Interakcija čovek-računar

Laboratorijske vežbe – III termin

Unity

Šta je Unity?

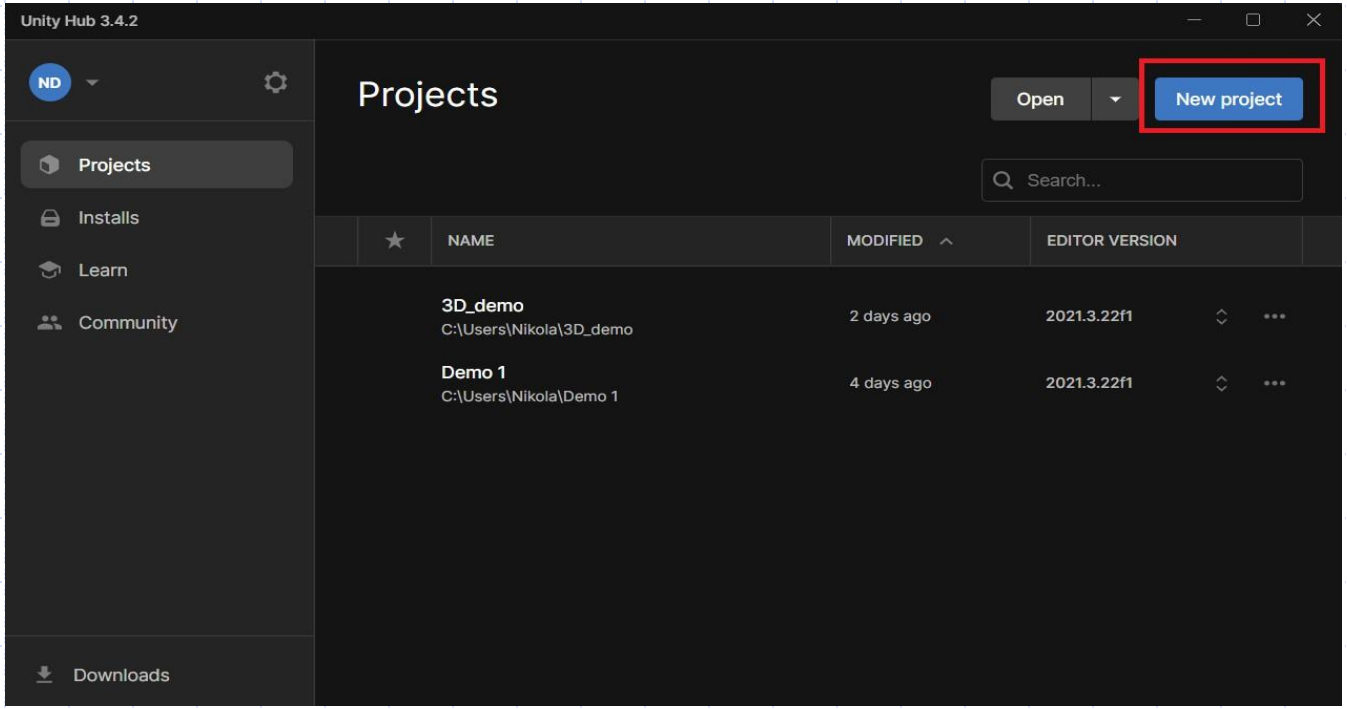
- Unity je cross-platform game engine
- Omogućava razvoj 2D i 3D interaktivnih multimedijalnih aplikacija i igara
- Predstavljen je 2005. godine od strane Unity technologies

Preuzimanje i instalacija

- Preuzeti i instalirati Unity hub (https://unity.com/download)
- Napraviti Unity ID nalog, i prijaviti se preko istog u okviru Unity hub-a
- U okvru Unity hub-a izabrati opciju za instaliranje Unity editora
- Izabrati neku od verzija editora iz LSM (Long Term Support) liste i instalirati je (poželjno je instalirati verziju 2022.3.24, jer je ona instalirana u laboratoriji na fakultetu)

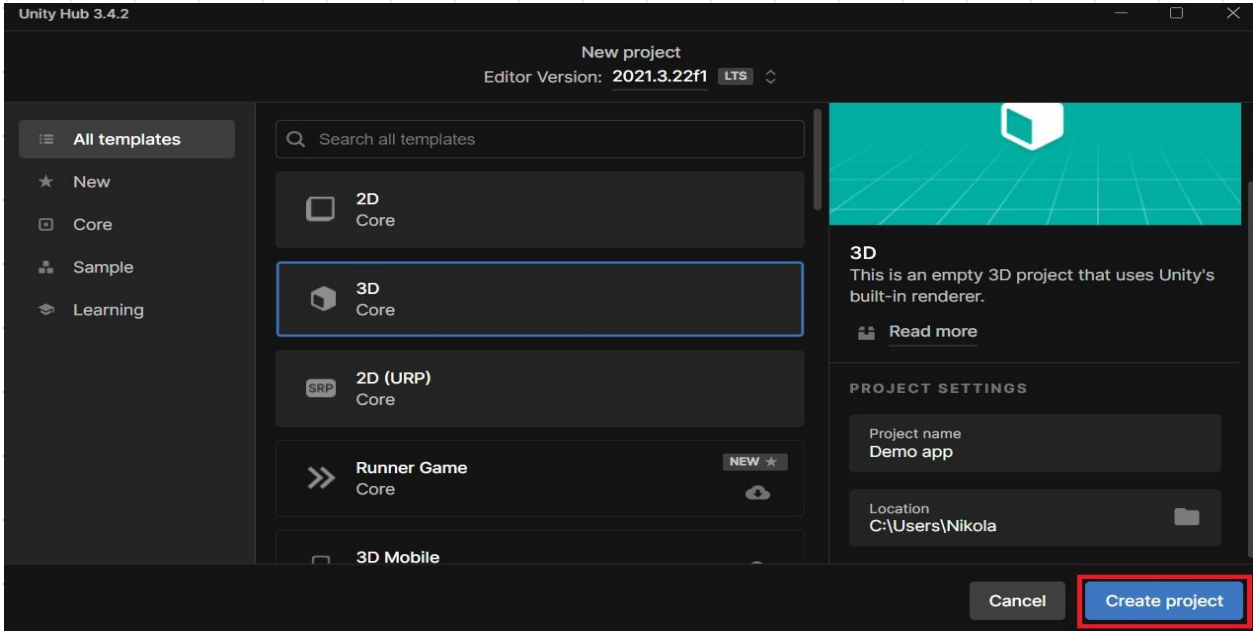
Kreiranje projekta

- Otvoriti Unity hub i izabrati opciju New project



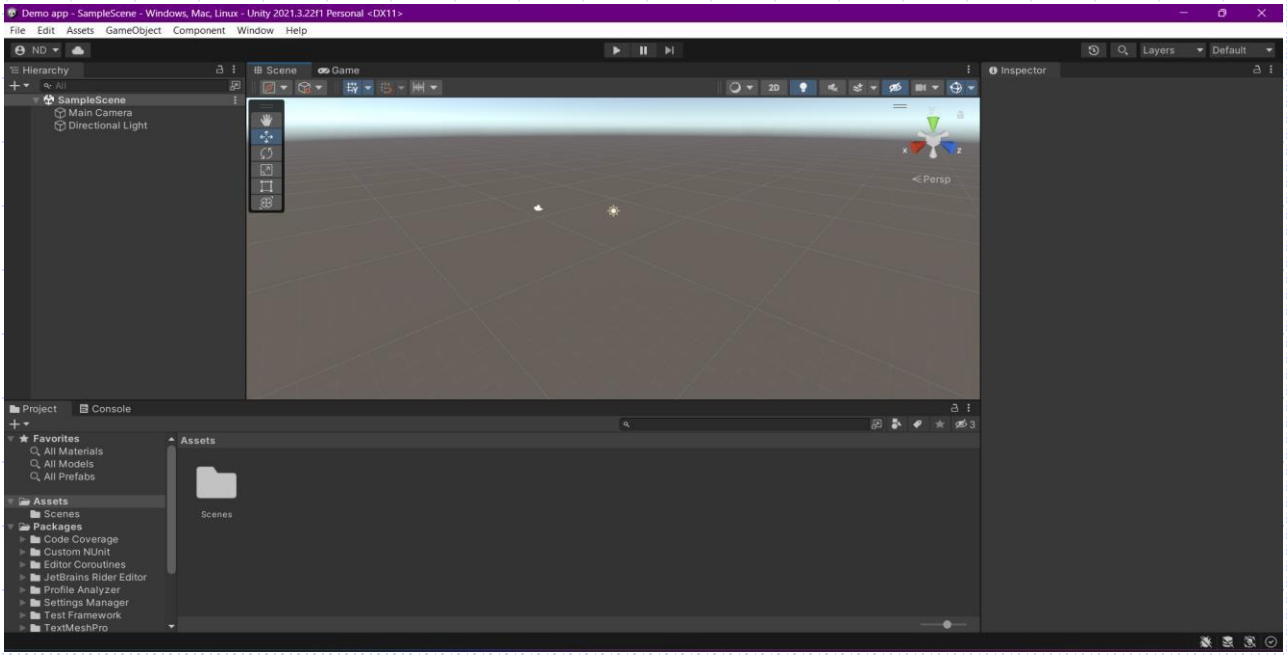
Kreiranje projekta

- Iz liste dosutpnih šablona, izabrati 3D template
- Postaviti ime projekta, lokaciju na kojoj će biti smešten i izabrati opciju Create project



