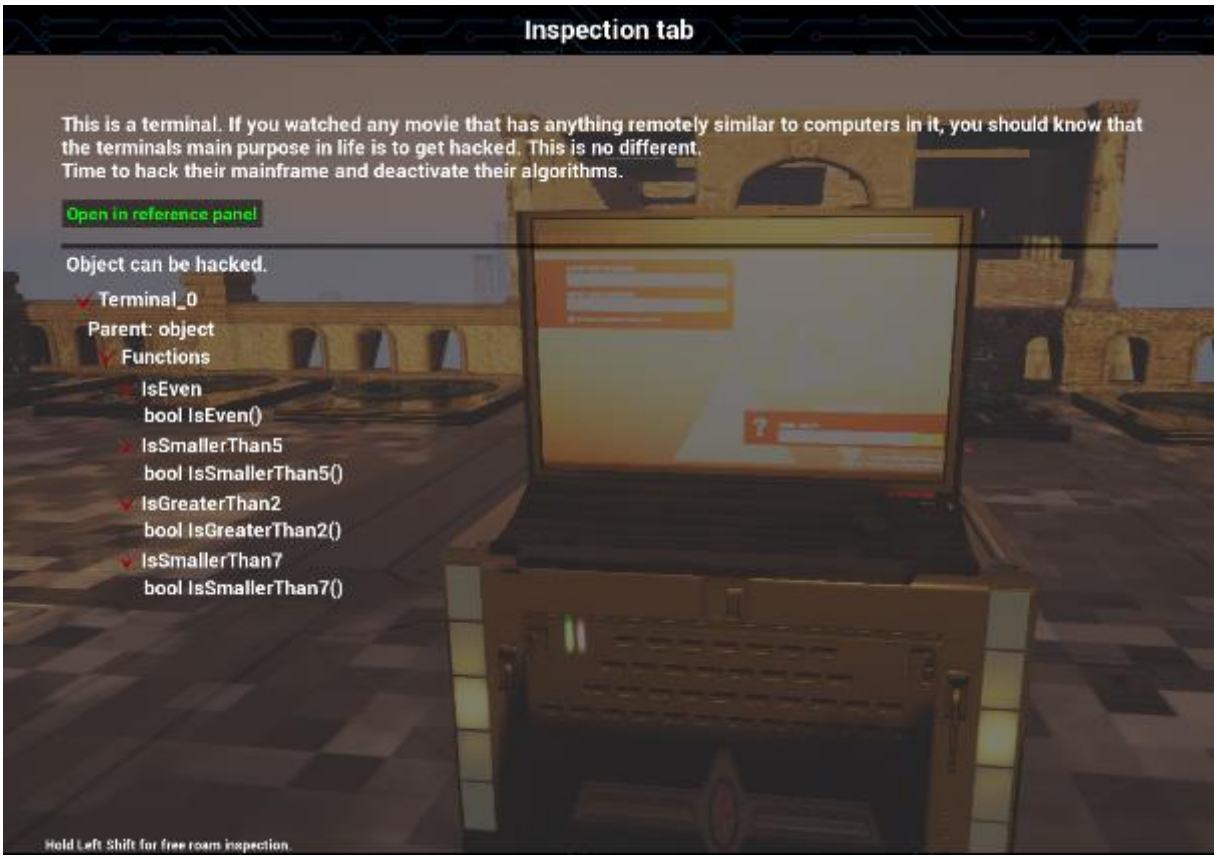
vršiti interakciju. Konceptni dizajn je najprije napravljen u Adobe Photoshopu, te je taj koncept pokazan na sljedećoj slici (slika 22).



Slika 22 – HackerVision konceptni dizajn
Izvor: Vlastita izrada

Napravljeni koncept je potom dorađen i implementiran, te je na koncu postao jedan od glavnih igrinih sustava. HackerVision je ovime postao elegantno rješenje za problem nedostatka informacija o tome s kojima se objektima može vršiti interakcija.

Zadnji veliki problem koji je bilo potrebno riješiti je na koji način igraču dati do znanja na kakve načine može vršiti interakciju sa objektima uz pomoć koda. Bilo je potrebno pronaći rješenje koje igraču može na jednostavan način dati sve potrebne informacije, slično poput ideje iza HackerVisiona. Rješenje toga problema bilo je stvaranje sustava inspekcije (engl. inspection). Sustav inspekcije omogućava igraču da pritiskom na potrebnu tipku vidi sve potrebne informacije o objektu s kojime želi vrišiti interakciju. Te informacije su opis objekta, poveznica za detaljniji opis objekta ili koncepta unutar igrine baze znanja i sve njegove funckionalnosti koje se mogu koristiti, taj sustav je prikazan na sljedećoj slici (slika 23).



Slika 23 – Sustav inspekcije

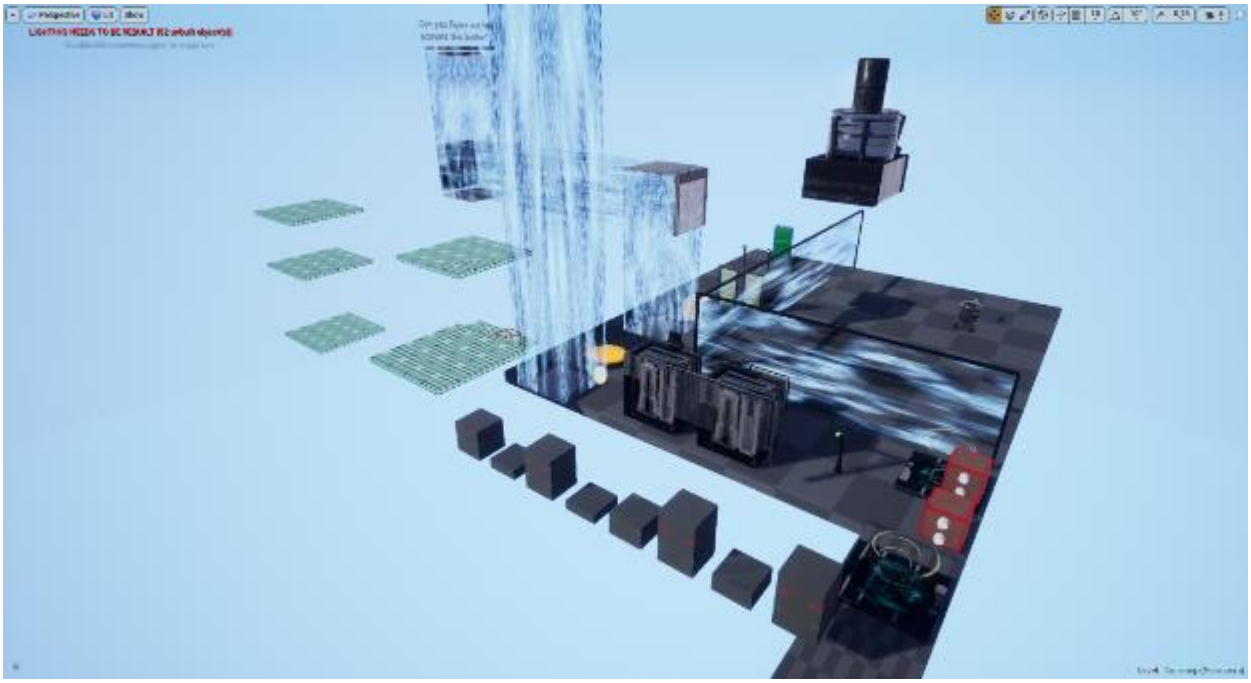
Izvor: Vlastita izrada

Kreiranjem ova tri osnovna sustava automatski je također i stvorena osnovna igrina petlja igranja. Igrač najprije uz pomoću HackerVisiona identificira objekte u svome okruženju s kojima može vrišiti interakciju, potom koristeći sustav inspekcije dobiva dodatne informacije o tome objektu, te na kraju uz pomoću sustava hakiranja pokušava riješiti zagonetku.

Tijekom igrinog razvoja došlo je do potrebe da se stvore pomoćni sustavi za igrača kako ne bi došlo do toga da igrač zapne na nekoj zagonetci zbog toga što se ne može sjetiti nekoga od prije naučenih koncepata. Za te potrebe razvijen je panel sa referencama na sve prijašnje naučene koncepte, taj panel predstavlja bazu programerskog znanja koju igrač može koristiti u bilo kojemu trenutku kako bi naučio nešto novo ili se podsjetio na prijašnje naučene koncepte. Idući sustav koji je uveden kao pomoć za igrača je sustav pomoćnih poruka, Hint sustav. Ovaj sustav za svaku zagonetku ima određeni broj poruka koje se prema potrebi daju igraču kako bi mu se pomoglo u rješavanju zagonetki. S obzirom da je jedan od ciljeva bio da se što više smanje frustracije igrača pri rješavanju zagonetki također je uveden sustav koji

igračima omogućava da preskoče zagonetke na kojima su beznadno zapeli jednostavnim pritiskom na tipku za automatsko rješavanje zagonetke.

Potrebno je također napomenuti da su svi novi sustavi i funckionalnosti nakon implementacije ponajprije testirani na testnoj mapi, čiji je izgled prikazan na sljedećoj slici (slika 24).



Slika 24 – Mapa za testiranje

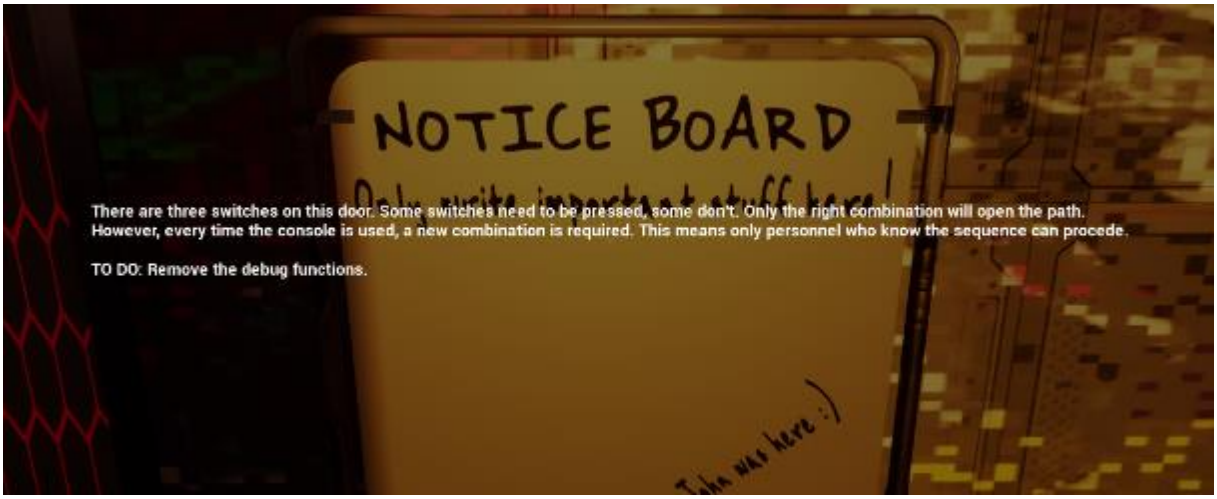
Izvor: Vlastita izrada

Testna mapa ima jako jednostavan vizualni dizajn i služi kako bi se razni novi sustavi i funkcionalnosti mogli testirati u kontroliranom okruženju. Ukoliko se prilikom testiranja naiđe na neke bugove ili neispravna ponašanja, ta odstupanja od željenog ponašanja se ispravljaju i ponovno se provodi proces testiranja. Nakon što je ustanovljeno da sustav ili funckionalnost ispravno rade, tada ih se počinje koristiti pri izradi igrinih pravih razina.

