Projekt R: Squid Game

Tehnička dokumentacija

Verzija <1.0>

Studentski tim: Voditelj: Borna Majerić

Ema Nekić

Timon Rogić

Klara Peričić

Anja Obuljen

Nastavnik: Igor Sunday Panđić

Sadržaj

1. Opis razvijenog proizvoda 4

2. Tehničke značajke 5

3. Upute za korištenje 6

4. Literatura 7

Opis razvijenog proizvoda

Na predmetu Projekt R naš tim odlučio se za izradu igrice u kojoj će kontrole biti pokreti glave, odnosno implementirat će se Visage SDK tehnologija za praćenje lica.

Inspiraciju za koncept i izgled igre dobili smo od popularne Netflixove korejske serije Squid game. Ova horor serija o izmišljenoj ubojitoj zaludila je svijet, te smo se i mi odlučili iskušati u izazovima koje igra stavlja pred vas.

Squid Game se usredotočuje na niz nevinih igara iz djetinjstva koje su postale mračne

Ideja je bila napraviti 3 miniigre koje predstavljaju stvarne levele igre u seriji, odabrali smo igre koje su nam najzaimljivije. Svaka miniigrica razvijena je kao 3D igra u programskom alatu Unity, a sve funkcije i mehanika pisani su u C# programskom jeziku

Home page (Menu)

* \*\*single player or multiplayer (još za raspraviti)
* Start
* Instructions (objašnjenje svake igre, pravila i upravljanje)
* Quit

3 igre /levela ( nemoguće preci na sljedeću igru dok ne prođeš prethodnu, 'umireš' ako ne uspiješ pobijediti u igri, povratak na prvu igru)

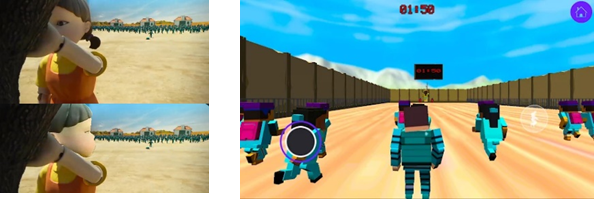
* 3 života koja možeš iskoristiti u bilo kojoj fazi igre
* first person igre

ovdje još treba dodat slike i onjasnjenja svih ne glavnih prozora i instrukcija

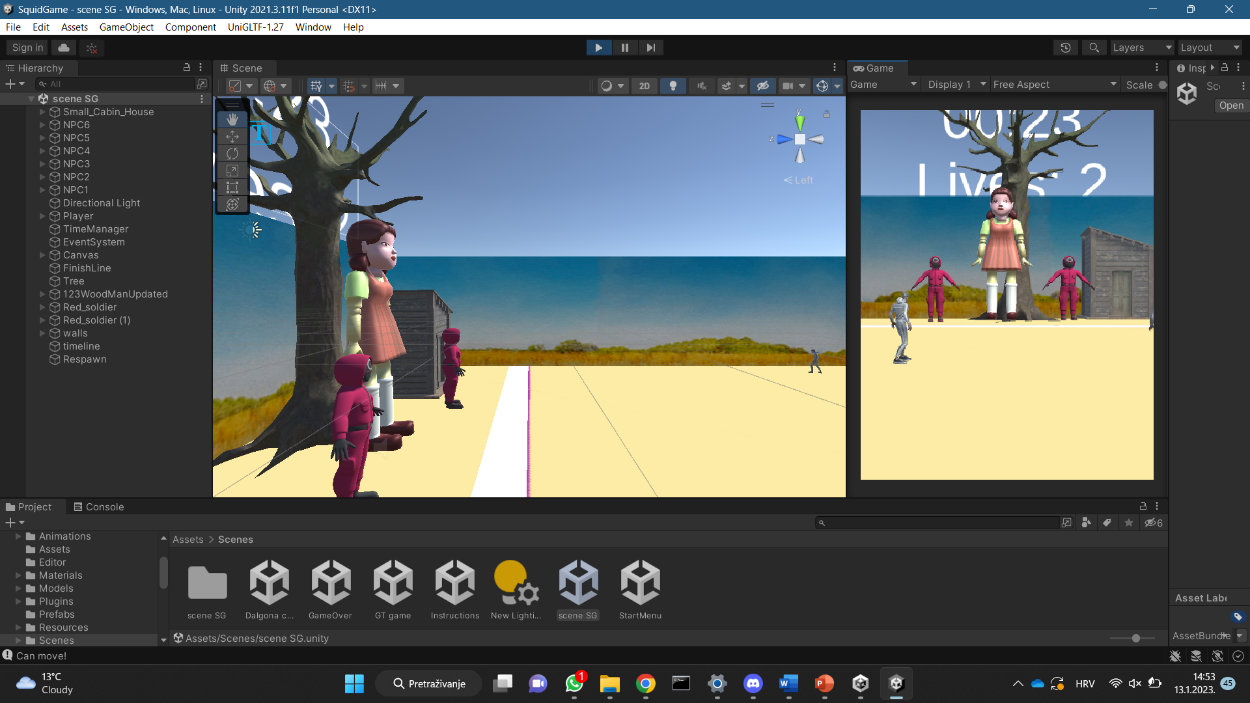
1. **Red Light Green Light (Crvena kraljica jedan, dva, tri)**