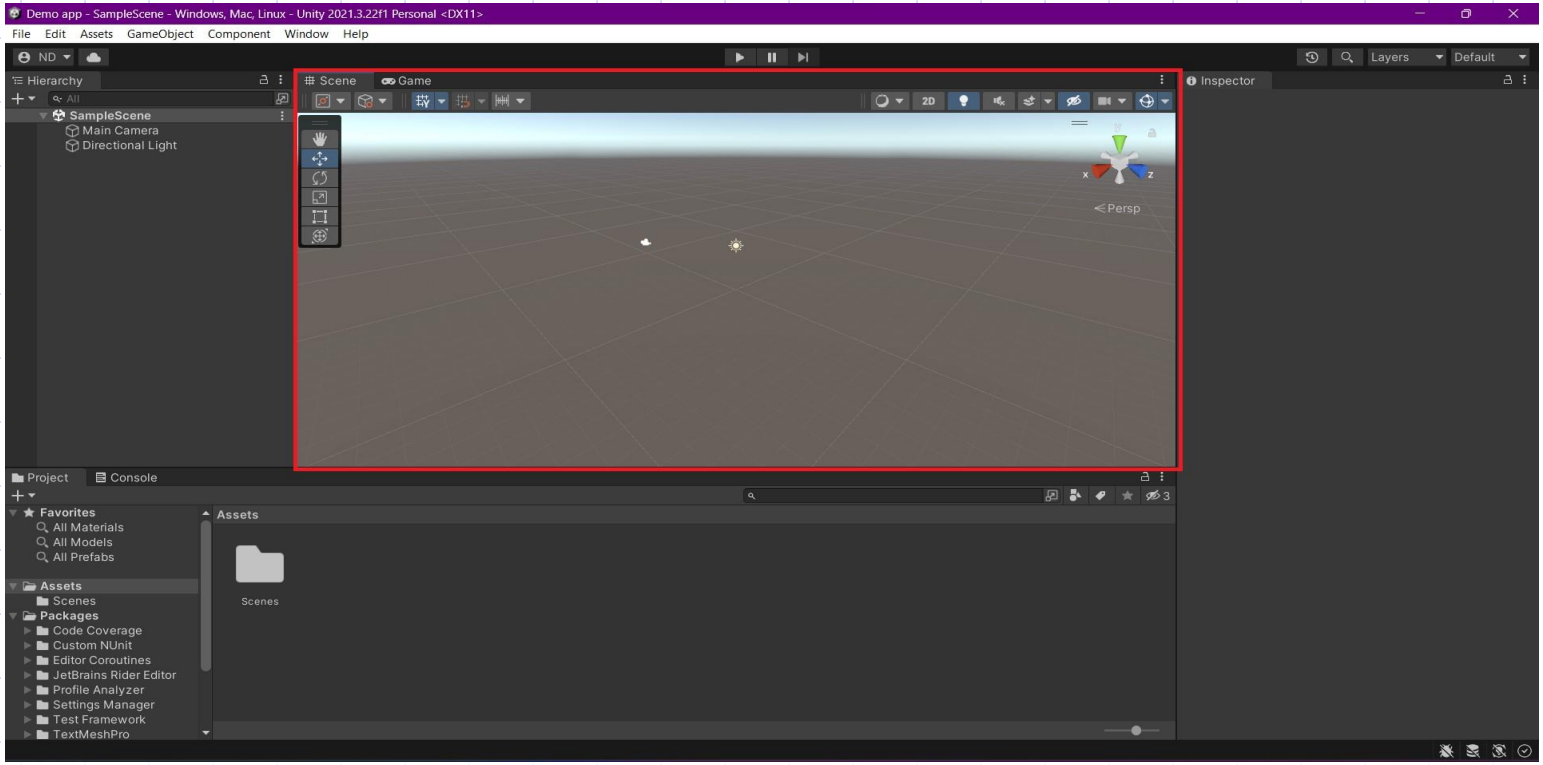
Upoznavanje sa editorom

- Kada je projekat kreiran, dobija se sledeći prikaz, u okviru kog možemo uočiti 4 regiona



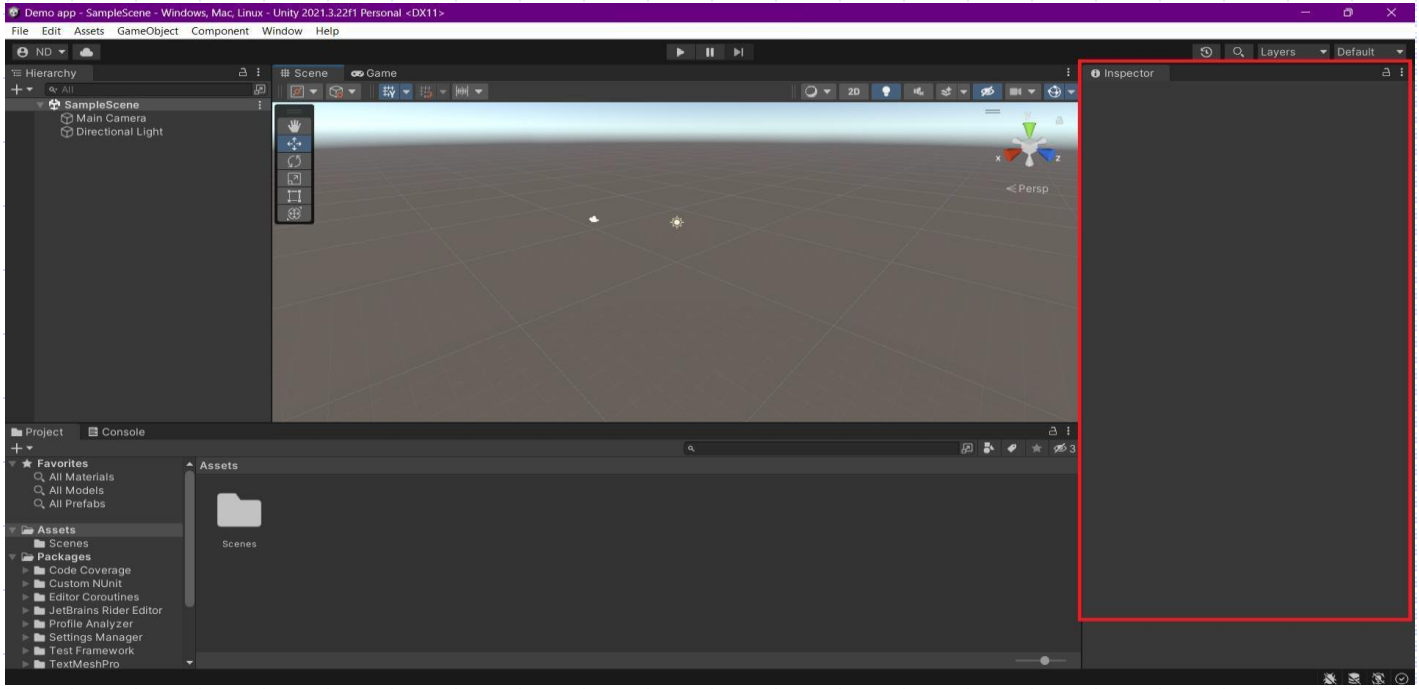
Upoznavanje sa editorom

- Uokvireni region je prozor u kome pravimo scene



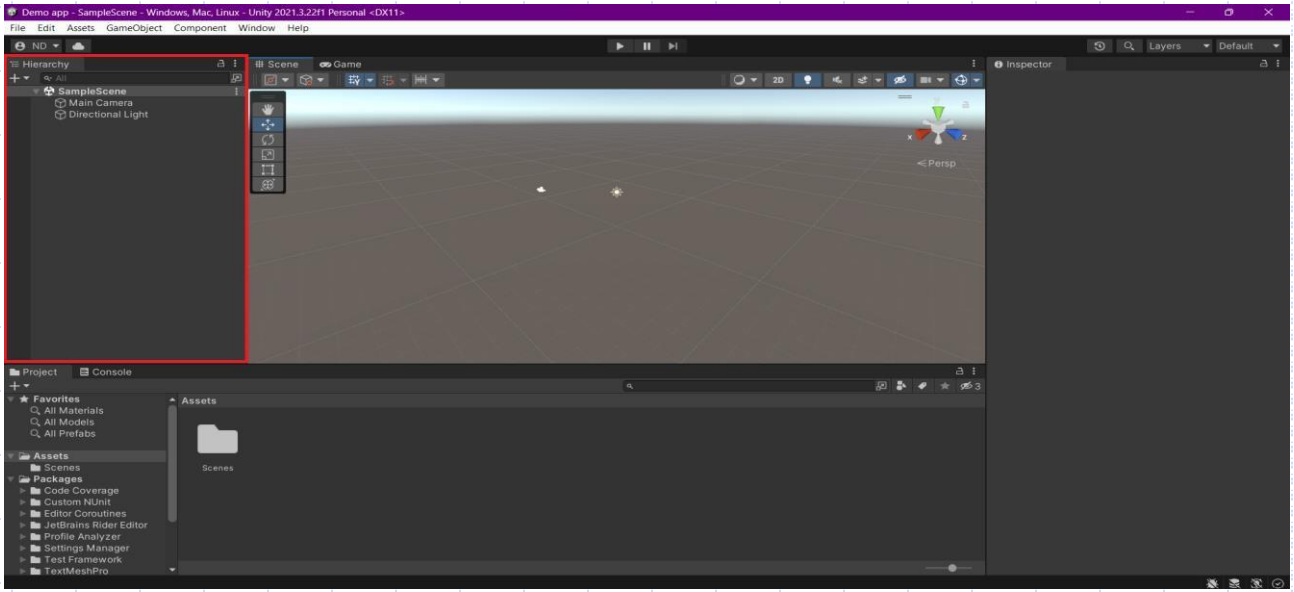
Upoznavanje sa editorom

- Uokvireni region je Inspector prozor, u okviru kog možemo videti i podešavati svojstva izabranog objekta



