



SVEUČILIŠTE U ZAGREBU



Fakultet
elektrotehnike i
računarstva

Diplomski studij

Računarstvo

Akademska godina
2022/2023



Umrežene igre

Dijeljenje ekrana videoigre. Uvod u
pogonski sustav Unity

Posjet Gamepires - potvrđen

- Posjet će se održati polovicom 12. mjeseca
- Vjerojatno za do 30tak studenata
- Teme
 - Povijest tvrtke
 - Motion capture
 - Umreženi aspekti Scuma
- Što bi željeli čuti? Javite teme za koje ste zainteresirani pa ćemo ih dostaviti tvrtki.

Predmet Uvod u dizajn i produkciju videoigara

- Održava se na Akademiji dramskih umjetnosti
- Nastava se održava petkom od 10 do 12 sati u prostoriji 104 (prizemlje u glavnoj zgradi, Trg Republike Hrvatske 5)
- U ponudi je kao izborni predmet za studente na BA i MA razinama
- <https://www.adu.unizg.hr/uvod-u-dizajn-i-produkciju-videoigara-novi-izborni-kolegij-na-adu-u/>

Sadržaj

- Višekorisničke igre na jednom uređaju
- Uvod u pogonski sustav Unity
 - Scene i objekti
 - Sučelje
 - Komponente

Višekorisničke igre na jednom uređaju

- Više igrača povezano na isti uređaj (najčešće računalo ili konzolu)
- Osnovni pristupi prikazu
 - Dijeljeni ekran
 - Naizmjenično igranje
- Osnovni pristupi ulaznim uređajima
 - Ulazni uređaji najčešće istog tipa (primjerice kontroleri)
 - Moguće dijeliti jedan uređaj (primjerice tipkovnicu)
 - Neke igre omogućavaju različite uređaje (primjerice miš i tipkovnica ili tipkovnica i kontroler)
- Za razvoj ovakvih igara je jako bitno:
 - Razdvojiti unose po igračima
 - Omogućiti unose različitim uređajima



Dijeljenje ekrana videoigre

- **Dijeljenje ekrana** ili zaslona je tehnika prikaza u računalnoj grafici u kojoj se prikaz, slika, video i/ili tekst dijeli na susjedne dijelove.
- U videoigrama je krenulo najranije 1970-ih, s arkadnim igrama poput *Drag Race*
- Način dijeljenja ekrana može ovisiti o žanru videoigre, kontrolama, mehanikama, a općenito se može podijeliti na **naizmjenično** i **istovremeno** dijeljenje ekrana



Naizmjenično dijeljenje ekrana

- **Naizmjenično dijeljenje ekrana** obično je prisutno kod videoigara na poteze, koje spadaju u već spomenute **hotseat** igre
- Igrači koriste samo **jednu ulaznu napravu** tj. jednu tipkovnicu i miš ili jedan igraći kontroler
- Izmjena se događa kada jedan igrač završi svoj potez i prepušta kontrole i zaslon drugome – u igri se tada prebacuje **kamera i kontekst igre** na idućeg igrača
- Ovakav način igranja je **danās rijetko zastupljen** u svom izvornom obliku te zbog praktičnosti skoro u potpunosti zamijenjen s umreženim igrama na poteze

Istovremeno dijeljenje ekrana

- **Istovremeno dijeljenje ekrana** može funkcionirati tako da svi igrači vide isti zaslon ili da je ekran podijeljen na više područja jednake veličine
- Igre kod kojih se sve događa na nepodijeljenom zaslonu prilagođavaju mehanike igre ili pomicanje kamere tako da svaki igrač ima jednaku vidljivost svojeg lika u igri – npr. igra *Trine*



Istovremeno dijeljenje ekrana

- Budući da se podjela ekrana vrlo rijetko radi za neparan broj igrača, iduća vrlo česta podjela je na četiri jednaka dijela zaslona
- Dijeljenje ekrana na više od četiri dijela nije uvijek praktična u videoigrama jer **zahtijeva vrlo veliki ekran** kako bi igrači uopće imali normalnu vidljivost svojeg pogleda na videoigru



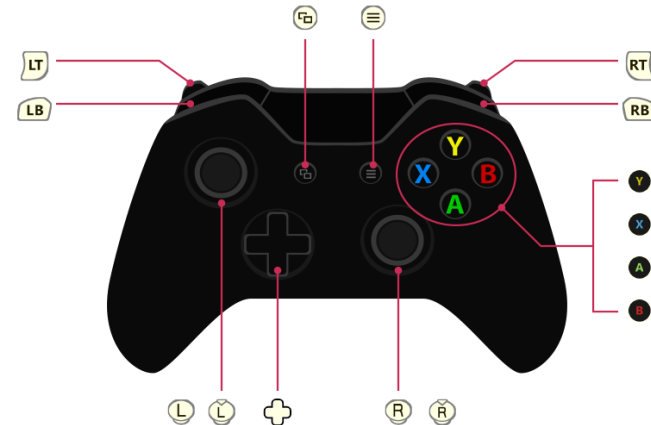
Tipkovnica za lokalno višekorisničko igranje

- **Tipkovnica za lokalno višekorisničko igranje** se obično koristi kada igrači na osobnim računalima ne posjeduju igraće kontrolere
- Ovisno koliko videoigra ima ulaznih kontrola, tipkovnica se dijeli na područja koja će svaki zasebni igrač koristiti za igranje
- Primjer: ako imamo igru koja ima četiri tipke kao ulazne kontrole za kretanje lika (gore, dolje, lijevo, desno), za četiri igrača moramo na tipkovnici pronaći tipke u sličnom rasporedu kao strelice na tipkovnici
- **Potencijalni problemi:**
 - Nepraktičnost igranja u više od dva igrača (zamislite 4 ljudi na jednoj tipkovnici)
 - Problem sa složenijim kontrolama – što ako treba miš ili više tipki?
 - *Key rollover* – sposobnost računala da ispravno rukuje s nekoliko istovremenih pritisaka na tipke – za većinu tipkovnica to je 6 tipki



Upravljači za više igrača

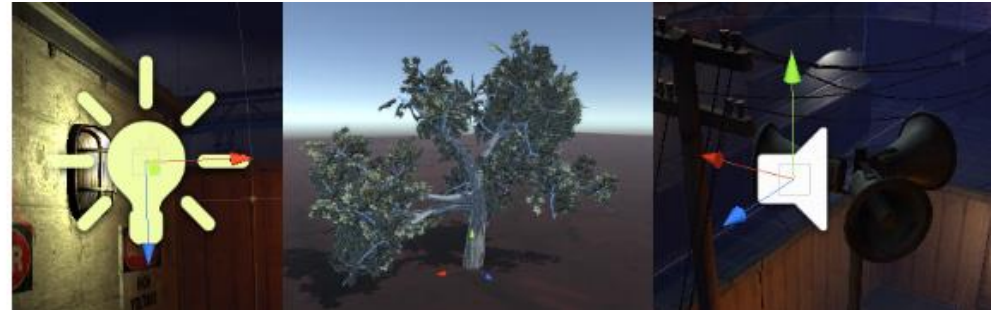
- Kako bi igrači na osobnim računalima dobili **najbolje iskustvo** u igranju lokalnih višekorisničkih videoigara, potrebni su im igrači kontroleri (engl. joystick, gamepad, gaming controller)
- Danas najpopularniji kontroleri na osobnim računalima su Xbox kontroleri, Sony Playstation kontroler (DualShock i DualSense), Razer i Logitech kontroleri
- Većina kontrolera slijedi **određeni standard rasporeda i broja tipki**
- Prednosti kontrolera nad tipkovnicom za više igrača:
 - Svaki igrač može koristiti svoj kontroler, lakše je dodati više od 2 igrača – nema „gužvanja” i svi imaju jednako raspoređene kontrole
 - Prikladna zamjena za tipkovnicu i miša – ne mora se sjediti za stolom



Uvod u pogonski sustav Unity

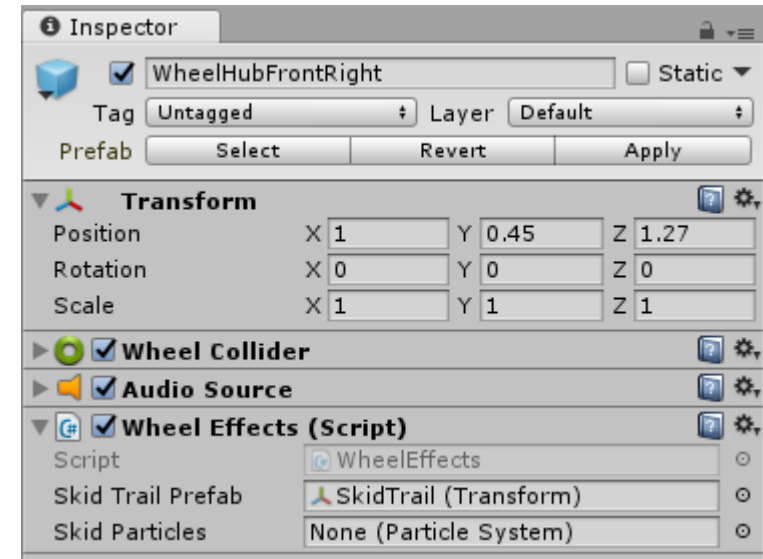
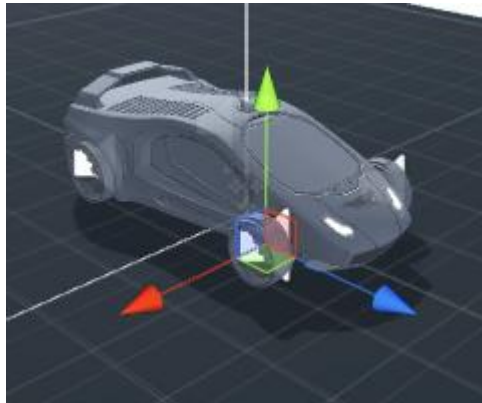
Scene i objekti

- Scena u Unityu se može definirati kao jedan nivo igre ili jednu zasebni virtualni svijet
 - Scene se snimaju u Assets direktorij
 - Scene sadrže objekte
 - Svaka početna scena ima dva objekta – glavnu kameru i izvor osvjetljenja
- Objekti (GameObjects) su osnovni koncept u Unity alatu – sve unutar virtualnog svijeta igre je sastavljeno od objekata
- Objekti se mogu tumačiti kao kontejneri koji mogu imati različite komponente
 - Zvuk
 - 3D informaciju
 - Skriptu
 - ...



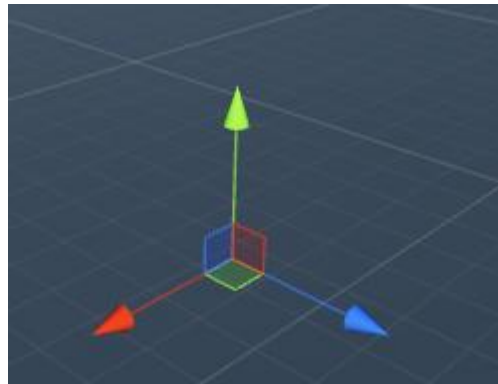
Primjer objekta

- Objekt je na početku prazna posuda
- Svaki objekt ima samo jednu komponentu – transform koji indicira njegovu lokaciju, rotaciju i skalu na X, Y i Z osi
- Objekt točka na autu posjeduje:
 - Transform
 - Collider
 - Audio source
 - Script



Primjer objekta

- Objekt može imati više istovrsnih komponenata
- Ne mora imati 3D informaciju
- Kamera
 - Transferom
 - Camera
 - Flare Layer
 - GUI Layer
 - Audio Listener
 - Scripts



Praktični dio nastave

- Kreiranje Unity računa
- Pripremite računala i pokrenite Unity
- Stvaranja projekta
- Stvaranje nove scene
- Uvoz sredstava
- Modifikacija objekata

Kreiranje Unity računa

- <https://id.unity.com/>
- Možete kreirati račun putem svog postojećeg facebook, google ili apple računa, a možete kreirati i novi unity račun samo s e-mail adresom i lozinkom

Create a Unity ID


If you already have a Unity ID, please [sign in here](#).

Email	Password
<input type="text"/>	<input type="password"/>
Username	Full Name
<input type="text"/>	<input type="text"/>

☐ I have read and agree to the [Unity Terms of Service](#)(required).

☐ I acknowledge the [Unity Privacy Policy](#) [Republic of Korea Residents agree to the [Unity Collection and Use of Personal Information](#)](required).





☐ I agree to have [Marketing Activities](#) directed to me by and receive marketing and promotional information from Unity, including via email and social media(optional).

☐ I'm not a robot
reCAPTCHA
Privacy - Terms

Create a Unity ID

Already have a Unity ID?

