Razvoj edukativnih računalnih igara korištenjem LibGDX okvira

Pićan, Lea

Master's thesis / Diplomski rad

2018

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:137:414413

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-01-30**

Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula

Sveučilište Jurja Dobrile u Puli

Fakultet informatike

LEA PIĆAN

Razvoj edukativnih računalnih igara korištenjem LibGDX okvira

Diplomski rad

Pula, lipanj 2018. godine

Sveučilište Jurja Dobrile u Puli

Fakultet informatike

LEA PIĆAN

Razvoj edukativnih računalnih igara korištenjem LibGDX okvira

Diplomski rad

JMBAG: 0303032968, redoviti student

Studijski smjer: Informatika

Predmet: Mobilne aplikacije

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informacijske i komunikacijske znanosti

Znanstvena grana: Informacijski sustavi i informatologija

Mentor: doc. dr. sc. Siniša Sovilj

Pula, 1. lipnja 2018. godine

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisana Lea Pićan 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 seminarskog 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.

U Puli, 1.06.2018. godine

Student

_____

IZJAVA

o korištenju autorskog djela

Ja, Lea Pićan, dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom „Razvoj edukativnih računalnih igara korištenjem LibGDX okvira“ 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, 1. lipnja 2018. godine

Potpis

_____

SVEUČILIŠTE JURJA DOBRILE U PULI
ODJEL ZA INFORMACIJSKO-KOMUNIKACIJSKE TEHNOLOGIJE

Pula, 1. ožujka 2017.

DIPLOMSKI ZADATAK

Pristupnik:	**Lea Pićan (0303032968)**
Studij:	Sveučilišni diplomski studij Informatike
Naslov (hrv.):	**Razvoj edukativnih računalnih igara korištenjem LibGDX okvira**
Naslov (eng.):	Educational video games development using LibGDX framework
Opis zadatka:	Zadatak je razviti edukativnu 2D računalnu igru korištenjem LibGDX programskog okvira. U sklopu razvoja istražiti prednosti i nedostatke LibGDX-a te ga usporediti sa nekoliko drugih popularnih programa za razvoj računalnih igara: Unity, Unreal Engine, Cocos2D.

Zadatak uručen pristupniku: 1. ožujka 2017.
Rok za predaju rada: 1. veljače 2018.

Mentor:

Siniša Sovilj

_____

doc.dr.sc. Siniša Sovilj

Sadržaj

UVOD

Edukativne igre sve su češći način učenja. Djeca uče kroz igru. Igre su zabavne, motivirajuće i okupiraju pažnju u potpunosti. Ukoliko je igra zabavna djeca će provesti više vremena igrajući je te tako naučiti više o temi.

U Istri maslina i maslinovo ulje poznati su proizvodi. Veliki broj kućanstava koristi maslinovo ulje. Proizvodi od masline su jedan od najboljih promotora turizma Istre, dobra osnova za gospodarski razvoj i naposljetku jedan od najistaknutijih simbola Istre, svojevrsni brand. Iako svi znaju što je maslinovo ulje, rijetko tko zna kako se ono prerađuje. Za dobivanje ulja visoke kakvoće ključni segmenti su vrijeme i način berbe.

Cilj ovoga rada je izraditi edukativnu igru koja će pratiti proces branja maslina uz zabavne igre. Igra je namijenjena djeci posjetitelja Kuće maslinovog ulja u Puli te je prilagođena svim generacijama.

Rad se sastoji od šest glavnih poglavlja. Prvo poglavlje opisuje ideju i motivaciju za odabir ove teme. Spajanje istarske tradicije sa modernim načinom učenja za najmlađe posjetitelje muzeja.

Istraživanje programskih okvira i programskih pogona nalazi se u drugom poglavlju, definirane su glavne razlike te prednosti i nedostaci u odnosu na *LibGDX*. Kroz istraživanje biti će opisane temeljne karakteristike odabranih programskih okvira kao što su: podržane platforme, cijene, potrebno znanje programskih jezika i slično. *LibGDX* nije odabran za ovaj projekt na temelju istraživanja, nego radi usklađivanja teme sa kolegijem.

Kroz treće poglavlje opisuju se odabrane tehnologije te njihova primjena u projektu. Igra je razvijena za više platformi u Java programskom jeziku koristeći *IntelliJ Idea* razvojno okruženje i *LibGDX* programski okvir. Dizajn igre izrađen je u Adobe *Illustratoru*.

Četvrto poglavlje opisuje način planiranja igre kroz dokument dizajna igre. Igra se sastoji od dva nivoa, prvi nivo je *Falling olives* u kojem igrač mora sakupiti što više maslina. *Memory* je drugi nivo koji je klasična igra pamćenja sa motivima vezanim uz masline, njihovu obradu te korištenje.

U petom se poglavlju analizira razvoj igre po korištenim *LibGDX* klasama. Igra je razvijana prateći Udemy tutorijal - *The Complete LibGDX Game Course Using Java*. Poglavlje je podijeljeno u pet potpoglavlja: postavljanje projekta, klase za crtanje objekata, vizualni efekti, korisničko sučelje i zvuk. Kroz navedena potpoglavlja objašnjen je kod igre.

U zadnjem poglavlju prije zaključka, sažetka i bibliografije opisane su prilike za poboljšanje igre.

1. IDEJA I MOTIVACIJA

Motivacija i ideja za izradu ovakve vrste igre dolaze se više strana. Istarsko maslinarstvo jedno je od najuspješnijih u zemlji i šire, a naše maslinovo ulje najbolje na svijetu prema svojoj kvaliteti. O njemu zna mali broj ljudi. Od kada je u Puli otvoren muzej maslinovog ulja (*poglavlje 1.2*), sve više posjetitelja dolazi i želi naučiti nešto o tome. Međutim, teško je zabaviti najmlađe, kojima nije zabavno slušati dugačak razgovor o maslinama.

Edukativna igra *Olivia* osmišljena je na način na djeca uče temeljne pojmove o maslinama kroz igru i zabavu. Nakon što prođu igru te krenu ponovo razgledavati muzej, sjetit će se onoga što su igrali.

1.1 Edukativne računalne igre

Potpuno integrirane u svakodnevni život milijuna mladih diljem svijeta, video igre su vitalni dio suvremene kulture i društva. No, reakcija mnogih vlasti i većine odgojitelja bila je diskreditirati video igre uz pretpostavku da imaju negativan učinak na djecu. Međutim, nakon više od dva desetljeća istraživanja objavljeno je mnogo studija koje su postupno dovele do složenijih, nijansiranijih i korisnijih razumijevanja video igara.

U današnje vrijeme multimedijske računalne igre predstavljaju najčešći oblik digitalne zabave koja je uvelike zamijenila ostale oblike zabave. Igre su ušle u sve pore svakodnevnog života i ne poznaju dobne granice. Razvoj takvog oblika zabave vrlo je zahtjevan jer multimedijski oblik obuhvaća integraciju videa, animacije, zvuka, teksta, pokretne i/ili nepokretne slike i interakciju bez koje ne bi postojale. Računalne igre često od igrača zahtijevaju brzu reakciju, rješenje nekog problema te mogu pozitivno utjecati na razvoj mnogih kompetencija djeteta. Dijete se, igrajući igru, susreće s novim i zahtjevnijim zadacima. Mnogi stručnjaci se slažu s time da kroz računalne igre djeca razvijaju svoje psihomotorne, mentalne i perceptivne sposobnosti.

Važno je naglasiti kako video igre utječu na proces učenja djece i adolescenata, kao i njihove učinke na obrazovni proces općenito. Dakle, video igre, kao i bilo koji drugi tehnološki uređaj, jednostavno su mediji preko kojih se mladi ljudi bave određenim aktivnostima. Ili, vidljivi iz drugog kuta, oni su samo jedna druga značajka sa simboličkim, ekonomskim i tehnološkim dimenzijama u složenom društvenom kontekstu koji se stalno podvrgava intenzivnom i ubrzanom procesu promjene koji

utječe na sve sfere svakodnevnog života. Internet koji je sveprisutan u našim životima uveden je u posljednja dva desetljeća tijekom kojeg je današnja mlada generacija rođena. Tehnologija u životu djeteta svakako može imati dobre i loše posljedice. Danas treba objašnjavati kako koristiti sustav za poboljšanja, a ne zabranjivati korištenje. (Noguero, 2003)

Većina tih istraživanja dijele stajalište da uporaba videoigara može biti korisna za stjecanje sposobnosti i vještina kao što su: (Noguero, 2003)

- Prostorna percepcija i prepoznavanje
- Razvoj vizualnog razlučivanja i odvajanja vizualne pozornosti
- Razvoj logike
- Razvoj u znanstvenim / tehničkim aspektima
- Razvoj složenih vještina

Glavna karakteristika obrazovne igre je činjenica da je obrazovni sadržaj isprepleten s karakteristikama igre. Igra bi trebala motivirati učenika da ponavlja cikluse unutar konteksta igre. Za vrijeme ponavljanja, npr. igrajući igru, očekuje se da će učenik pokazati željeno ponašanje temeljeno na emocionalnim ili spoznajnim reakcijama koje proizlaze iz interakcije s igrom i povratnom informacijom od igranja igre. (Pivac, 2006)

Slika 1: Model učenja kroz igru

Izvor: [Maja Pivec: Igra i učenje: Potencijali učenja kroz igru]

1.2 Kuća istarskog maslinovog ulja

Kuća istarskog maslinovog ulja u središtu grada Pule vodi vas kroz povijest i sadašnjost istarskog maslinarstva. U muzeju ćete saznati zašto su još Rimljani izrazito cijenili istarsko maslinovo ulje te kako su ga prerađivali; što se događalo s maslinarstvom tijekom srednjeg vijeka, kako su masline prerađivali naši djedovi te kako se ono prerađuje danas i koje su tajne u stvaranju vrhunskog ekstra djevičanskog maslinovog ulja. Svakog posjetitelja očekuje i vođena degustacija istarskog maslinovog ulja uz edukaciju pravilnog kušanja. Saznat ćete zašto su istarska ekstra djevičanska maslinova ulja među najboljima u Hrvatskoj i svijetu, probati ekstra djevičanska ulja raznih istarskih proizvođača, degustirati različite sorte istarskih ulja. U prodajnom prostoru možete kupiti najveći izbor vrhunskih istarskih ekstra djevičanskih maslinovih ulja od preko 25 istarskih proizvođača te druge istarske proizvode..

(Museum Olei Histriae, 2018)

2. USPOREDBA PROGRAMSKIH OKVIRA

Usporedba programskih okvira sastoji se od tri glavne cjeline

- Definiranje pojmova „*Engine (pogon igre)*“, „*Framework (programski okvir)*“, „Library (biblioteka)“
- Opis odabranih programskih okvira
- Statistike, analize i usporedbe programskih okvira u odnosu na *LibGDX*.

U prošlosti se nerijetko za svaku računalnu igru radio poseban *pokretač igre*, no to je bilo vrlo zahtjevno i skupo rješenje. Danas, mnogo godina kasnije, imamo niz pokretača igre kao što su *Cryengine*, *Construct 2*, *Game Maker Studio*, *Frostbite* i mnogi drugi pomoću kojih timovi i tvrtke stvaraju svoje igre. No, većina tržišta u rukama je dva konkurentska proizvoda, zvanih Unity 3D i Unreal Engine.

Programske okvire za izradu igara koristi sve manje razvojnih programera zbog njihove kompleksnosti te razine potrebnog znanja. (Game from scratch, 2018)

Biblioteka (eng. Library)

Biblioteka je jednostavno rečeno kolekcija koda i podataka namijenjena ponovnoj upotrebi. U razvoju igara, biblioteka je općenito zbirka kodova za obavljanje zadataka iz određene domene. Na primjer, sviranje zvuka, izvođenje fizike, rukovanje ulaznim jedinicama i slično. (Game from scratch, 2018)

Programski okviri (eng. Framework)

Programski okvir je kolekcija biblioteka i alata koji zajedno rješavaju dati zadatak, u ovom slučaju razvijanje igre. (Game from scratch, 2018)

Neki od poznatih programskih okvira za razvijanje igara su:

- LibGDX
- Cocos2dx
- SFML

Pokretač igre (eng. Game engine)

Engine je kao i *framework* kolekcija biblioteka i alata. Najveća razlika je što *engine* sadrži još dvije bitne stavke:

- Graf scene – struktura podataka koja sadrži svijet igre. Uključuje brojne funkcije za upravljanje, pretraživanje i spremanje grafa scene iako su te funkcije često implementirane od strane same igre.
- Uređivač svijeta / nivoa igre – najveća razlika između *engine-a* i *framework-a*.