Omiljena dječja igra na igralištima širom svijeta, Crveno svjetlo, zeleno svjetlo poznata je po mnogim imenima, uključujući Kipove, Bakine stope i, u Koreji, Mugunghwa kkochi pieotseumto nida. To ime je i pjesma koju pjeva jeziva lutka ubojica u *Igra lignje* , što otprilike znači Cvijet Mugunghwa je procvjetao. Simbolično, Crveno svjetlo, zeleno svjetlo ima puno smisla kao prva igra jer natjecateljima na istinski traumatičan način pokazuje da će umrijeti ako iskorače iz reda čak i malo.

Inscpiracija:



Proizvod:



Korišteni modeli, funkcije, elementi igre Itd:

* Set speed
* Semafor
* Timer?
* Ostali ljudi modeli?
* Gledatelji?
* Model žene koja se okreće
* Ruke

Kontrole:

Upute

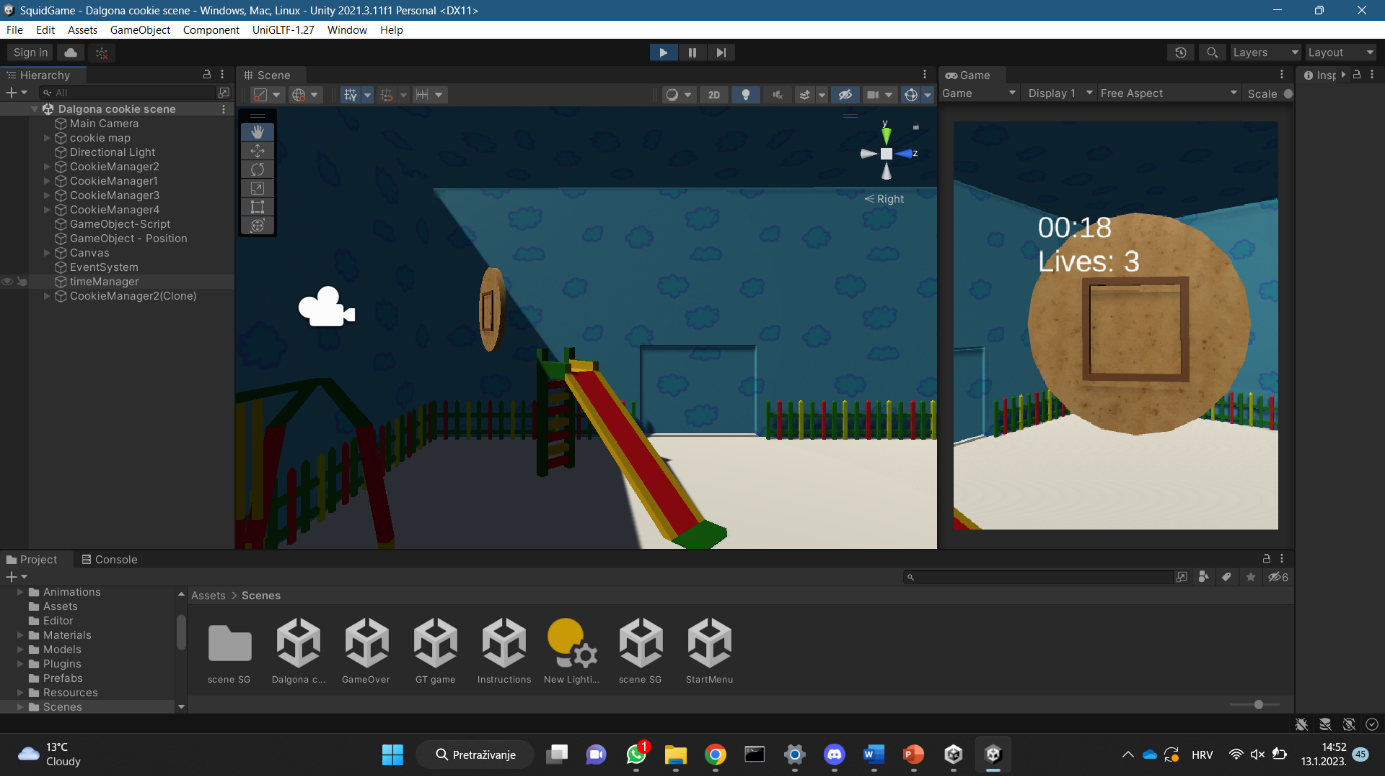
1. **Dalgona cookie game – Rezanje kolačića**