OR



Kreiranje novog projekta

Unity 2017.1.0f3




Projects

Learn

 NEW

 OPEN

 MY ACCOUNT

Project name*

Alice

☒ 3D ☐ 2D


Add Asset Package

Location*

C:\Users\msuznjevic\Desktop



ON ☒

Enable Unity Analytics 

Organization*

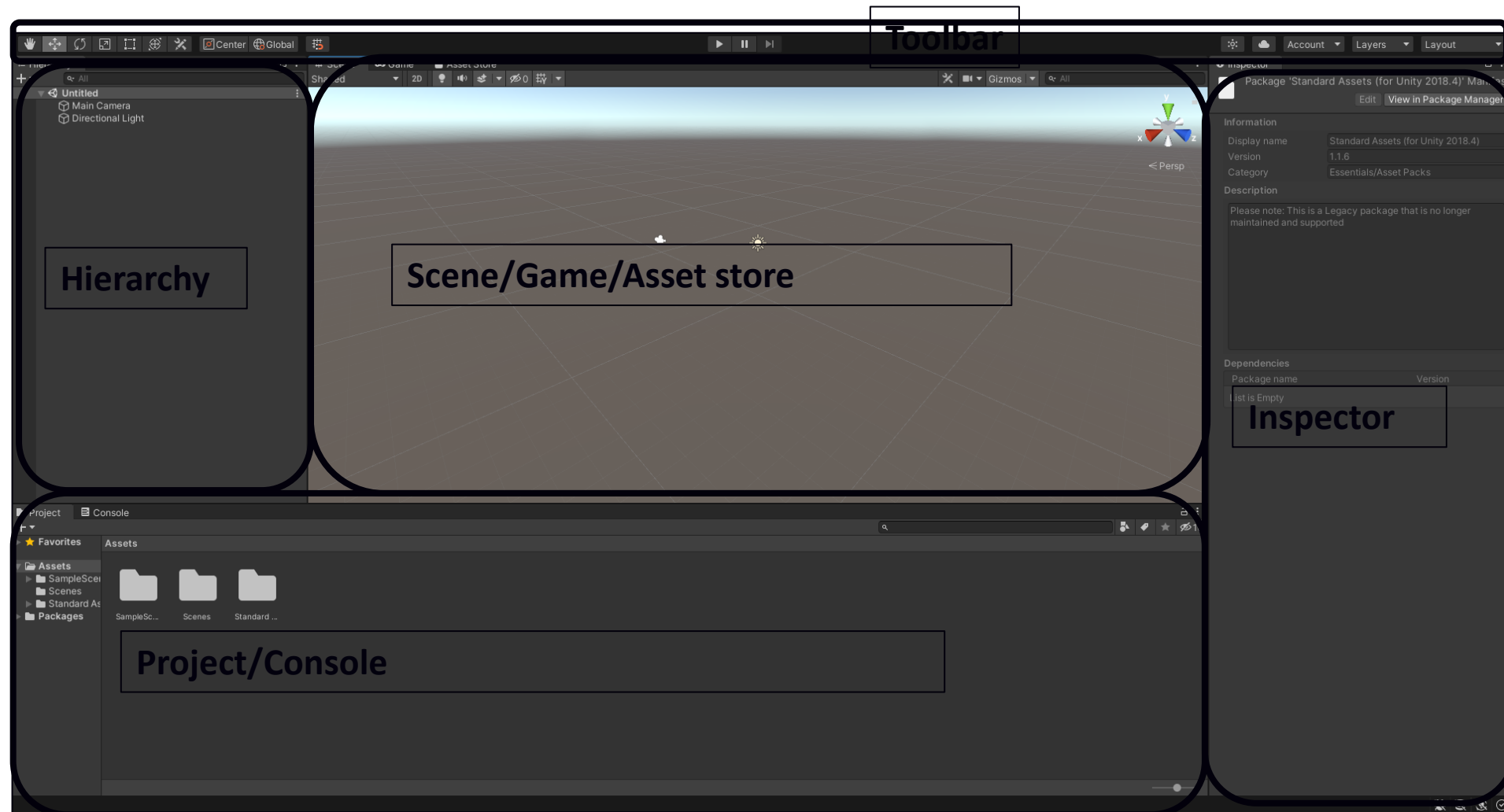
msuznjevic



Cancel

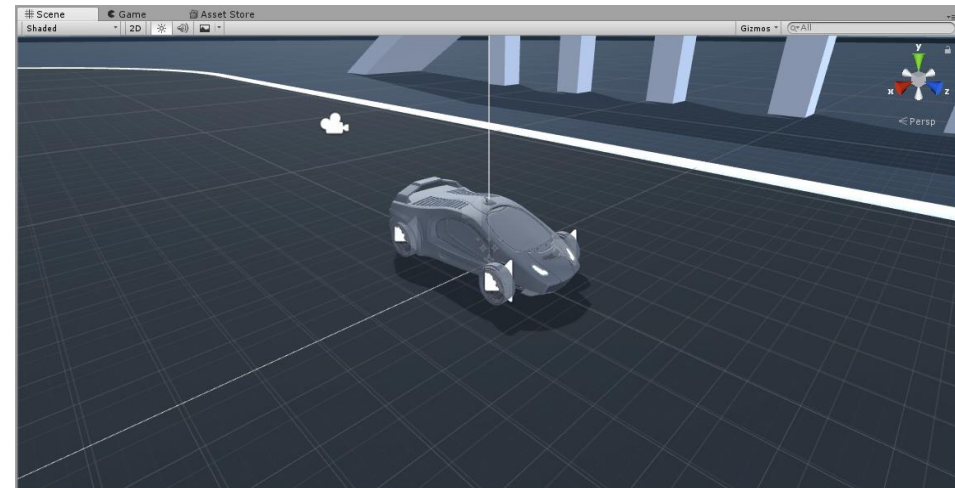
Create project

Unity – osnovno korisničko sučelje



Pogled Scene

- Omogućuje direktnu interakciju sa svim objektima unutar scene projekta
 - Označavanje
 - Rotacija
 - Translacija
 - Mijenjanje veličine
 -
- Može prikazati 2D
ili 3D perspektivu
- Lako prebacivanje perspektive korištenjem widgeta u desnom gornjem kutu (x, y, z ravni)

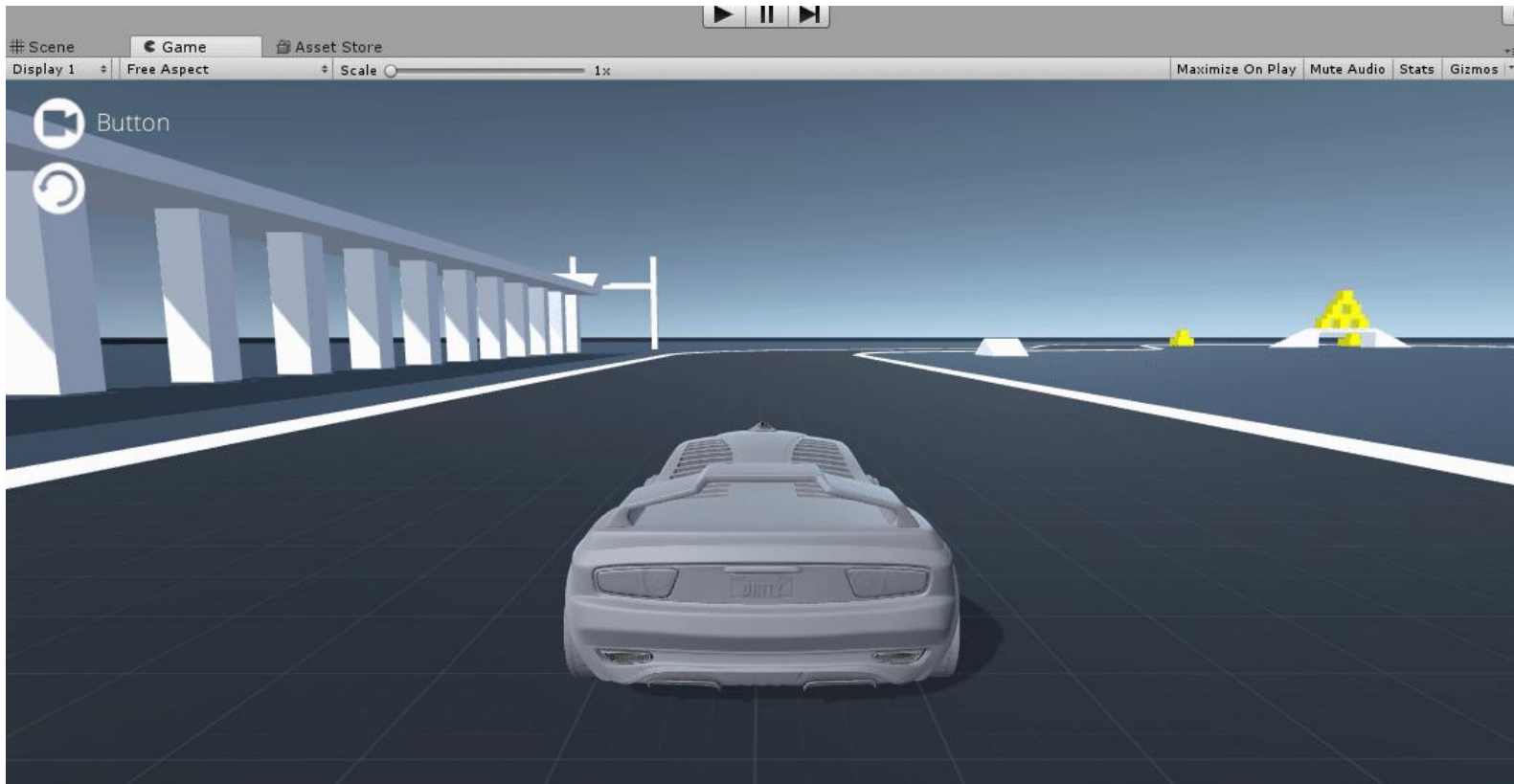


Pogled Scene - interakcija

- Ako niti jedan objekt u sceni nije selektiran (demonstrirati)
 - Držanje desne tipke miša i povlačenje – mijenja se točka na koju se gleda
 - Lijevi klik – selektira objekt
 - Držanje lijeve tipke miša i povlačenje – stvara se pravokutnik u kojem se selektiraju objekti koji leže u njemu
 - Držanje srednje tipke miša (skrol dugme) i povlačenje mijenja točku gledišta (točku iz koje se gleda na scenu)
 - Skrol naprijed – zumira se unutar scene
 - Skrol nazad – odzumira se unutar scene