Danas su *engine*-i najpopularniji za razvoj igara zbog jednostavnog korištenja. Neki od najpoznatijih *engine*-a su:

- Unity
- Unreal engine 4
- Godot Engine

2.1 Unity3d

Unity je razvijen 2004. godine nakon što su trojica prijatelja, čija se igra nije uspjela prodati, shvatili da je pokretač koji su za nju napravili bio vrjedniji i od same igre. Od tada su uspjeli demokratizirati razvoj igara. Iako je to pridonijelo "gužvi" na tržištu igara, donijelo je i sve više zanimljivih igara i priča. (Unity technologies, 2017)

Unity ima fantastičnu zajednicu fanova, a njihov poslovni model pokazuje kako izdavanje igara nije jedini način zarade u industriji igara. Neki razvojni programeri su se u potpunosti prebacili sa razvoja igara na razvoj modela ili skripti za članove zajednice u *Asset Storeu*, prodajom tečajeva na platformi *Udemy* ili pokretanjem *Patreon* kampanja. (Unity technologies, 2017)

Podržava 2D i 3D grafiku, te koristi sljedeće grafičke pokretače:

- Direct3D za windows i Xbox one
- Open GL za Linux, macOS i Windows
- Open GL za Android i iOS

Osim samog pokretača dostupni su još i sljedeći alati: (Unity, 2017)

- **Unity Ads** - Oglasi izdavačima omogućuju integraciju video oglasa u mobilne igre na način da se povećava angažiranje igrača te da igrica više zarađuje.
- **Unity Analytics** – alat za analizu ponašanja igrača specifičnih za jednu igricu. Moguće je vidjeti statistike u realnom vremenu da bi se moglo raditi na poboljšanju korisničkog iskustva.
- **Unity Certification** – edukacija / test za dobivanje certifikata za Unity razvojnog programera, dizajnera, animaciju ili efekte.
- **Unity Cloud Build** – olakšava izradu i dijeljenje igre. Automatski kompilira, implementira i testira igru. Postavljanje traje nekoliko sekundi i radi s postojećim repozitorijem za upravljanje izvornim kodom.
- **Unity IAP** – omogućuje prodaju različitih stavki izravno u besplatnoj ili plaćenoj igri, uključujući dodatni sadržaj, virtualne proizvode i pretplate.
- **Unity Multiplayer** – je najjednostavniji način kreiranja igre za više igrača
- **Unity Performance Reporting** – automatski prikuplja aplikacijske pogreške neovisno o korištenim uređajima i platformama, tako da se problemi mogu pronaći i riješiti u najkraćem roku
- **Unity Collaborate** – jednostavan način spremanja, dijeljenja i sinkronizacije projekta za timove.

Danas je Unity najpopularniji *pokretač igre.* Koristi se za razvoj igara za više sustava i više platformi. Unity se ne ograničava samo na mobilne igre nego ima *engine* i za web, konzole, desktop te VR[1] i AR[2] igre. Jedan od slogana Unityja je „Build once, deploy anywhere“ što bi značilo da se jednom napisani kod može izvršavati na bilo kojoj platformi.

[1] Virtual reality – igre virtualne realnosti
[2] Agumented reality – igre digitalno proširene stvarnosti

U sljedećoj tablici prikazane su sve platforme koje Unity podržava. (Unity, 2017)

Tablica 1: Unity - podržane platforme

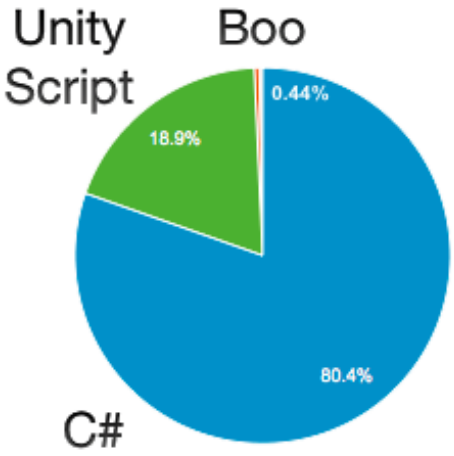
Mobilne	**Virtualna i proširena stvarnost**	**Operativni sustavi**	**Konzole**	**Web**
iOS	Oculus Rift	Windows	Ps4	Web GL
Windows Platform	Goole Cardboard	Mac	psVita	
Android	Steam VR	Linux	xbox one	
Fire OS	Playstation VR		Nintendo Switch	
	Gear VR		3DS	
	Windows Mixed Reality			
	Daydream			
	Apple AR Kit			
	Google AR Core			

Izvor: Istraživanje autorice

Unity podržava „povuci i ispusti“ tehniku stoga se jednostavne igre mogu razviti bez velikog programerskog znanja. Podržana su tri programska jezika C#, UnityScript i Boo.

Po Unity statistici korištenja programskih jezika 80.4% razvijatelja koristi jezik C# (c sharp), zatim UnityScript sa 18.9% (JavaScript za Unity) te Boo sa tek 0.44%.

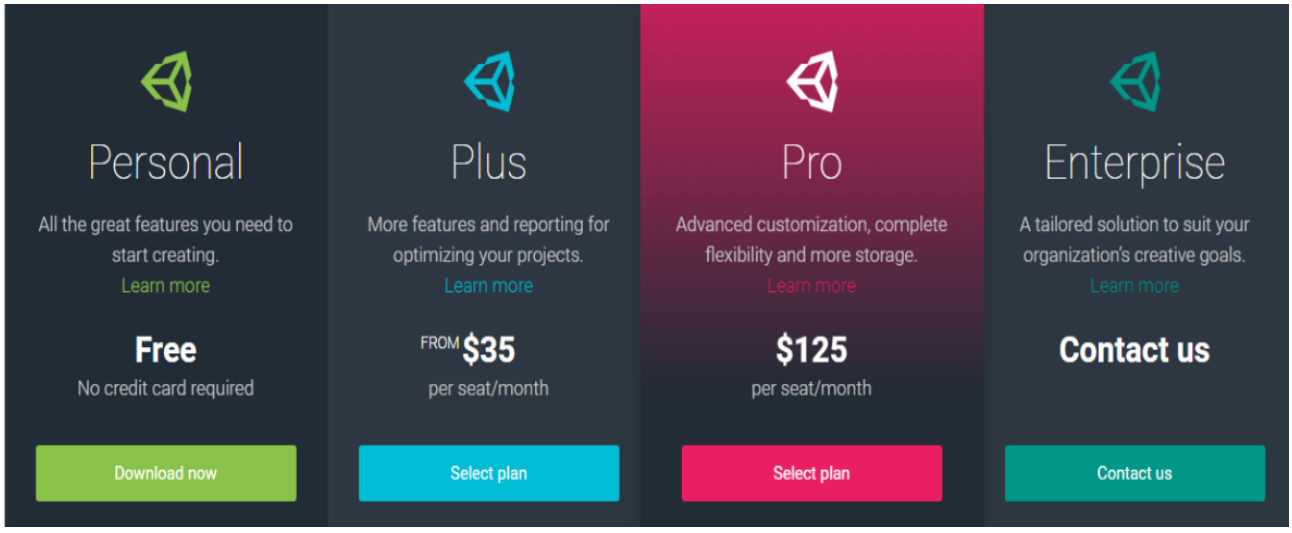
Graf 1: Unity - statistika korištenja programskih jezika



Izvor:https://thenextweb.com/gaming/2016/03/24/engine-dominating-gaming-industry-right-now/

Unity je besplatan pokretač igre samo za osobnu uporabu za početnike te za osobe koje se bave izradom igrica kao hobijem. Plus verzija koja se naplaćuje 35$ mjesečno nudi mogućnosti sakrivanja početnog zaslona, analize te dodavanja reklama. Za 125$ mjesečno *Unity* nudi još dodatnih pogodnosti. (Unity technologies, 2017)

Slika 2: Unity – plan pretplate



Izvor: https://store.unity.com/?_ga=2.165326652.2089007252.1524998743-324393899.1497360475

Na stranicama *Unitya* može se pristupiti *Asset store-u* gdje se može kupiti ili besplatno skinuti mnogo gotovog sadržaja kao što su modeli, pozadine slike, zvuk te alate koji se mogu koristiti kasnije u razvoju igre.

Zahtjevi sustava za instalaciju i korištenje *Unity* pokretača za razvijanje igara (Unity, 2017)

OS:

- Windows 7 SP1+, 8, 10, 64-bitne verzije
- Mac OS X 10.9+.

Procesor: SSE2

Grafički procesor: Grafička kartica sa DX10 mogućnostima

Android: Android SDK i Java Development Kit

Sljedeća tablica izrađena je zapisivanjem zaključaka istraživanja autorice u tabličnom obliku.

Tablica 2: Prednosti i nedostaci Unity-a

Prednosti	**Nedostaci**
Unity asset store	Unity može biti skup ukoliko trebate sve značajke.
Integracija za sve platforme	Spora priprema podataka za prikaz *(eng. Rendering)*
Velika zajednica	Dokumentacija nije ažurirana
Dokumentacija	Kod Unitya nije javno dostupan
Stabilnost	
Jednostavan za korištenje	
UI, Sprite i Audio editori	
Dodatni alati	

Izvor: Istraživanje autorice

Igra napravljena u Unity-u

Osvajač nagrade za najbolju VR (virtualna stvarnost) igru „Owelchemy labs“ u njihovom najnovijem projektu rekreirao je svjetski poznate Ricka i Mortya u VR svijet koristeći Unity 3d.

Slika 3: Rick and Morty: Virtual Rick-ality

Izvor 1: https://unity3d.com/games-made-with-unity

2.2 Unreal engine 4

Unreal Engine 4 (UE4) je pokretač i uređivač igre koji je razvila kompanija *Epic Games* još 1998. godine te je dobio ime po igri *Unreal Tournament*. Kroz povijest *Unreal Engine* pojavio se u četiri verzije: (Cookson, et al., 2016)

- **Unreal engine 1** - Razvoj prve generacije *Unreal Enginea* predvodio je osnivač *Epic Games*, Tim Sweeney. Inspiriran John Carmackovim razvijanjem igara Doom i Quake, Sweeney je 1995. pokrenuo pokretač igre koji će kasnije postati poznat kao *Unreal*, igra postavljena u srednjovjekovni svijet s izvanzemaljskim elementima. Nakon tri godine razvoja, debitirao je s izdavanjem igre 1998. S pokretačem dolazio je i alat za dizajniranje nivoa pod imenom *UnrealEd*.
- **Unreal engine 2** - Druga verzija debitirala je 2002. godine s igrom „*America's Army*“, besplatnom akcijskom igrom za više igrača koju je razvila američka vojska kao alat za regrutiranje. Razvijena na temelju prethodne tehnologije ova

generacija donijela je nove značajke kao što su alat za uređivanje kinematografije *Matinee,* dodatke za izvoz 3D Studio Max i Maya, Karma fizički pogon te alat *Math Engine*.

- **Unreal engine 3** - Tijekom cijelog životnog ciklusa UE3 ugrađene su značajne nadogradnje uključujući poboljšanu destruktivnu okolinu, dinamiku tijela, simulaciju mase, funkcionalnost za iPod Touch, integriranje Steamworks i stereoskopski 3D na Xbox 360. Potpora za DirectX 11 demonstrirana je demonstracijom igre „*Samaritan*“. Iako je Unreal Engine 3 bio prilično otvoren za suradnike, sposobnost objavljivanja i prodaje igara napravljenih putem UE3 bila je ograničena samo na licencirane osobe pokretača. Međutim, u studenom 2009. Epic je objavio besplatnu verziju UE3
- **Unreal engine 4** - izašao je 2012 godine te je donio ogromne promjene u samom softveru ali i u svijetu video igara zbog vrhunske vizualne kvalitete finalnih proizvoda i performansi. Unreal Engine 4 također je donio i mogućnost programiranja sa C++ ili Blueprints dok je igra pokrenuta. To omogućava programerima da vrlo brzo testiraju promjene nakon programiranja, u samoj igri bez dugih čekanja da se cijeli kod ponovno učita.

Njegova namjena je razvijanje igara na Windows, OS X ili Linux sustavu. Unreal engine simulira stvarnost te koristeći zakone fizike omogućuje izradu interaktivnog sadržaja.

Kod se može pisati u programskom jeziku ili koristeći Blueprints – sustav vizualnog skriptiranja koji omogućava osobama bez velikog znanja programiranja da stvore interaktivne elemente, a sastavni je dio Unreal Engine-a 4. Za takav način programiranja nije potrebno pisati sam kod, već je on unaprijed definiran te je na pojedincu da prema sistemu *povuci i ispusti* već predefinirane kodove slaže i povezuje

Slično kao i Unity, Unreal engine ima svoj online dućan za kupovinu modela, zvukova, likova i slično koji se zove „Marketplace“. Ima manje sadržaja od Unity Storea te, iako je sadržaj kvalitetniji, istovremeno je i cjenovno nepristupačniji.

Platforme koje Unreal Engine 4 podržava: (Unreal, 2018)

Tablica 3: Ureal engine 4 - Podržane platforme

Mobilne	**Virtualna i proširena stvarnost**	**Operativni sustavi**	**Konzole**	**Web**
Android	Oculus Rift	Windows	PlayStation 4	HTML 5
iOS	Goole Cardboard	Linux	Xbox One	
SteamOS	Steam VR / HTC Vive	Mac OS X		
	Playstation VR			
	Samsung Gear VR			
	Windows Mixed Reality			
	Google VR			
	Apple AR Kit			
	Leap Motion			

Izvor: Istraživanje autorice

Unreal ima politiku otvorenog koda – kod se može proširiti, mijenjati, zalijepiti ili integrirati s drugim softverom ili bibliotekama, uz iznimku: Ne može se kombinirati Unreal Engine kod s kodom koji pokriva licencni ugovor "*Copyleft*" koji bi izravno ili neizravno zahtijevao da Unreal Engine ne postupa po EULA (*End User Licence Agreement*) uvjetima.

Potpuna verzija UE4 je besplatna za korištenje. Ukoliko igra razvijena njime počne zarađivati, *Epic games* će potraživati 5% zarade.