3.5 Dizajn zagonetki i proces učenja

S obzirom da je CodeBreaker igra koja je bazirana na rješavanju problema kako bi se napredovalo, jako je bitno napraviti te probleme zanimljivima za rješavanje. Prema rječima kreatora popularnih puzzle igara The Witness, Braid i Sokobond, napraviti dobru puzzle igru je jako teško i ne postoji nikakav specifičan

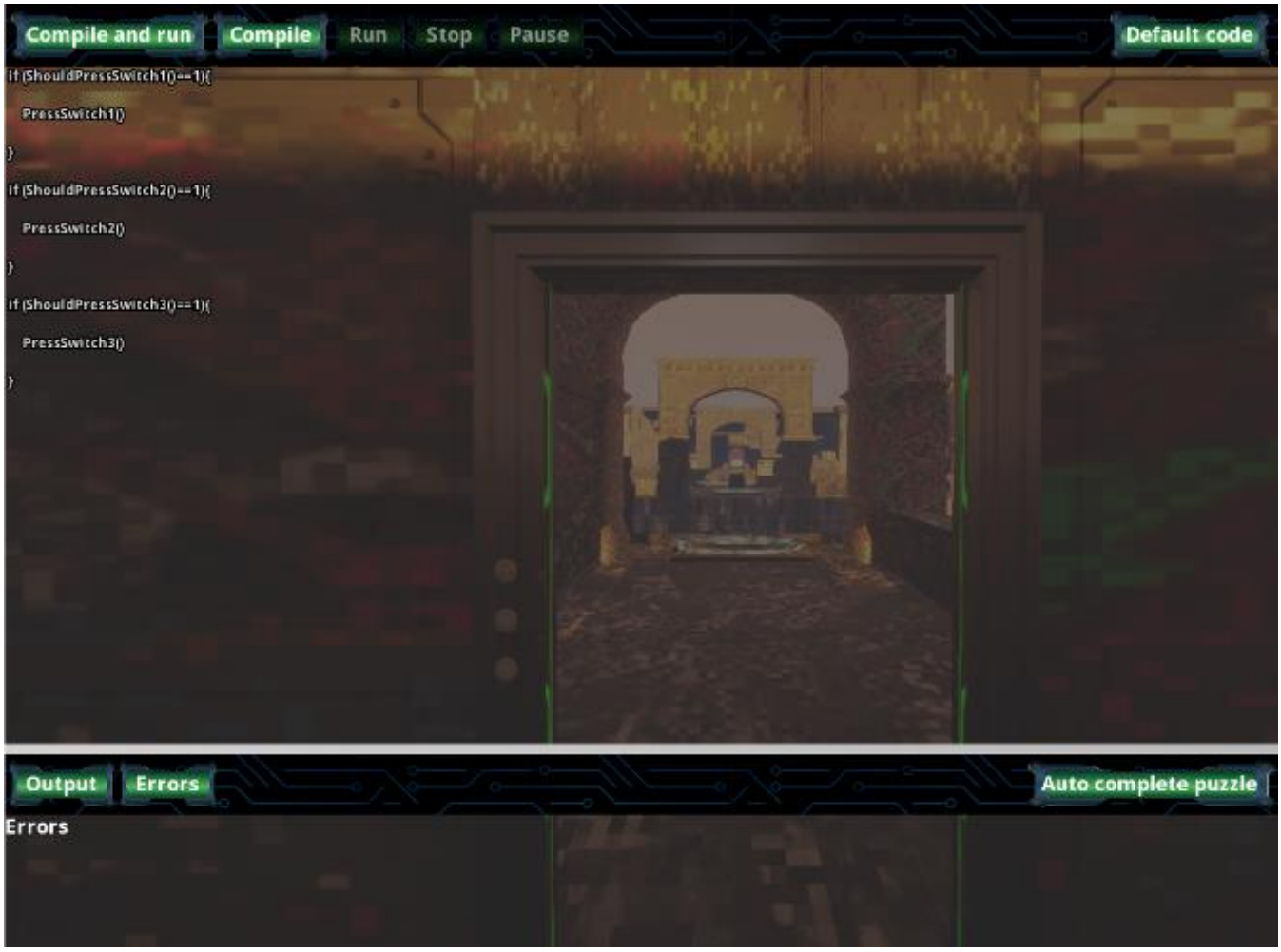
recept za uspjeh (Davies 2015). Činjenica da je CodeBreaker puzzle igra čiji je cilj da od igrača napravi programera čini izradu zanimljivih zagonetki za nju još težim. Glavni cilj prilikom dizajniranja tih zagonetki je bio da budu zanimljive za rješavanje i da postepeno igrača uče nove koncepte i tjeraju ga da te nove koncepte koristi u kombinaciji sa onima koje je prije naučio. Dakle fokus bi trebao biti na procesu učenja. U igri CodeBreaker zagonetke su postavljene na sljedeći način: igrač uđe u zonu koja sadrži zagonetku i mora uspješno riješiti tu zagonetku kako bi mu se otvorio put dalje. Zagonetka je nerijetko opisana uz pomoć igrinog objekta koji je nazvan oglasna ploča (engl. notice board) na kojemu je definiran zadatak kojega igrač treba riješiti. Primjer takvoga opisa prikazan je na sljedećoj slici (slika 25).



Slika 25 – Opis jedne od igrinih zagonetki
Izvor: Vlastita izrada

Na ovoj slici vidi se opis prve zagonetke sa razine koja se bavi uvjetnim grananjima. Čim započne ovu razinu igrač je obaviješten da sada ima pristup IF funkcionalnosti za uvjetno grananje čime mu se daje do znanja kako bi tu funkcionalnost trebao koristiti za rješavanje ove zagonetke. Gore prikazana slika objašnjava igraču kako se na vratima nalaze tri prekidača i da je potrebno otkriti potrebnu kombinaciju kako bi se vrata otvorila. Inspekcijom tih vrata igrač može vidjeti da na raspolaganju ima dvije funkcije s kojima može raditi, a to su ShouldPressSwitch() i PressSwitch(). Ukoliko prouči što koja funkcija radi može vidjeti da će mu ShouldPressSwitch() funkcija dati do znanja koji od prekidača treba biti pritisnut, a PressSwitch() funkcija će pritisnuti željeni gumb. U tome trenutku igrač

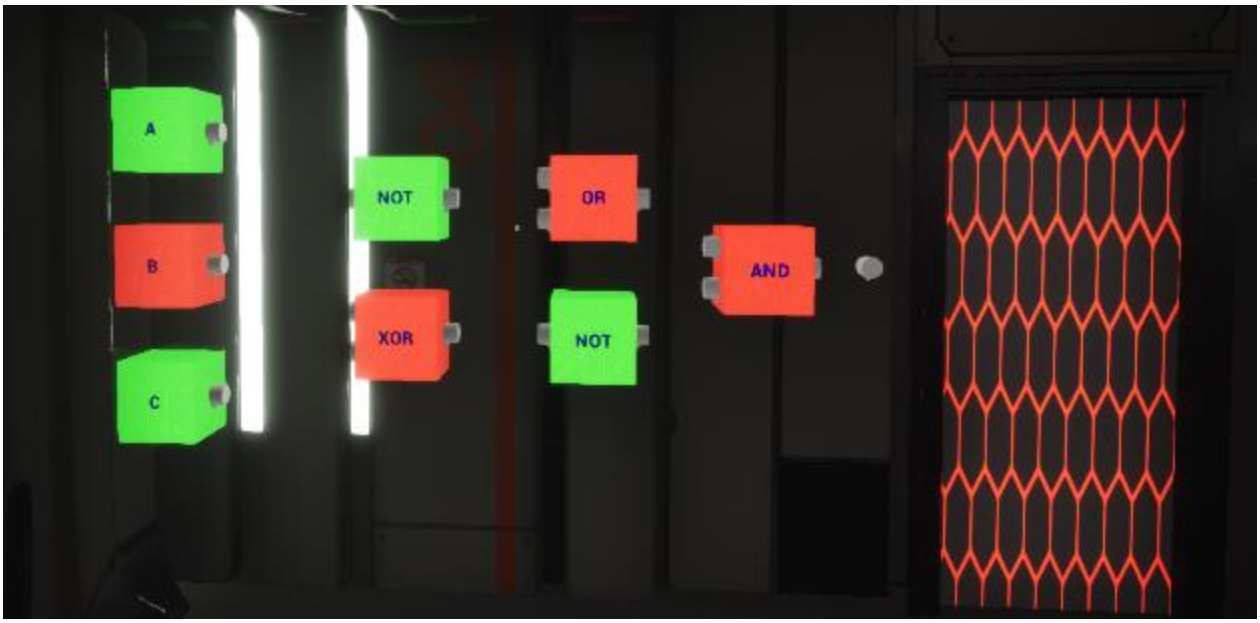
ima sve potrebne informacije kako bi riješio zagonetku i otvorio vrata. Rješenje te zagonetke prikazano je na sljedećoj slici (slika 26).



Slika 26 – Rješenje opisane zagonetke
Izvor: Vlastita izrada

Dakle rješenje problema je bilo koristiti IF uvjetno grananje kako bi se otkrilo koji od gumba je potrebno stisnuti i na temelju povratnog rezultata te funkcije odlučiti da li stisnuti određeni gumb. Ukoliko je igrač smatrao da zagonetka nije dovoljno dobro objašnjena ili je jednostavno nije razumio mogao je koristiti prije navedeni Hint sustav kako bi dobio poruke koje bi mu dodatno objasnile kako riješiti ovu zagonetku.

Razine u ovoj igri su većinom podjeljene na nekoliko sekcija kako bi se proces učenja igrača mogao provesti u nekoliko koraka. Tako primjerice na prvoj razini prva stvar koja se igrača uči su osnove logičkog programiranja. Primjer ove zagonetke prikazan je na sljedećoj slici (26).



Slika 26 – Primjer zagonetke sa logičkim programiranjem

Izvor: Vlastita izrada

Ovdje se od igrača traži da međusobno poveže logičke čvorove na ispravan način kako bi otvorio vrata. Zagonetka na slici predstavlja zadnju zagonetku toga tipa i pokazuje igračevu kreativnost i razumjevanje tematike s obzirom da za ovu zagonetku postoji nekoliko različitih rješenja. Tek kada je rješavanjem nekoliko takvih zagonetki dokazao da razumije kako logičko programiranje funkcionira ga se počinje učiti pravom programiranju, počevši od korištenja jednostavne funkcije za ispisivanje rečenice „Hello, World!“. Ovakav način učenja gdje se igraču postepeno predstavljaju novi koncepti se koristi za sve zagonetke koje se nalaze u ovoj igri.