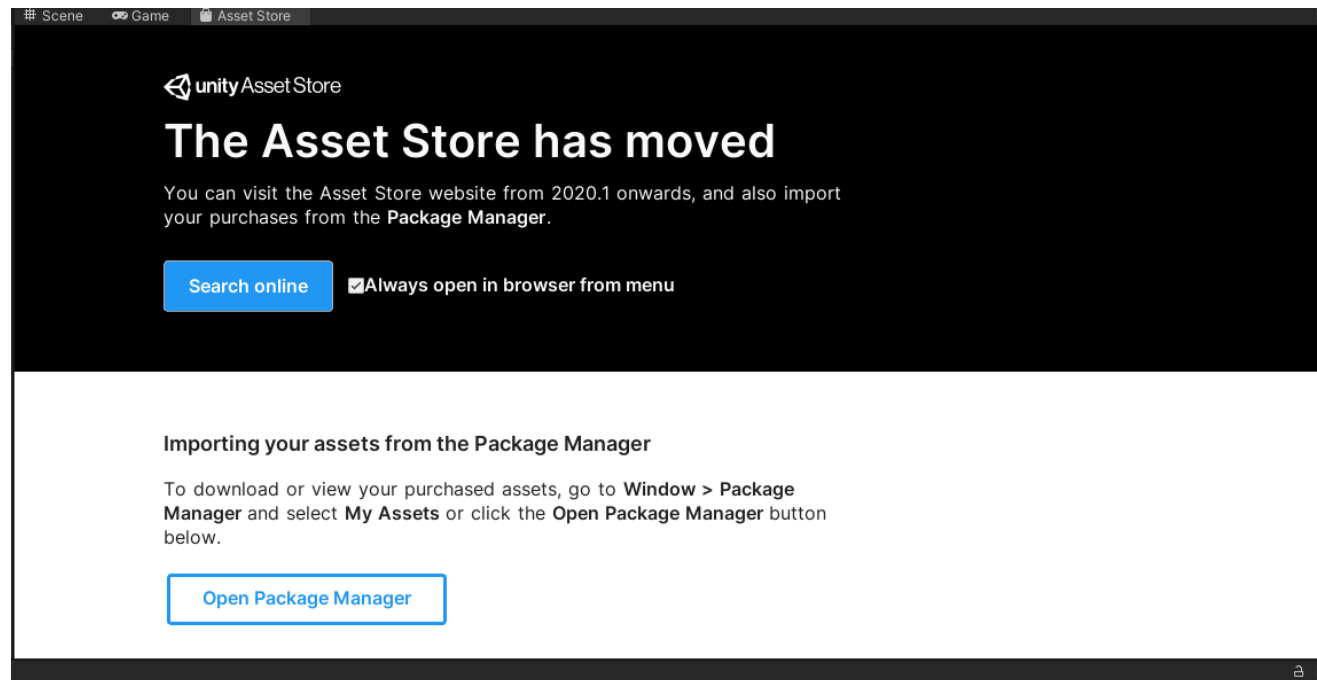
Pogled Game

- Omogućava jednostavno testiranje igre



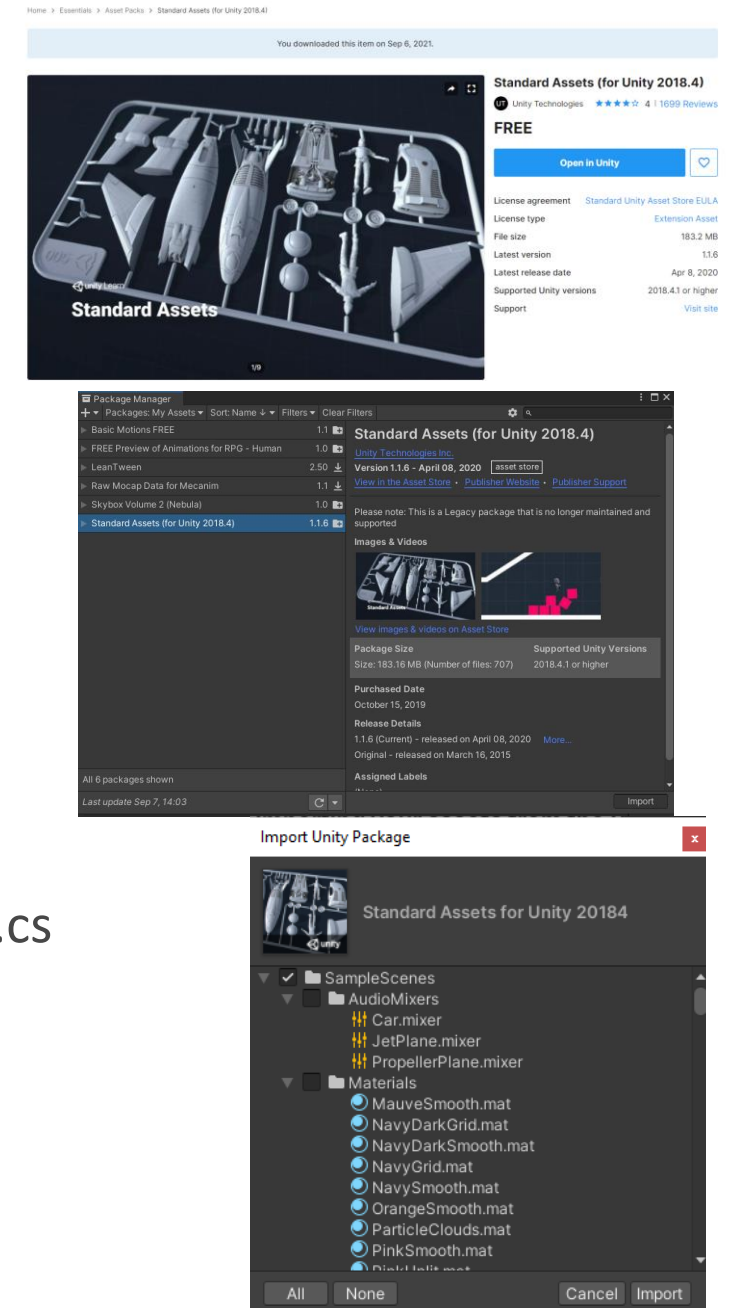
Prozor Asset Store

- Nekad otvarao preglednik s trgovinom za prodaju sadržaja za izgradnju igre
- Kroz isti se moglo lako uvesti sadržaje u igru
- Zbog ubrzanje samog IDE-a, isključena opcija u 2020 te se kroz ovaj prozor može otvoriti preglednik za online pretragu ili Package Manager kroz koji se može upravljati postojećim i dodavati nove pakete sadržaja



Instalacija Standard assets paketa

- Odaberite u trgovini opciju Search online
- Otvorena je stranica <https://assetstore.unity.com/>
- Ulogirajte se s prethodno napravljenim računom za Unity
- U polje za pretraživanje ukucajte „Standard assets”
- Odaberite opciju Open in Unity -> Open Unity editor
- U Package Manager prozoru odaberite Standard Assets (for Unity 2018.4) te opciju Import
- U novom prozoru ponovno odaberite Import
- U slučaju greške u paketu izbrišite datoteku SimpleActivatorMenu.cs ili je popravite ukratko koristeći ovaj video: <https://youtu.be/0K6m-jJZ8QQ?t=277>



Prozor Project

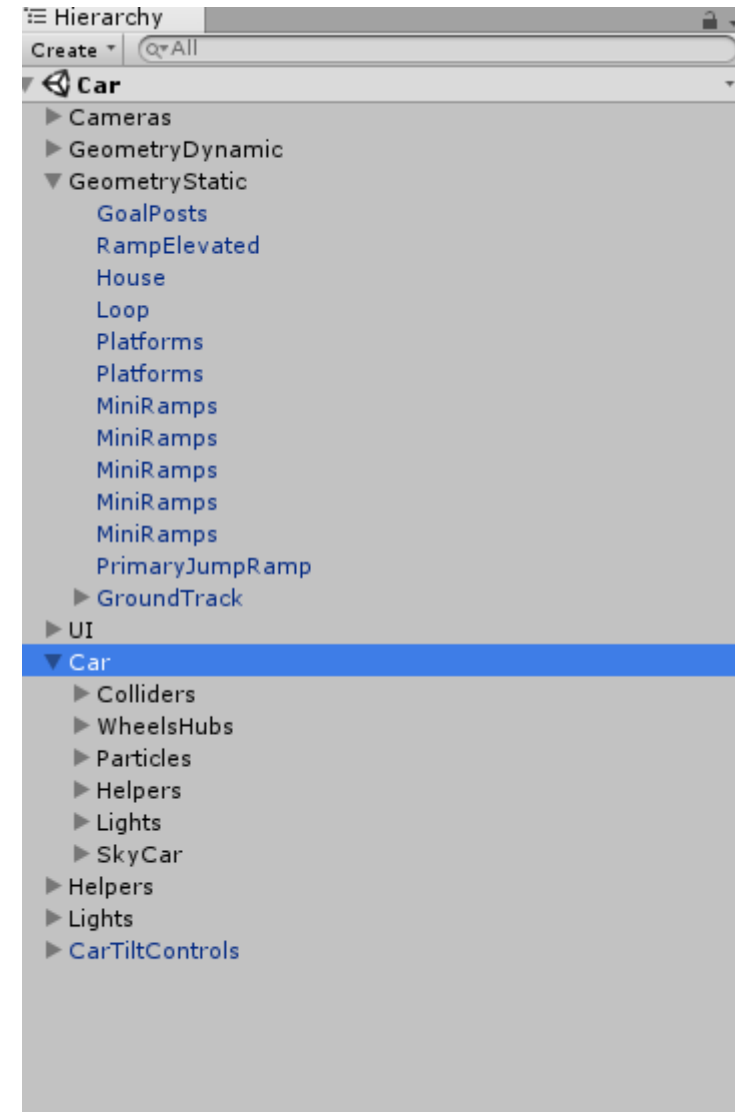
- Assets (sredstva) – datoteke sa sadržajem koji se koristi u izradi igre
- Projekt se sastoji od različitih tipova datoteka
 - Animation
 - Audio
 - Material
 - Mesh
 - Model
 - Prefab – objekt koji služi kao predložak za kreiranje više objekata
 - Scene – scena unutar projekta odnosno jedan nivo unutar igre
 - Script
 - Texture

Prozor Console

- Konzola u kojoj se ispisuju poruke sustava greške i slično
- Primarno služi za pronalazak grešaka u kodu
- Koristeći posebne funkcije možemo ispisivati svoje poruke
 - [Debug.Log](#),
 - [Debug.LogWarning](#) i
 - [Debug.LogError](#) .
- Clear – briše konzolu
- Collapse – prikazuje samo prvu instancu greške koja se ponavlja
- Error Pause pauzira igru kada god se pozove Debug.LogError

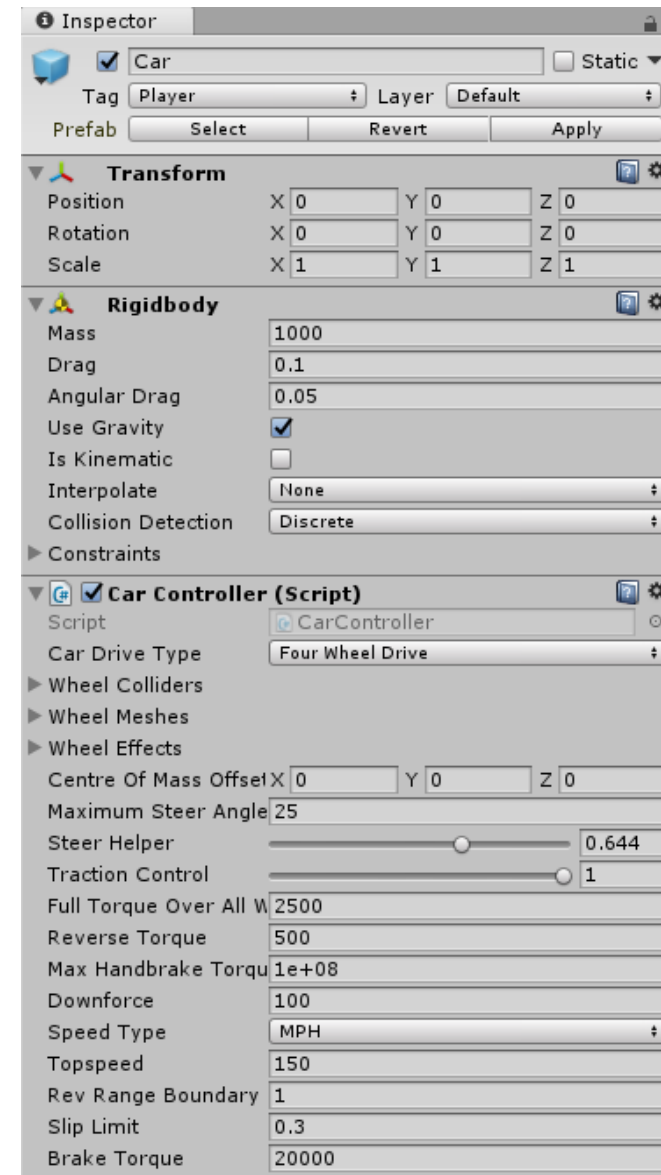
Prozor Hierarchy

- Služi za pregled objekata u sceni
- Svi GameObjects (objekti) koji su instancirani u sceni se pojavljuju u prozoru hijerarhije
- Objekti mogu biti instancirani direktno iz asset datoteka ili iz prefab datoteka (predložaka)
- Lako selektiranje objekata u sceni
- Odnos roditelj – dijete omogućuje grupiranje sličnih objekata slično sustavu direktorija
- Unity renderira objekte redom kojim su navedeni u listi, ali prvo ulazi u objekte roditelje i izvodi sve unutar njih



Prozor Inspector

- Služi za pregled komponenata unutar objekta
- Kada se objekt selektira iz prozora scene ili prozora hijerarhije detalji o objektu se prikazuju u prozoru inspektor
- Svaki objekt definiran je određenim brojem komponenti
- Komponente mogu biti – pozicija, 3D podaci, skripte, audio, animacije, čestice...
- Svaka od karakteristika ima svoje opcije – detaljnije na sljedećim predavanjima



Traka s alatima

- Alati za transformaciju:

- Pomicanje u sceni
- Translaciju objekta
- Rotaciju objekta
- Mijenjanje veličine odnosno povećavanje/smanjivanje objekta
- Rastezanje objekta
- Translacija/rotacija/mijenjanje veličine

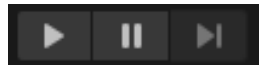


- Pozicioniranje lokacije i orijentacije alata za upravljanje (primjerice strelica osi)

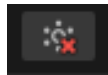
- Opcije – centar (centar selektiranog objekta) i pivot (centar samog mesha odnosno pivot točka oko koje se rotira objekt) te lokalni (relativno u odnosu na objekt) i globalni (u odnosu na centar scene) – omogućuje pomicanje po globalnoj mreži što je Grid Snapping alat



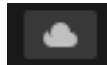
- Pokretanje igre / pauza / izračun jednog okvira



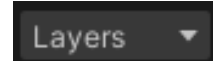
- Alati za suradnju



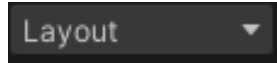
- Unity usluge (reklame, analitika, itd.)