Ovaj sustav ima ponekih ograničenja, no idealan je za početnike koji žele čim prije napraviti svoj prvi projekt i spremni su prilagoditi se ponuđenim funkcionalnostima. (Cookson, et al., 2016)

Zahtjevi sustava za instalaciju i korištenje Unreal Engine 4 pokretača za razvijanje igara:

- Windows 7 64-bit ili više
- Mac OS X 10.9.2 ili viši
- 8 GB RAM
- Intel ili AMD procesor,
- Grafičku karticu kompatibilnu sa DX11.
- UE4 može se pokrenuti i na nižim standardima, ali će performanse biti limitirane
- Unreal Editor može se pokrenuti na Windows, OS X i Linux sustavima.

Sljedeća tablica prikazuje prednosti i nedostatke Unreal 4

Tablica 4: Unreal Engine 4 - Prednosti i nedostaci

Prednosti	**Nedostaci**
Besplatan do zarade	Marketplace
Blueprints	Nema razne alate kao Unity
Otvoreni kôd	Loš Marketplace (Internet dućan)
Jednostavan za korištenje	Plaćanje 5% zarade
Uređivač levela	
Dokumentacija	

Izvor: Istraživanje autorice

Jedna od trenutno najpopularnijih igara koja koristi Unreal engine 4 zove se „Fortnite“ tvrtke Epic Games. Radnja se odvija na akcijskom svijetu Fortnite te timovi s do četiri igrača moraju istraživati ogroman, razrušeni svijet, skupljati resurse i raditi zajedno u cilju izgradnje ogromnih utvrda i raznovrsnih oružja kako bi što duže preživjeli.

Nakon što je tajanstveni kataklizmički događaj poznat kao „Oluja“ poharao 98% svjetskog stanovništva, surova skupina preživjelih mora se udružiti kako bi obranili planet od naleta čudovišta i jedni od drugih.

Slika 4: Fortnite

Izvor 2: http://www.alfabetajuega.com/noticia/fortnite-battle-royale-podria-anunciar-su-version-para-nintendo-switch-durante-el-e3-d-123146

2.3 Cocos2d-x

Cocos2d-x je programski okvir otvorenog koda za izradu igara napisan u C++, s tankim slojem koji ovisi o platformi. Koristi se za izradu igara, aplikacija i drugih interaktivnih programa temeljenih na GUI platformama.

Cocos2d-x kombinira prednosti korištenja jednog od najpopularnijih programskih okvira za razvijanje igara sa programskim jezikom C++. Programski okvir je razvijen da bude brz te jednostavan za korištenje i dozvoljava kodu da se izvršava na više platformi, koje se mogu vidjeti kroz sljedeću tablicu:

Tablica 5: Cocos2dx - Podržane platforme

Mobilne	**Operativni sustavi**	**Web**
Android	Windows	HTML 5
iOS	Linux	
Windows Phone	Mac OS X	
Blackberry		

Izvor: Istraživanje autorice

Cocos Creator je kompletan paket alata za razvoj igara i tijek rada, uključujući igraće pogone (na temelju Cocos2d-x), upravljanje resursima, uređivanje scene, pregled igara, ispravljanje i objavljivanje jednog projekta na više platformi. Cocos Creator pruža inovativan i jednostavan za korištenje skup alata kao što su UI sustav i uređivač animacija. SDKBOX olakšava razvojnim programerima Cocos2d-x integraciju SDK-ova treće strane u svoje igre. Svi servisni dodatci su testirani i certificirani.

Koristi se, kako mu i ime govori, za izradu 2D igri. Cocos2D-x nije namijenjen za početnike bez velikog znanja programiranja ili one koji nisu upoznati sa programskim jezikom C++. (Engelbert, 2013)

U sljedećoj tablici prikazane su prednosti i nedostaci Cocos2dX okvira.

Tablica 6: Cocos2d-x: Prednosti i nedostaci

Prednosti	**Nedostaci**
Odličan programski okvir za iPhone aplikacije	Kompleksno za početnike
Podržava 3D	Loša dokumentacija i malo materijala za učenje
Podržava skriptne jezike	Nije dugo održavan
Integracija alata treće strane	Zastarjelo

Izvor: Istraživanje autorice

2.4 LibGDX

LibGDX je besplatan programski okvir za razvijanje igara za više platformi. Trenutno podržava Windows, Linux, Mac OS X, Android, Blackberry, iOS, i HTML5. Omogućuje da se kod napiše jednom te se onda razvije za više platformi bez modifikacije. Početkom 2010 godine, pojavom prvih smartphonea Mario Zechner htio je izrađivati igrice za svoj smartphone. 2014 godine izbacio je prvu verziju LibGDX-a. Zadnja verzija izašla je 2016. godine.

Budući da LibGDX koristi Java programski jezik, a mogu se i koristiti manje popularni programski jezici (Kotlin, Scala itd..), LibGDX dozvoljava da se sve radi na niskom levelu programiranja, ali i ima svoje aplikacijsko programsko sučelje (*engl. application programming interface, API*) koje olakšava zadatke razvijanja igara. U aplikacijskom programskom sučelju su funkcije za prikaz spriteova - slike koje se integriraju kao likovi za igrice, teksta, izgradnju korisničkih sučelja, puštanje zvučnih efekta, fizike u igri i slično. (Libgdx, 2017)

Tablica 7: LibGDX - podržane platforme

Mobilne	**Operativni sustavi**	**Web**
Android	Windows	HTML 5
iOS	Linux	
Blackberry	Mac OS X	

Izvor: Istraživanje autorice

LibGDX alati:

- **Bullet:** 3D biblioteka otvorenog koda koja služi da detekciju kolizije i fiziku
- **FreeType Scallable font** – Biblioteka za manipuliranje tekstom
- **Controller Library** - Bilblioteka za rad sa upravljačima (npr. upravljač za XboX)
- **Box2d** - biblioteka za fiziku
- **Box2dlights** - 2D rasvjetni okvir koji koristi box2d
- **Ashley** – Mali okvir entiteta
- **Ai** - Okvir za umjetnu inteligenciju

LibGDX može se integrirati sa više alata trećih strana, neki od njih su:

- **Overlap2d** - Izrađen za odvajanje kodiranja od sadržaja, omogućuje razvojnim programerima stvaranje bogatih sadržaja pomoću slika, animacija, efekata čestica, svjetlosnog sustava, fizike i složenih grupiranih stavki.
- **VisUI** - omogućuje stvaranje lijepog korisničkog sučelja u LibGDXu pomoću scene2d.ui. Knjižnica sadrži scene2d.ui *skinove,* korisne *widgete* poput birača boja i odabira datoteka, a sadrži i modificirane *widgete* scene2d.ui
- **gdx-facebook** - proširenje pruža podršku za Facebook Graph API

Prednosti i nedostaci LibGDX-a mogu se vidjeti u sljedećoj tablici.

Tablica 8: LibGDX - prednosti i nedostaci

Prednosti	**Nedostaci**
Sloboda i fleksibilnost otvorenog koda	Težak za početnike
Integracija alata trećih strana	Zastarjela dokumentacija
Besplatan za korištenje	Nema gotove predloške
Ugrađeni alati	Nema integrirano razvojno okruženje
Lagan rad na više platformi	Gradle

Izvor: Istraživanje autorice

2.5 Usporedba i statistike

Usporedbe i statistike karakteristika programskih okvira i pogona igre koje slijede u potpoglavljima izrađene su na temelju istraživanja autorice. Radi preglednosti i lakšeg razumijevanja istraživanje je prikazano kroz tablice.

2.5.1 Podržane platforme

Različiti programski okviri i pogoni za razvoj mobilnih igara podržavaju različite platforme. Za razliku od programskih okvira (LibGDX i Cocos2dx) pogoni igre (Unity, Unreal) podržavaju više platformi. Igre za mobilne uređaje i desktop verzije igre zbog svoje jednostavnosti mogu se razviti na svim uspoređenim okvirima. Najpopularnije platforme za razvoj igara i aplikacija su mobilne platforme.

Tablica 9: Podržane platforme razvoja igre

Platforme	Unreal Engine	Unity	Cosos2dx	LibGDX
Android	✓	✓	✓	✓
iOS	✓	✓	✓	✓
SteamOS	✓	✓		
WindowsPhone	✓	✓	✓	
Oculus Rift	✓	✓		
Goole Cardboard	✓	✓		
Steam VR	✓	✓		
Playstation VR	✓	✓		
Gear VR	✓	✓		
Windows Mixed Reality	✓	✓		
Daydream	✓	✓		
Apple AR Kit	✓	✓		
Windows	✓	✓	✓	✓
Linux	✓	✓	✓	✓
Mac OS X	✓	✓	✓	✓
HTML5	✓	✓	✓	✓
Ps4	✓	✓		
psVita	✓	✓		
xbox one	✓	✓		
Nintendo Switch	✓	✓		
Nintendo 3DS	✓	✓		

Izvor: Istraživanje autorice

2.5.2 Cijene korištenja programskih okvira

Cijene su različite ovisno o programskom okviru ili pogonu. LibGDX i Cocos potpuno su besplatni budući da ne nude sve što nudi i njihova konkurencija te zarađuju isključivo kroz donacije. Programski okviri su najčešće besplatni dok se igrači pogoni različito naplaćuju.

Tablica 10: Cijene korištenja programskih okvira

	Besplatan	Različita	Provizija zarade
Unreal Engine			✓
Unity		✓	
Cocos 2d	✓		
LibGDX	✓		

Izvor: Istraživanje autorice

2.5.3 Programski jezici u razvoju igre

Za razvoj igre mogu se koristiti različiti programski jezici. Svaki programski okvir koristi svoj programski jezik. Koriste se objektno orijentirani (najčešće u programskim okvirima) te skriptni programski jezici (u pogonima igre).

Tablica 11: Programski jezici u razvoju igre

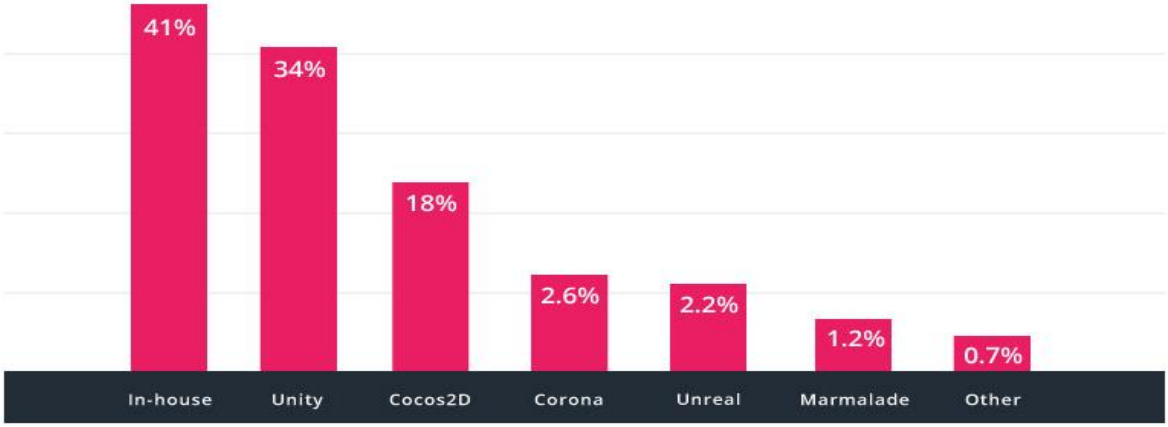
Programski jezici	Unreal Engine	Unity	Cosos2dx	LibGDX
Java				✓
C#		✓		
Unity Script		✓		
Blueprint	✓			
C++			✓	
Lua			✓	
Boo		✓		
Javascript	✓		✓	

Izvor: Istraživanje autorice

2.5.4 Najpopularniji programski okviri

Prema istraživanju Unityja za 2016. godinu najviše igra napravljeno je „in house“ programskim okvirom što znači da su kompanije unutar sebe razvile svoj vlastiti programski okvir. Zatim slijedi Unity sa impresivnih 34%. Cocos2d je zauzeo treće a Unreal engine 5 mjesto. LibGDX nije na toj listi jer za sada je njime razvijeno samo 4346[3] igara.

Graf 2: Igre razvijene game enginima u 2016 godini



Izvor: https://unity3d.com/public-relations 2016

U 2018. godini zbog zahtjeva tržišta koje očekuje često nove mobilne igre najviše se koriste pogoni igre zbog njihove jednostavnosti.

[3] Prema galeriji igara na službenoj stranici https://LibGDX.badlogicgames.com/gallery.html

2.5.5 Zaključak istraživanja

U ovom poglavlju uspoređeni su programski okviri i pogoni za izradu mobilnih igara. Kroz istraživanje pokazalo se da su pogoni igre (eng. Engine) prikladniji za početnike zbog očekivane niže razine znanja programiranja. Budući da imaju implementiranu „povuci i ispusti“ tehniku, najjednostavnije igre mogu se napraviti i bez znanja programiranja. Unreal i Unity besplatni su za studente i početnike koji žele učiti kako razvijati igre.

Unity je trenutno najpopularniji za razvijanje igara. Ima veliku zajednicu korisnika te puno primjera igri, objašnjenja igre korak po korak, te je dizajniran za zabavne kratke igre. Njegov glavni nedostatak je što ukoliko trebate sve značajke može biti vrlo skup.