Tijekom igranja ove igre preko zagonetki se ovim redoslijedom igrača uče sljedeći programerski koncepti:

- Logičko programiranje
- Pozivanje funkcija
- Varijable i tipovi podataka
- Uvjetna grananja – if, else, ugnježđeni if

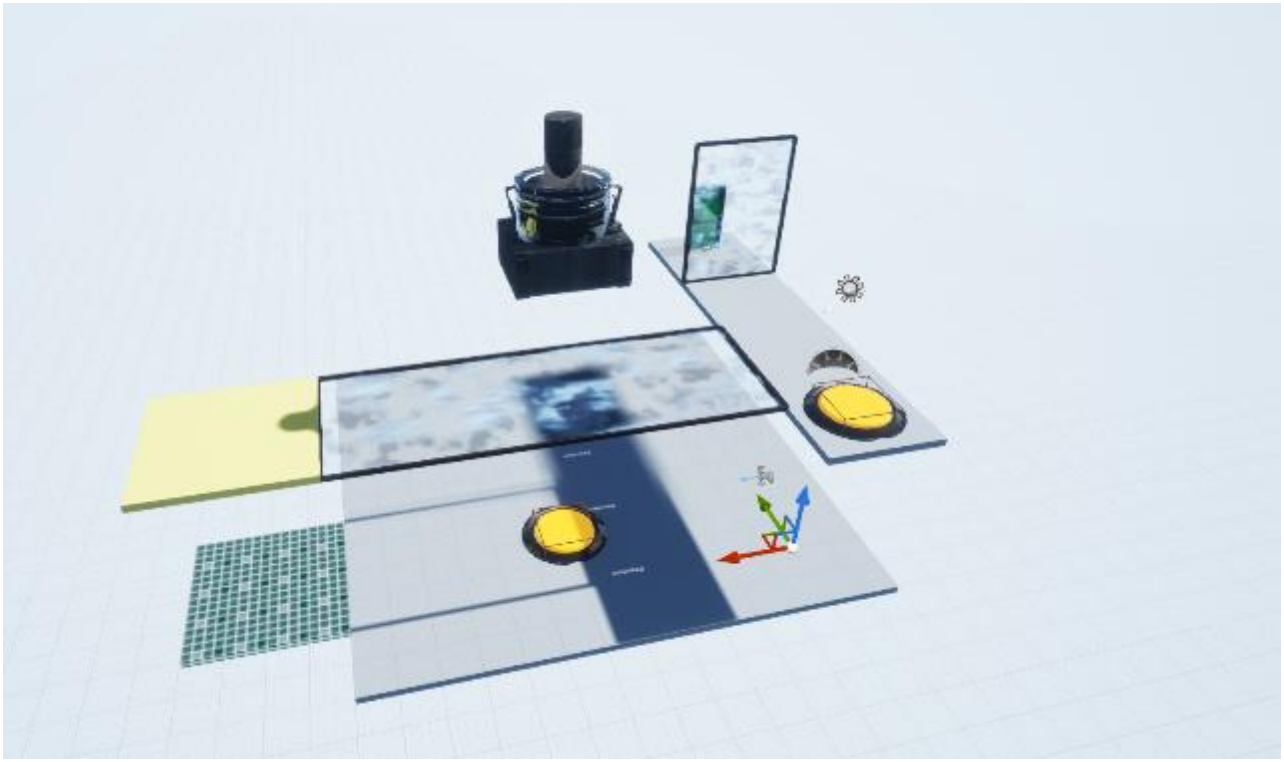
- Petlje – for, do while, while
- Polja
- Izrada funkcija i korištenje funkcija
- Izrada i korištenje klasa
- Nasljeđivanje
- Korištenje referenci

Neke od koncepata koji su trebali biti presdstavljeni igraču nije bilo moguće predstaviti u obliku zagonetki. Zbog toga su se ti koncepti igraču predstavljeni u obliku interaktivnih primjera. Koncepti koji su na takav način predstavljeni igraču su sljedeći:

- Binarno pretraživanje
- Bubble sort
- Quick sort
- Insertion sort
- Selection sort
- Vektori
- Operatori
- Konstante
- Scope
- Pointeri

3.6 Dizajn razina

Kada je poznata igrina tematika i priča, te kada su dizajnirani osnovni igrini sustavi potrebno je započeti rad na igrinim razinama. Uz igrine zagonetke, razine predstavljaju skoro podjednako bitan dio igre zbog toga što igrač predstavlja dio svijeta i potrebno je taj svijet dizajnirati na zanimljiv način. Prije nego što je započeto za izradom igrinih stvarnih razina, napravljeno je nekoliko prototipnih razina, jedna od njih prikazana je sljedećom slikom (slika 27).

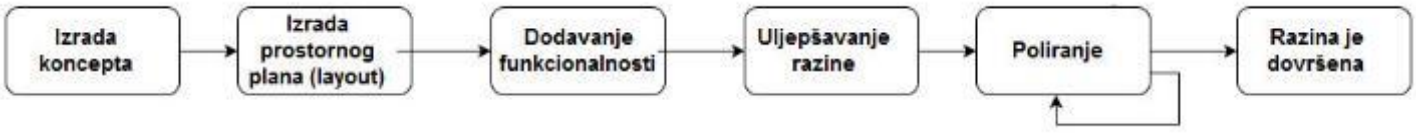


Slika 27 – Izgled prototipne razine

Izvor: Vlastita izrada

Prototipne razine su prema funckionalnosti i izgledu slične kao i prije navedena testna razina. Svrha prototipnih razina bila je izrada različitih zagonetki kako bi se vidjelo da li bi funkcionirale unutar sustava razvijenih za ovu igru. Na ovaj način bilo je moguće napraviti i testirati velik broj ideja za izradu zagonetki, što je naposljetku bilo korisno pri izradi igrinih finalnih razina.

Razine za ovu igru kreirane su iterativnom metodologijom dizajniranja. Iterativan pristup znači da je dizajn razina podjeljen na nekoliko koraka, to jest postoji nekoliko iteracija neke razine prije nego što je ona finalizirana. Navedeni proces prikazan je dijagramom na sljedećoj slici (slika 28).



Slika 28 – Dijagram iterativnog pristupa izradi razina

Izvor: Vlastita izrada

Kao što je na dijagramu vidljivo prvi korak u iterativnom procesu je izrada koncepata za igrine razine. Ponajprije je potrebno ustanoviti koliko je razina za igru potrebno napraviti. Nakon što je poznat broj razina potrebno je ukratko definirati koji bi programerski koncepti trebali biti pojašnjeni igraču na pojedinim razinama, te ustanoviti vizualnu tematiku pojedinih razina. Ukoliko razvojni tim sadrži umjetnika u ovome se koraku izrađuju konceptni crteži izgleda tih razina. Pri izradi CodeBreaker-a unutar ovoga koraka ustanovljeno je kako će igra sadržavati dva glavna tipa razina. Prvi tip razina predstavljaju standardne razine unutar kojih igrač riješava zagonetke kako bi napredovao. Drugi tip razina su tako zvane centralne razine (engl. hub) koje imaju sufiks Nexus, izgled jedne od tih Nexus razina prikazan je na sljedećoj slici (slika 29).



Slika 29 – Igrina prva Nexus razina

Izvor: Vlastita izrada

Unutar tih razina igrač može pomoću portala započeti igranje iduće potrebne razine ili ukoliko to želi, može ponovno odigrati neku od prije pređenih razina. Prelaskom potrebnog broja razina otvara se put do igrine iduće Nexus razine i novog seta razina sa zagonetkama. Unutar ovoga koraka ustanovljeno je kako će igra sadržavati tri Nexusa i devet razina sa zagonetkama čime smo došli do završnog broja od 12 razina.

Nakon što su izrađene konceptne ideje za sve razine potrebno je započeti sa izradom tih razina. Unutar ovoga koraka izrađuje se prostorni plan (engl. layout) za sve razine. Ovdje se određuje veličina razine, način na koji će se igrač njome kretati i definiraju se mjesta na kojima će se nalaziti zagonetke uz kratke opise navedenih zagonetki. Unutar ovoga koraka se mape ne uljepšavaju, te se uglavnom koriste osnovni privremeni modeli (engl. placeholders) poput kocke i pravokutnika kako bi se reprezentirao objekt koji će se na to mjesto nadodati u budućim iteracijama. Izgled takve iteracije razine za učenje petlji prikazan je na sljedećoj slici (slika 30).



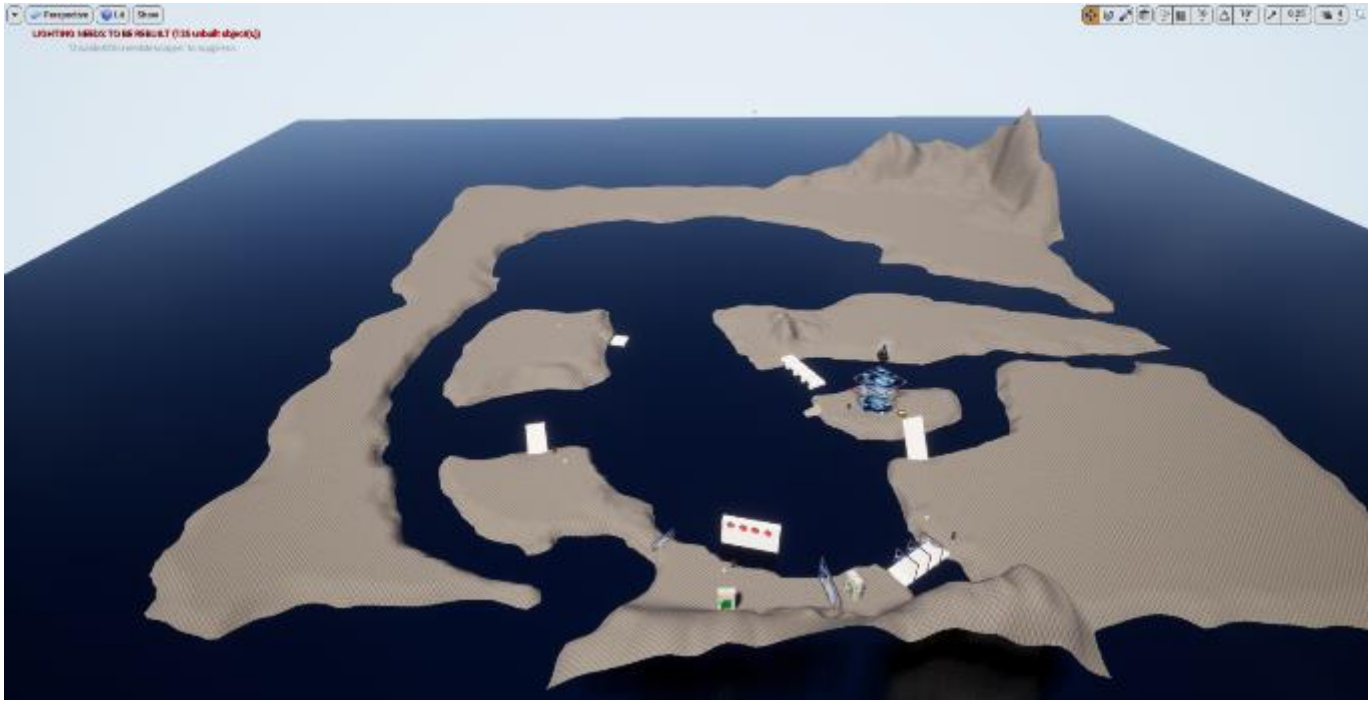
Slika 30 – Prva iteracija razine za učenje petlji

Izvor: Vlastita izrada

Dakle prikazana razina je napravljena za predstavljanje koncepta petlji. Unutar konceptnog koraka navedeno je da će se ova razina nalaziti u eksterijeru. Osnovna ideja bilo je da se u sredini mape nalazi veliko jezero, te da tijekom prelaska razine igrač radi krug oko toga jezera (engl. loop), što je tematski prikladno za ovu razinu. Kao što je vidljivo sa slike ta ideja je bila očuvana u odnosu na koncept te su nad njome napravljene neke modifikacije. Bijeli pravokutnici na slici predstavljaju mostove i platforme koje igrač koristi kako bi navigirao kroz razinu.

Kada su dovršeni prostorni planovi za sve razine ulazi se u idući korak dizajniranja razina, a to je dodavanje funkcionalnosti. Unutar ovoga koraka dodaju se

zagonetke i sve potrebne funkcionalnosti na razine. Cilj ovoga koraka je da se nakon dodavanja svih potrebnih funkcionalnosti razina može odigrati od početka do kraja bez ikakvih većih problema. Izgled mape nakon takve iteracije prikazan je sljedećom slikom (slika 31).