- Slojevi odnosno kontrola prikaza objekata u prozoru scene



- Organizacija korisničkog sučelja

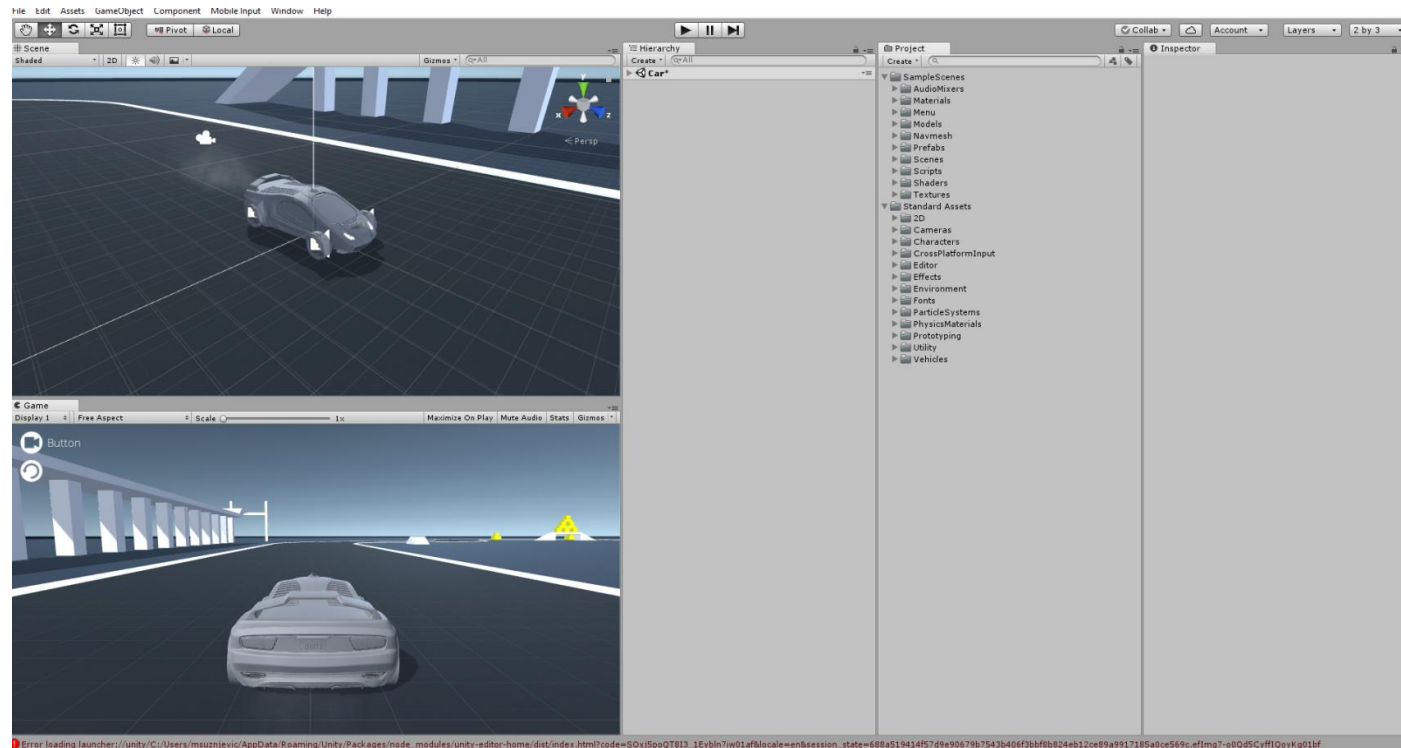


- Korisnički račun



Izgled korisničkog sučelja

- Vrlo konfigurabilan
- U predavanjima će se koristiti 2 by 3 ili nazivni raspored
- U 2 by 3 rasporedu istovremeno aktivni prozori igre i scene

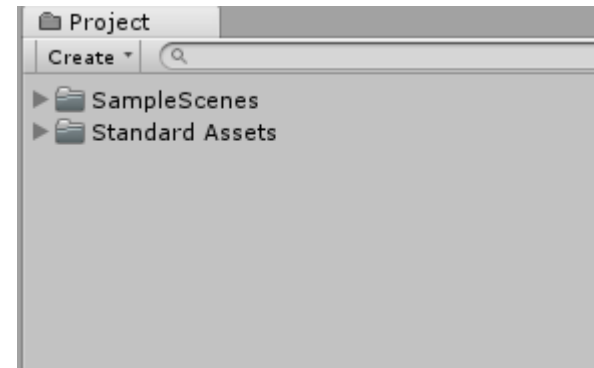
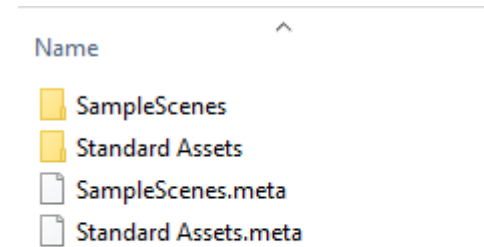


Korišćenje asseta (sredstava)

- Unity prihvaća uvoz raznih tipova datoteka (3D modeli, slike, audio datoteke, skripte itd.)
- Mogućnost kreiranja određenih tipova sredstava u Unityiu (3D objects Animator Controller, Audio Mixer i Render Texture)
- 3D oblici koji se mogu kreirati direktno u Unityiu kroz naredbe (**GameObject > 3D Object**):
 - Kocka
 - Sfera
 - Kapsula
 - Valjak
 - Ploha
 - Quad (jednostavna vertikalna ploha)

Uvoz sredstava

- Datoteke u asset direktoriju se direktno prenose u projektni prozor u Unityiu
- Možemo unijeti nova sredstva u Unity projekt kopirajući ih direktno u Assets folder ili koristeći sučelje (**Assets > Import New Asset**)
- Svaki uvezeni asset ima određene postavke
 - Slika biti pohranjena kao tekstura, mapa ili 2D sprite
 - Audio kodiran određenim bitrateom
 - Postavke uvoza se mogu postaviti u inspektor prozoru
- Tipovi asseta:
 - 3D modeli (.max, .blend, .mb, .ma)
 - Slike
 - 3D mesh (mreže)
 - Animacije
 - Audio

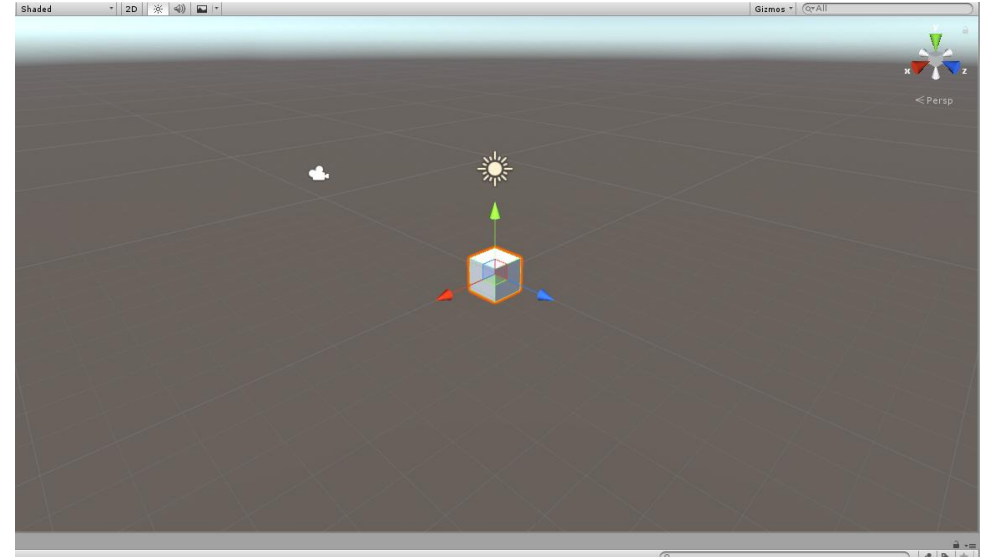


Uvoz paketa

- Za upravljanje paketima koristi se Package Manager
- Veliki broj asseta koji su zapakirani u određeni paket
- Standard assets – dolazili nekada predefinirani s Unity, sada se mora posebno skinuti iz trgovine
- Custom asset packages – mogu doći s Unity Store ili iz drugih izvora
- Komanda Assets -> Import Package - > Custom Package

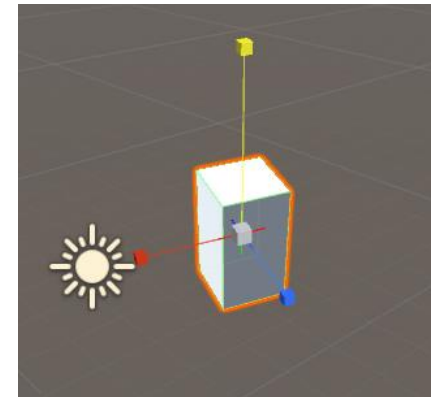
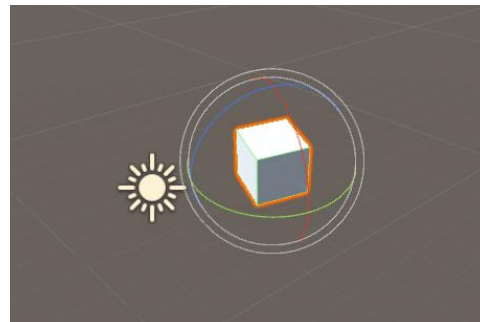
Kreiranje predefiniranog objekta

- GameObject > 3D object > Cube
- Podsjetnik na interakciju
 - Lijevi klik selektira objekt
 - Držanje srednjeg gumba na mišu pomiče poziciju s koje korisnik gleda na svijet
 - Držanje desnog gumba mijenja točku na koju korisnik gleda
- Probajte „napraviti krug” oko kocke
- Dok je objekt selektiran pritiskom na gumb „F” na tipkovnici se zumira na objekt (za lakše pronalaženje objekata i restartiranje pogleda)



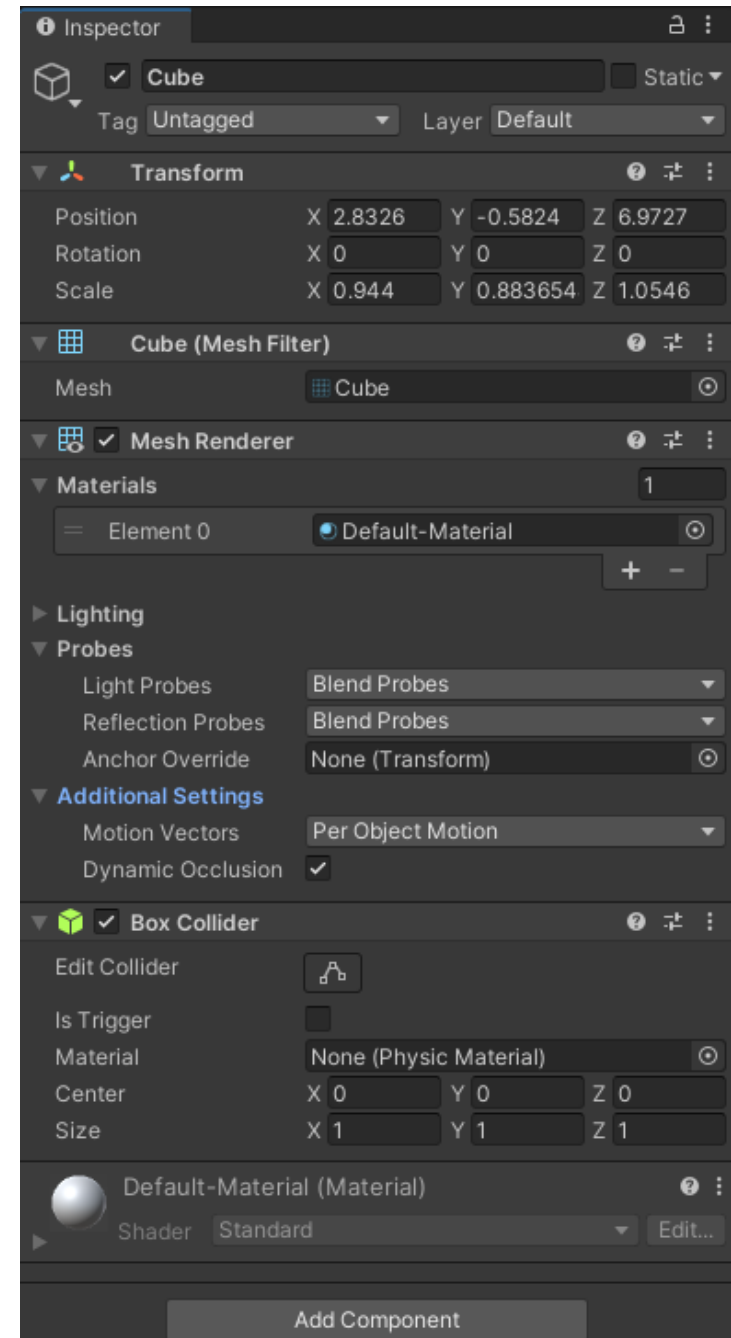
Transformacija objekta

- Translacija – korištenjem strelica na kocki
 - Odaberite jednu od osi te lijevim klikom miša i povlačenjem pomjerate objekt po toj osi
 - Odabir opcije Global i Grid Snapping te povlačenje mijenja vrijednost translacije u koracima od 1
- Rotacija – korištenjem kružnica oko kocke
 - Rotirajte objekt prvo selektirajući osu rotacija lijevim gumbom miša te pomjerajući miša
- Skaliranje na istom principu kao translacija
- Rastezanje – istovremeno mijenjanje skale na dvije osi kasnije korisno za sučelja