Cocos2d i LibGDX programski okviri namijenjeni su programerima sa iskustvom zbog toga što koriste programske jezike poput Jave i C++. Nemaju uređivače nivoa stoga je teže urediti nivoe po želji, te ih se treba izgraditi koristeći za to predviđene klase i funkcije. Iako su sada svi programski okviri podržavaju više platformi, za potrebu ovog rada razvoj će se bazirati samo na desktop i android platformama.

Svaki programski okvir ima svoje prednosti i mane, te su na kraju programski jezik, podržane platforme te jednostavnost korištenja presudni za odabir programskog okvira.

Za potrebe ovog rada odabran je LibGDX programski okvir zbog programskog jezika Java koji je u skladu sa odabranim kolegijem. Nakon istraživanja smatram da bi Unity bio bolji izbor za jednostavnu igru zbog njegove jednostavnosti i učinkovitosti.

3. ODABIR TEHNOLOGIJA I KORIŠTENI MATERIJALI

Igra je razvijena za više platformi u Java programskom jeziku koristeći IntelliJ Idea razvojno okruženje i LibGDX programski okvir. Dizajn igre izrađen je u Adobe Illustratoru i After Effectsu.

3.1 IntelliJ IDEA

Svaki aspekt IntelliJ IDEA posebno je dizajniran kako bi se povećala produktivnost razvojnog programera. Nakon indeksiranja izvornog koda, IntelliJ IDEA nudi plasiranje brzog i inteligentnog iskustva dajući relevantne prijedloge u svakom kontekstu: brzo i pametno predviđanje završetka kôda, analiza koda na licu mjesta i pouzdani alati za refaktoriranje. (Anon., 2018)

IntelliJ razvojno okruženje odabrala sam na temelju istraživanja, preporuka i prethodnog iskustva.

3.2 Java

Java je objektno orijentirani programski jezik koji je razvila tvrtka Sun Microsystems. Java se koristi za izradu desktop aplikacija, web aplikacija, mobilnih aplikacija, pametnih kartica, robotike, igri i slično.

Najveća vrlina Jave je to što nije limitirana na samo jednu platformu te se može koristiti u različitim domenama. Java je programski jezik baziran na klasama što znači da Java podržava nasljeđivanje kao odliku objektno orijentiranog programiranja.

Programski jezik Java izvršavat će se na bilo kojem uređaju koji koristi JVM (Java Virtual Machine) što znači da se jednom napisani kod može izvršavati na bilo kojoj platformi (Windows, Mac, Linux..). (Oracle, 2017)

Iako se u programiranju LibGDX mogu koristiti i drugi programski jezici (Clojure, Kotlin, Scala, Phyton, Ruby, Ceylon) odabrana je Java zbog prijašnjeg iskustva. (Libgdx, 2017)

LibGDX sadrži nekoliko modula koji pružaju servise za svaki korak tipične arhitekture igre. Po dokumentaciji to su:

1. Ulaz – ulazni model za sve platforme. Podržava ulaz sa tipkovnice, dodirnika, akcelometra i miša.

2. Grafika – omogućuje crtanje slika koristeći hardver
3. Datoteke – Apstraktni pristup datotekama na svim platformama te metode za pisanje i čitanje
4. Audio – Snimanje i puštanje pjesama/melodija na svim platformama
5. Mrežna povezanost – omogućuje metode za operacije na mrežama kao što su HTTP dohvat i postavljanje i TCP poslužitelj – klijent komunikacija

(Libgdx, 2017)

3.3 Adobe Illustrator i Adobe After Effects

Adobe Illustrator je vektorski grafički uređivač, odnosno vektorski baziran računalni program za crtanje, kojega je razvila američka tvrtka Adobe Systems.

Adobe After Effects je program, razvijen i izdan od strane američke tvrtke Adobe Systems. Program se koristi za vizualne efekte, pokretne grafike i tipografiju. (Adobe, 2018)

Adobe programi korišteni su za izradu dizajna igre, odabrani su na temelju preporuke i istraživanja autorice. Za razvijanje dizajna korištena je probna verzija oba programa.

3.4 Udemy

Udemy služi kao platforma koja omogućava instruktorima izgradnju online tečajeva o temama po njihovom odabiru. Pomoću Udemyjevih alata za razvoj tečaja mogu učitati videozapise, PowerPoint prezentacije, PDF datoteke, zvučne zapise, zip datoteke i klase uživo za stvaranje tečajeva. Instruktori se također mogu angažirati i komunicirati s korisnicima putem internetskih zajednica.

Nude se tečajevi više kategorija, uključujući informatiku. Udemy također nudi Udemy for Business, omogućujući tvrtkama pristup ciljanom paketu od preko 2.000 tečajeva za teme poslovanja.

Za potrebe ovog rada korišten je tečaj „*The Complete LibGDX Game Course Using Java*“ koji se sastoji od 92 lekcije koji je prilagođen početnicima u LibGDX svijetu.

Ciljevi tečaja su: (Udemy, 2018)

- Korištenje Jave za rješavanje problema
- Razumijevanje koncepta objektno orijentiranog programiranja
- Razumijevanje rada LibGDX-a
- Razumijevanje rada Box2D
- Kreiranje izbornika LibGDX
- Rad sa klasama zvuka
- AnimiranjeSpriteova
- Korištenje LibGDX akcija
- Razumijevanje fizike

3.5 Patreon

Patreon je platforma za članstvo koja pruža kreativne alate za pokretanje usluge pretplate, kao i načine za stvaranje odnosa i pružanje ekskluzivnih iskustava svojim pretplatnicima ili "pokroviteljima".

Patreon je popularan među snimateljima videa za YouTube, web stripovima, piscima, podcasterima, glazbenicima i drugim kategorijama kreatora koji redovito objavljuju internetske sadržaje. Omogućuje umjetnicima da izravno dobivaju sredstva od svojih obožavatelja ili pokrovitelja, na temelju ponavljanja ili po umjetničkom djelu.

Mark Rise, kreator stranice Gigantic snima tutorijale o dizajnu likova i nivoa koristeći Adobe Illustrator. Kanal je prilagođen početnicima u razvoju igara i dizajna koji žele da lak i učinkoviti način dizajnirati svoju igru. Mark za Patreon pretplatnike daje module s kojima se stvara dizajn te objašnjenja korak po korak kroz videa. Osim za Ilustrator dostupni su i tečajevi animacije koristeći Adobe After Effects.

4. DOKUMENT DIZAJNA IGRE

Dokument dizajna igre (*GDD – Game design document*) je vrlo deskriptivni dokument dizajna i životnog ciklusa igre. Dokument je izrađen od strane razvojnog tima kao rezultat suradnje između svojih dizajnera, umjetnika i programera kao vizija koja se koristi tijekom cijelog procesa stvaranja igre.

Dokument može sadržavati tekst, slike, dijagrame, konceptualne umjetnosti ili bilo koji medij kako bi ilustrirao odluke o dizajnu. Iako je zahtjev mnogih tvrtki, GDD nema postavljen standardni obrazac. Svaki razvojni programer ili razvojni tim može imati svoj GDD obrazac.

U nastavku slijedi skraćeni GDD obrazac za razvoj igre Olivia.

Analiza igre

Olivia, edukativna igra s tematikom prikupljanja, obrade i konzumiranja maslina. Svrha igre je da se igrač zabavi i istovremeno uči činjenice o maslinama. Trajanje igre je desetak minuta do pola sata.

Misija

Igra nazvana Olivia namijenjena je djeci posjetitelja kuće maslinovog ulja u Puli. Sastoji se od dva nivoa koja su vrlo jednostavna za igru. Edukativnog je karaktera te je prilagođena djeci svih uzrasta.

Žanr

Edukativna dječja igra

Platforme

Desktop i Android

Ciljana skupina

Igra je primjerena za sve uzraste, najviše djecu od 4 do 15 godina.

Jezik sučelja igre

Engleski jezik

Priča i likovi

Glavni lik igre je djevojčica Olivia koja u svojoj uzbudljivoj avanturi uči o maslinama te procesu prerade maslina.

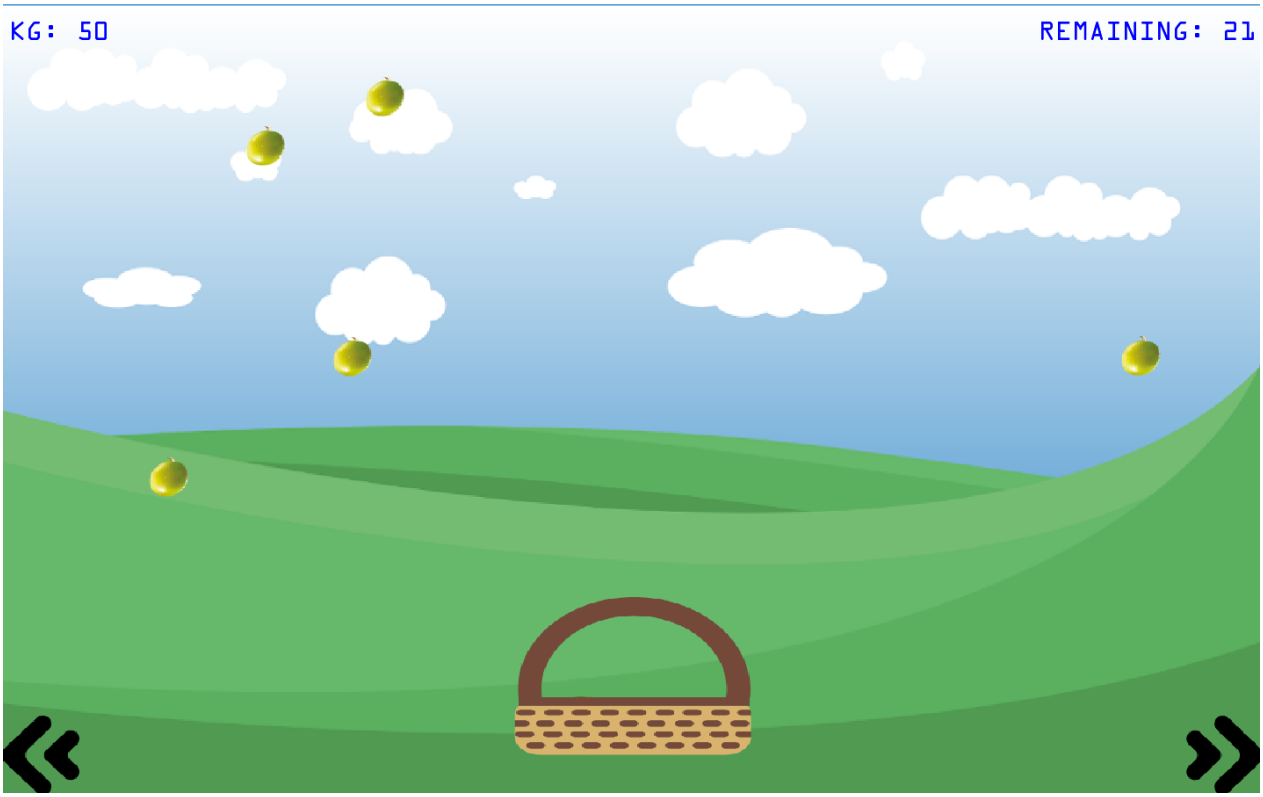
Dizajn nivoa

1. FALLING OLIVES

Igra branja i sakupljanja maslina. Skupljaju se masline koje padaju sa stabla pomicanjem košare lijevo ili desno. Svaka sakupljena maslina predstavlja deset kilograma ubranih maslina. Nakon pada 30 maslina sve prikupljene se pretvaraju u ulje. U lijevom kutu svijeta igre prikazuje se sakupljen broj bodova a u desnom kutu prikazuje se broj preostalih maslina.

Sučelje nivoa *Falling olives* vrlo je jednostavno te pruža ugodno korisničko iskustvo. Sučelje se sastoji od 5 objekata: pozadina, košara, masline, strelica lijevo i strelica desno. Za navigaciju po ekranu potrebno je koristiti strelice.

Slika 5: Falling olives - gameplay



Izvor: Snimka zaslona autorice

Nakon završene igre pojavljuje se ekran na kojemu piše koliko kilograma maslina je sakupljeno te koliko prosječno litara ulja možemo za to dobiti. Litre ulja se računaju na način da se na ekran ispiše prikupljena kilaža podijeljena sa 100 te rezultat pomnoži s 15 (u prosjeku 100 kilograma maslina daje 15 litara ulja, što ovisi o vrsti, načinu i vremenu branja).