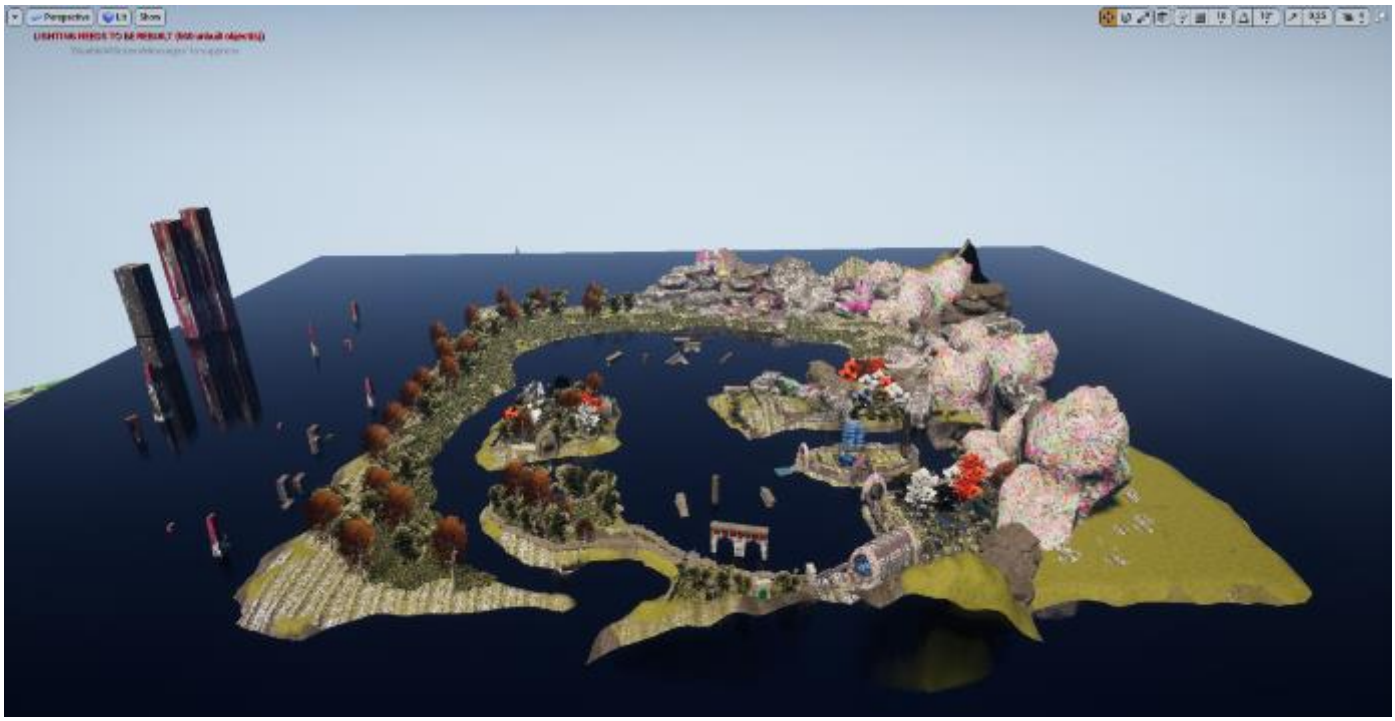


Slika 31 – Druga iteracija razine za učenje petlji
Izvor: Vlastita izrada

Unutar ovoga koraka na navedenu razinu su implementirane zagonetke i svi potrebni elementi kako bi se te zagonetke mogle uspješno izvršiti. Nadodani elementi su bili portal za početak i kraj razine, energetska polja koja igraču blokiraju napredak dok ih ne onesposobi i terminali koje igrač mora hakirati kako bi mogao napredovati. Prije postavljenim mostovima i platformama su također nadodane animacije koje se pokreću kada su određeni uvjeti zadovoljeni.

Nakon što je dodavanje funkcionalnosti na sve razine dovršeno dolazi se do koraka unutar kojega se razine dotjeravaju korištenjem pripremljenih tekstura i 3D modela, te se ovdje također podešava osvijetljenje i dodaju potrebni efekti. Prilikom uljepšavanja razina potrebno je na umu imati prije napravljeni koncept za svaku od njih kako bi tematski imale smisla. Unutar ovoga koraka se također provodi prvi optimizacijski prolaz. Optizacija se vrši tako da se kroz razinu prolazi sa upaljenim

mjeračem za broj sličica u sekundi (engl. fps counter) i svaki put kada dođe do većega pada broja sličica se to područje analizira korištenjem raznih alata koji se nalaze unutar Unreal Engine-a. Nakon što se provede analiza ti se problemi rješeavaju na potreban način i optimizacijski prolaz se nastavlja. Izgled ove iteracije razine prikazan je na sljedećoj slici (slika 32).



Slika 32 – Treća iteracija razine za učenje petlji

Izvor: Vlastita izrada

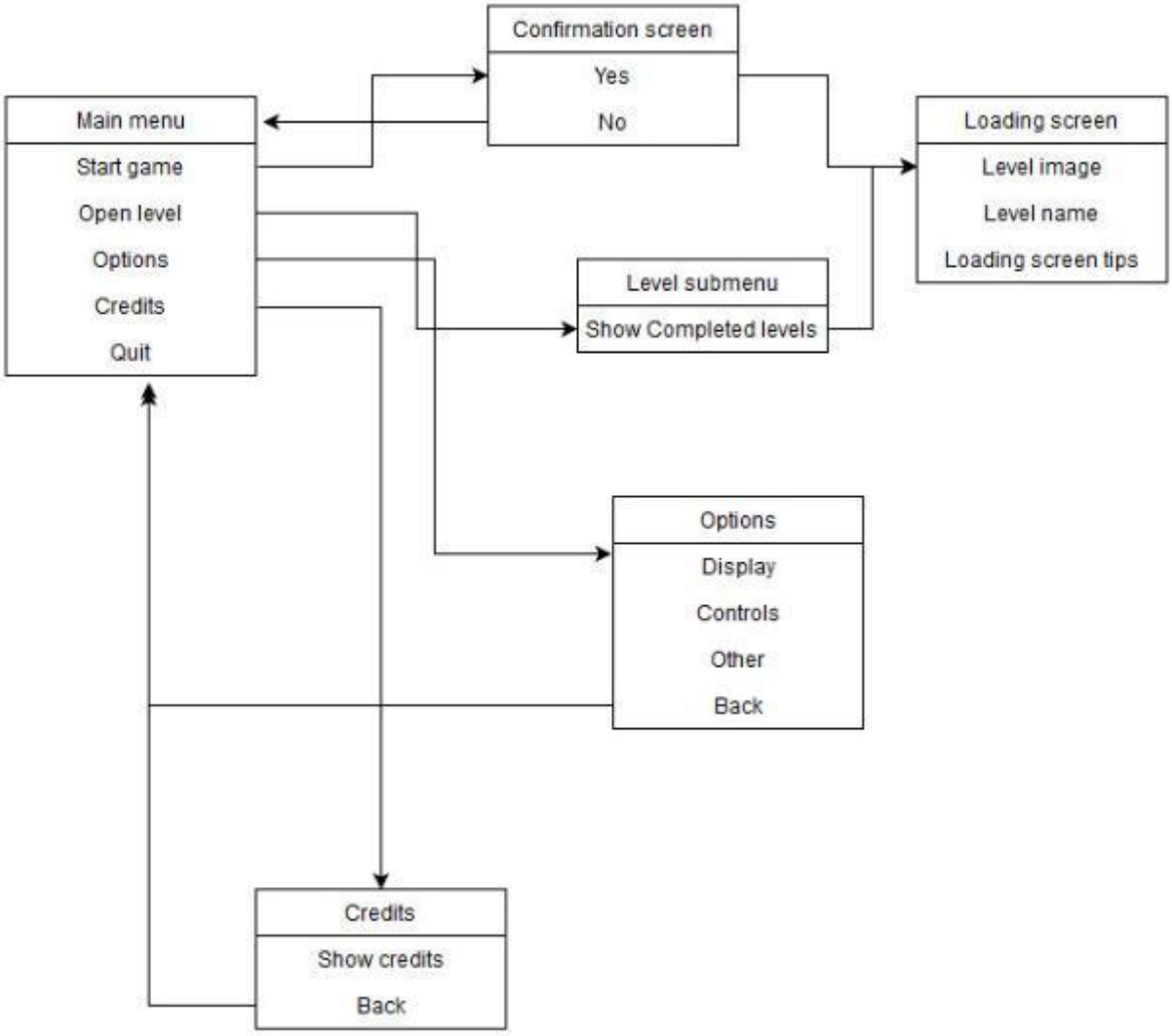
U ovome koraku na prikazanu razinu je nadodano puno zelenila i drveća jer je tematika ove razine bio vanjski park sa velikom količinom flore. Osim toga nadodano je kamenje, kameni objekti te mnogo manjih detalja poput lišća koje leti zrakom.

Zadnji korak prije nego što je razina u potpunosti dovršena je poliranje. U ovome koraku se mnogo puta razine odigravaju te se rješavaju svi problemi na koje se naiđe prilikom tih odigravanja. Neki od tih problema mogu biti zagonetke koje ne funkcioniraju kako je zamišljeno, problemi sa optimizacijom, ukoliko igrač propadne kroz igrin svijet ili prolazi kroz predmete kroz koje ne bi smio itd. Unutar ovoga koraka se također prema potrebi dorađuje vizualna strana razina i prema potrebi se dorađuju zagonetke bez da se mjenjaju osnovne ideje iza njihovoga dizajna. Ovaj se korak nerijetko izvršava više od jednoga puta zbog toga što se često nakon prolaza koji bi trebao biti zadnji pronađe neki problem koji nije bio pronađen na prijašnjemu prolasku ili je uveden kada je popravljen neki prijašnji problem. Tek kada je razvojni tim zadovoljan sa razinom, ili kada ostanu bez vremena za daljnja poboljšanja, je razina proglašena dovršenom.

3.7 Dizajn korisničkog sučelja

Zadnja bitna stvar do koje se dolazi kod dizajniranja igre je dizajn korisničkog sučelja. Unutar svake igre igrači provode veliku količinu vremena koristeći igrina korisnička sučelja, bili oni svjesni te činjenice ili ne. Dizajn korisničkog sučelja je često stvar koja se ostavlja za sam kraj razvojnog ciklusa zbog toga što mnogi projektni menadžeri smatraju da je izrada korisničkog sučelja jednostavan proces, te da ga bilo tko može na brzinu napraviti. Takav pristup je velika pogreška koja dođe do izražaja čim igrači dobiju pristup igri i vide ružno i neupotrebljivo korisničko sučelje. Na koncu se dolazi do toga da je teško prodati igru koja loše izgleda (Fox 2005). Dakle na korisničkom sučelju potrebno je raditi od samoga početka te vršiti izmjene u skladu sa izmjenama koje se događaju tijekom igrinog razvoja. Kod izrade korisničkog sučelja potrebno je na umu imati dvije stvari, sučelje ne smije ružno izgledati i sučelje mora biti jednostavno i intuitivno za korištenje. Razlog tomu je da ukoliko sučelje nije atraktivno za gledanje i ako ga je naporno koristiti, igrač će htjeti vršiti što manje interakcije s njime i potencijalno bi mogao u potpunosti odustati od igranja.

Prva stvar koju će igrač vidjeti kod pokretanja igre je igrin glavni meni. Zbog te činjenice potrebno je glavni meni učinit atraktivnim, jednostavnim za navigaciju, te treba sadržavati sve potrebne funkcionalnosti. Za te potrebe napravljen je UML dijagram sa sljedeće slike (slika 33).