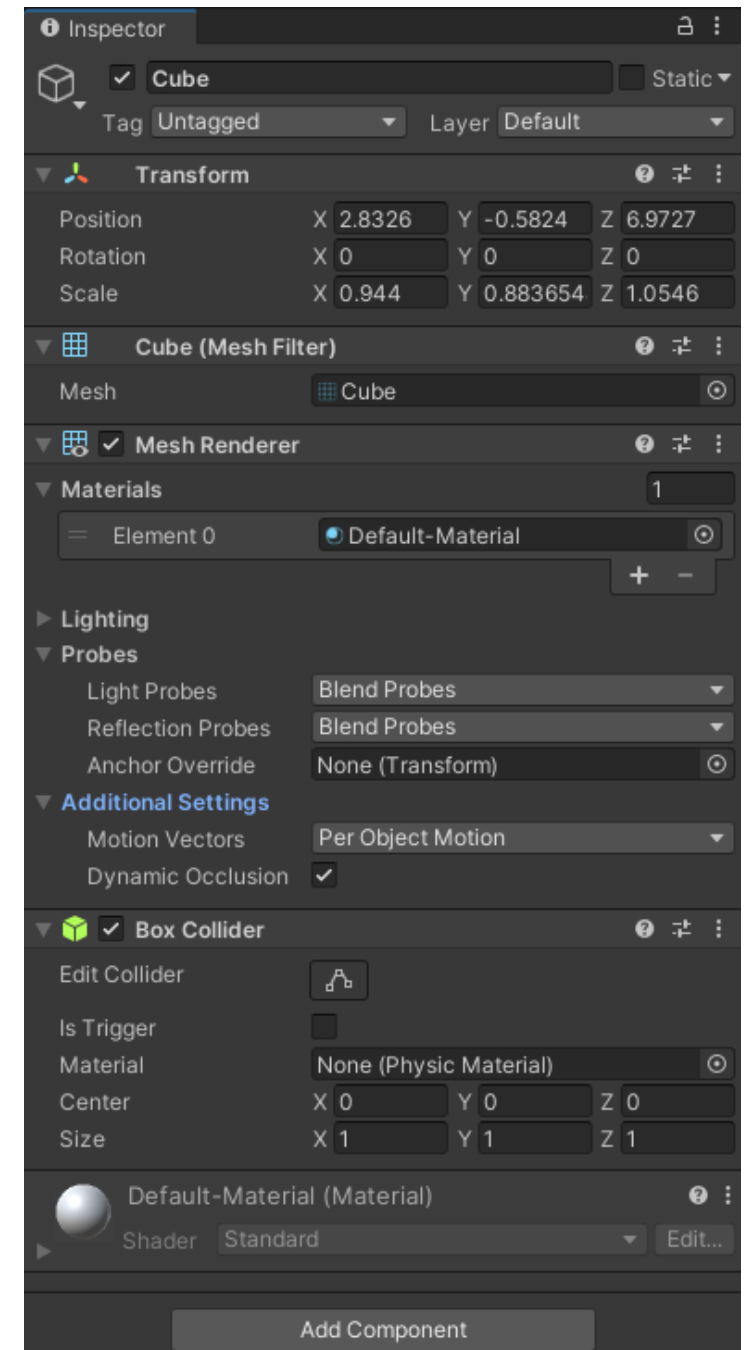
Komponente objekta

- Transform – ima svaki objekt
 - Resetiranje objekta na početne postavke settings (zupčanik u desnog gornjem kutu) > Reset
 - Resetiranje objekta moguće i postavljanjem svih vrijednosti na 0 ručno
- Cube (Mesh Filter)
 - Preuzima iz sredstava oblik mesha te ga dostavlja Mesh Rendereru
 - Odabir mesha – moguće odabrati drugi mesh (primjerice sferu)
- Box Collider
 - Služi za detekciju sudara
 - Is Trigger opcija omogućava pokretanje određene reakcije ako dođe do kolizije (u tom slučaju ne reagira kao čvrsto tijelo)
 - Materijal opcija postavlja fizikalni materijal (kako reagira na koliziju s drugim tijelima – detaljnije o fizici u idućem predavanju)
 - Opcije center i size – položaj i veličina



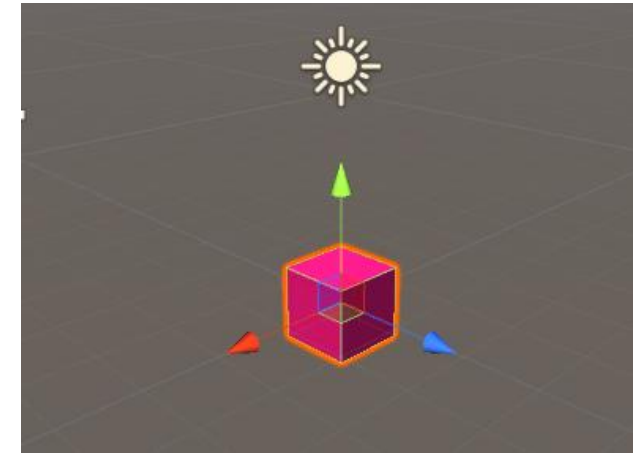
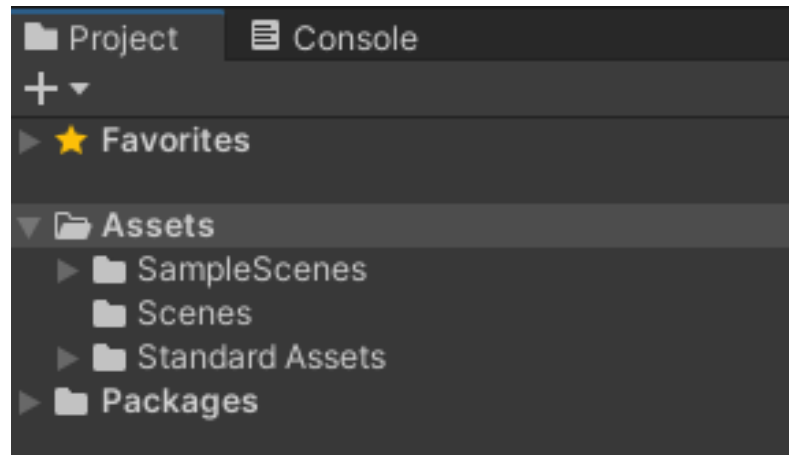
Komponente objekta

- Mesh Renderer
 - Uzima podatke o obliku mesha od Mesh Filtera te ga iscrtava
 - Materials – lista materijala koji služe za renderiranje objekta
 - Materijali određuju kako će biti objekt renderiran: teksture, popločavanje, nijanse boja i slično
 - Niz opcija vezanih za osvjetljenje (uključujući probes) – detaljnije u priručniku:
<https://docs.unity3d.com/Manual/class-MeshRenderer.html>
 - Shader – skripta koja računa na koji način će se renderirati boja svakog pojedinačnog piksela ovisno o izvoru osvjetljenja i konfiguraciji materijala



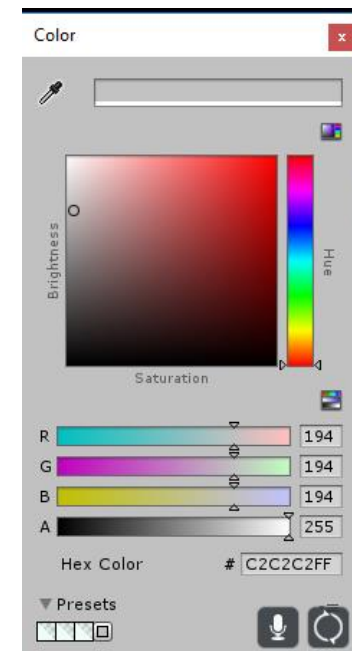
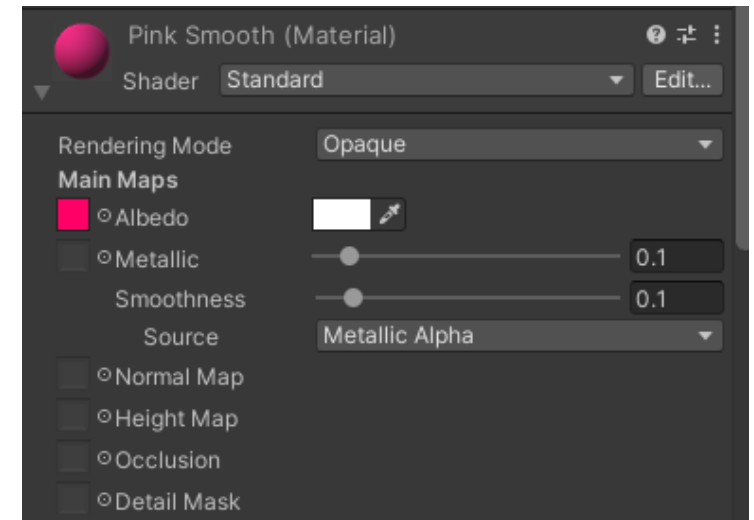
Uvoz objekta iz Standard assets direktorija

- Uvezimo standard assets
(ako niste prije): Window> Package Manager -> Standard Assets -> Import
- Novi izgled prozora projekt
- Pristup novim materijalima
 - Selektirajte kocku (kroz prozor hijerarhije ili prozor scene)
 - Odaberite na Mesh rendereru u materials dijelu točkicu s krugom
 - Odaberite proizvoljan materijal
 - Novi izgled kocke s PinkGrid materijalom



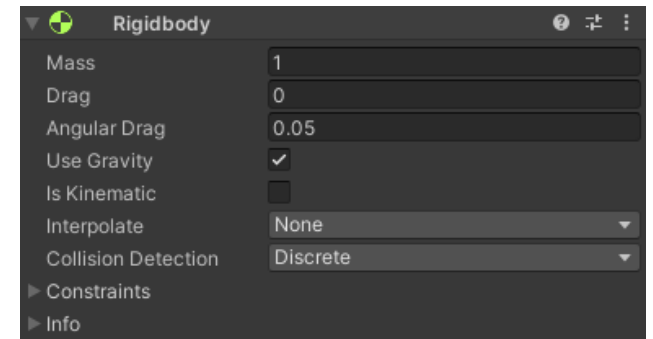
Podešavanja standard shadera

- U izborniku shader odaberite Standard
- Albedo
 - Diktira osnovnu boju površine predmeta (uz već definirani materijal)
 - Često se asocira tekstura umjesto osnovne boje
 - Izbor boje omogućuje definiranje i numerički (RGB) te i A (alpha) odnosno transparentnost
 - Više detalja na predavanju o shaderima
- Metallic – kontrolira koliko je materijal metalan odnosno količinu odsjaja