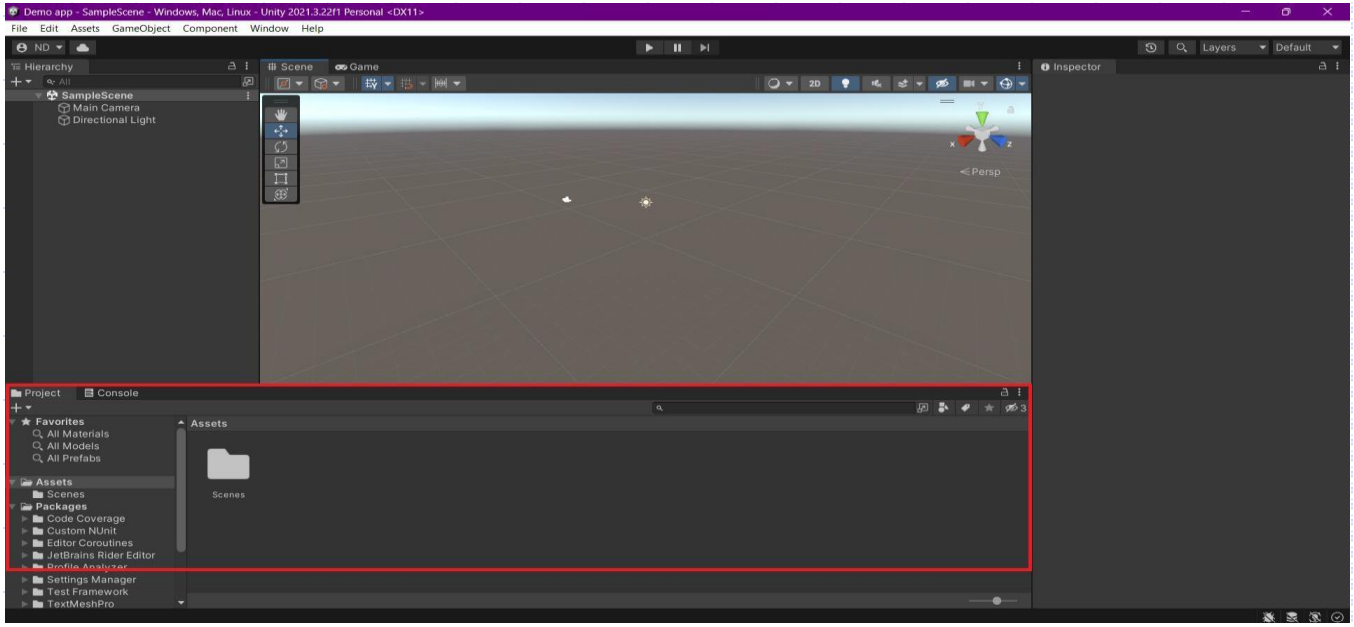
Upoznavanje sa editorom

- Uokvireni region je Scene Hierarchy prozor, u okviru kog se prikazuju svi objekti koji se nalaze u trenutno otvorenoj sceni

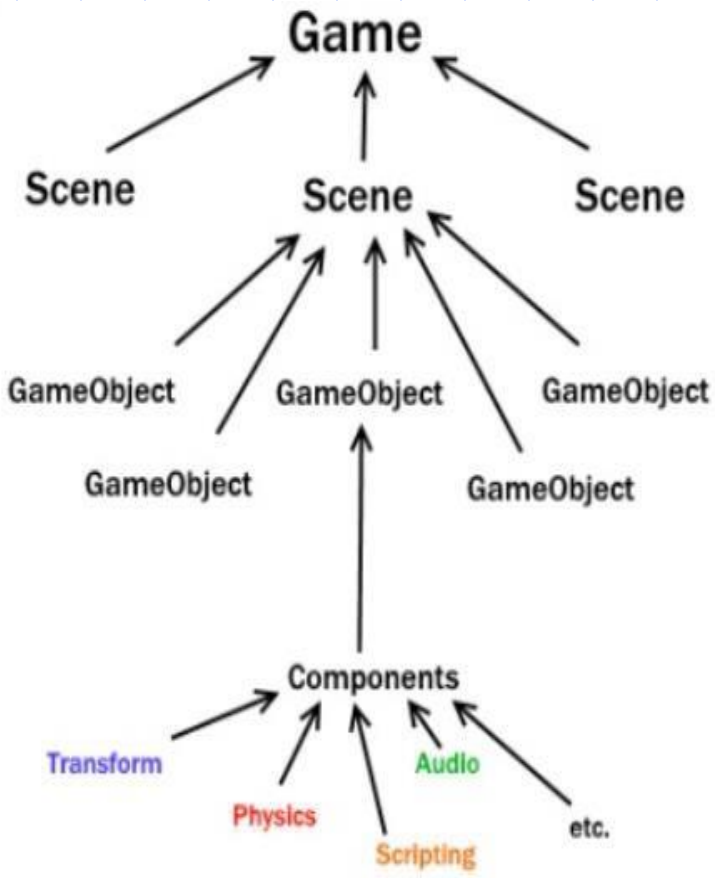


Upoznavanje sa editorom

- Uokvireni region je Project Assets prozor, u okviru kog se nalaze gradivni blokovi koje smo sami kreirali ili importovani eksterni resursi, poput tekstura, fontova, audio fajlova, animacija i slično



Hijerarhija igre



Scene

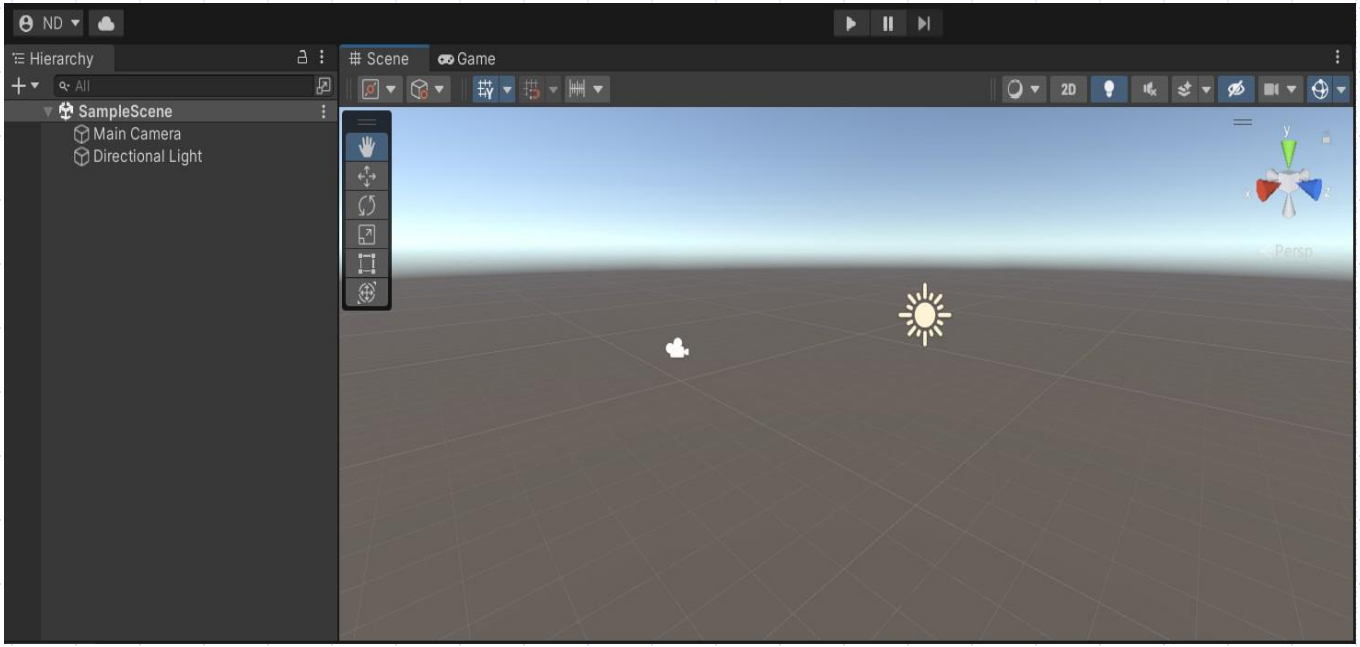
- Scene su prostor u kome se odvija igra
- U Unity-u se celokupna igra odvija u scenama
- Scene sadrže sve objekte koji su prikazani na ekranu
- Scena može biti nivo naše igre, naslovni ekran, meni i slično

Game objects

- Scena se sastoji od objekata, koji se nazivaju game objects
- Game object može biti bilo šta, od modela karaktera do elemenata GUI-a
- Zahvaljući ovom koncpetu, igra se može podeliti na delove kojima se lako upravlja, i koji se jednostavno povezuju i podešavaju