Slika 33 – UML dijagram sa funckionalnostima glavnoga menija

Izvor: Vlastita izrada

Na prikazanom UML dijagramu prikazani su entiteti koji označavaju različite menije i dijelove korisničkog sučelja. Za svaki od entiteta definirane su njihove funkcionalnosti koje u nekim slučajevima se nalaze unutar gumbova menija. Ovakav se postupak ponavlja za sve manje igrine menije dok su za one kompleksnije potrebne drugačije metode s obzirom da sa povećanjem broja entiteta, dijagram postaje sve manje čitak. Nakon što su ustanovljene osnovne funkcionalnosti, u Adobe Photoshopu je napravljeno nekoliko koncepata za glavni meni, te je naposljetku jedan od njih odabran, dorađen i implementiran. Izgled jednoga od izrađenih koncepata prikazan je sljedećom slikom (slika 34).



Slika 34 – Konceptni dizajn glavnog menija

Izvor: Vlastita izrada

Nakon glavnoga menija bilo je potrebno redizajnirati korisničko sučelje za sustav koje je nazvan konzola, s kojom će igrač provoditi jako veliku količinu vremena. Rad na tome redizajnu započeo je kada je igra bila u svom prototipnom stadiju sa samo jednom razinom koja se sastojala od kontroliranja platformi pomoću koda. Tadašnji izgled konzole vidljiv je na sljedećoj slici (slika 35).



Slika 35 – Konzolino koriničko sučelje tijekom prototipiranja igre

Izvor: Vlastita izrada

Kao što je vidljivo sa slike tadašnji dizajn sučelja konzole je bio jako ružan i neupotrebljiv. Zbog toga je bilo potrebno što ranije početi raditi na njegovom redizajnu. S obzirom je ovdje riječ o jako kompleksnom korisničkom sučelju nije bilo moguće napraviti dijagram za njega već je u Photoshopu napravljen cijeli niz konceptnih slika. Jedan od tih koncepata prikazan je na sljedećoj slici (slika 36)



Slika 36 – Konceptni dizajn izgleda korisničkog sučelja za konzolu

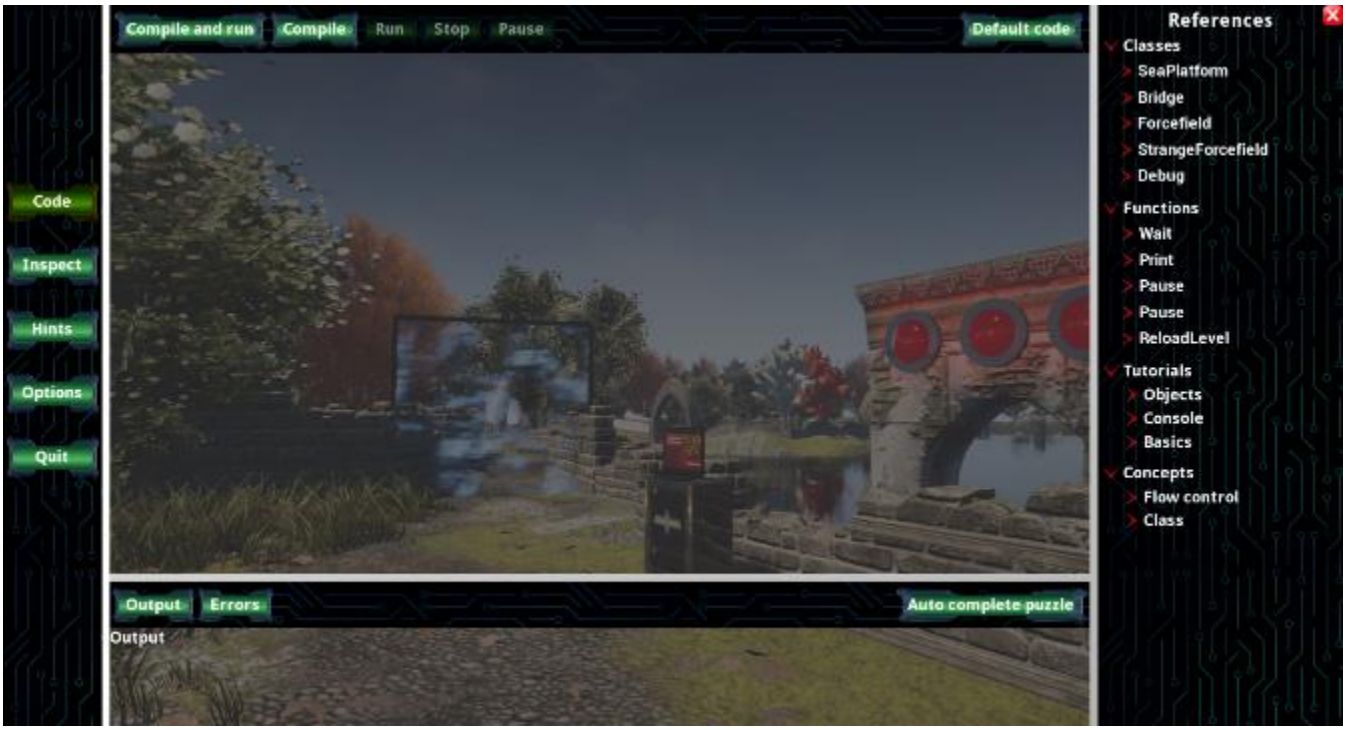
Izvor: Vlastita izrada

Te su konceptne slike potom dodane unutar jednoga dokumenta u kojemu su sve funkcionalnosti pojedinih dijelova sučelja bile detaljno objašnjene. Nakon što su na konceptima napravljene potrebne izmjene došlo je vrijeme za implementaciju tog korisničkog sučelja unutar Unreal Engine-ovog UMG (Unreal Motion Graphics) sustava. Na sljedećoj slici je vidljiv rezultat te implementacije (slika 37).



Slika 37 – Prva implementacija redizajniranog sučelja za konzolu
Izvor: Vlastita izrada

Redizajnirana verzija korisničkog sučelja za konzolu je definitivno bilo veliko poboljšanje u odnosu na prijašnju verziju, ali još uvijek nije bila dovoljno dobra, vizualno nije bila atraktivna i nedostajalo joj je funkcionalnosti. Ova verzija je međutim predstavljala jako dobar temelj na kojemu se moglo dalje graditi konzolino sučelje. Tijekom igrinog razvoja izgled konzole je prošao kroz mnogo iteracija i njena završna verzija vidljiva je na idućoj slici (slika 38).



Slika 38 – Završni izgled sučelja za konzolu

Izvor: Vlastita izrada

Kao što je vidljivo sa slike završna verzija sučelja je vizualno puno atraktivnija u odnosu na prijašnje verzije. Kao što je prije napomenuto osnovni dizajn je predstavljao temelj za izradu završnog, što je vidljivo iz činjenice da se sam dizajn nije drastično mijenjao već je samo bio konstantno nadograđivan kroz cijeli niz iteracija tijekom igrinog razvoja. Što se tiče funkcionalnosti i lakoće korištenja za tu informaciju je potrebno pogledati rezultate korisničke ankete koji će biti prikazani u jednom od sljedećih poglavlja.

4.Korisnička dokumentacija

Prijašnje poglavlje je na detaljniji način pokrilo razne igrine sustave, dok će se ovo baviti prikazivanjem i objašnjavanjem načina na koji bi prosječni igrač koristio te igrine sustave.

4.1 Pokretanje igre

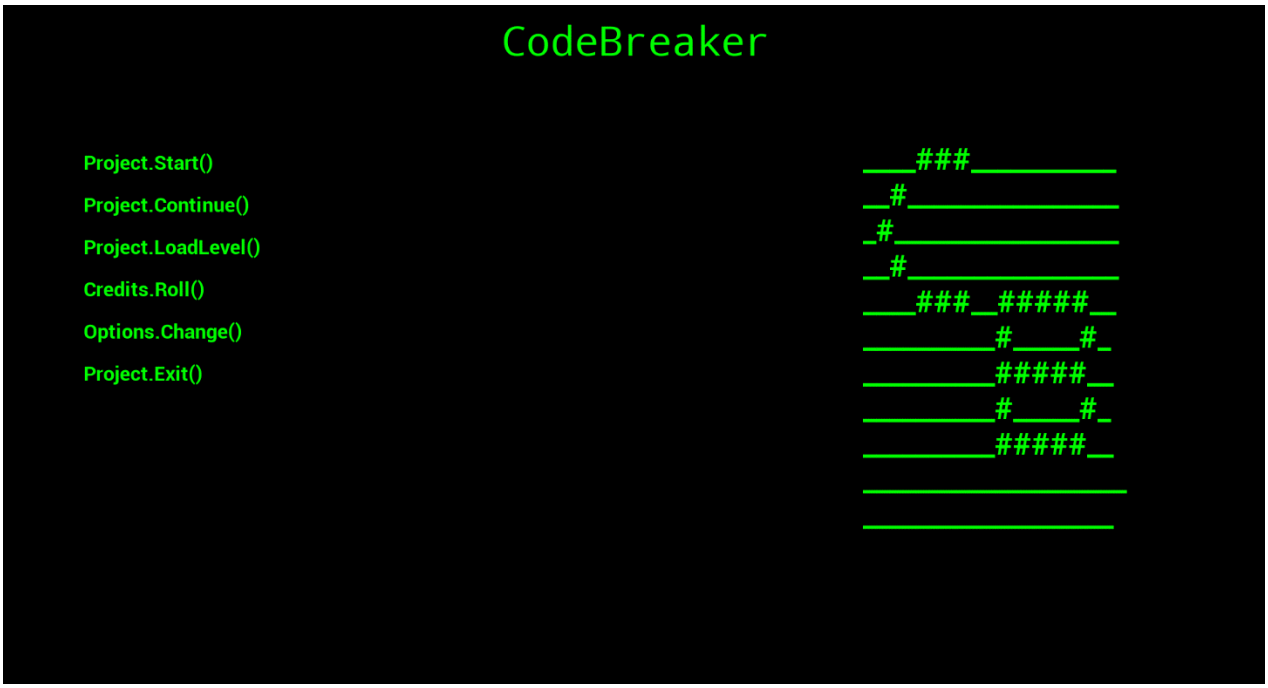
Prilikom pokretanja igre igrač je pozdravljen sa uvodnom animacijom koja odmah daje do znanja kako je tematika ove igre interakcija sa računalima i pisanje koda. Završni rezultat ove animacije je pokazan na sljedećoj slici (slika 39), potrebno je također napomenuti činjenicu da je moguće preskočiti tu animaciju ukoliko igrač je igrač nema želju gledati.



Slika 39 – Rezultat igrine uvodne animacije

Izvor: Vlastita izrada

Nakon što igrač pritisne bilo koju tipku na prikazanom ekranu otvara se igrin glavni meni koji je prikazan sljedećom slikom (slika 40).



Slika 40 – Završni izgled igrinog glavnog menija

Izvor: Vlastita izrada

Glavni meni ima simplistički dizajn za koji je zamišljeno da imitra korisničko sučelje zastarjelog računalnog terminala. Na desnoj strani menija je animacija koja konstantno ispisuje igrin ascii logo, dok se s lijeve strane nalaze gumbi od toga menija.