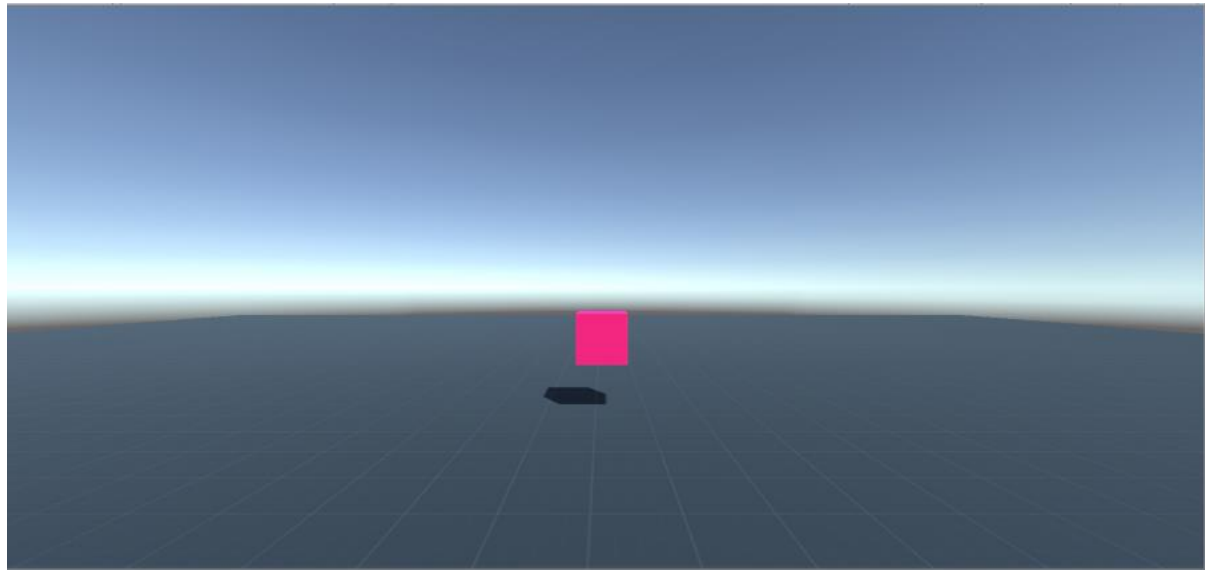
Dodavanje osnovne fizike

- Odaberite opciju play – što se desi?
- Zaustavite igru (***sve promjene dok je igra aktivna su privremene!!!!***)
- Za kocku odaberite opciju Add component > Physics > Rigidbody
- Rigidbody dodaje komponentu koja omogućuje gravitaciju, utjecaj sila na objekt te općenito realistično kretanje i djelovanje sila na objekt
 - Mass – masa objekta u kg
 - Drag – otpor zraka
 - Angular Drag – otpor zraka pri rotaciji
 - Use Gravity – je li na objekt djeluje gravitacija
 - Is Kinematic – ako je opcija uključena na objekt ne djeluje fizika
- Odaberite opciju play – što se sada desi?
- Kako ćemo napraviti da naša kocka ne propadne “kroz svijet”?



Uvoz objekta iz Standard Assets

- Dodajte pod u scenu kroz opciju: Prozor project > Assets> Standard Asstes > Prototyping > Models > FloorPrototype64x1x64
- Ako pokrenete igru hoće li kocka propasti kroz pod?
- Ponovno pokrenite igru
 - Propada li kocka kroz pod?
 - Zašto?



Postavljanje funkcionalnog poda

- Primijetite da pod na sebi nema BoxColider - dodajemo ga kroz Add component
- Ponovite eksperiment
- Što bi se desilo da smo dodali Rigidbody?
- Dodajte Rigidbody te ponovite eksperiment. Što se dešava?
- Isprobajte i opciju da komponenti Rigidbody oduzmete gravitaciju – što se dešava?
- Kako postići istu da pod ne propadne kad kocka udari u njega iako pod ima Rigidbody?

Kreiranje predloška (Prefab)

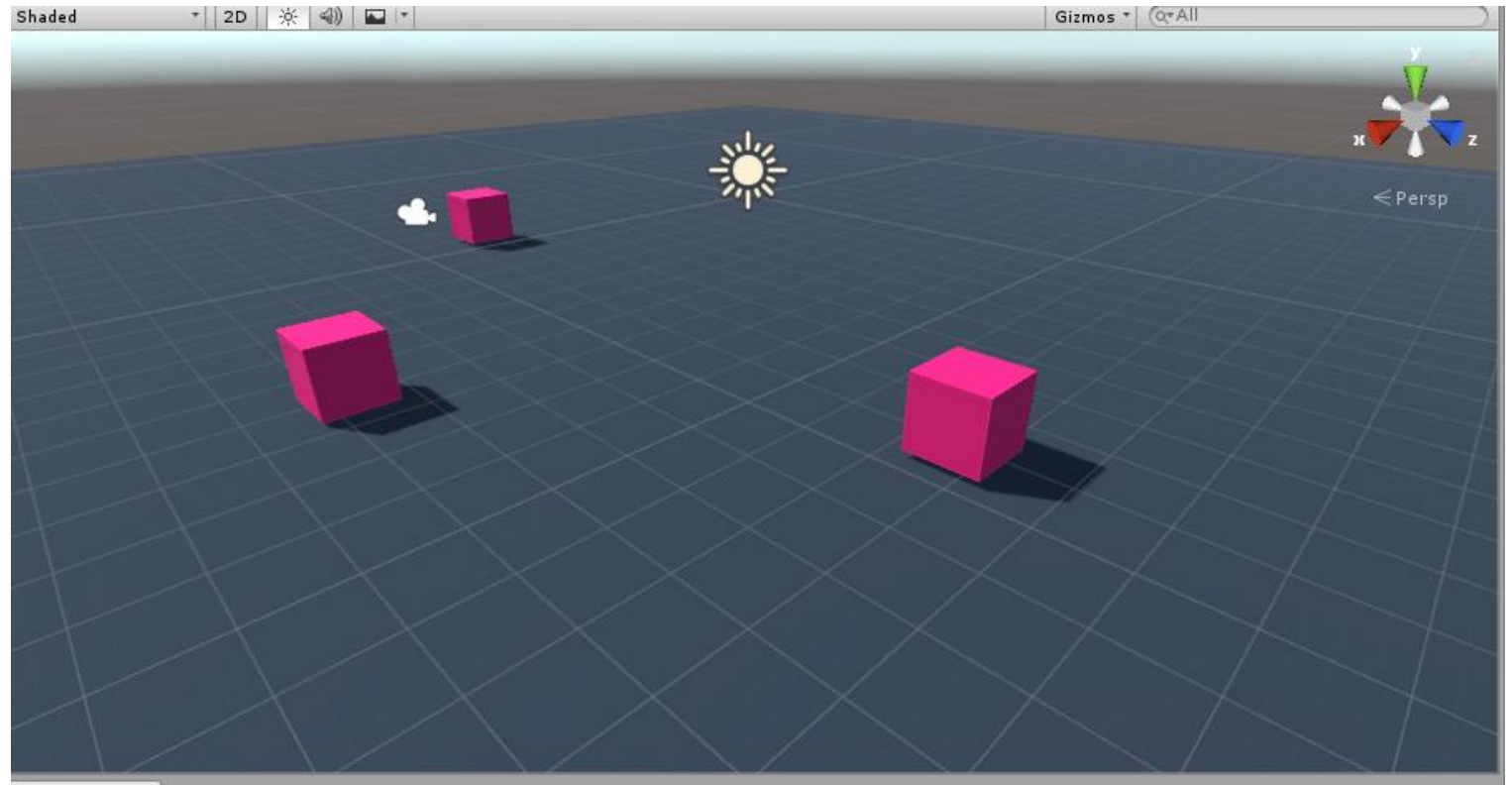
- Osnovna karakteristika predložaka (prefabs) je da promjenom na predlošku promjene se apliciraju na sve instance predloška
- U projektu kreirajte direktorij pod imenom Materials i direktorij pod imenom Prefabs unutar direktorija Assets (desni klik na pozor projekta u folderu Assets -> Create -> Folder)
- Preimenujte pod u “pod” te cube u “kocka” u prozoru hijerarhije
- Lijevim klikom miša povucite kocku iz prozora hijerarhije u Prefabs direktorij
- Obrišite kocku iz prozora hijerarhije

Dupliciranje materijala

- Odaberite kocku u prozoru projekt
- Odaberite PinkSmooth material (direktno klikom na ime materijala) u Mesh Render komponenti > otvara Materials direktorij u Standard Assets direktoriju
- Duplicirajte materijal (ctrl + D) te prevucite duplikat u direktorij Materials u Assets direktoriju u prozoru projekta
- Preimenujte materijal iz PinkGrid1 u Kocka Materijal
- Odaberite Kocku u Prefabs direktoriju i sada za materijal odaberite upravo napravljeni Kocka Material (odabirete klikom na kružić s točkicom)
- Na ovaj način odvajamo materijal od standardnih asseta te možemo na njemu raditi određene promjene

Instanciranje objekata iz predloška

- Prvo resetirajte pod na poziciju 0 0 0
- Prevucite nekoliko objekata kocka iz Prefabs direktorija u scenu te ih pozicionirajte na pod

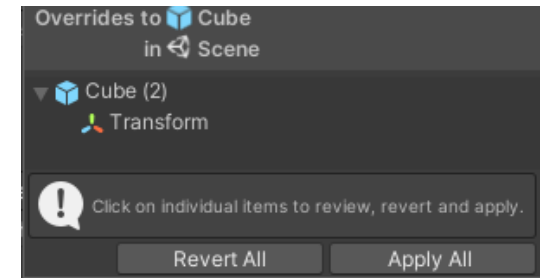


Organizacija prozora hijerarhije

- Prozor hijerarhije se sada bespotrebno puni s istovrsnim objektima -> potrebno ga organizirati
- Kreirajte novi prazan objekt `GameObject > Create Empty` te ga nazovite `Spremiste za kocke`
- Sve kocke prevucite u taj novi objekt
 - Novi objekt postaje roditelj, a kocke djeca
 - Pozicije kocaka su relativne u odnosu na njihovog roditelja – dobro je prazne objekte koji služe kao roditelji uvijek pozicionirati na 0 0 0
 - Ovaj princip omogućava bolju organizaciju kod složenijih scena

Mijenjanje predloška i instanci

- Promjena svih instanci istovremeno
 - Odaberite materijal Kocka Material u projekt prozoru
 - Promijenite Albedo na materijalu u drugu opciju, primjerice SwatchTeal (opet odaberite novi materijal kroz kružić s točkom)
 - Primijetite da su sve promjene direktno aplicirane na sve instance objekta
- Promjena jednog objekta te propagiranje na predložak
 - Odaberite objekt Kocka (2) iz prozora hijerarhije
 - Promijenite mu skalu na 2, 2, 2
 - Primijetite da se promijenila samo jedna instanca – Kocka (2)
 - Odaberite opciju Overrides > Apply all – promjena se propagira na predložak
 - Moguće i samo pojedinačnu karakteristiku promijenjenu primijeniti na prefab kroz desni klik na ime parametra (u našem slučaju na parametar Scale u komponenti transform) i Apply to prefab



Fizika u Unityu

- Cilj – omogućiti u igri da ima „uvjerljivu” fiziku odnosno fiziku koja je dovoljno uvjerljiva simulacija stvarne fizike
- Osnovna komponenta takve fizike je utjecaj sila
 - Gravitacija
 - Sudari čvrstih tijela
 - Utjecaji ostalih sila (trenje, elastična sila, i sl.)
 - Utjecaj fizikalnih materijala
- Unity posjeduje odvojene simulacije fizike (2D i 3D)
 - Koncepti su identični u obje simulacije
 - Različite komponente (primjerice Rigidbody za 3D simulaciju fizike i Rigidbody 2D za simulaciju 2D fizike)

Komponente za osiguranje fizike

- Osnovne komponente za osiguranje fizike
 - **Rigidbody** (čvrsto tijelo) – omogućuje djelovanje sila na objekt
 - **Collider** (sudarač) – omogućuje detekciju sudara objekta s drugim objektima koji imaju sudarače
 - Joints (zglobovi) – komponenta za povezivanje dvaju objekata koji imaju Rigidbody komponentu; postoji više različitih implementacija ove komponente ovisno o ograničenjima pojedinog zgloba
 - Character controllers (kontroleri likova) – komponenta koja daje jednostavan sudarač liku u obliku kapsule koja je uvijek vertikalna te ima posebne funkcije za definiciju brzine i usmjerenja tako da komponenta Rigidbody nije potrebna

Rigidbody

- Osnovna komponenta koja omogućuje simulaciju fizike na pojedinom objektu (utjecaj sila i obrtnog momenta)
- Automatski uključuje gravitaciju na objektu (moguće isključiti označavanjem odgovarajuće opcije na komponenti)
- Ponekad je potrebno da nam objekt posjeduje Rigidbody komponentu, ali da ga ne kontrolira simulacija fizike
 - Primjerice hoćemo da je naš objekt kontroliran posebnim skriptama, ali želimo omogućiti korištenje okidača (engl. triggers)
 - Ovo omogućuje opcija Is Kinematics koja ukida kontrolu od simulacije fizike i omogućuje kontrolu dodatnim skriptama
 - Opcija koja dosta utječe na performanse – donosi dodatno izračunavanje te je treba oprezno koristiti

Rigidbody

- Utjecajem sila kroz Rigidbody utječemo na komponentu Transform (mijenja se pozicija i rotacija)
- Nemojte koristiti skriptno mijenjanje Transform komponente u isto vrijeme u kojem se koristi fizikalna simulacija kroz Rigidbody komponentu!!!
- Dodavanje sila na objekt koristi metode `AddForce()` i `AddTorque()`
- Veličina mesha je važna za izračun fizike
- Masa definira kako će objekti reagirati kada se sudare
- Otpor zraka (drag) definira koliko će zrak utjecati na brzinu padanja objekta (što je niža vrijednost manji je otpor zraka 0.001 za metal dok je 10 za pero)
- Ne detektiraju sudare – za to je nadležne Collider komponente

Collider

- Collider komponenta definira oblik objekta u svrhu detekcije sudara s drugim objektima
- Unutar Unity sustava imamo sljedeće definirane oblike 3D Collidera te njihove pandane u 2D sustavu
 - Box
 - Sphere
 - Capsule
 - Terrain
 - Wheel
- Oblike koji poprima Collider komponenta može, a ne mora biti istovjetan 3D meshu objekta na koji je postavljen
 - Izračun detekcije sudara za složenije objekte može biti vrlo zahtjevan
 - S optimalnim pozicioniranjem jednostavnih Collider komponenti moguće je definirati dosta precizno složenije objekte
 - Treća opcija – kreirati jednostavniju verziju mesha na temelju koje će se definirati collider koji je istovjetan meshu
- Statički sudarači – objekti bez Rigidbody komponente (zidovi, podovi...)
- Dinamički sudarači – objekti sa Rigidbody komponentom (projektili, vozila...)