Game objects

- Inicijalno, nova 3D scena sadrži dva objekta – glavnu kameru i izvor svetlosti

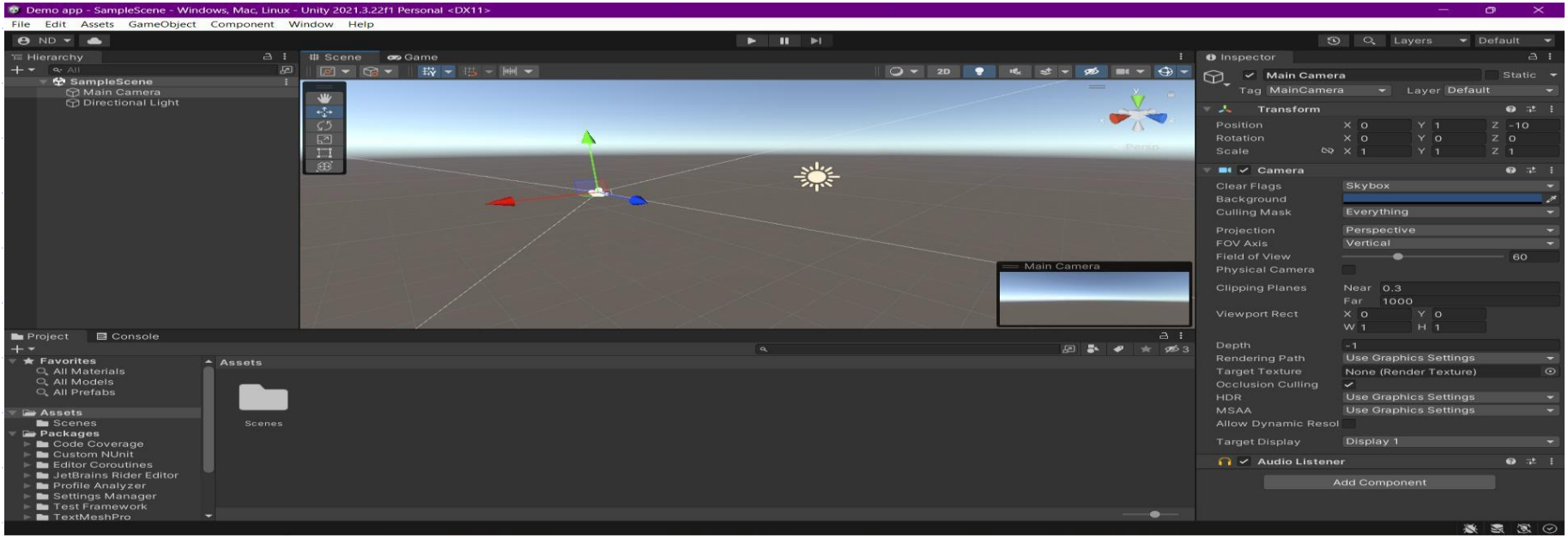


Game objects - komponente

- Game object ima pridružen skup **komponenti** koje definišu njegovo ponašanje
- Komponente sa sastoje od promenljivih koje predstavljaju podešavanja kojima kontrolišemo datu komponentu

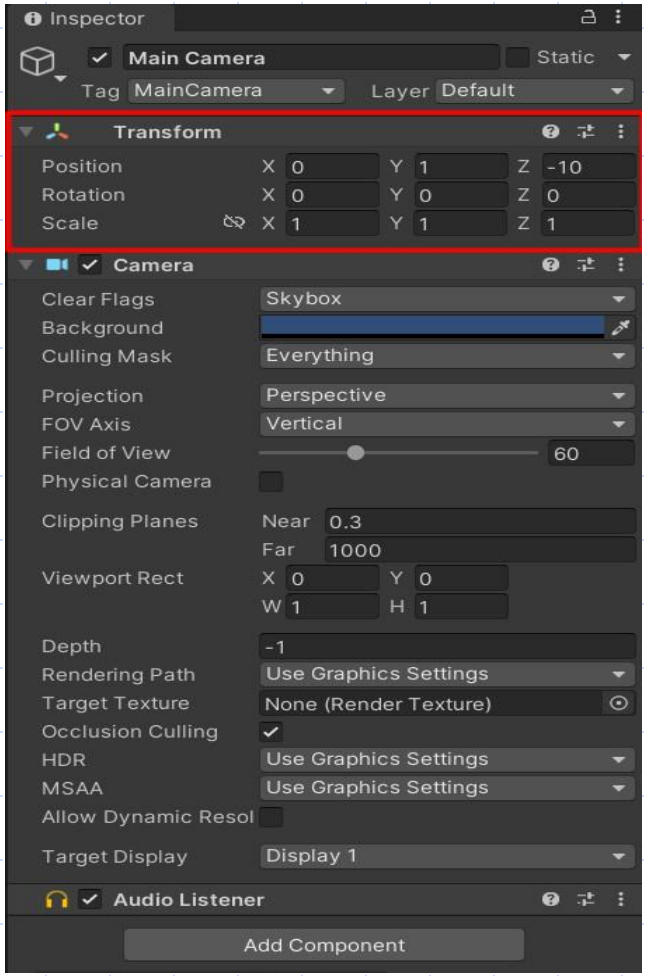
Game objects - komponente

- Klikom na objekat u sceni (ili na objekat u Scene Hierarchy prozoru) u Inspect prozoru se prikazuju komponente tog objekta
- Klikom na objekat Main camera dobijamo sledeći prikaz u Inspector prozoru



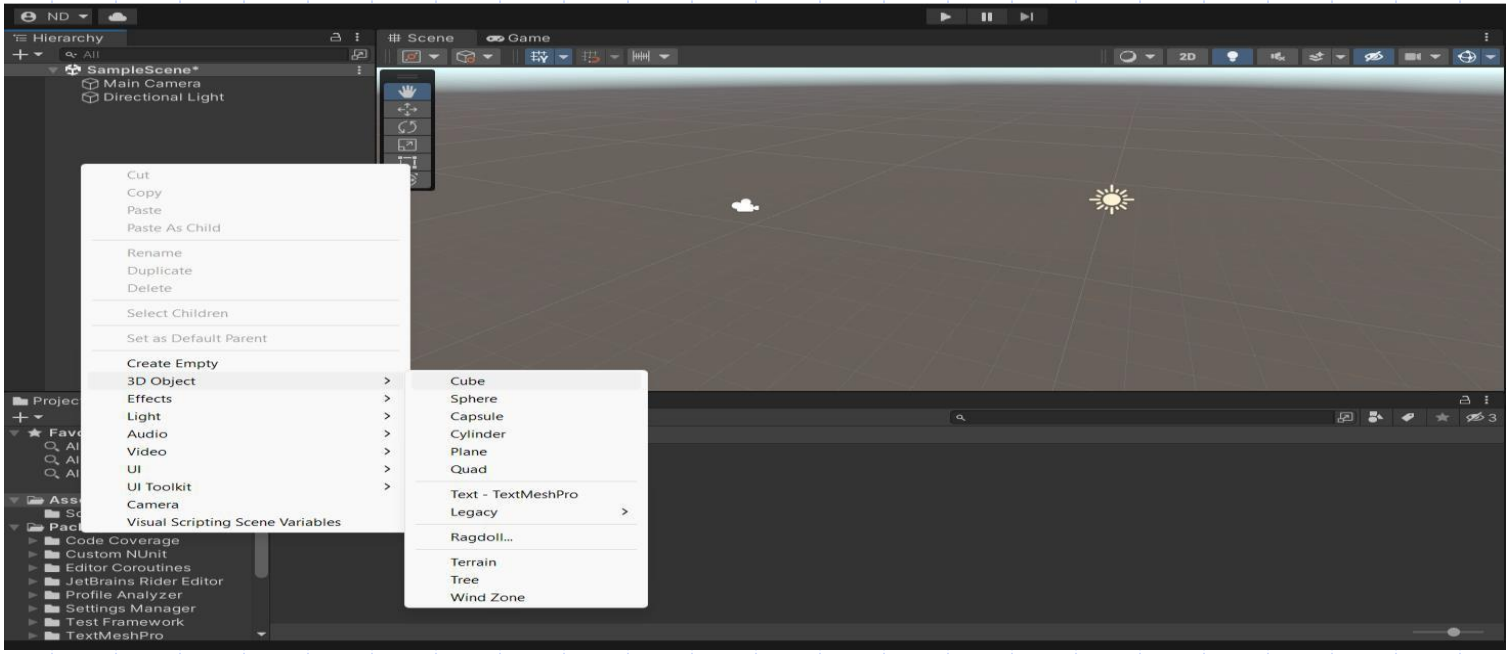
Game objects - komponente

- Jedna od najvažnijih komponenti svakog objekta je **Transform**
- **Transform** definiše poziciju, rotaciju i skaliranje objekta