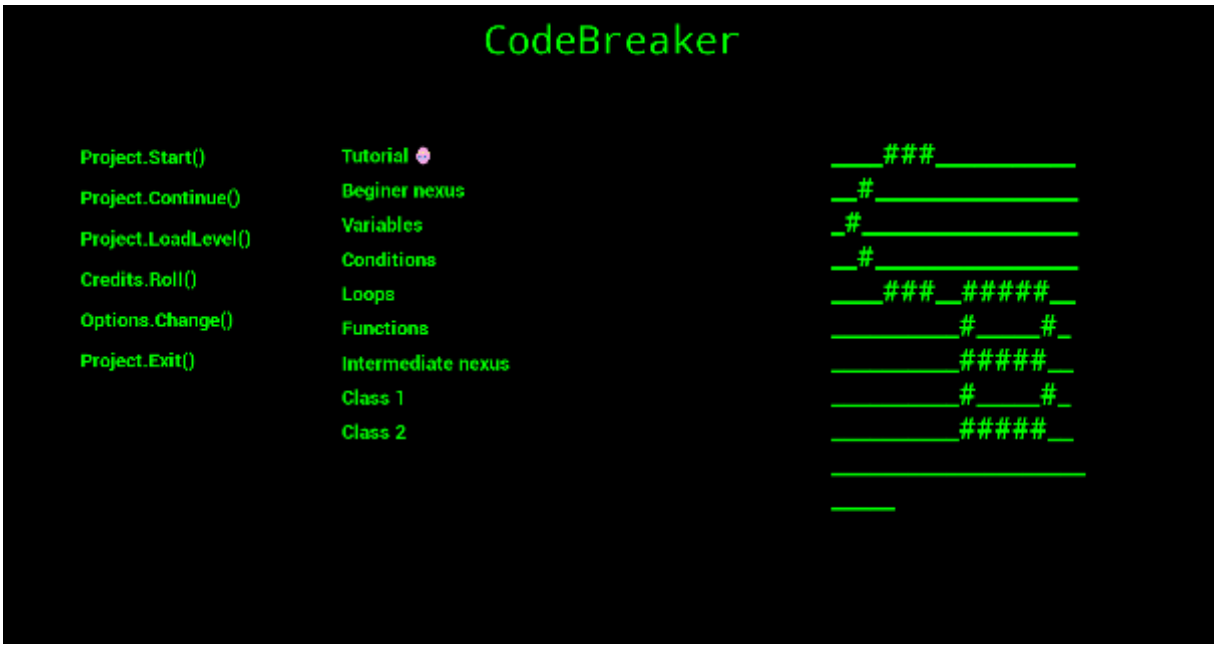
4.1.1 Project.Start() i Project.Continue()

Prtiskom na gumb Project.Start() igrač započinje igru i to mu je jedini način započinjanja igre kada prvi puta pokrene aplikaciju. Ukoliko je igrač već igrao igru ranije i ima spremljene prijeđene razine tada pritiskom na ovaj gumb igrač dobiva upozorenje da će mu, ukoliko nastavi, sve dosadašnje razine koje je prešao biti izbrisane, te da će morati cijelu igru odigrati iznova. Ukoliko igrač nema spremljenih razina gumb Project.Continue() će biti zatamnjen i neće ga biti moguće pritisnuti. Ukoliko igrač ima spremljene razine pritiskom na ovaj gumb biti će otvorena zadnja razina na kojoj se igrač nalazio prije nego što je zadnji put izašao iz igre.

4.1.2 Project.LoadLevel()

Kao i što je bio slučaj sa ProjectContinue() gumbom, ovaj gumb je također zatamnjen ukoliko igrač nije odigrao niti jednu igrinu razinu. Ukoliko je igrač odigrao

barem jednu igrinu razinu pritiskom na ovaj gumb se otvara lista svih razina kojima igrač ima pristup. Svaka od razina ima na sebi skriveno jedno uskršnje jaje koje igrač može pokupiti, te se pored naziva svake od razina se pojavljuje ikona uskršnjeg jaja ukoliko ga je igrač pokupio na toj razini. Pritiskom na naziv željene razine ta će razina biti otvorena nakon kraćega učitavanja. Izgled toga podmenija je prikazan sljedećom slikom (slika 41).



Slika 41 – Izgled Project.LoadLevel() podmenija

Izvor: Vlastita izrada

4.1.3 Options.Change()

Pritiskom na gumb Options.Change() gumb se otvara meni koji igračima omogućava da izmjene neke od igrinih opcija. Opcije koje igrač može izmijeniti su grafičke, poput rezolucije i kvalitete igrine grafike, izmjena igrinih kontrola na način koji igračima više odgovaraju i naopsljetku zadnja opcija omogućava igraču da izmjeni igrinu svijetlinu, glasnoću zvukova i upali mogućnost preskakanja razina. Nakon pritiska na gumb automatski se otvara meni sa grafičkim opcijama i igrač može pritiskom na odgovarajuće gumbe otvoriti ostale kategorije. Izgled menija sa opcijama prikazan je na sljedećoj slici (slika 42).



Slika 42 – Izgled menija sa opcijama

Izvor: Vlastita izrada

4.1.4 Credits.Roll() i Project.Exit()

Pritiskom na gumb Credits.Roll() otvara se ekran sa imenima osoba koje su sudjelovale pri izradi igre, te su navedene njihove uloge u igrinoj kreaciji. Pritiskom na gumb Project.Exit() igra se zatvara.

4.2 Igranje igre

Prilikom prvog pokretanja igre, igrač započinje igranje na igrinoj Tutorial razini koja je dizajnirana kako bi ga naučila sve osnovne igrine koncepte i da bi ga se upoznalo sa predviđenim načinom igranja. Uz pomoć uputa koje se pojavljuju u gornjem desnom kutu ekrana igraču se postepeno objašnjava način igranja. Prva takva poruka koju igrač dobije prikazana je na sljedećoj slici (slika 43).



Slika 43 – Igrina prva tutorial poruka
Izvor: Vlastita izrada

Osim za tutorial poruke ovakav način obavještavanja igrač se koristi i kada nauči neki novi koncept o kojemu može čitati u svojoj konzoli. Unutar ovoga tutoriala se igrača uči o važnosti HackerVision sustava, koji je prikazan sljedećoj slikom (slika 44) ,o tome kako da vrši interakciju sa igrinim objektima i upute za korištenje konzole.



Slika 44 – Završni izgled HackerVision sustava
Izvor: Vlastita izrada

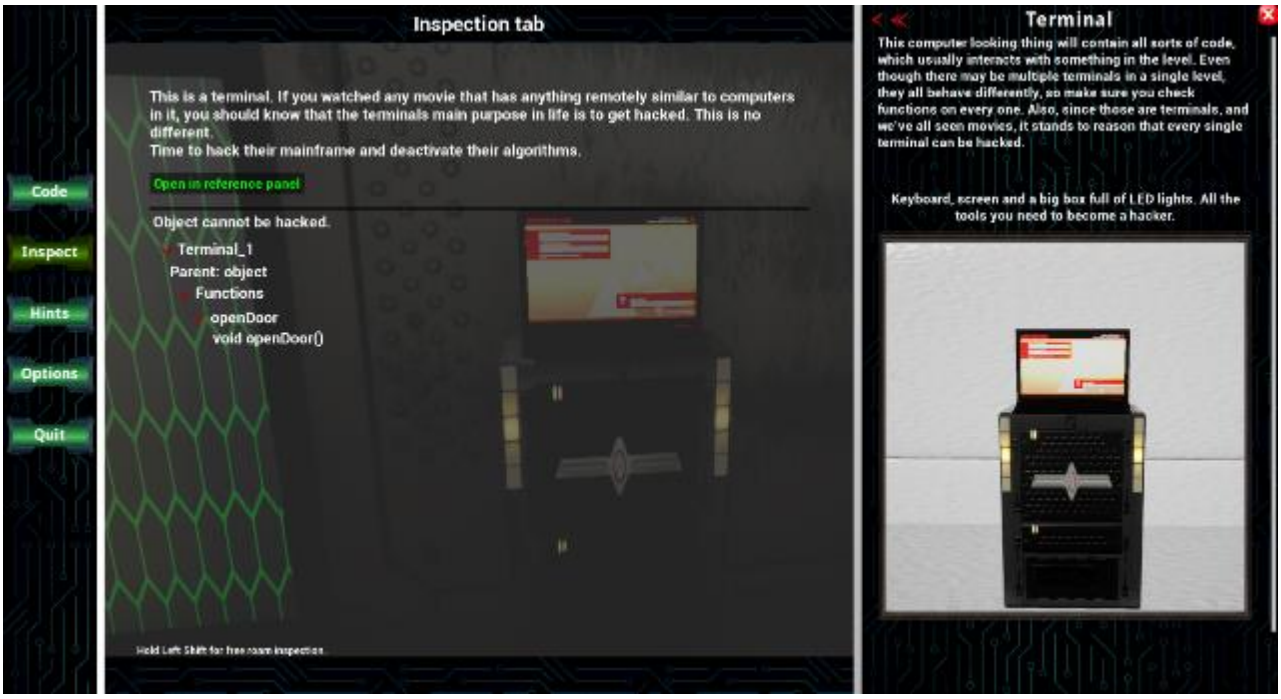
4.2.1 Interakcija sa objektima

Unutar igre CodeBreaker igrač može vršiti tri tipa interakcije sa određenim igrinim objektima. Igrač može otkriti s kojim objektima i na kakav način može vršiti interakciju sa objektima na dva različita načina, od kojih je prvi korištenjem prije navedenog i objašnjenog HackerVision sustava. Drugi način na koji to može otkriti je tako što gledanjem u određene objekte se u gornjem lijevom kutu pojavljuje jedna ili više ikona koje igraču daju do znanja kakav tip interakcije može vršiti sa promatranim objektom. Prvi takav oblik interakcije je inspekcija i ukoliko igrač pogleda u objekt nad kojime se može vršiti inspekcija, tada će se u gornjemu lijevom kutu prikazati ikona za inspekciju, što je prikazano sljedećom slikom (slika 45).



Slika 45 – Objekt nad kojim se može vršiti inspekcija
Izvor: Vlastita izrada

Ukoliko je nad objektom moguće vršiti inspekciju igrač može pritisnuti tipku za inspekciju te će biti otvoren prozor sa informacijama o gledanom objektu. Ovdje igrač ima pristup generalnom opisu toga objekta i ako objekt unutar sebe ima neke funkcije igrač ih ovdje može vidjeti. Inspekcijski prozor za gore prikazani objekt prikazan je sljedećom slikom (slika 46).



Slika 46 – Inspekcijski prozor

Izvor: Vlastita izrada

Idući način interakcije predstavlja klasična interakcija. U klasični oblik interakcije spadaju akcije poput pritiskanja gumbiju i korištenja kablova u zagonetci sa logičkim vratima. S obzirom da je nad praktički svakim objektom s kojim je moguće vršiti ovaj tip interakcije također i moguće vršiti inspekciju, gledanjem prema objektu s kojim je moguće vršiti ovakvu interakciju će biti prikazane ikone za inspekciju i interakciju. Izgled tih dviju ikona vidljiv je na sljedećoj slici (slika 47).



Slika 47 – Objekt s kojim je moguće vršiti klasičnu interakciju
Izvor: Vlastita izrada

Zadnji oblik interakcije koji igrač može vršiti sa nekim objektom je hakiranje. Hakiranje predstavlja igrinu glavnu mehaniku i njenim korištenjem igrač prelazi veliku većinu igrinih zagonetki. Kao što je prije navedeno, unutar ove igre moguće je vršiti inspekciju nad velikom količinom objekata te će se tako više manji svaki objekt koji je moguće hakirati također moći koristiti i za inspekciju. Ikone koje se pojavljuju pri promatranju objekta koji se može hakirati su prikazane sljedećom slikom (slika 48).



Slika 48 – Objekt koji je moguće hakirati

Izvor: Vlastita izrada

Pritiskom gumba za hakiranje prilikom gledanja u smjeru objekta kojega je moguće hakirati otvara se konzola, točnije prozor za pisanje koda. Unutar toga prozora za pisanje koda će se ponekad nalaziti funkcije i komentari vezani za hakirani objekt, dok će ponekad taj prostor biti u potpunosti prazan i na igraču je da u potpunosti samostalno napiše svoj vlastiti kod. Osim koda igrač sa desne strane može vidjeti sve funkcije koje je moguće pozivati prilikom rada sa tim specifičnim objektom. Izgled toga prozora za hakiranje prikazan je na sljedećoj slici (slika 49).