Slika 6: Falling olives - rezultat

Izvor: Snimka zaslona autorice

2. MEMORY

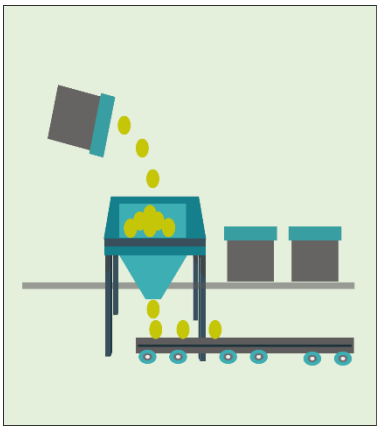
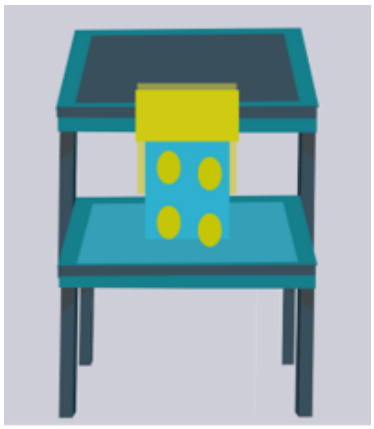
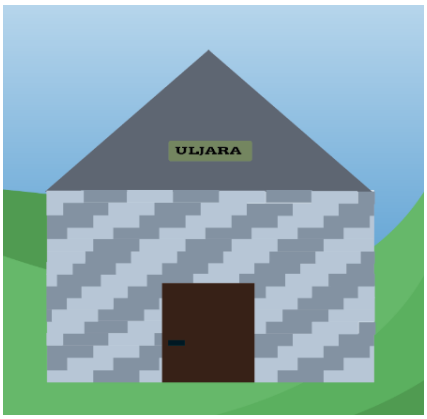
Memory je klasična igra pamćenja. Igra pamćenja vježba koncentraciju djece. Sastoji se od 6 parova sličica. Svi motivi su povezani s maslinama te njihovom obradom. Igra završava kada se sve sličice upare i ostanu otkrivene.

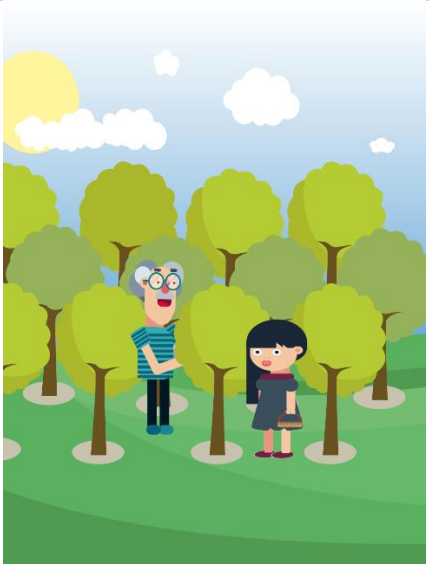
Slika 7: Memory - gameplay

Izvor: Snimka zaslona autorice

Memory sličice posebno su dizajnirane za igru Olivia te prikazuju motive vezane uz masline. Sljedeća tablica prikazuje motive kartica te njihova edukativna objašnjenja.

Tablica 12: Memory - kartice i opisi

Sličice	Opis
	*Mljevenje maslina* Maslinini plodovi se moraju prozračivati od prašine, te odvojiti od grančica i lišća. Nakon toga se plodovi peru hladnom vodom radi odstranjivanja ostalih mogućih nečistoća. Plodovi zatim prolaze kroz vibrirajuće rešetke kako bi se plodovi odvojili od vode i ostalih nečistoća.
	*Pranje maslina* Za dobivanje homogene mase potrebno je drobiti i usitniti plod masline koji će biti spreman za daljnju obradu. Mljevenje plodova omogućava oslobađanje kapljica ulja iz staničnih vakuola. Mljevenje se obavlja uz pomoć dva tipa strojeva: • kamenim mlinovima • mlinovima čekićarima
	*Uljara* Uljara je pogon za proizvodnju ulja u kojem se prerađuju plodovi maslina u maslinovo ulje.

	*Branje maslina* Čovjek bere maslinu više od 5 tisuća godina, ali još nije proniknuo u tajnu njezina dozrijevanja. Ispitivanja su pokazala da isto stablo svake godine na isti datum nema istu količinu i istu kvalitetu ulja. Količina ulja u plodu ovisi i o količini oborina u vrijeme dozrijevanja.
	*Košara maslina* Maslina (uljika, lat. Olea europaea) ime je za suptropsku zimzelenu biljku, porodice maslina (Oleaceae). Maslina razvija stablo, koje je nepravilno, kvrgavo i razgranato. Listovi su kožnati i ovalni, dok je boja listova na naličju tamno zelene boje, dok je donja strana lista bjelkasto - srebrne boje. Kada je u cvatu, maslina razvija bijele cvjetove u grozdovima, a plod je ovalnog oblika tamnozelene do crne boje.
	*Maslinovo ulje* Proces proizvodnje maslinovog ulja odvija se u dvije faze. U prvoj fazi se zdravi plod masline drobi gnječenjem, a potom se u drugoj fazi tiještenjem smjese zdrobljenih plodova cijedi čisto maslinovo ulje. Maslinovo ulje tradicionalno je temeljno ulje na području Mediterana, koristi se u gotovo svim kulinarskim aplikacijama koje zahtijevaju izvor masnoća.

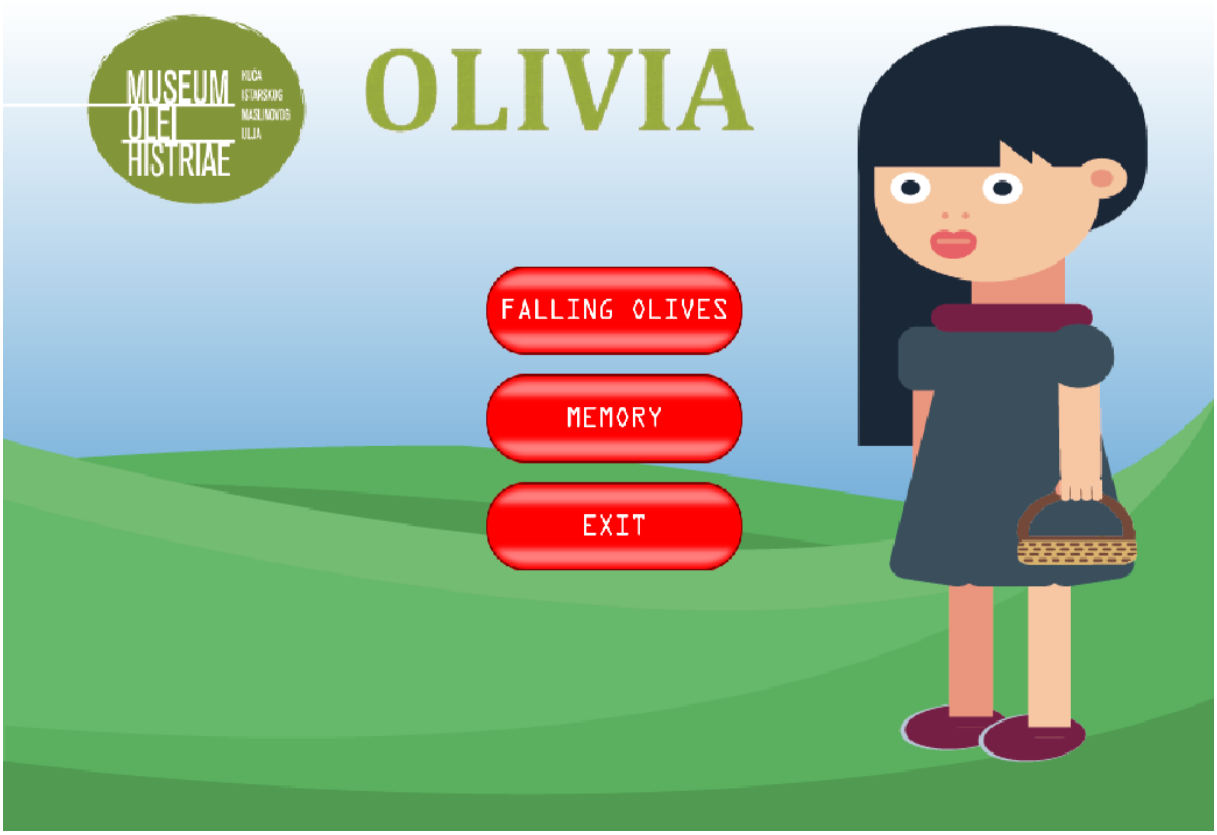
Izvor: rad autorice

Korisničko sučelje

Korisničko sučelje je način interakcije igrača sa igrom. U igri Olivia korisničko sučelje je jednostavno i u potpunosti primijenjeno korisniku, sastoji se samo od dizajna te tri gumba za interakciju:

- Falling olives
- Memory
- Exit

Slika 8: Olivia - korisničko sučelje igre



Izvor: Snimka zaslona autorice

5. RAZVOJ IGRE

Razvoj igara je proces stvaranja video igre. U razvoj igre uključeni su razvojni programeri igre. Tradicionalne komercijalne igre za računala i konzole obično financira izdavač te njihova izrada može potrajati godinama dok za igre financirane od strane pojedinca treba manje vremena. Neovisna industrija igara u posljednjih nekoliko godina znatno je povećala rast novih online distribucijskih sustava, kao što su Steam i Uplay, kao i tržište mobilnih igara, kao što su Android i iOS.

5.1 Postavljanje projekta

Da bi se krenulo razvijati igre u LibGDX programskom okviru potrebno je skinuti LibGDX postavke projekta sa linka. Otvorit će se sljedeća aplikacija

Slika 9: LibGDX postavke projekta

Izvor 3: https://LibGDX.badlogicgames.com/documentation/gettingstarted/Creating%20Projects.html

U postavkama treba postaviti sljedeće:

- **Name** – Ime projekta
- **Package** – Java paket za aplikaciju, obično domena koju imate
- **Game class** – Ime glavne klase
- **Destination** – Folder u kojeg se sprema projekt
- **Android SDK** – Putanja do Android SDK
- **SubProjects** – Označiti platforme za koje se radi igra
- **Extensions** – Označiti ukoliko se integriraju neke od platformi. Platforme su opisane u poglavlju 2.4.

LibGDX projekti generirani pomoću alata za postavljanje imaju modularniji izgled. Raspored će se malo promijeniti, ovisno o ciljanim platformam, ali općenito izgledaju vrlo slično.

Nakon što je projekt generiran imat će sljedeće konfiguracijske datoteke

- ***settings.gradle*** – definira pod-module: jezgreni, desktopa, android, html i ios
- ***build.gradle*** – glavna Gradle datoteka – definira ovisnosti i dodatke
- ***gradlew*** – skripta koja pokreće Gradle na Unix sustavima
- ***gradlew.bat*** - skripta koja pokreće Gradle na Windows sustavima
- ***gradle*** – lokalni Gradle
- ***local.properties*** – Gradle datoteka, definira put do Android SDK lokacije
- **/core**
    - ***build.gradle*** – Gradle datoteka za primarni projekt
    - ***src/*** - folder za kod igre
- **/ desktop /android / html / ios**
    - ***build.gradle*** – Gradle datoteka za / desktop /android / html / ios projekt
    - **src/** - sadrži launcher klasu

(Libgdx, 2017)

Gradle je sustav za upravljanje ovisnošću, jednostavan način za privlačenje biblioteka trećih strana u projekt bez potrebe za pohranjivanjem biblioteka. Omogućuje dodavanje, uklanjanje i promjenu verzije biblioteke treće strane te izmjene konfiguracijske datoteke. Ovaj sustav je odgovoran za sastavljanje, testiranje, pokretanje i pakiranje programa.

5.1.1 Resursi korišteni u igri Olivia

Za opis svih korištenih resursa kao što su slike, zvukovi, skinovi, teksture korištena je klasa *Asset Descriptor*.

Slika 10: Prikaz klase Asset Descriptor

```
public final class AssetDescriptors {

    public static final AssetDescriptor<BitmapFont> FONT =
            new AssetDescriptor<~>(AssetPaths.SCORE_FONT, BitmapFont.class);

    public static final AssetDescriptor<TextureAtlas> GAME_PLAY =
            new AssetDescriptor<~>(AssetPaths.GAME_PLAY, TextureAtlas.class);

    public static final AssetDescriptor<Sound> PICKUP =
            new AssetDescriptor<~>(AssetPaths.PICKUP, Sound.class);

    public static final AssetDescriptor<Skin> SKIN =
            new AssetDescriptor<~>(AssetPaths.SKIN, Skin.class);

    public static final Array<AssetDescriptor> ALL = new Array<~>();
```

Izvor: Snimka zaslona autorice

Klasa *Asset Descriptor* opisuje resurse koje treba učitati u igru s nazivom i tipom datoteke. U klasi *Asset Paths* pohranjene su putanje do resursa koji odgovaraju klasi *Asset Descriptor*.

Slika 11: klasa Asset Paths

```
public final class AssetPaths {

    public static final String SCORE_FONT = "ui/fonts/score.fnt";

    public static final String GAME_PLAY = "gameplay/gameplay.atlas";

    public static final String PICKUP = "sounds/pickup.wav";

    public static final String SKIN = "ui/skin.json";
```

Izvor: Snimka zaslona autorice

Klasa RegionNames sadrži konstante s opisima regija unutar atlasa slika.

Slika 12: Klasa Region Names

```
public final class RegionNames {
    public static final String BACKGROUND = "pozadina";
    public static final String BACKGROUND_OVER = "background";
    public static final String BASKET = "basket";
    public static final String MENU_BG = "pocetni";
    public static final String OLIVE = "olive";
    public static final String MEMORY_BG = "memcard";
    public static final String BACKGROUND_MEMORY = "pozadinaM";
    public static final String CARD1 = "card1";
    public static final String CARD2 = "card2";
    public static final String CARD3 = "card3";
    public static final String CARD4 = "card4";
    public static final String CARD5 = "card5";
    public static final String CARD6 = "card6";
    public static final String ARROW_LEFT = "arrow-left";
    public static final String ARROW_RIGHT = "arrow-right";
```

Izvor: Snimka zaslona autorice

U klasi *GameConfig* nalaze se vrijednosti konstanta koje se koriste u igri. Kroz klasu definiraju se dimenzije svijeta igre, početnog ekrana igre te ekrana unutar nivoa. Za dimenzije svijeta definirano je slijedeće:

- desktop veličina prozora u pikselima
- širina unutarnjeg prozora za prikaz teksta
- središnja os prostora

Za nivo *Falling Olives* definirane su sljedeće dimenzije:

- **Košara** (Basket) – Za košaru je definirana visina, širina, njezina početna pozicija u sredini ekrana i brzina pomicanja
- **Masline** (Olives) – Za stvaranje maslina definiran je pomak od ruba ekrana, veličina maslina, bodovi koje donosi svaka prikupljena maslina, broj maslina koje će padati te brzina pada masline.
- **Strelice** za pomicanje košare (Arrows) – strelicama za pomicanje košare po svijetu igre definirana je veličina.