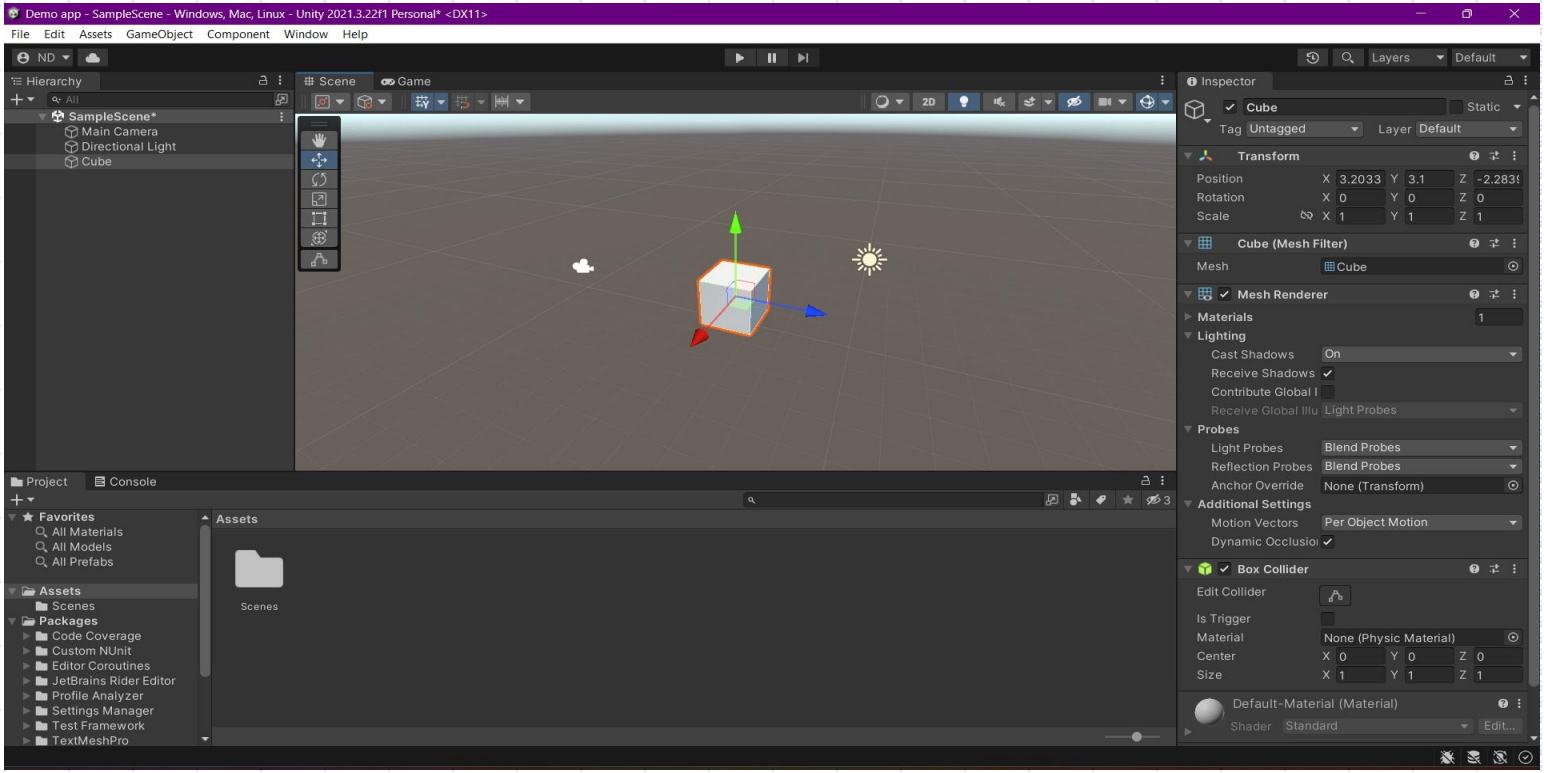
Kreiranje i dodavanja game object-a

- Desnim klikom u okviru Scene Hierarchy otvara se padajući meni
- Iz menija biramo opciju 3D object i nakon toga biramo željeni 3D objekat



Kreiranje i dodavanja game object-a

- Izborom opcije Cube, kocka je kreirana i dodata u scenu



Kreiranje i dodavanja game object-a

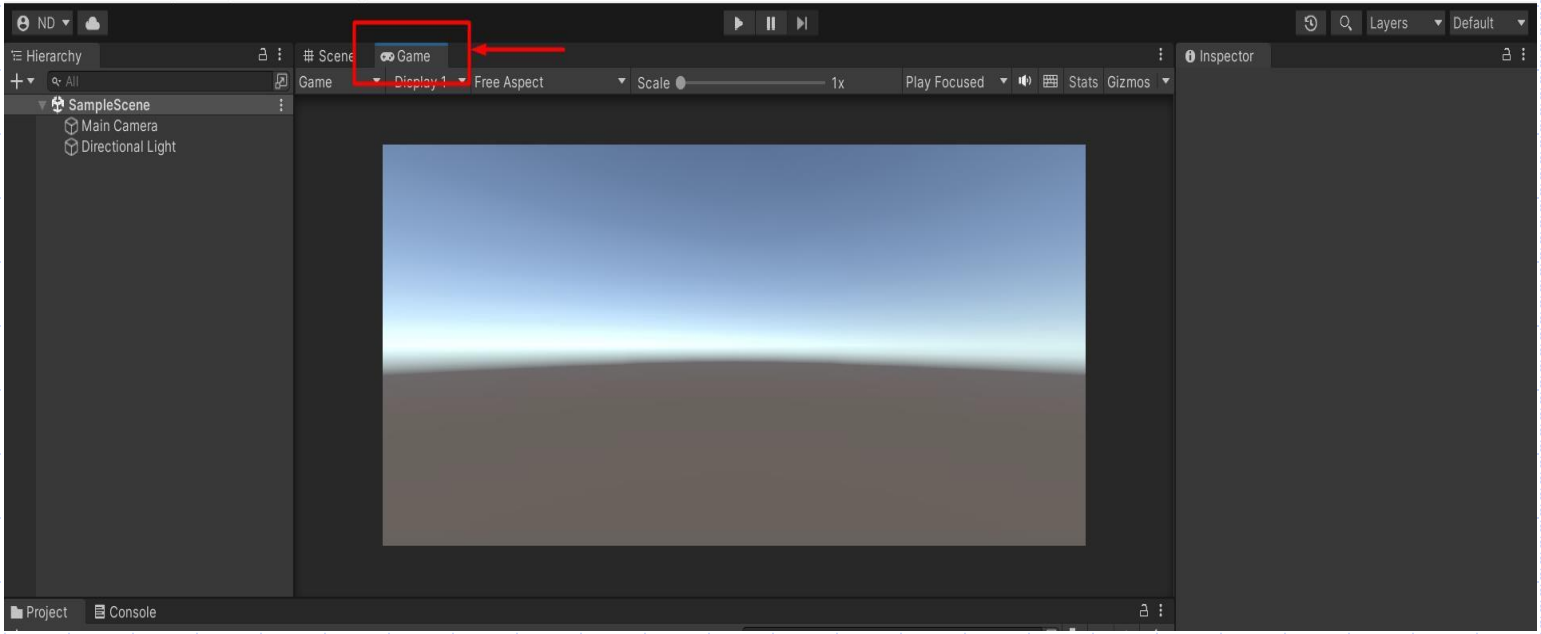
- Postojeći objekat možemo instancirati tako što ga selektujemo u Scene Hierarchy prozoru, i iskoristimo Ctrl + D prečicu sa tastature
- Postojeći objekat možemo prevući iz Scene Hierarchy prozora u Assets prozor. Nakon toga, objekat iz Assets prozora možemo instancirati prevlačenjem na scenu

Kamera

- Kamera je objekat preko kojeg korisnig vidi "svet"
- Menjanjem položaja kamere, menja se ono što korisnik vidi

Kamera

- Izborom opcije game može se videti kako scena izlgeda iz perspektive igrača



Toolbar

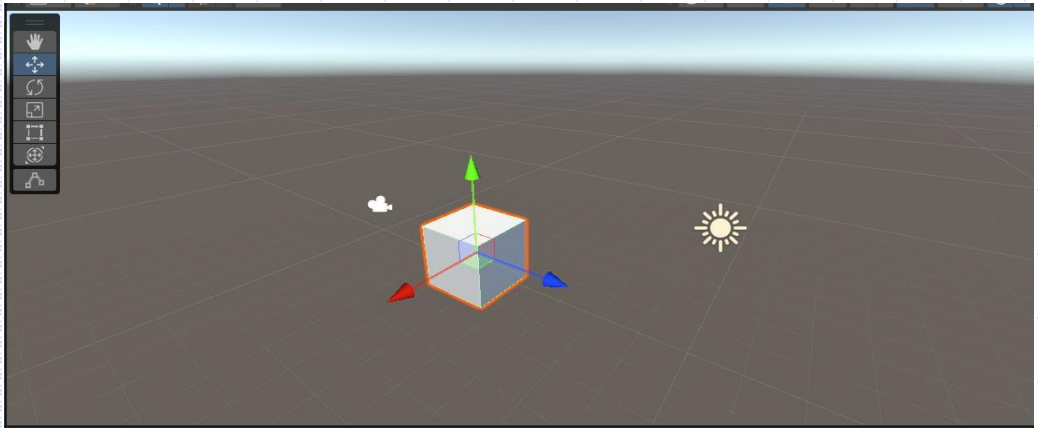
- U gornjem levom uglu Scene prozora, može se uočiti toolbar sa nekoliko opcija

Toolbar

- Prva opcija je View tool
- View tool nam omogućava da vršimo pomeranje prikaza korišćenjem miša
- Moguće je i zumiranje korišćenjem skrol-tastera ili tastature (strelice)

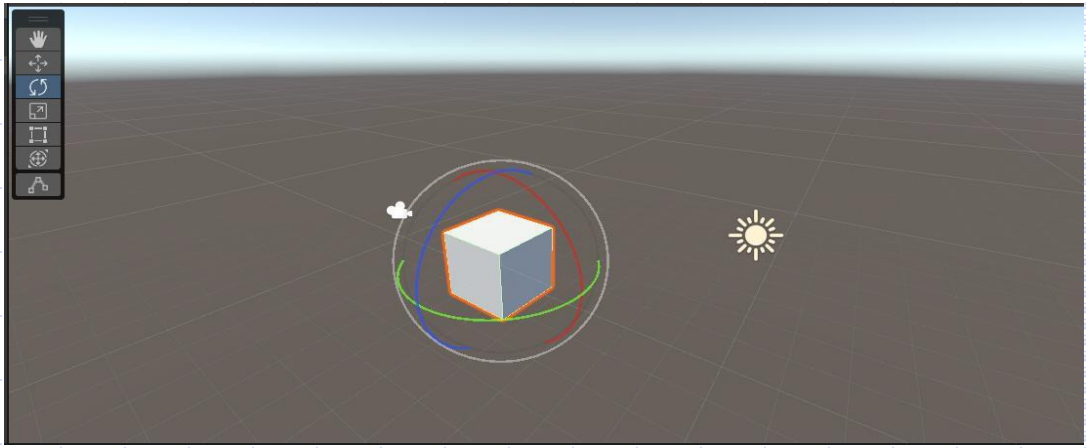
Toolbar

- Druga opcija je Move tool
- Ova opcija nam omogućava da vršimo pomeranje, odnosno translaciju selektovanog objekta duž x, y ili z ose
- Translaciju je moguće vršiti i menjanjem **Position** parametra Transform komponente