Slika 49 – Izgled prozora za hakiranje objekta

Izvor: Vlastita izrada

4.2.2 Korištenje konzole

Nakon što su pojašnjeni igrini glavni sustavi potrebno je objasniti možda i najvažniji element s kojim će igrač vršiti interakciju, a to je konzola. Prilikom igranja igrač u bilo kojemu trenutku može pritisnuti gumb za otvaranje konzole, koja predstavlja njegov glavni alat za rješavanje zagonetki. Na sljedećoj slici (slika 50) prikazan je osnovni izgled konzole, te su na njoj crvenom bojom označeni njezini glavni dijelovi.



Slika 50 – Glavni dijelovi konzole

Izvor: Vlastita izrada

Kao što je vidljivo na slici sučelje konzole se sastoji od tri glavna dijela. Dio označen sa brojem jedan predstavlja bočni meni (engl. side bar) i korištenjem gumbiju koji se nalaze na njemu moguće je pristupiti konzolinim prozorima sa različitim funkcionalnostima. Kada se konzola prvi puta otvori automatski se otvara prozor za pisanje koda, koji bi se inače otvorio pritiskom na gumb sa nazivom Code. Pritiskom na gumb Inspect otvara se prozor za inspekciju koji izgleda identično poput onoga prikazanog na jednoj od prijašnjih slika (slika 46). Ukoliko prilikom pritiska na tipku Inspect igrač ne promatra objekt nad kojime je moguće vršiti inspekciju, tada otvoreni prozor bude u potpunosti prazan. Pritiskom na gumb Hints otvara se prozor na kojemu je moguće zatražiti pomoć pri rješavanju određene zagonetke i pročitati prijašnje poruke. Izgled Hint prozora prikazan je na sljedećoj slici (slika 51)



Slika 51 – Izgled Hints Prozora

Izvor: Vlastita izrada

Ostali gumbi koji se nalaze na tom meniju su Options i Quit. Pritiskom na gumb s nazivom Options otvara prozor sa opcijama koji izgleda poput prozora koji je opisan u ranijem dijelu rada, te je tamo i prikazan slikom (slika 42). Naposljetku dolazimo do Quit gumba čijim pritiskom se igrač vraća na igrin glavni meni.

Prostor koji je na prijašnjoj slici (slika 50) označen sa brojem dva nazvan je Code Area, to jest prostor za kodiranje. Taj se prostor sastoji od tri glavna dijela, a to su zaglavlje u kojemu se nalaze gumbi, centralni prostor za pisanje koda i prostor za ispis poruka. Unutar zaglavlja se nalaze sljedeći gumbi:

- Compile and run – omogućava korisniku da u jednome koraku kompajlira i pokrene kod
- Compile – omogućava korisniku da samo kompajlira kod i tako provjeri njegovu ispravnost
- Run – pokreće kod ukoliko je kompajliran
- Stop – u potpunosti zaustavlja izvođenje koda
- Pause – privremeno zaustavlja izvođenje koda
- Default code – vraća originalni kod koji se nalazio unutar hakiranog objekta

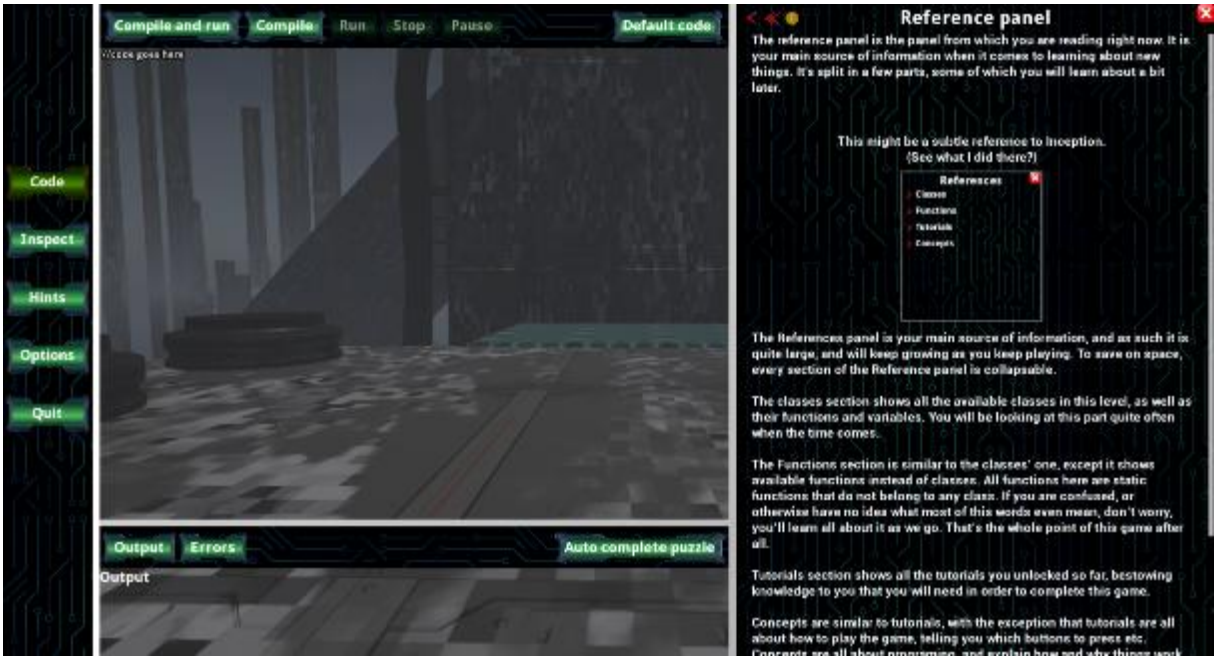
Prostor koji se nalazi na dnu ovoga područja koristi se za ispis raznih povratnih poruka i sastoji se od dva prozora između kojih se može prebacivati pritiskom na odgovarajuće gumbe. Prvi prozor za poruke je Output prozor. Unutar njega će biti ispisani podatci koje je korisnik upisao unutar Print() funkcije ili neke slične funkcije. Tako primjerice ukoliko korisnik upiše Print(5) output prozor će ispisati broj pet. Drugi prozor ovdje služi za ispisivanje grešaka. Ukoliko korisnik napiše neispravan kod i pokuša ga kompajlirati tada će se u ovome prozoru napisati poruka koja će specificirati koji tip greške je napravljen.

Zadnji dio konzolinog sučelja se nalazi na njenoj desnoj strani i nazvan je Reference panel. Reference panel predstavlja igraćevu bazu znanja unutar koje se nalaze svi potrebni podatci za igranje igre, učenje novih programerskih koncepata i samim time rješavanje igrinih zagonetki. Unutar ovoga područja nalaze se četiri glavne kategorije podataka, svaka sa svojim vlastitim podkategorijama. Te glavne kategorije su sljedeće:

- Classes – klase predstavljaju sve objekte koje je moguće hakirati unutar razine na kojoj se igrač trenutno nalazi. Osim točnih naziva objekata, ovdje je također moguće vidjeti točne definicije pojedinih funkcija koje se nalaze unutar tih objekata.
- Functions – unutar ove kategorije se nalaze generičke funkcije koje igrač može koristiti na bilo kojoj od igrinih razina. Primjerice neke od tih generičkih funkcija su funkcija za ispisivanje Print() i funkcija Pause() koja korisniku omogućuje da pauzira izvođenje koda na specificirani broj sekundi.
- Tutorials – u ovoj kategoriji se nalaze sve upute za igranje igre i korištenje njezinih sustava.
- Concepts – unutar ove kategorije nalaze se generalizirani opisi svih objekata s kojima igrač vrši interakciju prilikom igranja i još važnije, ovdje se nalaze opisi i primjeri svih programerskih koncepata koji se igraču prezentiraju tijekom igranja.

Classes i Functions se po svojoj funkcionalnosti razlikuju od Tutorials i Concepts kategorija. Kada se otvore podkategorije od prve dvije navedene kategorije igraču je prezentirana definicija neke funkcije. Pritiskom na neku od podkategorija koncepata ili tutoriala se otvara potpuno novi prozor koji će sadržavati opis tog koncepta ili tutoriala, dok će za programerske koncepte sadržavati primjere koda. Tutoriali i opisi

igrinih objekata će uz same opise sadržavati i njihove slike. Izgled jednoga od tih opisa koncepata prikazan je sljedećom slikom (slika 52).



Slika 52 – Opis Reference Panel-a unutar Reference Panel-a

Izvor: Vlastita izrada

5.Testiranje i povratne informacije

Nakon što su unutar igre implementirane sve bitne funkcionalnosti i kada se ona može odigrati od početka do kraja, igra ulazi u beta verziju. Kada se igra nalazi u beta verziji potrebno je početi provoditi sve veći i veći broj testiranja (engl. playtesting). Proces testiranja je takav da se kroz igru i kroz njezine razine prolazi veliki broj puta, te se zabilježavaju svi bugovi i odstupanja od predviđenog ponašanja. Nakon što se zabilježe svi problemi, taj popis se uzima i stvara se prioritetna lista prema kojoj je tada potrebno rješavati navedene probleme. Testiranja se ponajprije provode od strane razvojnog tima, te se onda nakon određenog vremena počne igru davati na testiranje sve većem i većem broju ispitanika, ta testiranja se obično nazivaju zatvorena i otvorena beta testiranja.

5.1 Interno testiranje

Prva provedena verzija testiranja je bilo interno testiranje. Tijekom internog testiranja članovi razvojnog tima mnogo puta prolaze kroz igru i zabilježavaju probleme na koje naiđu, ukoliko je pronađene probleme lako popraviti oni se odmah popravljaju i nije ih potrebno stavljati na listu, već samo unutar kontrole verzija zabilježiti što je izmjenjeno. Kroz igru se ponajprije prolazi na način na koji bi novi igrač igrao, dakle način koji je predviđen od strane razvojnog tima. Nakon toga kroz igru se prolazi tako da ju se pokušava što više polomiti igrajući ponašanjima koja nisu predviđena za dane razine, poput skakanja po dijelovima razina koji nisu predviđeni za to i pisanjem lošega koda kako bi se otkrilo da li će on dovesti do rušenja igre. Unutar ovoga testiranja se popravlja jako veliki broj problema koji variraju od manjih bugova i predmeta bez kolizije kroz koje igrač onda može prolaziti, pa sve do rušenja igre u određenim uvjetima ili nemogućnosti nastavljanja daljnjega igranja jer neka funkcionalnost ne radi kako treba.