Za nivo *Memory* definirane su iduće dimenzije:

- Vrijeme prije skrivanja/okretanja nakon neuspješnog uparivanja kartica
- Brzina sužavanja kartica po X osi za privid animacije okretanja kartica
- Broj kartica koje će se prikazivati u retku

5.2 Obrasci dizajna

Objektno orijentirano programiranje – ili kraće OOP – je jedan od mogućih pristupa programiranju računala. Za razliku od ostalih pristupa, u kojima je težište na akcijama koje se vrše na podatkovnim strukturama, ovdje je težište na projektiranju aplikacije kao skupa objekata koji izmjenjuju poruke između sebe.

Programiranje orijentirano klasama je stil objektno orijentiranog programiranja u kojem se nasljeđivanje postiže definiranjem klase objekta. Najpopularniji i razvijeniji model OOP-a je model koji se temelji na klasi, za razliku od modela temeljenog na objektu. U ovom su modelu objekti entiteti koji kombiniraju stanje (tj. Podatke), ponašanje (tj. Postupci ili metode) i identitet (jedinstveno postojanje među svim ostalim objektima). Struktura i ponašanje objekta određuje klasa, koja je definicija ili nacrt svih objekata određene vrste.

Oblik dizajniranja (*Creational design pattern*) podrazumijeva obrasce dizajna koji se bave mehanizmima stvaranja objekta, pokušavajući stvoriti objekte na način prikladan za situaciju. Osnovni oblik stvaranja objekta može rezultirati problemima dizajna ili u kompleksnosti dizajna. Oblik dizajniranja rješava taj problem tako što upravlja stvaranjem objekta. (SourceMaking, 2018)

5.2.1 Factory

U programiranju orijentiranom klasama, *factory* obrazac dizajna koristi *factory* metode za rješavanje problema stvaranja objekata bez potrebe za određivanjem točne klase objekta koji će se stvoriti. To se postiže stvaranjem objekata pozivom *factory* metode - bilo specificirano u sučelju i implementirano od strane klasa koje nasljeđuju, ili implementirano u osnovnoj klasi. (Gamma, et al., 1994)

Slika 13: Klasa Entity Factory

```
public EntityFactory(AssetManager assetManager) {
    this.assetManager = assetManager;
    init();
}

private void init() {
    //inicijalizacija pool-a za masline
    olivePool = Pools.get(Olive.class,  max: 10);
}
```

Izvor: Snimka zaslona autorice

U klasa *Entity factory* je klasa za kontrolirano instanciranje objekata. U njoj se inicijalizira *pool* za masline. *Pool* služi za efikasnije kreiranje objekata maslina tako što štedi memoriju. Masline se stvaraju na ekranu koristeći *random* funkciju - nasumično postavljanje x koordinate masline iznad vrha ekrana. Masline imaju i nasumičnu brzinu pada. Kada je maslina kreirana sprema se u polje (*array*) maslina.

Kod kreiranja maslina koristi se pool umjesto klasičnog načina kreiranja objekata. Klasa za maslinu nasljeđuje osnovni entitet iz *util* modula i *Poolable* za bolju kontrolu memorije prilikom stvaranja.

Osim maslina, u klasi *Entity factory* kreira se i pozadina te košara za masline. Košara se stvara korištenjem dimenzija iz klase *Config* opisane u odlomku 4.1. Pozadina se kreira korištenjem *util* modula *parallax*.*ParallaxLayer* te njegove funkcije *createbackground().* Kao kod stvaranja košare, dimenzije pozadine su definirane u klasi *Config*.

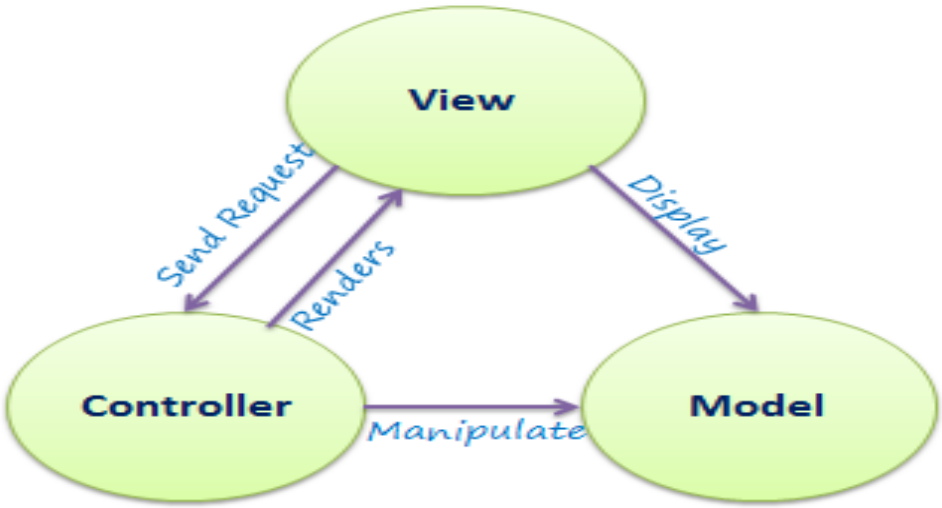
5.2.2 Model – View – Controller (MVC)

MVC obrazac dizajna već dugo postoji u programskom inženjerstvu. Većina programskih jezika koriste MVC s malom varijacijom, ali konceptualno ostaje isti. Obrazac dizanja MVC sastoji se od tri komponente: (Tutorials Teacher, 2018)

- **Model** - predstavlja oblik podataka i poslovne logike. On održava podatke aplikacije. Model objekti preuzimaju i pohranjuju stanje modela u bazi podataka
- **Prikaz** *(View)* - korisničko sučelje. Omogućuje prikaz i izmjenu podataka.
- **Kontroler** *(Controller)* - obrađuje zahtjev korisnika. Korisnik šalje inpute kroz prikaz koji podiže odgovarajući URL zahtjev. Kontrelor zatim prikazuje taj zahtjev s odgovarajućim modelom podataka.

Sljedeća slika prikazuje interakciju između modela, prikaza i kontrolera.

Slika 14: MVC obrazac dizajna



Izvor: http://www.tutorialsteacher.com/mvc/mvc-architecture

U igri Olivia MVC obrazac dizajna korišten je za nivo Falling Olives.

Kontroler klasa (*GameController*) za padajuće masline koji služi za aktivaciju igre prilikom prvog dodira. Klasa za prikaz (*GameRenderer*) sastoji se od inicijalizacija objekata, polja i funkcija za prikaz teksta, maslina, košare, pozadine i strelica za kretanje. Model se nalazi u klasi *GameScreen* koja služi kao model podataka svega u igri Falling olives i sadrži glavninu logike koja će biti opisana kasnije kroz rad.

5.3 Grafika

Grafički modul pruža informacije o trenutnom zaslonu uređaja i prozoru aplikacije, kao i informacijama i pristupu trenutnom kontekstu *OpenGL*. Na taj se način mogu pronaći svi podaci o veličini zaslona, gustoći piksela i svojstvima okvira kao što su dubine boja, dubine matrice i slično. Kao i kod ostalih uobičajenih modula, pristup se osigurava putem statičkih polja LibGDX klase.

Slika koja je dekodirana iz njenog izvornog formata (npr. PNG) i prenesena u GPU naziva se tekstura. Za crtanje teksture, potrebno ju je opisati geometrijom, crta se tako da svaki vrh u geometriji odgovara teksturi. Na primjer, geometrija može biti pravokutnik i tekstura se može primijeniti tako da svaki kut pravokutnika odgovara kutu teksture. Pravokutnik koji je podskup teksture naziva se površinom teksture.

Nakon što je napravljena tekstura, tada se geometrijama dodjeljuje OpenGL. Veličina i položaj teksture na zaslonu određuju se geometrijom i načinom na koji je konfiguriran OpenGL preglednik. Mnoge 2D igre konfiguriraju prikaz igre tako da odgovara rezoluciji zaslona. To znači da je geometrija navedena u pikselima, što olakšava crtanje tekstura u odgovarajućoj veličini i položaju na zaslonu.

Teksture se čuvaju u *TextureAtlas*-u. Budući da se pozivi za crtanje na ekran ponavljaju veliki broj puta u sekundi, dobra je praksa zapakirati sve korištene teksture iz zasebnih datoteka u jednu. TextureAtlas služi za mapiranje pozicija zasebnih tekstura u zajedničkoj slici i omogućava njihovo učinkovitije i brže crtanje.

Kroz iduća potpoglavlja opisane su LibGDX klase za crtanje koje se najčešće koriste te u poglavlju 5.3.5 primjena nekih od klasi u igri Olivia.

5.3.1 Texture

Tekstura obuhvaća standardnu *OpenGL ES* teksturu. OpenGL ES je podskup aplikacijskog programskog sučelja računalnog grafičkog prikazivanja OpenGL (API[4]) za prikazivanje 2D i 3D računalne grafike koje koriste igre, obično hardverski ubrzane pomoću grafičke (GPU). Namijenjen je ugrađenim sustavima kao što su pametni telefoni, tableti, konzole i PDA uređaji. OpenGL ES je najčešće postavljen 3D grafički API u povijesti.

Tekstura mora biti vezana putem *GLTexture.bind ()* metode kako bi se mogla primijeniti geometrija. Tekstura će biti vezana za trenutno aktivnu strukturu teksture navedenu putem *GL20.glActiveTexture (int).* Tekstura mora biti uništena *(engl disposed)* kada se više ne koristi.

Klasa texture obuhvaća i klasu Pixmap koja predstavlja sliku u memoriji. Ima širinu i visinu izraženu u pikselima te Pixmap Mapu koja navodi broj i redoslijed komponenti boja po pikselu. (Stemkoski, 2015)

Neke od korisnih metoda klase Texture su: (Libgdx, 2017)

- ***Texture(int width, int height, Pixmap.Format format)*** – kreiranje teksture
- ***draw(Pixmap pixmap, int x, int y)*** – crta danu mapu piksela na poziciju x,y
- ***get funkcije*** – vraća dubinu, visinu, podatke teksture, širinu i slično

[4] Aplikacijsko programsko sučelje

5.3.2 Shape2D

Prilikom kreiranja resursa potrebno je implementirati i jednu od podklasa klase Shape2D koja olakšava detekciju kolizije. Enkapsulira 2D oblik definiran kutnom točkom u lijevom donjem kutu i njegovim dimenzijama x (širina) i y (visina). (Stemkoski, 2015) :

- **Rectangle** - enkapsulira 2D pravokutnik definiranu kutnom točkom u lijevom donjem kutu s i koordinatama x (širina) i y (visina).
- **Polygon** - *e*nkapsulira 2D poligon definiran njezinim vrhovima u odnosu na točku izvora
- **Ellipse** – kreira odgovarajuću 2d elipsu temeljenu na klasi Circle
- **Circle** – kreira odgovarajući 2d krug
- **Polyline** - kontinuirana linija koja se sastoji od jednog ili više segmenata krajnjih točaka svakog segmenta.

Neke od metoda klasa Rectangle / Polygon / Ellipse / Circle / Polyline su: (Libgdx, 2017)

- ***Konstruktor*** – Konstruira novi objekt na određenim koordinatama i s određenim dimenzijama
- ***fitInside*** – postavlja jedan objekt u drugi uz održavanje omjera slike
- ***get funkcije*** – funkcija može vratiti visinu, širinu, koordinatu x, koordinatu y, omjer slike, centar objekta i slično.

5.3.3 Batch

Batch se koristi za crtanje 2D pravokutnika koji se odnose na teksturu. Klasa će složiti naredbe za crtanje i optimizirati ih za obradu GPU-a. Da biste nacrtali nešto s *Batchom*, najprije morate pozvati metodu *start()* koja će postaviti odgovarajuće stanja renderiranja. Kada završite s crtanjem morate pozvati metodu *end()* koja će transformirati projekciju onako kako želite.

Sve naredbe za crtanje serije izvode se u koordinatama zaslona. Koordinatni sustav zaslona ima x-os koja pokazuje desno i y os koja pokazuje prema gore.