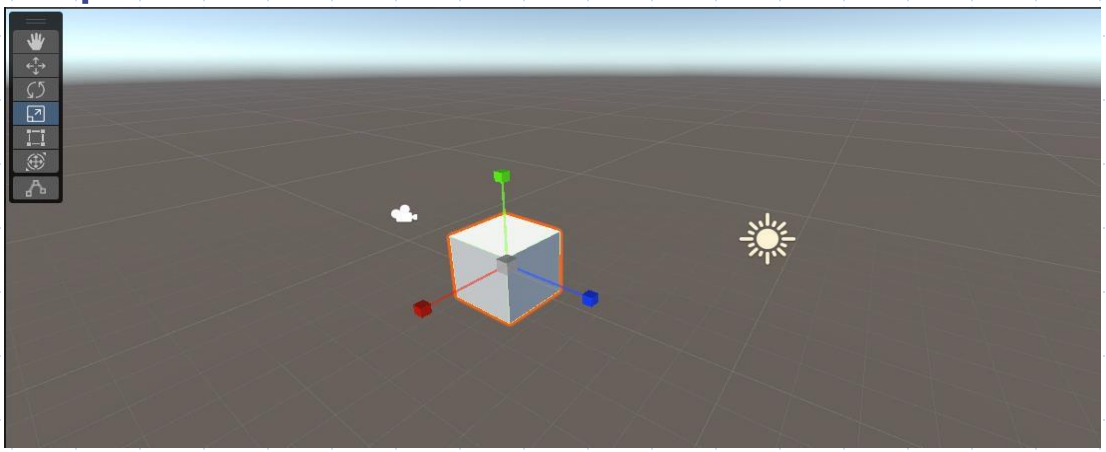
Toolbar

- Treća opcija je Rotate tool
- Ova opcija nam omogućava da vršimo rotaciju selektovanog objekta oko x, y ili z ose
- Translaciju je moguće vršiti i menjanjem **Rotation** parametra Transform komponente



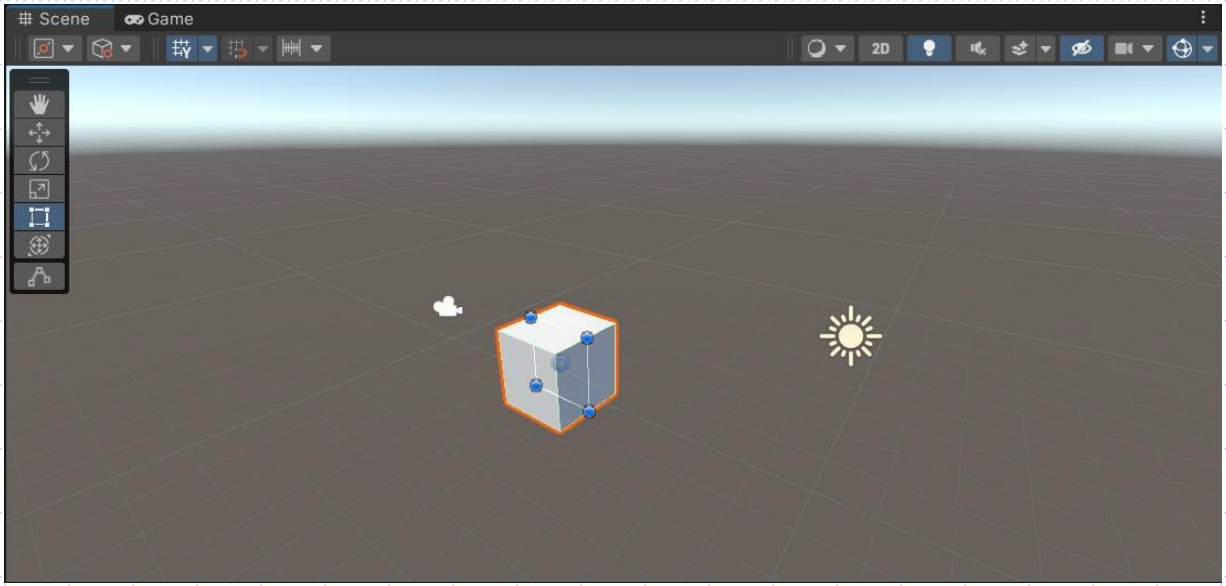
Toolbar

- Četvrta opcija je Scale tool
- Ova opcija nam omogućava da vršimo skaliranje selektovanog objekta po x, y ili z osi
- Skaliranje je moguće vršiti i menjanjem **Scale** parametra Transform komponente



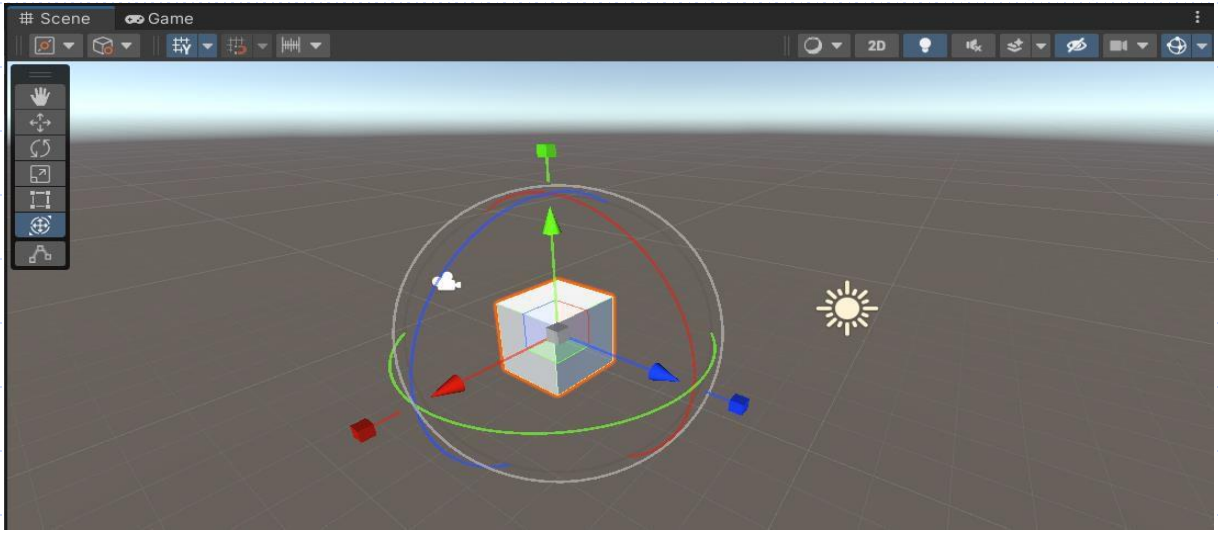
Toolbar

- Peta opcija je Rect tool
- Ova opcija predstavlja kombinaciju Move i Scale opcija



Toolbar

- Šesta opcija je Transform tool
- Ova opcija predstavlja kombinaciju Move, Rotate i Scale opcija

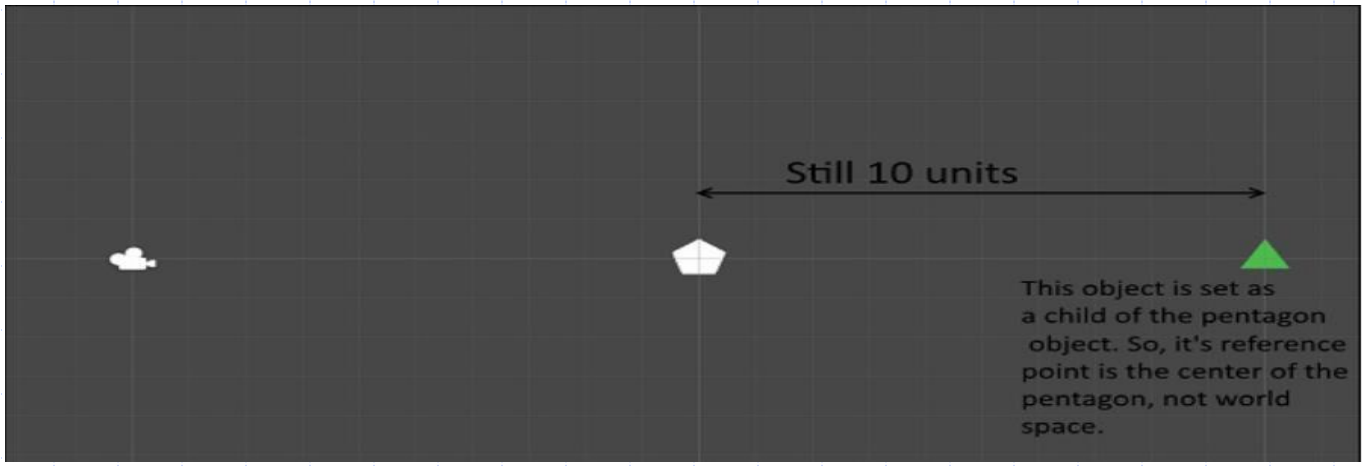


Object parenting

- Jedan game object može da bude roditelj drugog game object-a
- Ukoliko objekat ima roditelja, onda će se sve transformacije primenjivati u odnosu na roditeljski objekat

Object parenting

- Neka je pozicija kamere (0, 0) i neka kamera nema roditeljski objekat.
- Neka je pozicija petougla (10, 0) i neka petougao nema roditeljski objekat.
- Neka je trougao dete-objekat petougla,i neka je njegova pozicija (10, 0).