Drugi krug Squid game-a je igra koja uključuje Dalgona cookie, odnosno zašećereni keks od karamele koji je nekoć bio popularna korejska ulična hrana. Tradicionalno, ako bi osoba koja je kupila dalgonu uspjela pojesti oko oblika u sredini, a da je ne razbije, dobila bi drugi, besplatni slatkiš. Igrač ove igre mora izrezbariti oblik koji je narctan na keksu, s obrzirom da je kolačić jako krhak ako imalo pogriješi u tome, kolačić se lomi i igrač gubi igru.

Inspiracija:



Proizvod:



Elementi igre:

* 2D
* Margine za odstupanje od linija
* Timer
* Oblici: Kišobran, krug, kvadrat, trokut

Kontrole:

Upute:

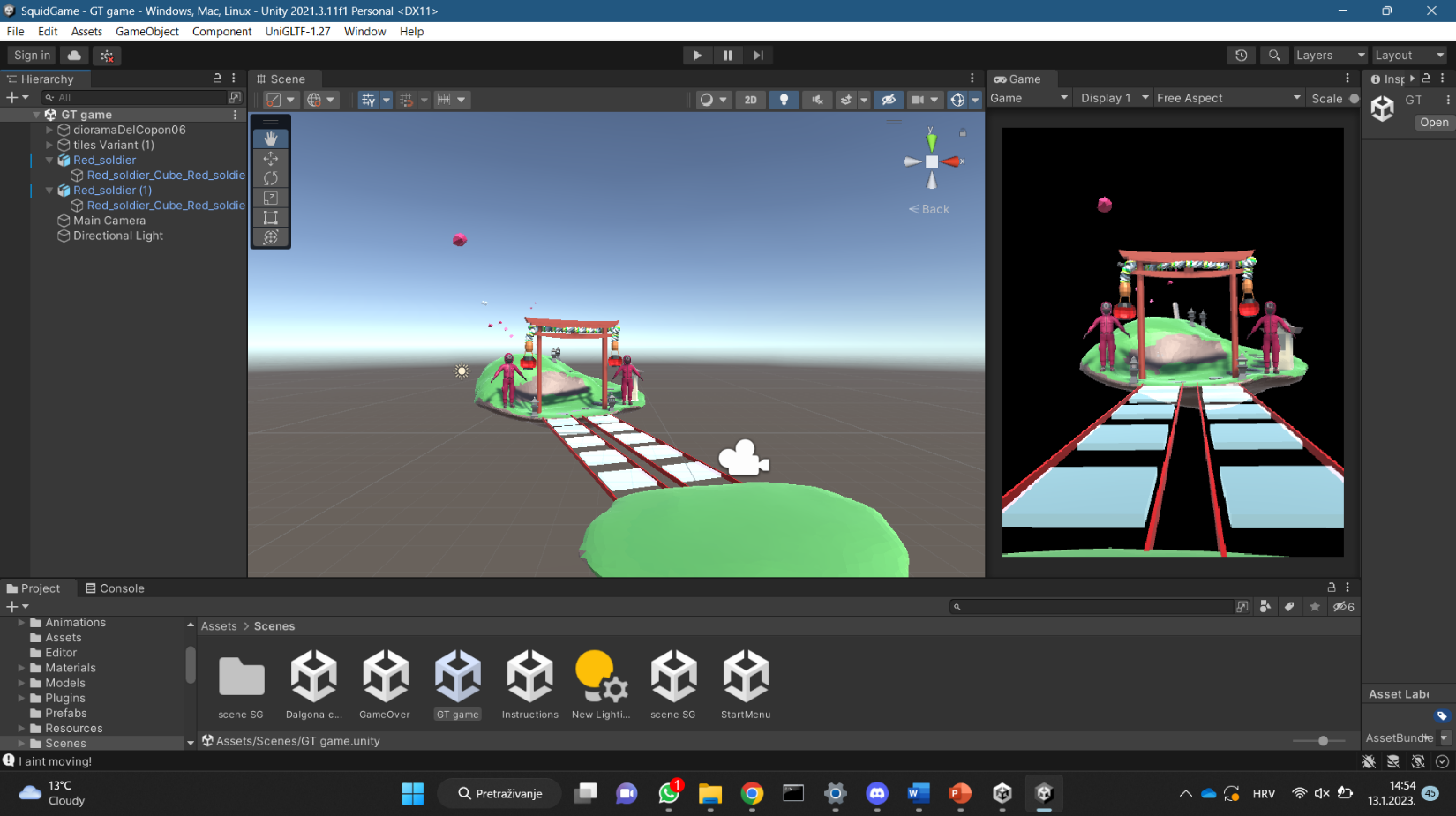
1. **The Glass Tile Game (Igra staklenih ploča)**

Jedina igra u Squid Game koja nije igra stvarnom svijetu je 5. krug, u kojoj se natjecatelji probijaju podmuklim putem od staklenih platformi - od kojih će neke zadržati svoju težinu, a neke će se razbiti. Ipak, igra sa staklenim bridžom Igra lignje 7. epizoda inspirirana je poskocima i drugim sličnim igrama skakanja, koje datiraju stoljećima unatrag iz drevne Indije i Rima.

Inspiracija:



Proizvod:



Elementi igre:

* 3D
* Kontrole: Glava lijevo desno za odabir lijeve ili desne ploče + skok
* Nakon gubitka života pamti ispravne ploče
* 4-5 ploča
* Highlight na odabranu ploču

Kontrole:

Upute:

# Tehničke značajke

Na ovom projektu glavni alat/program koji smo koristi je Unity.To je razvojno okruženje za igre i računalnu grafiku, odnosno alat za izradu video igara. Razvojno okruženje je u cijelosti napisan u C i C++ programskim jezicima. Mi smo programirali u C# programskom jeziku.

Unity razvojno okruženje podržava integraciju s mnogo alata za 3D modeliranje i uređivanje slika i slično.