Batch mora biti uništen *(engl disposed)* ako se više ne koristi. (Stemkoski, 2015)

Neke od metoda Batch klase su: (Libgdx, 2017)

- ***begin()*** – Postavlja Batch za crtanje
- ***draw(Texture texture, float x, float y)*** – crta pravokutnik u donjem lijevom kutu sa visinom i širinom teksture
- ***end()*** – kraj renderinga

5.3.4 Sprite

Klasa *Sprite* sadrži polja za spremanje koordinata, tekstura i dodatnih informacija kao što su rotacija i skalabilnosti. Uz pomoć metoda te klase može se stvoriti detekcija kolizije. Svako od tih polja koristi standardne funkcije za čitanje i postavljanje (eng. get and set). Sprite je uvijek pravokutnik i ima svoju poziciju (x,y) i reprezentativnu matricu. (Stemkoski, 2015)

Neke od korisnih metoda klase Sprite su: (Libgdx, 2017)

- ***Sprite(Texture texture)*** – Kreira Sprite sa visinom i širinom jednakom veličini tekture.
- ***translateX() i translateY ()*** - koje mijenjanju vrijednosti x i y koordinate Sprite objekata te ga crtaju na postavljenim koordinatama.
- ***Get funkcije*** – vračaju boju, os x ili y, rotaciju, skalu, visinu, širinu i slično
- **Set funkcije** – postavljaju granice, centar pravokutnika, boju, ljestvicu, veličinu itd.

5.3.5 Primjena klasa za crtanje u igri Olivia

U nivou igre ***Memory*** klasa *TextureRegion* koristi se za kreiranje karti a klasa *Batch* koristi se za iscrtavanje ispravno okrenutih karti na ekran.

Unutar klase *MemoryCard* deklarirane su dvije boolean varijable

- isCovered - određuje je li kartica okrenuta ili skrivena
- isChecked - određuje je li trenutna kartica već uparena sa svojim duplikatom

Kreira se novi *TextureRegion* objekt koji se sastoji od dvije varijable

- face – prava slika kartice, pokazuje se ukoliko je pronađen par
- back – slika kartice koja je uvijek ista, da se ne mogu znati parovi unaprijed

Zatim se kreiraju dvije varijable koje prate početne dimenzije karte, one se dobivaju funkcijama:

- getScaleX()
- getScaleY()

Zatim se na ekran crtaju karte ovisno o varijablama *isCovered* i *isChecked*. Ako je karta pokrivena (*isCovered*=***true***) na ekran se crta stražnja strana karte (*back*) inače se crta prednja strana karte (*face*).

Koristeći funkciju *batch.draw* crta se karta (*region*) sa koordinatama x i y, određenom širinom i visinom te rotacijom.

U nivou igre ***Falling Olives*** u klasi *BasketInputController*, kreirana su dva objekta klase Rectangle

- Left arrow key – strelica u lijevom kutu ekrana
- Right arrow key – strelica u desnom kutu ekrana

Strelice služe za pomicanje košare po ekranu. Pravokutnik se kreira za svaku od strelica koje su prikazane na rubovima ekrana. Pomicanje je definirano za desktop i mobilnu verziju. Funkcija provjerava koja je od tipki na tipkovnici (desna ili lijeva strelica) pritisnuta te na temelju toga poziva funkciju za pomak u desnu ili lijevu stranu.

5.4 Scene2d

Scene2d je grafikon 2D scene za gradnju aplikacija i korisničkog sučelja pomoću hijerarhije Actora. U Scene2D-u nalaze se samo Stage, Actor i Group klase. Za stvaranje gumba, tablice te ostalih elemenata korisničkog sučelja, treba koristiti klase koje nasljeđuju klasu Actor. Korisničko sučelje igre obično prikazuje informacije o svijetu igara ili statusu igrača koristeći kombinaciju grafike i teksta.

Općenito, elementi korisničkog sučelja crtaju se na pozadinu igre. Scena se crta na način da se najprije pojavljuju pozadinske slike, zatim glavni entiteti, te nakon toga tekst i ostali elementi korisničkog sučelja. Jednostavnija metoda je napraviti više objekata te ih zatim prikazati u željenom redoslijedu. (Stemkoski, 2015)

5.4.1 Stage

Stage zaprima ulazne podatke te po tome aktivira određene Actore. Postavljanjem Viewporta (5.4.1.1.) kontrolira koordinate korištene u igri te postavlja kameru koja pokreće nivoe. Neke od funkcija klase Stage su sljedeće (Libgdx, 2017)

- **Gdx.input.setInputProcessor** – primanje ulaznih podataka
- **Act –** poziva *Actor.act* metodu za svaki *Actor* objekt na *Stageu*
- **addAction** – dodaje akciju *Actoru*
- **addActor** – dodaje *Actor* objekt
- **hit (float stageX, float stageY, boolean touchable**) – vraća *Actor* objekt na određenim koordinatama

5.4.1.1 Viewport

Upravlja kamerom i određuje kako se koordinate svijeta prenose na i sa zaslona i podešava omjer slike (bez obzira je li slika raširena). Prikazni prozor također pretvara koordinate zaslona na i od koordinate nivoa. Viewport se definira konstruktorom koristeći funkciju setViewport. Neke od metoda klase Viewport su: (Libgdx, 2017)

- Metode za dohvaćanje (get) – kamera, visina i širina ekrana, visina i širina svijeta
- Metode za postavljanje (set) – granice svijeta, visina ekrana, veličina ekrana. Postavljanje koordinata,

5.4.1.2 Klasa Camera

Klasa *Camera* omogućuje pozicioniranje programa unutar svijeta igre. Kamera se postavlja u središnji dio svijeta igre. X koordinata kamere fotoaparata mora biti postavljena na *Width / 2* piksela od lijeve i desne granice svijeta. Dijeli se sa 2 zato što je kamera u središtu zaslona, pa stoga treba prostora samo pola širine sa svake strane. Slično tome, y koordinata mora biti barem vidljiva *na Height / 2* od gornje i donje granice svijeta. (Stemkoski, 2015)

Neke od metoda klase Camera su: (Libgdx, 2017)

- Postavljanje koordinata
- Projekcija vektora
- Rotacija
- Transformacija
- Ažuriranje

5.4.2 Actor

Actor je čvor 2D scene. Actor ima položaj, pravokutnu veličinu, podrijetlo, veličinu, rotaciju, indeks Z i boju. Položaj je nerotiran i neskaliran u donjem lijevom kutu *Actora*. Položaj je u odnosu na roditelja

Actor ima popis akcija koje se primjenjuju na njega. Akcije se koriste za promjenu prezentacije *Actora* (premještanje, promjenu veličine itd.). *Actor* ima dvije vrste listenera koji su povezani sa praćenjem događaja u igri. (Stemkoski, 2015)

Neke od korisnih metoda klase Actor su: (Libgdx, 2017)

- ***addCaptureListener(EventListener listener)*** – dodaje listenera koji se obavještava samo tijekom faze snimanja
- ***addListener(EventListener listener***) – metoda koja se koristi za identificiranje događaja – najčešće klika miša, dodira, ulaza s tipkovnice..
- ***draw*** funkcija - crta Actora
- ***get funkcije*** – vraća visinu, širinu, listenere, ime, roditelja, level, objekt, x i y os, itd.

- ***hit(float x, float y, boolean touchable)*** – vraća najdaljeg Actora koji sadrži specificiranu točku koja može biti dodirljiva i vidljiva ili nula ako nijedan Actor nije u točci
- ***set funkcije*** – postavljanje koordinata, visine, širine i z indeksa.

5.4.2.1 Klase za uređivanje korisničkog sučelja

Za uređivanje korisničkog sučelja koriste se LibGDX klase koje nasljeđuju klasu Actor: (Libgdx, 2017)

- Skin
- Table
- Button

Klasa Table

LibGDX ima klasu *Table* koja pojednostavljuje proces uređivanja elemenata korisničkog sučelja. Princip je sličan kao kod korištenja HTML tablica te su zbog toga jednostavne za uporabu.

Tablice se mogu oblikovati na način da se stvoreni objekt može odmah oblikovati pozivanjem jedne od sljedećih metoda klase Table: (Stemkoski, 2015)

- Vodoravno poravnanje
  - Left - Lijevo
  - Middle - Sredina
  - Right - Desno
- Okomito poravnanje
  - padLeft – razmak od lijeve strane ekrana
  - padRight – razmak od desne strane ekrana
  - padBottom – razmak od dna ekrana

Klasa Skin

Klasa *Skin* pohranjuje resurse za uporabu *widgeta* kao što su tekst, fontovi, boje i slično. Skin je odgovor na pitanje "Kako Scene2D Actors objekti dobiju stvarni, grafički oblik?". Definira se konstruktorom skin() te ima sljedeće metode:

- **add** – metode za dodavanje skina
- **dispose** – metoda za uklanjanje resursa koji se ne koriste
- **find** – vraća određeni skin
- **get, get all** – metode za dohvat boje, fonta, regije, sprita i slično

Klasa Button

Gumb je jedna od osnovnih kontrola korisničkog sučelja koja dobiva input od korisnika. Postoji više načina prilagodbe izgleda i ponašanja gumba kao i proširenja klase *Button*. Objekt *ButtonStyle* može pohraniti jednu ili više slika, od kojih će jedna biti prikazana, ovisno o trenutnom stanju gumba. Podaci o slici za elemente korisničkog sučelja moraju se pohraniti pomoću klase *Drawable* koja implementira sučelje koje se može nacrtati i ima metode koje mijenjaju veličinu i nacrtaju sliku kako bi se uklapale u igru. Najlakši način za inicijalizaciju takvog objekta je korištenje klase *Skin*, koja osim izvrsnog načina upravljanja resursima, sadrži mnoge metode za pretvaranje slikovnih podataka.

5.4.2.2 Vizualni efekti

Mnogi vizualni efekti mogu se postići kontinuiranim promjenama vrijednosti povezanih s entitetom igara, kao što su sljedeći: (Stemkoski, 2015)

- Efekt kretanja može se stvoriti promjenom vrijednosti koordinata položaja.
- Efekt vrtnje može se stvoriti promjenom vrijednosti rotacije
  Efekt boje u boji može se stvoriti promjenom vrijednosti crveno / zeleno / plave komponente u boji.
- Efekt stvaranja i nestajanja može se postići mijenjanjem alfa vrijednosti (transparentnosti)

Klasa Action

Navedeni efekti mogu se dodati u igru koristeći klasu *Action* koja se može dodati klasi *Actor*. Za kreiranje akcija koriste se metode koje su raspoložive u klasi *Action*. Statične praktične metode za korištenje zajedničkih akcija namijenjene su statičnom uvozu. (Stemkoski, 2015)

Kroz klasu Action imamo sljedeće mogućnosti: (Libgdx, 2017)

- Kreiranje akcije
- Postavljanje listenera
- Alpha akcija – promjena vidljivosti
- Akcija promjene boja
- Akcija kašnjenja (eng. Delay)
- Akcija pomicanja
- Popis ostalih akcija nalazi se u LibGDX dokumentaciji

5.4.3 Group

Čvor 2D scene koji može sadržavati druge Actore.

Actori imaju z-redoslijed jednak redoslijedu kojim su ubačeni u grupu. Actori ubačeni kasnije bit će nacrtani nakon Actora dodanih ranije. Klasa se najčešće koristi za detekciju i usporedbu Actora u svijetu igre. Neke od metoda Group klase su:

- Act – ažuriranje Actora bazirano na vremenu
- addActor – dodaje Actor objekt
- addActorBefore – kreira Actor objekt prije određenog Actor objekta
- draw – crtanje grupe
- setStage – postavlja grupu na Stage

5.4.4 Scene2d klase u igri Olivia

Scene2d klase koriste se u svim segmentima igre Olivia – korisničkom sučelju te oba nivoa Memory i Falling Olives. Scene 2d klase su bolje od klasa grafike za izradu sučelja jer nije potrebno voditi računa o širini ekrana i samostalno računati pozicije raznih elemenata sučelja.

Korisničko sučelje igre Olivia kreirano je u klasi *MenuScreen.*

Postavljanje veličine prikaza i pozornice definirano je u funkciji show(). Kreiranjem viewporta (određuje kako se koordinate svijeta mapiraju na i sa zaslona) definiraju se dimenzije početnog ekrana koje su definirane u klasi *GameConfig*.

Olivia korisničko sučelje ima nekoliko elemenata

- Gumbi za ulaz u nivoe – kreiranje skina za prikaz gumba
- Pozadina – poziva se iz klase TextureRegion (poglavlje 5.3.1)

Da bi se gumbi ispravno prikazali kreirana je tablica (klasa *Table*) te su u nju dodana sva tri gumba logičnim redoslijednom. Tablica je zatim centrirana te proširena da zauzme cijeli zaslon.

Gumbi Falling Olives i Memory pozivaju istoimene funkcije za ulazak u nivoe.

U ***Memory*** nivou igre Olivia klasa *Action* se koristi za nekoliko animacija:

- Show – Akcija za okretanje karte na vidljivu (*face*) stranu. Slika se skalira po širini te se sužava prema centru. U trenutku kada je širina prve slike (*back)* 0, slika se mijenja u aktivnu sliku (*face)* te se skalira do zadane dimenzije. Takva izmjena slika daje privid da se slika okreće.
    - Akcijom je definirano kašnjenje – vrijeme prije skrivanja nakon neuspješnog uparivanja kartica
    - Trajanje rotacije – brzina sužavanja kartica po X osi za privid okretanja

- Hide – Akcija ne radi ništa ako je kartica pogođena. Ako kartica nije pogođena skrivamo ju po uzoru na akciju Show.

Da bi se akcija izvršila ispravno, da se kartica „okrene“ prije nego što se prikaže iduća slika koristi se posebno izrađena klasa MemoryTurningAction koja nasljeđuje Action klasu.