Trigger

- Trigger (okidač) – događaj sudara može „okinuti” nekakvu drugu funkcionalnost zapisanu u skripti
 - Primjerice, pogodak projektilom u tijelo mete meti smanjuje zdravlje
 - Postavlja se tako da se označi odgovarajuća opcija u Collider komponenti
- Pozivi funkcija povezani s kolizijom
 - OnCollisionEnter – poziva se prilikom detekcije sudara
 - OnCollisionStay – poziva se u svakom osvježavanju dok je sudar aktivan
 - OnCollisionExit – poziva se u slučaju kada je kontakt između dva objekta izgubljen
- Okidači pozivaju OnTriggerEnter, OnTriggerStay i OnTriggerExit
- U slučaju detekcije kolizije barem jedan objekt mora imati Rigidbody komponentu (s kinematic opcijom isključenom) da bi se pozvao OnCollisionEnter, dok to ne vrijedi za slučaj okidača gdje se OnTriggerEnter poziva bez obzira

♦ Usporedba različitih opcija detekcije sudara

Collision detection occurs and messages are sent upon collision

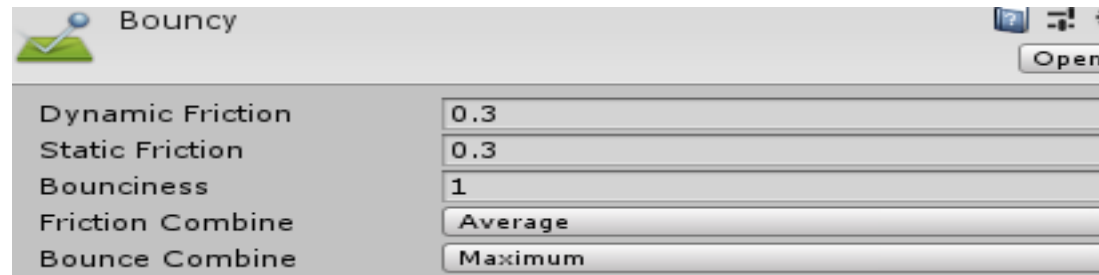
	Static Collider	Rigidbody Collider	Kinematic Rigidbody Collider	Static Trigger Collider	Rigidbody Trigger Collider	Kinematic Rigidbody Trigger Collider
Static Collider		Y				
Rigidbody Collider	Y	Y	Y			
Kinematic Rigidbody Collider		Y				
Static Trigger Collider						
Rigidbody Trigger Collider						
Kinematic Rigidbody Trigger Collider						

Trigger messages are sent upon collision

	Static Collider	Rigidbody Collider	Kinematic Rigidbody Collider	Static Trigger Collider	Rigidbody Trigger Collider	Kinematic Rigidbody Trigger Collider
Static Collider					Y	Y
Rigidbody Collider				Y	Y	Y
Kinematic Rigidbody Collider				Y	Y	Y
Static Trigger Collider		Y	Y		Y	Y
Rigidbody Trigger Collider	Y	Y	Y	Y	Y	Y
Kinematic Rigidbody Trigger Collider	Y	Y	Y	Y	Y	Y

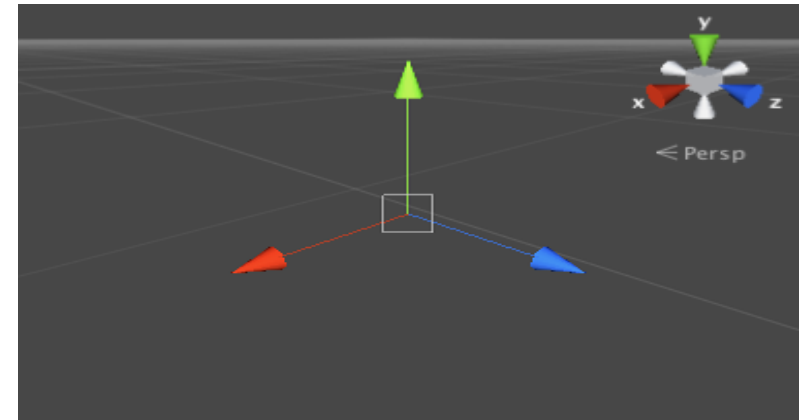
Physic Material

- Fizikalni materijal (Physic Material) definira kako se tijelo ponaša nakon što se dogodi sudar odnosno utjecaj sila poput sile trenja i elastične sile
 - Materijal leda imat će vrlo malo trenje i malu elastičnost
 - Materijal gume imat će veliko trenje i veliku elastičnost
- Tipovi materijala
 - Bouncy
 - Ice
 - Metal
 - Rubber
 - Wood
 - MaxFriction
 - ZeroFriction
- Trenje može biti:
 - Statičko – objekt miruje
 - Dinamičko – objekt se kreće
- Kada dva tijela dođu u kontakt trenje i sila elastičnosti se računaju na temelju modela koji su za njih definirani
- Prioritet modela: Average < Minimum < Multiply < Maximum.



Programsko dodavanje sile

- Kroz skriptiranje možemo jednostavno dodati utjecaj sile na objekt
- `Rigidbody.AddForce` (vektor (obvezan), tip sile (neobvezan))
- Vektor se sastoji od usmjerenja i intenziteta
- Može se koristiti orijentacija samog objekta za vektor
 - Za primjenu sile na Z osi koristimo `transform.forward`
 - Za primjenu sile na Y osi koristimo `transform.up`
 - Za primjenu sile na X osi koristimo `transform.right`
- Sila koja se primjenjuje indicira se brojkom
- Tip sile koji se primjenjuje:
 - Force – kontinuirana sila koja na koju djeluje masa ($F = m \cdot a$). Ako se izostavi parametar tip sile ovo je zadana vrijednost (engl. default)
 - Acceleration – kontinuirana primjena sile na koju ne djeluje masa
 - Impulse – instant primjena sile na koju djeluje masa
 - VelocityChange – instant primjena sile na koju ne djeluje masa



Programsko dodavanje okretnog momenta sile (eng. torque)

- Kroz skriptiranje isto možemo dodati okretni moment sile
- `Rigidbody.AddTorque` (vektor (obvezan), tip sile (neobvezan))
- Vektor se sastoji od usmjerenja odnosno osi oko koje će se objekt rotirati te od intenziteta rotacija
- Sila koja se primjenjuje indicira se brojkom
- Tip sile koji se primjenjuje:
 - Force – kontinuirana sila koja na koju djeluje masa ($F = m \cdot a$). Ako se izostavi parametar tip sile ovo je zadana vrijednost (engl. default)
 - Acceleration – kontinuirana primjena sile na koju ne djeluje masa
 - Impulse – instant primjena sile na koju djeluje masa
 - VelocityChange – instant primjena sile na koju ne djeluje masa
- Angular drag – otpor zraka čijim povećavanjem možemo zaustaviti „slobodno rotirajući” objekt

Fizikalna simulacija neba

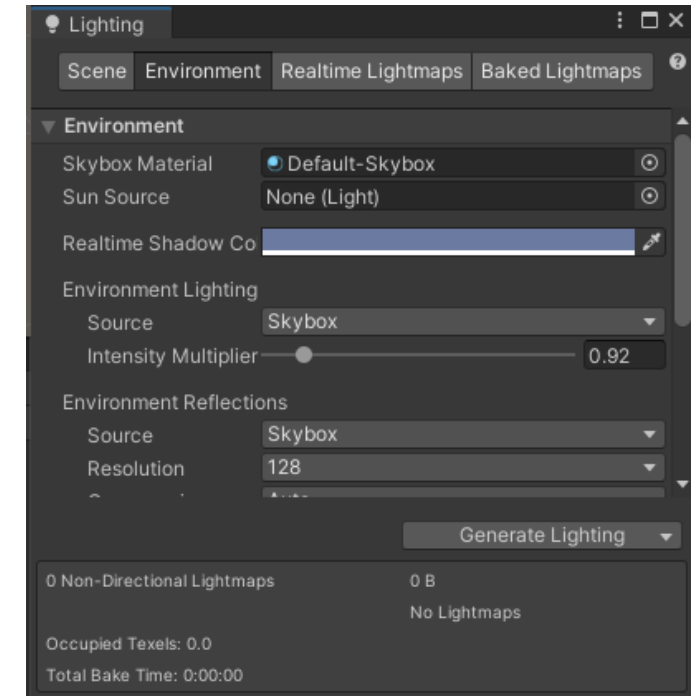
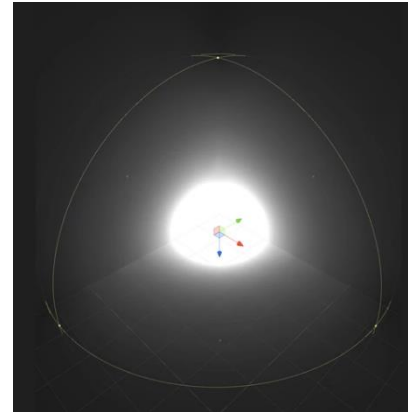
- Ne postoji – koristi se Skybox (nebeski svod) koji prikazuje kako svijet izgleda iza definirane geometrije (zapravo vrlo detaljna tekstura)
- Skybox se renderira oko cijele scene kako bi se prikazala kompleksan prizor koje se nalazi oko scene (primjerice može biti morska dubina, svemir, unutrašnjost nekog objekta itd...)
- Skybox se renderira preko oblika kocke, sfere ili sfere sastavljene od poligona
- Unutar Unitya Skybox ima poseban materijal – Skybox Material
- Skybox se može definirati na razini scene ili na razini kamere (u slučaju da želimo da kamera ima svoj specifičan Skybox)
 - Dodavanje na razini scene se vrši u Lightning Settings izborniku
 - Dodavanje na kameru se vrši pomoću komponente Skybox
- Jednostavno možete kreirati unutar Unitya svoj Skybox dodavanjem šest odvojenih tekstura i njihovim spajanjem

Zvuk

- U prirodi imamo izvore zvuka te točke slušanja – taj koncept je pretočen u Unity
 - Komponenta Audio Source (izvor zvuka)
 - Komponenta Audio Listener (slušač zvuka)
- Odbijanje zvukova od objekata (eho) se ne račina automatski već pomoću dodatnih audio filtara
- Prozor Audio Mixer omogućava kombiniranje i uređivanje raznih izvora zvuka te stavljanje efekata
- Unity omogućava uvoz raznih formata (AIFF, WAV, MP3, OGG...) u obliku asseta
- Unutar Unitya se može i snimiti određeni zvuk
- Snimljeno pozvano predavanje oko zvuka – Damjan Mravunac iz Croteama

Osvjetljenje u Unityu

- Dva tipa osvjetljenja
 - Dinamičko
 - Prethodno izračunato (eng. baked)
 - Kombinacija
- Tipovi svjetala:
 - Ambijentalno osvjetljenje (engl. Ambient lighting) – osvjetljenje objekta iz svih smjerova. Kontrolira se putem Lighting izbornika (Window -> Rendering -> Lighting -> Environment).
 - Svjetla kao objekti
 - Dodaju se kroz izbornik GameObject -> Light
 - Usmjereno svjetlo (engl. Directional lights) – osvjetljenje objekta iz određenog smjera.
 - Točkasto svjetlo (engl. Point lights) – točkasti izvor svjetla
 - Farovi (engl. Spotlights) – stožasti izvori svjetla
 - Plohe osvjetljenja (engl. Area lights) – plohe koje emitiraju svjetlo u svim smjerovima, ali samo s jedne strane plohe
- Unity omogućava dinamički izračun zrcaljenja, materijale koji zrače svjetlo, i mnoge druge opcije – posebno predavanje



Unos komandi

- Nazivni paket koji se u Unity pogonskom sustavu koristi je **InputManager**
- Omogućuje definiranje ulaznih osi na kojima se izvršavaju komande
- Omogućuje sljedeće tipove kontrola
 - „**Key**” – odnosi se na dugme na tipkovnici (primjerice „w”)
 - „**Button**” – odnosi se na dugme na upravljaču (primjerice „left b”)
 - „**virtual axis**” – odnosi se na os kretanja te se mapira na kontrolu definiranu s „key” ili „button” te prima vrijednosti u rasponu $[-1..1]$.