5.2 Zatvoreno testiranje

Nakon što je provedeno interno testiranje potrebno je napraviti manje zatvoreno kontrolirano testiranje, zatvoreno beta testiranje. Za potrebe toga testiranja odabrano je pet osoba koje su imale nisko ili nepostojeće znanje iz programiranja i nisko do osrednje iskustvo sa igranjem videoigara. Razlog tome je što je cilj ovakvoga testiranja bilo otkriti da li igra uspješno podučava koncepte programiranja,

te da li uspješno uči igrače kako je igrati. Kako bi se to moglo testirati ispitanicima je zadatak bio odigrati igrinu uvodnu, tutorial, razinu. Njihova sesija igranja (engl. playsession) je bila nadzirana od strane članova razvojnog tima, te je također bila snimana. Članovi tima koji su nadzirali te sesije su morali što manje utjecati na ponašanje ispitanika i što više se fokusirati na njihov način igranja i eventualne povratne informacije. Tijekom i nakon tih sesija sa ispitanicima su provedeni neformalni intervjui kako bi se dobilo njihovo mišljenje o igri. Odgovori sa tih intervjua su pokazali da su ispitanici smatrali kako igra ima atraktivnu vizualnu stranu i zanimljivu premisu, te su sve u svemu bili zadovoljni sa sveukupnim iskustvom. Glavne negativne stvari koje su svi ispitanici dali do znanja je činjenica kako igra zahtjeva previše čitanja i da je osvijetljenje unutar igre previše mračno. Potrebno je također napomenuti kako je ispitanicima koji nisu imali prijašnjega znanja iz programiranja bilo potrebno dati malo pomoći kako bi uspješno dovršili razinu, što je također vjerojatno povezano sa činjenicom da igra od igrača zahtjeva da pozorno čita sve dane upute. Osim korištenja samih korisničkih povratnih informacija također su bile i korištene snimke tih sesija. Navedene snimke su pregledane od strane razvojnog tima, te su uočeni određeni negativni uzorci ponašanja kod većine ispitanika. Jedan od tih uzoraka su bili problemi pri navigaciji igrinim korisničkim sučeljem gdje su korisnici ili u potpunosti ignorirali njegove određene dijelove, ili bi pokušavali kliknuti na dijelove sučelja za koje su mislili da predstavlja neki gumb, kada to zapravo nije bio slučaj. Nakon pregledavanja snimki i zabilježavanja najvažnijih problema koji su uočeni tijekom ovoga testiranja, bilo je potrebno te probleme popraviti. Tako su zahvaljujući ovome testiranju dorađeni razni igrini dijelovi; korisničko sučelje je poboljšano, tutoriali su poboljšani te je nadodana još jedna sekcija koja igraču eksplicitno daje do znanja na kakav način je predviđeno da igraju, povećana je svijetlina na određenim razinama i napravljeno je još mnogo manjih poboljšanja.

5.3 Otvoreno testiranje

5.3.1 Rezultati ankete

Zaključak

Literatura

Alonso, G, P. (2016). *What effects do puzzle video games have on the brain*.
URL: https://spritted.com/en/news/what-effects-do-puzzle-video-games-have-on-the-brain

Barseghian, T. (2013). *Can Digital Games Boost Student's Test Scores?.*
URL: https://www.kqed.org/mindshift/29361/can-students-learn-better-with-digital-games

CodeCombat. (2019). CodeCombat – *Learn how to code by playing a game*
URL: https://codecombat.com/

Davies, M. (2015). *A good Puzzle Game Is Hard To Build*.
URL: https://www.rockpapershotgun.com/2015/01/22/how-to-make-a-puzzle-game/

Drozina, A. (2019). *Implementacija igre za učenje koncepata programiranja u razvojnom okruženju Unreal Engine 4*. Fakultet informatike u Puli

Filmora. (2019). *Digital Video Game Trends and Stats for 2019*.
URL: https://filmora.wondershare.com/infographic/video-game-trends-and-stats.html

Fox, B. (2005). *Game Interface Design*. Thomson Course Technology

Gamasutra. (2019). *These are Gamasutra's favorite slides from GDC 2019*
URL:
http://www.gamasutra.com/view/news/339868/These_are_Gamasutras_favorite_slides_from_GDC_2019.php

Lofgren, K. (2017). 2017 video game trends and statistics – *Who's Playing What and Why?*.
URL: https://www.bigfishgames.com/blog/2017-video-game-trends-and-statistics-whos-playing-what-and-why/

Lynn, B. (2019). *Video Games Used as Learning Tools in Schools.*
URL: https://learningenglish.voanews.com/a/video-games-used-as-learning-tools-in-schools/4754113.html

Mubaslat, M, M. (2012). *The Effect of Using Educational Games on the Students' Achievementin English Language for the Primary Stage*. Amman – Jordan.
URL: https://files.eric.ed.gov/fulltext/ED529467.pdf

Osborn, G. (2017). *Male and Female Gamers: How Their Similarities and Differences Shape the Games Market*
URL: https://newzoo.com/insights/articles/male-and-female-gamers-how-their-similarities-and-differences-shape-the-games-market/

Q2L. (2019). *About Quest to Learn*.

URL: https://www.q2l.org/about/

Schell, J. (2008). *The Art of Game Design a Book of Lenses*. Morgan Kaufmann Publishers

Science Daily. (2014). *Puzzle games can improve mental flexibility, study shows*
URL: https://www.sciencedaily.com/releases/2014/06/140624092528.htm

Shapiro, J. (2014). *Games In The Classroom: What the Research Says.*
URL: https://www.kqed.org/mindshift/36482/games-in-the-classroom-what-the-research-says

Shieber, J. (2019). *Video Game Revenue Tops $43 Billion in 2018, an 18% Jump from 2017*.
URL: https://techcrunch.com/2019/01/22/video-game-revenue-tops-43-billion-in-2018-an-18-jump-from-2017/?guce_referrer_us=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_cs=5ZO1OC9JD9af-Fs8j2IeYQ&guccounter=2

Statista. (2019). *Number of games released on Steam worldwide from 2004 to 2018*
URL: https://www.statista.com/statistics/552623/number-games-released-steam/

SteamDB. (2019). *Steam Charts and Stats*.
URL: https://steamdb.info/graph/

Strickland, D. (2017). *Gamers care most about graphics, shooters reign supreme*.
URL: https://www.tweaktown.com/news/57199/gamers-care-graphics-shooters-reign-supreme/index.html

Super Data Research. (2018). *2017 Year in Review – Digital Games and Interactive Media*.
URL: http://strivesponsorship.com/wp-content/uploads/2018/02/SuperData-2017-year-in-review-digital-games-and-interactive-media.pdf

VGChartz. (2019a). *The Talos Principle Sales Numbers*.
URL: http://www.vgchartz.com/gamedb/games.php?name=talos+principle

VGChartz. (2019b). *Portal Sales Numbers*.
URL: http://www.vgchartz.com/gamedb/games.php?name=portal

Yee, N. (2016). *7 Things We Learned About Primary Gaming Motivations From Over 250,000 Gamers*.
URL: https://quanticfoundry.com/2016/12/15/primary-motivations/

Popis slika

Slika 1 – Broj igara objavljenih na Steam platformi kroz godine 1
Slika 2 – Broj istovremenih korisnika na Steam platformi 2
Slika 3 – Igra Portal 6
Slika 4 – Primjer zagonetke u igri The Talos Principle 7
Slika 5 – Rješavanje zagonetke u igri Glitchspace 8
Slika 6 – Kodiranje u igri else Heart.Break() 9
Slika 7 – Situacija iz igre CodeCombat 10
Slika 8 – Primjer Code Hunt zagonetke 11
Slika 9 – 10 najprodavanijih igara za osobna računala u 2017. godini 12
Slika 10 – Popularnost različitih žanrova po platformama 13
Slika 11 – Demografija igrača prema spolu i prema godinama 14
Slika 12 – Faktori koji utječu na odluku da igrač kupi neku igru 15
Slika 12 – Utjecaj streamera na razvoj igre 17
Slika 13 – Udio prodaje koji Epic Games i Steam dobivaju 18
Slika 14 – SWOT analiza igre CodeBreaker 19
Slika 15 – Razvojni ciklus videoigara 27
Slika 16 – Izgled osnovnog Unreal editor prozora 29
Slika 17 – Izgled Blueprint sustava 30
Slika 18 – Jedna od stranica CodeBreaker dizajn dokumenta 31
Slika 19 – Primjer igrinog svijeta pod utjecajem korupcije 33
Slika 20 – Primjer teksta sa pozadinskom pričom 34
Slika 21 - Prva verzija sustava za pisanje koda 35
Slika 22 – HackerVision konceptni dizajn 36
Slika 23 – Sustav inspekcije 37
Slika 24 – Mapa za testiranje 38
Slika 25 – Opis jedne od igrinih zagonetki 39
Slika 26 – Rješenje opisane zagonetke 40
Slika 26 – Primjer zagonetke sa logičkim programiranjem 41
Slika 27 – Izgled prototipne razine 43
Slika 29 – Igrina prva Nexus razina 44
Slika 30 – Prva iteracija razine za učenje petlji 45
Slika 31 – Druga iteracija razine za učenje petlji 46
Slika 32 – Treća iteracija razine za učenje petlji 48
Slika 33 – UML dijagram sa funckionalnostima glavnoga menija 50
Slika 34 – Konceptni dizajn glavnog menija 51
Slika 35 – Konzolino koriničko sučelje tijekom prototipiranja igre 52
Slika 36 – Konceptni dizajn izgleda korisničkog sučelja za konzolu 53
Slika 37 – Prva implementacija redizajniranog sučelja za konzolu 54
Slika 38 – Završni izgled sučelja za konzolu 55
Slika 39 – Rezultat igrine uvodne animacije 56
Slika 40 – Završni izgled igrinog glavnog menija 57
Slika 41 – Izgled Project.LoadLevel() podmenija 58
Slika 42 – Izgled menija sa opcijama 59
Slika 43 – Igrina prva tutorial poruka 60
Slika 44 – Završni izgled HackerVision sustava 60
Slika 45 – Objekt nad kojim se može vršiti inspekcija 61
Slika 46 – Inspekcijski prozor 62
Slika 47 – Objekt s kojim je moguće vršiti klasičnu interakciju 63

Slika 48 – Objekt koji je moguće hakirati 64
Slika 49 – Izgled prozora za hakiranje objekta 65
Slika 50 – Glavni dijelovi konzole 66
Slika 51 – Izgled Hints Prozora..... 67
Slika 52 – Opis Reference Panel-a unutar Reference Panel-a..... 69

Sažetak

Tema ovoga rada je dizajniranje videoigre za učenje koncepata programiranja unutar razvojnog okruženja Unreal Engine 4. Unutar ovoga rada biti će ponajprije opisana navedena igra i provedena analiza tržišta sličnih videoigara. Poglavlje nakon toga baviti će se praktičnim primjenama korištenja igara za potrebe edukacije. Glavni dio ovoga rada predstavlja poglavlje pod nazivom Razvoj igre u kojemu će biti pojašnjene metodologije i filozofije dizajniranja videoigre ovakvoga tipa. Unutar toga poglavlja su prikazani dizajnovi različitih igrinih dijelova unutar nekoliko stadija igrinog razvoja kako bi što efektivnije prikazalo na kakav način se igra i njezini pojedini dijlovi mijenjaju tijekom razvojnog ciklusa. Igra će također biti prikazana iz perspektive standardnoga korisnika unutar poglavlja koje se bavi korisničkom dokumentacijom, te će unutar nje biti pojašnjeno na kakav se način ona igra. Naposljetku biti će prikazani rezultati istraživanja nakon što je igra dana na igranje određenom uzorku ispitanika.

Ključne riječi

Unreal Engine 4, CodeBreaker, Dizajn igara, Dizajn razina, Dizajn korisničkog sučelja, Videoigre, Programiranje, Podučavanje, Igre u edukaciji, Edukacija, Razvojni ciklus, Industrija videoigara, Iterativni dizajn

Summary

The theme of this work is designing a videogame that teaches the concepts of programming, using the Unreal Engine 4 environment. Withing this work will be described the game that is being made and a market analysis of similar games will be conducted after that. The chapter after that one will talk about practical examples of using videogames as teaching tools within the educational system. The main part of this work is presented within a chapter that will be talking about methodologies and philosophies that are used when designing a game of this type. Within this cahpter different parts of the game will be shown and how they have changed through development cycle to showcase how much games change during development. Next chapter will be focused on user experience and it will explain in which way will the standard users interact with the game and its systems. Final chapter will be about different playtesting sessions that were conducted and the user feedback that was received from those playtests.

Key words

Unreal Engine 4, CodeBreaker, Game design, Level design, User interface design, Videogames, Programming, Teaching, Games in education, Education, Development cycle, Videogame industry, Iterative design