Assets

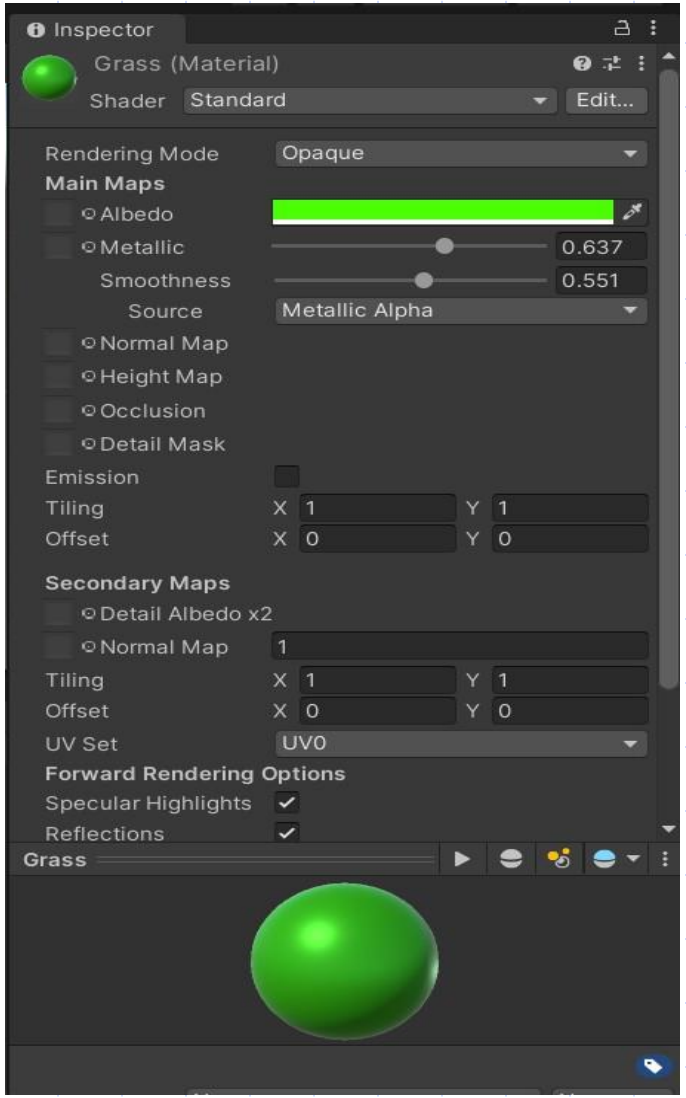
- External assets – importovani eksterni resursi poput animacija, audio fajlova, tekstura, slika, materijala itd. Mogu se dodati prevlačenjem željenog resursa u Assets prozor ili desnim klikom u isti, nakon čega iz otvorenog menija treba izabrati opciju *Import New Asset*
- Interal assets – gradivni blokovi koje smo sami kreirali. Mogu se kreirati desnim klikom u okviru Assets prozora (ili izborom opcije *Assets* iz toolbar-a) i izborom opcije *Create* iz otvorenog menija. Moguće je dodati i game object u Assets jednostavnim prevlačenjem do Assets prozora.

Materijali

- Uz pomoć materijala definišemo boju, sjaj, teskture koje se primenjuju nad objektom...
- Materijali se kreiraju desnim klikom u okviru Assets prozora (ili izborom opcije *Assets* iz toolbar-a), i izborom opcije *Create -> Material*

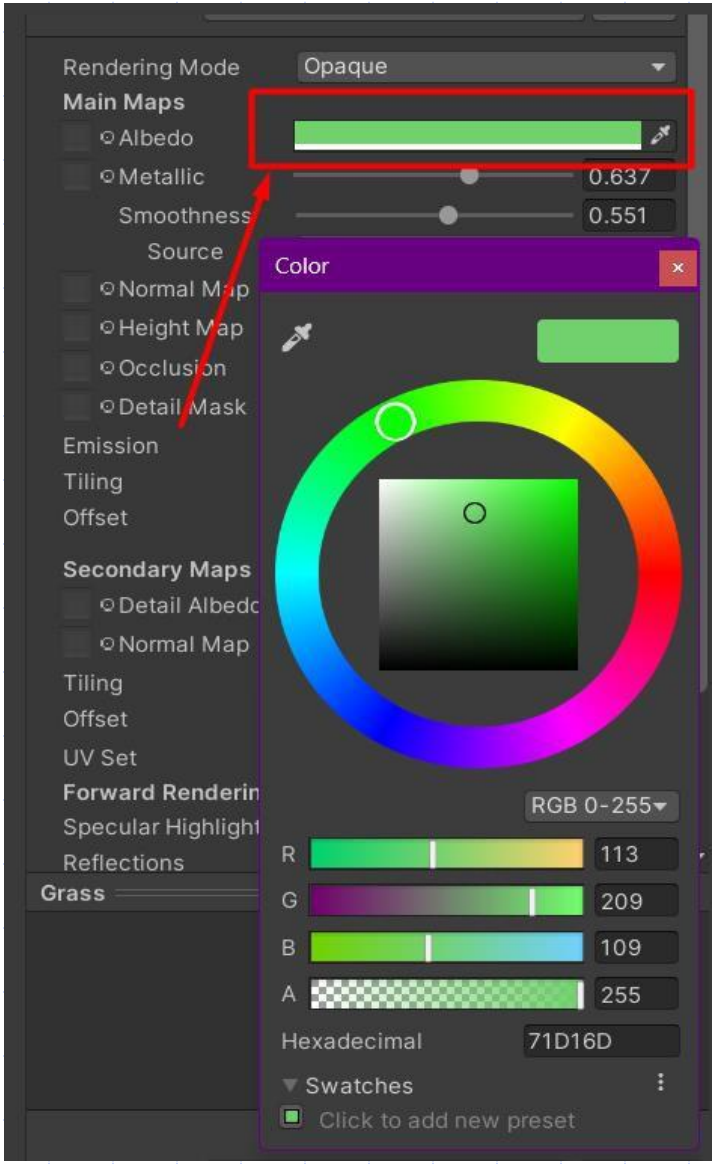
Materijali

- Nakon selekcije nekog materijala, u Inspector prozoru dobijamo prikaz kao na slici