Slika 15: Akcijska klasa MemoryTurningAction

```
private class MemoryTurningAction extends Action {
    public boolean act(float delta) {
        //castanje korisnika akcije u mem. karticu da možemo izmijenit isCovered
        MemoryCard card = (MemoryCard) actor;
        card.isCovered = !card.isCovered;
        return true;
    }
}
```

Izvor: snimka ekrana autorice

U ***Falling olives*** nivou igre korištene su Scene2d klase za kreiranje sljedećih funkcija

Košara

- **blockBasketFromTheWorld** – zaustavlja košaru uz lijevi ili desni rub ekrana
- **restart** - postavljanje košare na početnu poziciju i pauza igre

Masline

- **updateOlives** – ažurira pozicije svake masline, masline koje su pale ispod razine ekrana brišu se i recikliraju
- **checkBasketWithOlivesCollision** – provjerava koristeći dva *Polygon* objekta da li se rubovi košare preklapaju sa rubovima maslina. Ukoliko se koordinate nacrtanih likova preklapaju – svira zvuk za sakupljenu maslinu, dodaju se bodovi u klasu *Config*, prikupljena maslina se briše i reciklira.
- **startLevel** – stvaranje maslina iz *Factory* klase

Ekran kraja igre

- Tablica za sve elemente ekrana (ispis tri reda teksta i gumb za izlaz)

5.5 Zvuk

Ubacivanje zvuka u igru je jednostavan postupak zahvaljujući ugrađenim LibGDX bibliotekama. Podržane vrste datoteka uključuju MP3, OGG i WAV. (Stemkoski, 2015)

LibGDX nudi dva sučelja za ubacivanje zvuka u igru a to su: (Libgdx, 2017)

- Sound - male audio datoteke koje se reproduciraju pri diskretnim događajima igre, npr. Sakupljanje maslina. Zvučni efekti obično su kratki (nekoliko sekundi ili manje), a odgovarajuće datoteke ne smiju biti veće od 1 MB. Varijabla za zvuk je *float* koja određuje koliko će glasno zvuk biti reproduciran (0 je tih, a 1 puna glasnoća). Jedan zvučni efekt može svirati više puta za redom.
- Music - Sučelje glazbe omogućeno je za dulje audio sekvence, kao što su pozadinska glazba ili zvukovi okoline.

Zvukovi i glazba trebaju biti uklonjeni iz memorije kada je igra gotova, što se može postići koristeći njihove predviđene metode uklanjanja (engl. dispose).

Klasa *Sound* korištena je u igri Olivia da bi se ubacio zvuk za nivo *Falling Olives:*

- *PICK UP* – zvuk koji se producira kada je maslina sakupljena

U klasi Sound Controller definirane su dvije varijable:

- *Asset manager* – za upravljanje i učitavanje zvuka
- *Sound* – definiran je zvuk *pick up*

Nakon kreiranja zvuka poziva se funkcija *pickup()* koja služi za puštanje zvuka točno kada se dogodi akcija sakupljanja masline.

Slika 16: Implementacija kontrole zvuka

```
public class SoundController {

    private final AssetManager assetManager;
    private Sound pickup;


    public SoundController(AssetManager assetManager) {
        this.assetManager = assetManager;
        init();
    }


    private void init() {
        pickup = assetManager.get(AssetDescriptors.PICKUP);
    }

    public void pickup() { pickup.play(); }
}
```

Izvor: Snimka zaslona autorice

6. PRILIKE ZA POBOLJŠANJE

Budući da je ova igra stvorena u svrhu istraživanja mogućnosti LibGDXa te je primjenjiva samo u kući maslinovog ulja u Puli, ima mnoštvo prilika za poboljšanje.

Dodavanje više jezika – igra je trenutno na **engleskom** jeziku. Engleski jezik odabran je jer je međunarodan i više-manje sve su igre na engleskom jeziku. Pretpostavka je da će većina posjetitelja znati barem nekoliko osnovnih riječi engleskog koje se koriste u igri. Odlična prilika za poboljšanje bi bila da igra bude na više jezika tako da se korisnici u njoj još lakše snalaze.

Izmjena postojećih nivoa – igra je dizajnirana da ima samo dva nivoa iste težine te je trenutno vrijeme za završetak igre manje od desetak minuta. U oba nivoa moguća su unaprijeđena i poboljšanja te su dolje navedena samo neka od njih:

- Memory – dodavanje veće baze kartica, broj kartica ovisan o želji igrača, osmisliti način sakupljanja bodova u igri (broj poteza, vrijeme…)
- Falling Olives – dodavanje zamki (padanje grana umjesto maslina), povećanje brzine i učestalosti maslina nakon riješenog nivoa, vremensko ograničenje…

Dodavanje novih nivoa – budući da se igra Olivia sastoji od više malih igara, može ih se još dodavati. Olivia je zamišljena kao igra u kojoj će svaki igrač pronaći nivo koji mu odgovara. Ideje za nove nivoe:

- *OliveNinja* – igra prerade maslina, veće masline padaju po ekranu te ih je potrebno „presjeći“ prstom. Svaka prerezana maslina označavala bi količinu ulja te bi se na završnom ekranu prikazala količina ulja.
- *Super Olivia* – Olivia prolazi kroz niz zamki pri branju maslina koje treba izbjeći. Putem saznaje korisne informacije o načinu branja maslina. Na kraju puta dolazi u uljaru gdje predaje masline koje je sakupila kroz svoju avanturu.

Dodavanje liste najviših rezultata (Highscores) – budući da će igru koristiti djeca posjetitelja kuće maslinovog ulja, za svaki od nivoa mogla bi se kreirati lista najvećih rezultata u koju bi oni pored svog rezultata mogli zapisati svoje ime.

Prilagodba za sve uređaje – Iako LibGDX podržava više platformi, igra je dizajnirana za desktop i android za tablete. Igru bi trebalo testirati na svim platformama.

Ažuriranje dizajna igre – dizajn igre nastao je proučavanjem alata Adobe Illustrator te je početnički napravljen. Igra bi se trebala redizajnirati sa više znanja i iskustva te je napraviti zanimljivijom igračima. U igru bi se trebale dodati i animacije radi trodimenzionalnog efekta dvodimenzionalne slike.

Alternativna igra – razviti alternativnu igru u programskom okruženju Unity zbog jednostavnosti rada.

Igre po uzoru na ovu – gotovo svaki muzej mogao bi imati igru sličnu ovoj, koja bi bila edukativnog karaktera za najmlađe posjetitelje muzeja.

Više edukativnog sadržaja – Dodavanje više edukativnog sadržaja o maslinama da bi korisnici naučili što više iz igre

ZAKLJUČAK

Sve češći način učenja je kroz edukativne računalne igre. Znanstvenici odobravaju taj pristup jer razvija osjećaj za logiku, percepciju i prepoznavanje. Računalne igre svakodnevnica su djeci i mladima te je dobro iskoristiti ih na pozitivan način. U današnje vrijeme svi posjeduju pametne uređaje što olakšava pristup informacijama. Maslinarstvo je u Istri najrazvijenija grana poljoprivrede. Istarsko maslinarstvo jedno je od najuspješnijih u zemlji i šire, a naše maslinovo ulje najbolje na svijetu prema svojoj kvaliteti. Treba poticati turizam temeljen na tradiciji čak i za najmlađe. Pokazala se potreba za zabavljanjem najmlađih u Muzeju maslinovog ulja u Puli. Zamišljeno rješenje problema je kreiranje edukativne mobilne igre koja bi bila smještena u njihovom kutku za najmlađe.

Cilj ovog diplomskog rada bio je proučiti prednosti i nedostatke programskih okvira LibGDX te pomoću stečenog znanja razviti edukativnu igru. Rezultat ovog diplomskog rada je edukativna mobilna igra Olivia koja promiče znanje o maslinama te njihovoj preradi. Igra se sastoji od dva nivoa Falling Olives i Memory. Razvijena je koristeći LibGDX programski okvir i java programski jezik. Koncept edukativne igre može se primijeniti na ostale muzeje i kulturne ustanove da bi najmlađi posjetitelji kroz zabavu naučili o tome. Igra ima dosta mjesta za poboljšanje u segmentima dodavanja i izmjene nivoa, prevođenje igre na više jezika radi boljeg razumijevanja igre, dodavanja novih značajki te prilagodba za više uređaja.

Temeljem istraživanja u ovom radu smatram da bi igra bila puno kompleksnija i kvalitetnije izrađena koristeći Unity programski okvir, što je plan za budući razvoj ove igre. LibGDX je odličan programski okvir za razvojne programere koji žele izraditi igru koristeći Javu te pisati kod bez ikakvih ograničenja. Za jednostavnu igru kao što je Olivia idealno bi bilo koristiti programske okvire kao što su Unity ili Unreal Engine.

LITERATURA

Knjige

1. Cookson, A., DowlingSoka, R. & Crumpler, C., 2016. **Unreal Engine 4 Game Development in 24 Hours**. Indianapolis: Pearson Education.
2. Engelbert, R., 2013. **Cocos2d-x by Example Beginner's guide**. Mumbai: Packt publishing.
3. Gamma, E., Helm, R., Johnson, R. & Vlissides, J., 1994. **Design Patterns: Elements of Reusable Object-Oriented Software**. Boston: Addison-Wesley.
4. Stemkoski, L., 2015. **Beginning Java game development with LibGDX**. New York: Apress.

Članci

1. Noguero, A. M., 2003. **Research gate – Video games and education**. Dostupno na: https://www.researchgate.net/publication/220686511_Video_games_and_education_Education_in_the_face_of_a_parallel_school [Pristupljeno: 01.05.2018].
2. Pivac, M., 2006. **Potencijali učenja kroz igru.** Edupoint časopis.

Internet izvori

1. Museum Olei Histriae, 2018. **Kuća Istarskog maslinovog ulja**
2. Dostupno na: https://www.oleumhistriae.com/ [Pristupljeno 14.5.2018].
3. Adobe, 2018. **Adobe products**. [Web stranica]

4. Dostupno na: https://www.adobe.com/hr/products/catalog.html [Pristupljeno 14.5.2018].
5. Jetbrains., 2018. **IntelliJ IDEA.** Dostupno na: https://www.jetbrains.com/idea/ [Pristupljeno 14.5.2018].
6. Boo, 2018. **Boo Lang.** Dostupno na: http://boo-lang.org/ [Pristupljeno 1.5.2018].
7. Game from scratch, 2018. **Game from scratch**. Dostupno na: http://www.gamefromscratch.com/ [Pristupljeno 1.5.2018].
8. Libgdx, 2017. **Documentation**. Dostupno na: https://github.com/libgdx/libgdx/wiki/Project-Setup-Gradle [Pristupljeno 16.06.2017].
9. Oracle, 2017. **Java.** Dostupno na: https://www.java.com/en/download/faq/whatis_java.xml [Pristupljeno 16.06.2017].
10. SourceMaking, 2018. **Design patterns**. Dostupno na: https://sourcemaking.com/design_patterns/creational_patterns [Pristupljeno 01.04.2018].
11. Tutorials Teacher, 2018. **MVC Architecture.** Dostupno na: http://www.tutorialsteacher.com/mvc/mvc-architecture [Pristupljeno 20.05.2018].
12. Udemy, 2018. **The Complete LibGDX Game Course Using Java**. Dostupno na: https://www.udemy.com/the-complete-libgdx-game-course/ [Pristupljeno: 01.04.2018].
13. Unitiy technologies, 2017. **Unity.** Dostupno na: https://unity3d.com/unity [Pristupljeno: 01.04.2018].

14. Unity, 2017. **Documentation.** Dostupno na: https://docs.unity3d.com/ScriptReference/ [Pristupljeno 16.6.2017].
15. Unreal, 2018. **Unreal Engine 4.** Dostupno na: https://www.unrealengine.com/ [Pristupljeno: 01.04.2018].

POPIS SLIKA

POPIS TABLICA

SAŽETAK

Ovaj rad istražuje programske okvire i pogone za razvoj edukativnih računalnih igara bazirajući se na LibGDX programskom okruženju, te uspoređivanju istoga sa ostalima. Istraženi su predanosti i nedostaci LibGDX-a te je uspoređen sa konkurentima. Cilj istraživanja bio je otkriti trendove u razvijanju igara.

U radu je opisan i postupak kreiranja jednostavne igre Olivia – koja u sebi sadrži dvije manje igre: skupljanje maslina i igra pamćenja. Opis igara započinje sa planom nivoa i funkcionalnosti. Igre su opisane po klasama koje su se koristile za njihovu izradu.

Igra edukativnog karaktera te je namijenjena svim generacijama ali pretežito djeci posjetiteljima Kuće maslinovog ulja u Puli. Igra je izrađena početnički u svrhu istraživanja programskog okvira te daje puno mjesta za napredak i poboljšanje.

Ključne riječi: masline, muzej, LibGDX, Java, edukativna igra

SUMMARY

This thesis researches frameworks and engines used for developing educational computer games, but the main focus is on the LibGDX framework and comparing it to others while assessing its strengths and weaknesses. The goal of the research was to determine game development trends.

The thesis also describes the development of a simple game called Olivia which is comprised of two smaller games: collecting olives and a memory matching game. Game description starts with a level and functionality plan. Each game is analyzed by describing each class used to create it.

The game has an educational character and is aimed toward all generations, while the target audience are the visiting children of the House of Istrian Olive Oil in Pula. It is a beginner project developed while learning and researching the framework so there is much room for improvement.

Ključne riječi: olives, museum, LibGDX, Java, educational game