Unity je moćan alat koji puno olakšava izradu video igara zato što u sebi ima ugra**đ**ene mnoge funkcionalnosti koje su dostupne korisniku klikom miša.Unity je više platformski alat za razvoj video igara koji se koristi za izradu 2D i 3D igara. Svojom jednostavnošću i pristupačnošću i brojnim mogućnostima danas je jedan od vodećih me**đ**u sve brojnijom konkurencijom. Dovoljno jednostavan za početnike i dovoljno moćan za napredne korisnike omogućava da ga koriste svi te da zadovolji sve zahtjeve. Doći do 7 instalacijskog paketa je tako**đ**er jednostavno i može se besplatno preuzeti s njihovih službenih stranica gdje postoje mnoge upute, materijali i dokumentacija za rad. Danas Unity nudi gotovo sve mogućnosti kao i drugi alati za razvoj video igara te tako ne odstupa ni na koji način po kvaliteti, a najbolji dokaz tomu su mnogi partneri koji su ujedno i velike i poznate tvrtke koje sura**đ**uju na razvoju Unity alata kao što su: Microsoft, Sony, Samsung, Nitendo, Oculus ili pak Intel. Kao što je već spomenuto, Unity je besplatan alat, ali postoje i verzije koje se naplaćuju ovisno o zaradi i mogućnostima koje korisnik namjerava koristiti. U slučaju da se korisnik ili organizacija odluče za kupnju Unity alata, plaćanje se odvija na mjesečnoj bazi u obliku pretplate. Ovisno o količini mogućnosti postoje tri paketa koja se naplaćuju: Plus, Pro i Enterprise. Personal verzija je početna i besplatna

# Korisničko sučelje Unity-a možemo podijeliti na šest različitih prozora.



**5.**

**1.**

**2.**

**6.**

**4. i i 5**

**3.**

1. **Scene**

Ovaj prozor služi za kreiranje scene igre, odnosno za pozicioniranje objekata koje će se koristiti u igri. Ako se radi 2D igra, ovaj prozor je ograničen na 2D prostor, odnosno ima samo x i y os. Prebacivanjem prozora u 3D dobivamo z os, odnosno dubinu. Z os takoĎer može biti korisna u 2D igri jer pomoću nje možemo kontrolirati slojeve (eng. layer) u igri poput slojeva u Photoshopu. Prozor broj dva je prozor

1. **Game**

Ovaj prozor prikazuje izgled igre kada se pokrene. Točnije ovaj prozor prikazuje igru iz perspektive glavne kamere igre. U ovom slučaju vidimo kako igra izgleda kada se pokrene. Možemo primijetiti razliku izmeĎu izgleda scene i igre.

1. **Hirerachy**

Treći prozor možemo nazvati prozorom objekata. Ovaj prozor prikazuje sve objekte koji se nalaze u trenutnoj sceni. TakoĎer, unutar njega možemo kreirati nove objekte te ih grupirati unutar drugih objekata.

1. **Project –** ovdje treba ubacit strukturu projekta I kako smo sve raspodijelili po mapama

Četvrti prozor je prozor koji prikazuje sve mape za trenutni projekt

1. **Console**

ovaj prozor možemo grupirati s petim prozorom koji prikazuje sadržaj mape koja je trenutno odabrana. Ukratko ovi prozori nam služe za organiziranje resursa igre te pristupanjem njima kako bi mogli modificirati neke njihove karakteristike.

1. **Inspector**

Šesti prozor je prozor karakteristika (eng. properties), ovaj prozor u Unity-u služi za dodavanje, brisanje te kontroliranje svih karakteristika odabranog objekta. U ovom prozoru možemo dodavati karakteristike poput collidera koji služi da kontroliranje sudara s objektima, dodavanje tekstura ili sličica objekta za definiranje izgleda objekta, pa sve do dodavanja animacija i slično

OSNOVNI KONCEPTI:

1. **Objekti**

Objekti u igri su osnovni i najvažniji objekt. Ako korisnik želi i razumjeti i znati raditi u Unity alatu, mora znati što su objekti u igri i kako oni rade. Svaki objekt koji se nalazi u igri je objekt, ali sam objekt sam po sebi ne radi ništa. Kako bi imali odreĎenu funkciju nužno je da objekti imaju posebna svojstva koja im omogućuju da predstavljaju likove, dijelove okoliša ili efekte poput svjetla. Način na koji ti objekti funkcioniraju je jednostavan, oni su zapravo spremnici koji sadrže niz komponenata (eng. Component) koji svaki od njih ima odreĎenu svrhu. S obzirom na to kakav objekt želimo napraviti, koristimo razne kombinacije komponenata i sam Unity ima mnogo različitih vrsta koji su unaprijed ugraĎeni. Korisnici takoĎer mogu kreirati svoje vlastite komponente pomoću skripta i o tome će biti riječ u sljedećim poglavljima [16].