Primjer – korištenje gumba

```
public class ExampleClass : MonoBehaviour{
    void Update()
    {
        if (Input.GetKey("up"))
        {
            print("up arrow key is held down");
        }

        if (Input.GetKey("down"))
        {
            print("down arrow key is held down");
        }
    }
}
```

Primjer – korišćenje osi

```
float moveSpeed = 10;  
//Definiramo brzinu objekta
```

```
float horizontalInput = Input.GetAxis("Horizontal");  
//Učitavamo vrijednost unosa na horizontalnoj osi
```

```
float verticalInput = Input.GetAxis("Vertical");  
//Učitavamo vrijednost unosa na vertikalnoj osi
```

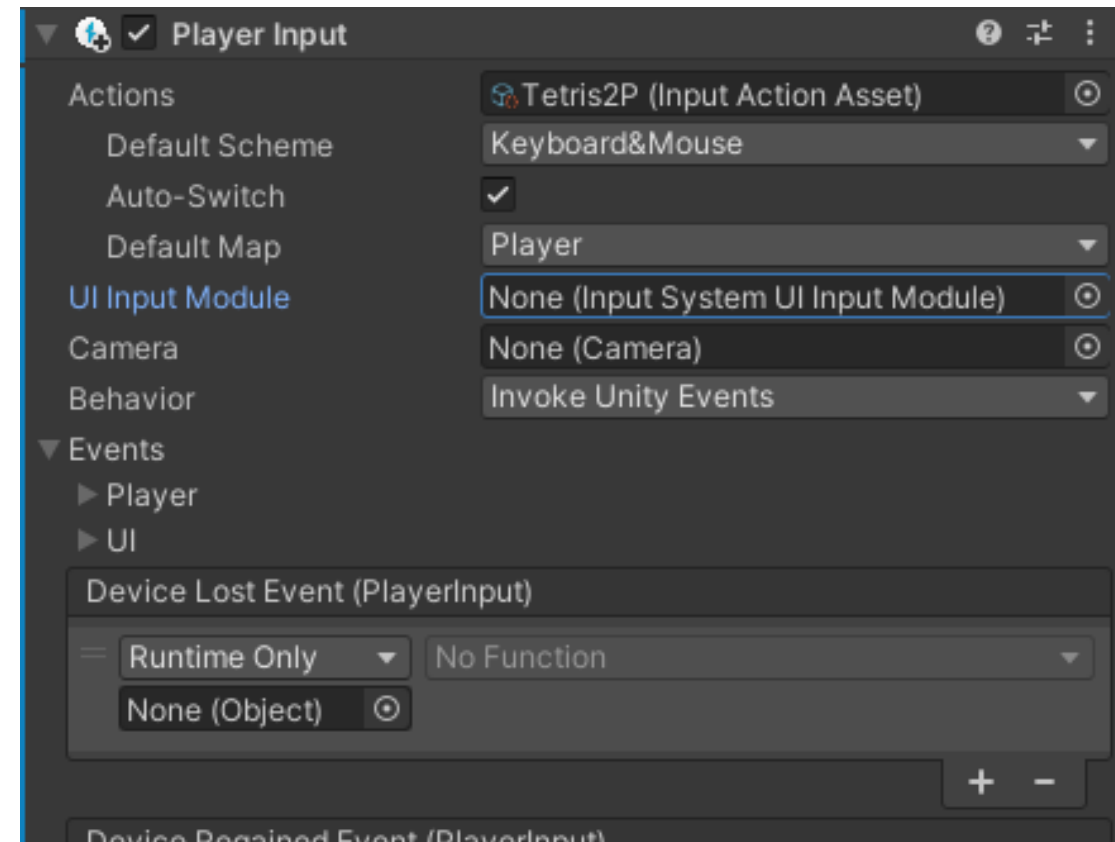
```
transform.Translate(new Vector3(horizontalInput, verticalInput, 0) *  
moveSpeed * Time.deltaTime);  
//pomjeranje objekta na temelju unosa, odnosno vektora definiranog  
istim te brzine kretanja
```

Unos komandi - Input System

- **Input System** je posebna biblioteka koja omogućuje korištenje bilo kakvog ulaznog uređaja za kontrolu igara razvijenih u Unity sustavu te nudi puno više opcija, fleksibilnosti i konfiguracija od Input Managera
 - Omogućuje kreiranje jednog kontrolera koji radi za različite ulazne uređaje
 - Omogućuje kreiranje jednog kontrolera koji radi za različite platforme (konzole, PC, mobilne uređaje)
 - Omogućuje kreiranje različitih kontrola ovisno o kontekstu igre (primjerice voda ili čvrsto tlo)
- **Input System** se instalira putem Package Managera
- Igre napravljene temeljem Input Managera neće funkcionirati nakon što je instaliran Input System

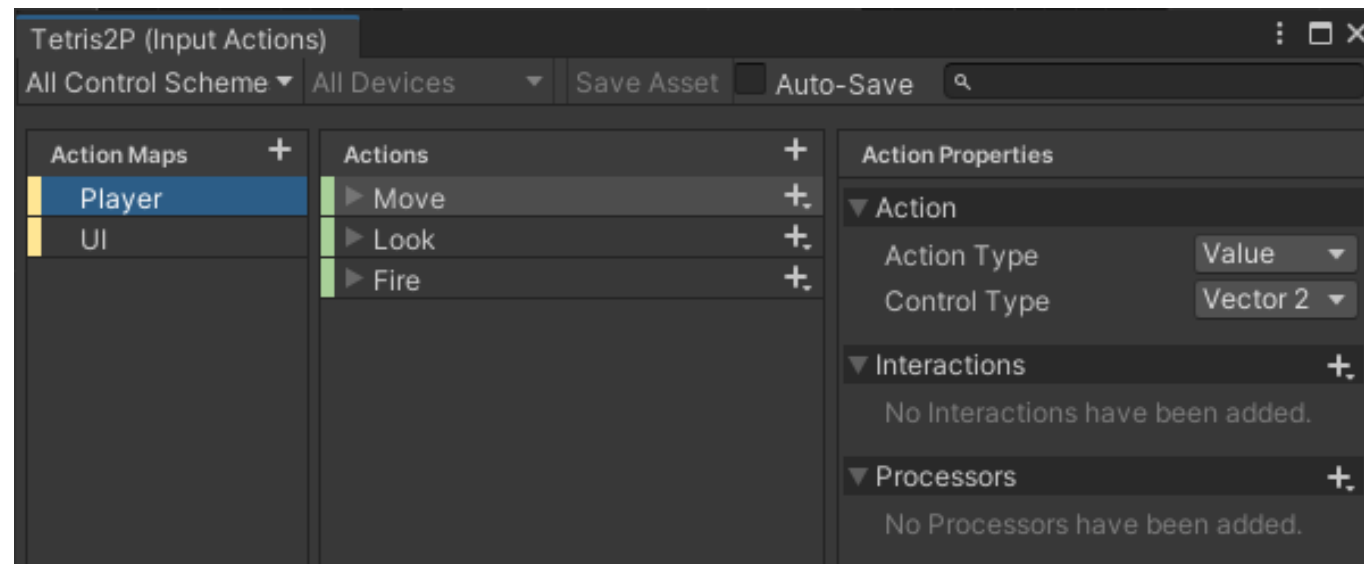
Komponenta Player Input

- Definira se za svakog igrača
- Na nju se veže sredstvo tipa Input Actions
- Može definirat Unity događaje koji mogu pozivati implementacije kroz Behavior komponentu
- Sve akcije su potom dostupne kao događaji na koje se mogu pretplatiti Callback funkcije



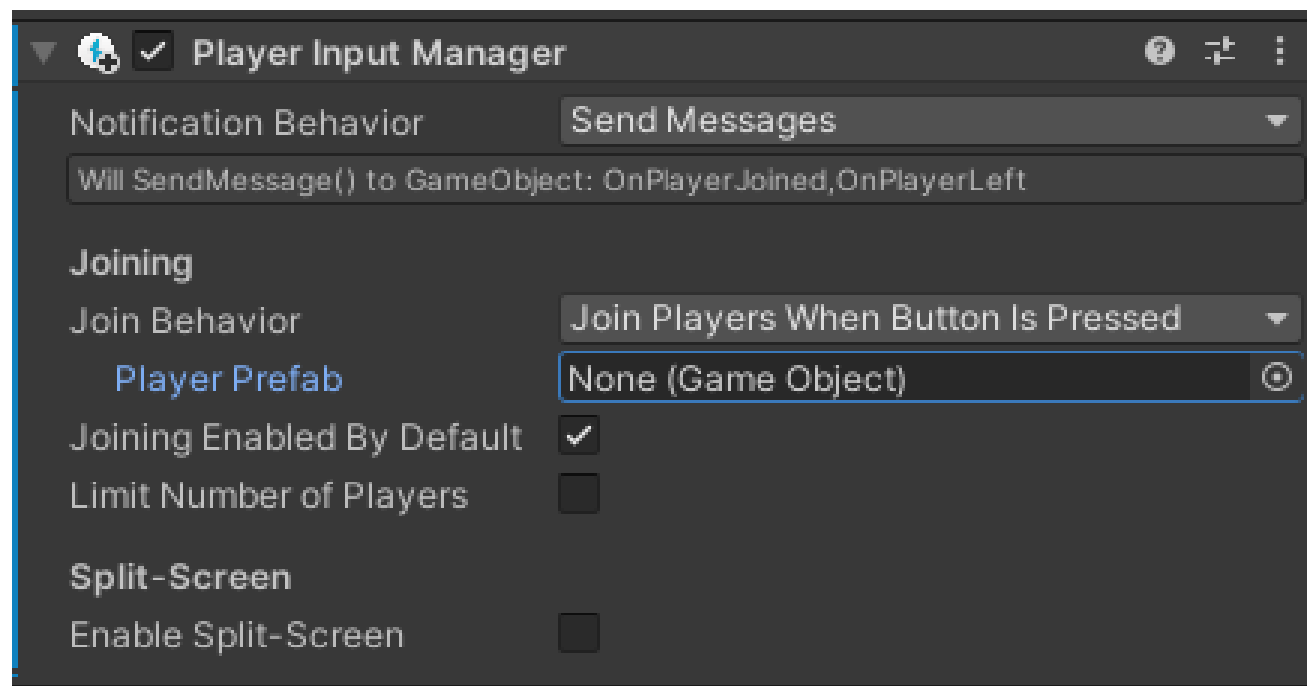
Sredstvo InputActions

- Action Map – set kontrola za jedan aspekt igre (različite kontrole za vožnju ili hodanje, ili za glavnu scenu i glavni izbornik)
- Actions – generičke akcije koje igrač može raditi (skakati ili trčati) koje ne moraju biti vezane direktno za neki gumb
- Properties – definiramo kakav tip ulaza ide za određenu akciju (samo pritisak gumba, kontinuirana vrijednost i slično)



Komponenta Input Manager

- Kontrolira dolazak i odlazak igrača iz igre
- Samo jedna instanca je aktivna u igri
- Omogućuje pridruživanje igrača na pritisak gumba
- Ako je uključen podijeljeni ekran, automatski dijeli ekran kako dolaze novi igrači



Što biste trebali znati?

- Osnovne uloge elemenata korisničkog sučelja Unityia
- Načini kreiranja 3D objekata
 - Kreiranje osnovnih objekata unutar Unityia
 - Uvoz objekata iz Standard Assets
 - Uvoz vanjski generiranih objekata
- Osnove objekata u Unityiu
- Osnove rada s prozorom inspektor
- Podešavanje osnovnih komponenti objekta
- Kreiranje predložaka
- Organizacija prozora hijerarhije
- Osnove fizike u Unityu
- Osnove unosa u Unityu

Zadatak za iduće predavanje

- Podsjetite se predavanja iz Komunikacijskih mreža vezanih za mrežni sloj: <https://www.fer.unizg.hr/download/repository/KM-2022-05.pdf>