# Kamera

# Kamere su objekti koji bilježe i prikazuju svijet igraču. PrilagoĎavanjem i manipulacijom komponenata kamere, korisnik može prezentirati igre na razne jedinstvene načine. U jednoj sceni može biti neograničen broj kamera i mogu se primjenjivati u bilo kojem redoslijedu ili mjestu na ekranu. Kamere su neophodne za prikazivanje igre samom igraču [24].

1. **Gotovi resursi**

Prilikom izrade odreĎene scene prikladno je kreirati objekt i dodati mu komponente i podesiti im svojstva. U slučaju kada želimo imati više takvih objekata poput lika kojim ne upravlja igrač ili dio okoliša scene, tada nam može biti komplicirano ponovno kreirati takve objekte u slučaju da želimo više istih. Kako bi se taj problem što lakše riješio, Unity nam nudi kreiranje gotovih (eng. Prefab) resursa koji omogućuju spremanje već kreiranih objekata kako bi ih mogli po potrebi ubacivati u scenu. Gotove resurse spremamo u mape projekta i kada želimo instancirati u scenu imaju odreĎene vrijednosti komponenata objekta. Te vrijednosti se mogu prilagoĎavati po potrebi i svaki novi objekt ubačen u scenu može imati na taj način jedinstvena svojstva [18].

1. **Scene**

Scene sadržavaju odreĎen broj i raspored objekata u odreĎenom stadiju igre. Mogu se koristiti za izradu izbornika, pojedinih razina (eng. level) igre ili bilo čega drugog. Jednu individualnu scenu možemo gledati kao jednu jedinstvenu razinu. U svakoj sceni korisnik može polagati dijelove okoliša, likove, dijelove korisničkog sučelja ili bilo koje druge tipove objekata kako bi izradio željeni dio same igre. Scene se spremaju u obliku datoteka u mapama projektnih resursa [19]

1. **Skripta**

Skripte su jedne od najvažnijih dijelova svih igara. One sadržavaju programski kod koji pokreće odreĎene akcije u igri te čak i najjednostavnije igre trebaju neku vrstu programskog koda. Najčešći primjer korištenja skripta je reagiranje na unos kontrola za kretanje igrača ili stvaranje dogaĎaja u odreĎenim trenucima igre. Osim toga skripte se koriste i za kreiranje grafičkih efekata, kontrolu ponašanja objekata, pisanje umjetne inteligencije za likove kojima ne upravlja igrač itd. Ovo je možda najteži dio kreiranja igara te je potrebno općenito znanje programiranja i uloženo vrijeme da se savladaju funkcije i logika pisanja programskog koda za 19 Unity. Kako bi korisnik kreirao skriptu za Unity, potrebne su mu funkcije Unity alata i dokumentaciju koja objašnjava način njihovog korištenja. Dokumentacija se može naći na službenim stranicama Unity alata. Unity alat nam omogućuje pisanje skripta u tri jezika, UnityScript (JavaScript), C# i Boo. Kada želimo kreirati novu skriptu program nudi samo UnityScript i C# iz razloga što žele izbaciti programski jezik Boo zbog njegovog slabog korištenja meĎu kreatorima igara [20].

1. **GUI elementi**

Slike korisničkog sučelja i tekstovi mogu biti prikazani pomoću „GUI Text“ i „GUI Texture“ objekata u Unity alatu. Svaki GUI element može imati svoju veličinu skalabilnost i poziciju. GUI elementi su zapravo objekti koji se prikazuju ispred svih ostalih objekata u sceni [22].

1. **Zvuk**

Baš kao i u stvarnom životu zvukovi imaju svoje izvore, a osluškivači ih primjećuju i slučaju. Zvukove koje želimo koristiti ubacujemo u projektne resurse kao i ostale, a možemo i unutar Unity alata izraĎivati nove. Kako bi zvukove mogli iskoristiti potrebno je objektu koji treba biti izvor navedenog zvuka dodati Audio Source komponentu [15]

**+Visage SDK**

# Upute za korištenje

# Literatura