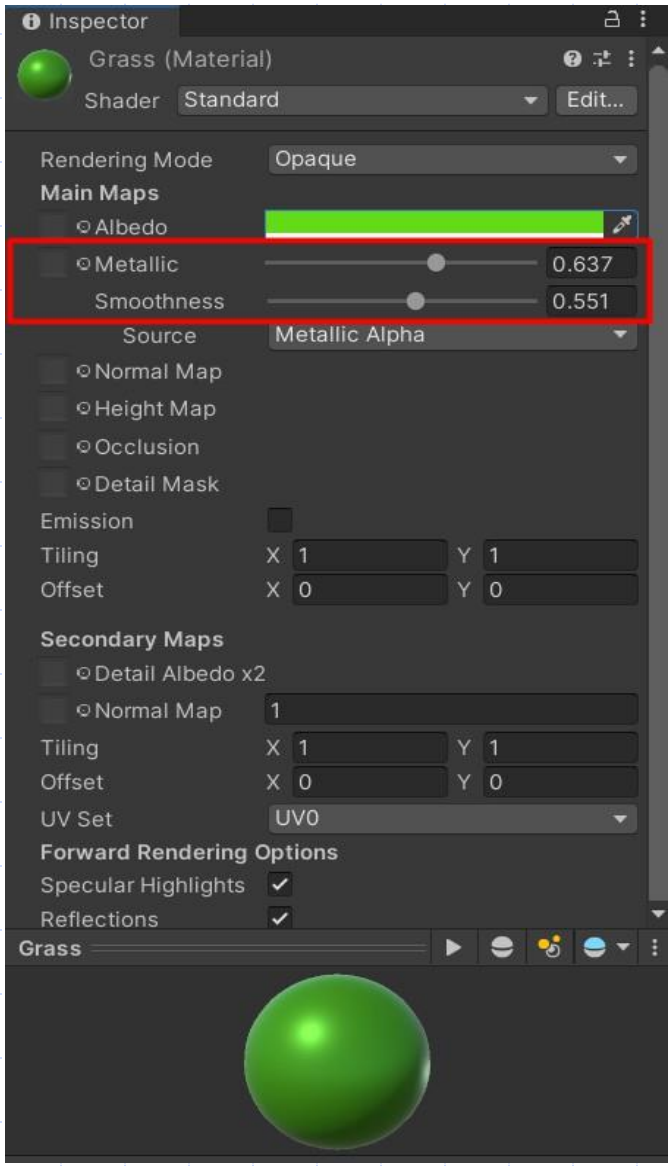
Materijali

- Klikom na pravougaonik sa bojom otvara se prozor za definisanje boje



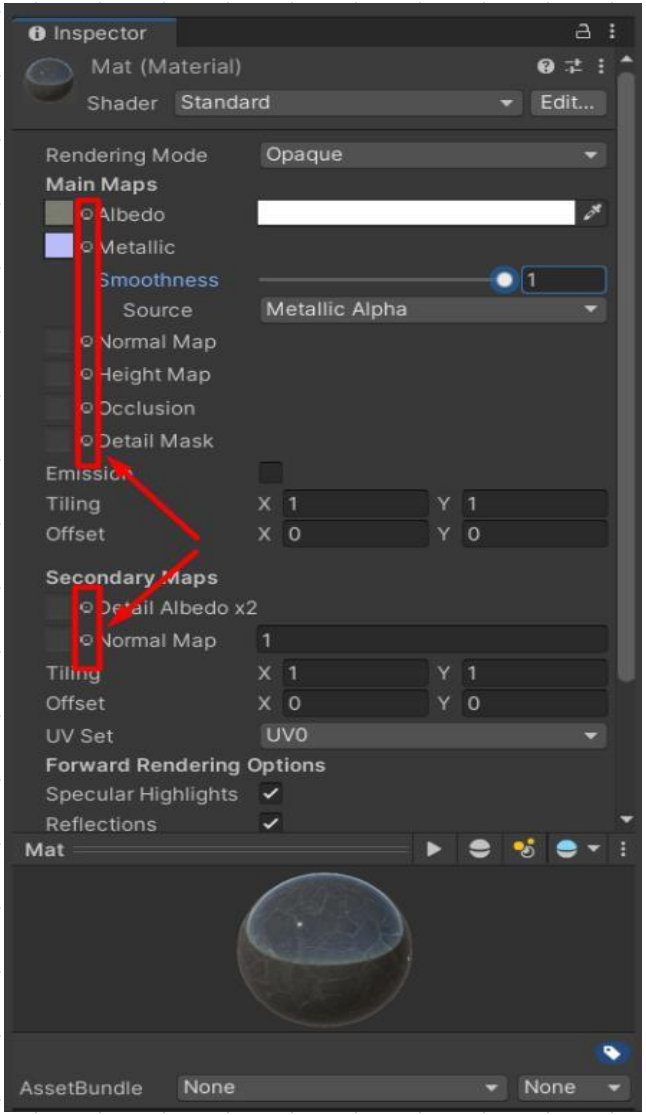
Materijali

- Ispod opcije za definisanje boje, nalaze se parametar *Metalic* koji definiše u kojoj meri materijal ima svojstva metala i paremetar *Smoothness* koji definiše koliko je površina glatka. Kombinacijom ovih parametara možemo dobiti razne efekte, od obične boje do refelektujuće površine



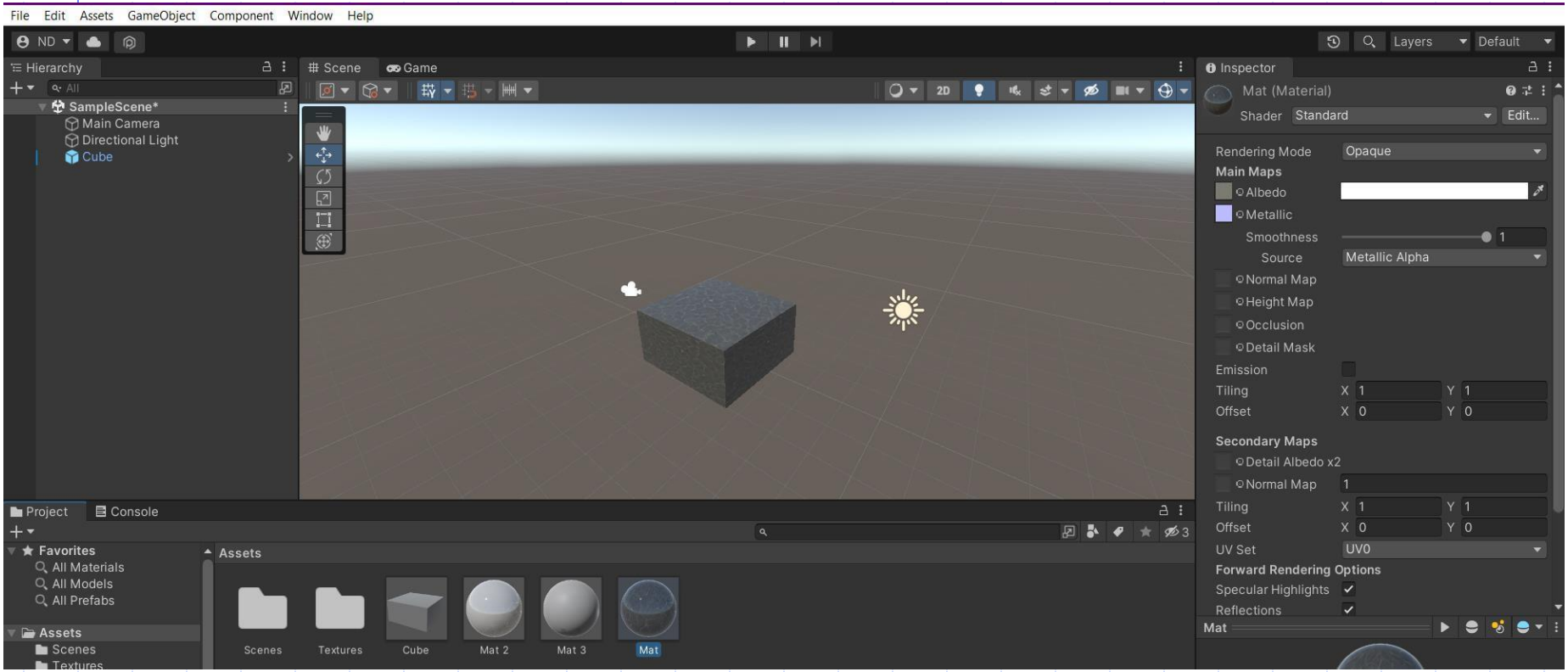
Materijali

- Teksture se koriste da bi se imitirao izgled objekata iz realnog sveta
- Teksture se mogu dodati klikom na mali krug prikazan na slici, nakon čega treba selektovati željentu teksturu



Materijali

- Materijal se primenjuje nad objektom tako što se jednostavno prevuče do istog

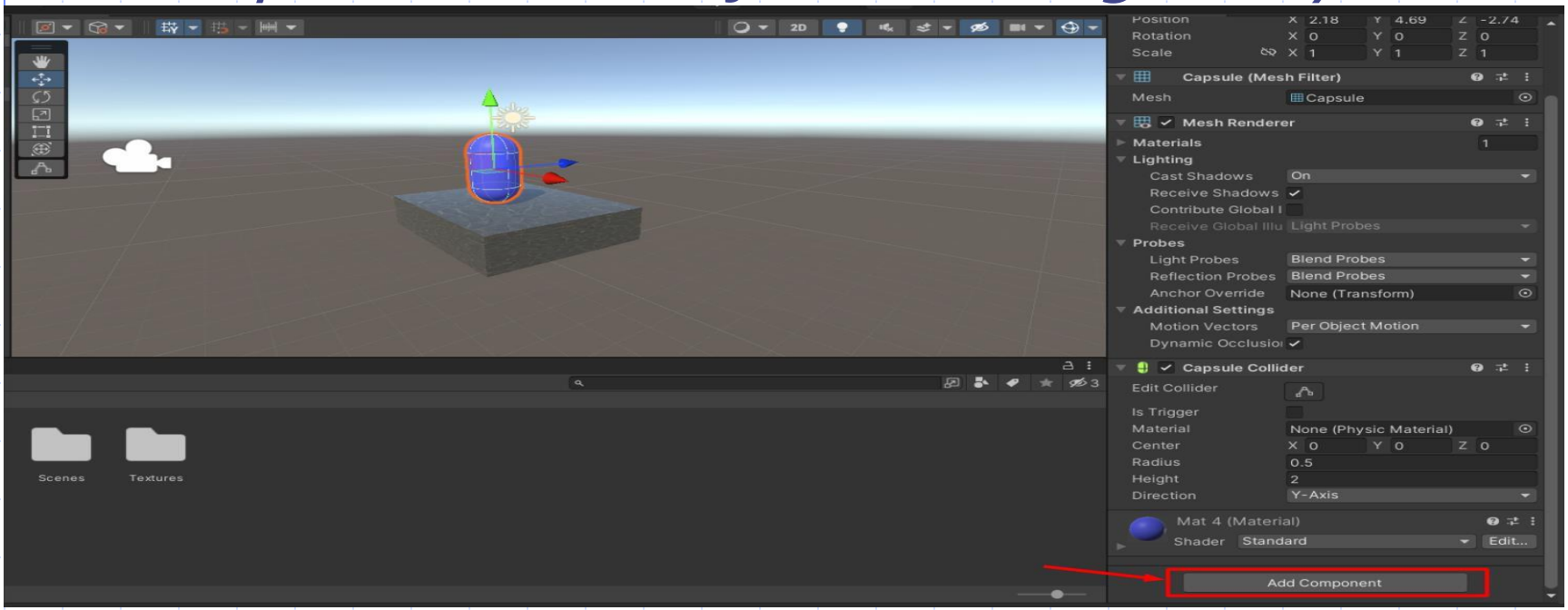


Rigidbody komponenta

- Fizika je važna jer omogućava simulaciju realnog sveta
- Inicijalno, fizika na kreirane objekte ne utiče
- Fiziku objektima uključujemo dodavanjem **Rigidbody komponente**
- Na ovaj način objektu možemo dodeliti osobine mase, gravitacije, ubrzanja, trenja...

Rigidbody komponenta

- Da bismo dodali Rigidbody komponentu objektu, potrebno je u Inspector prozoru selektovanog objekta izabrati opciju *Add component* i iz menija izabrati *Rigidbody*



Kamera koja prati igrača

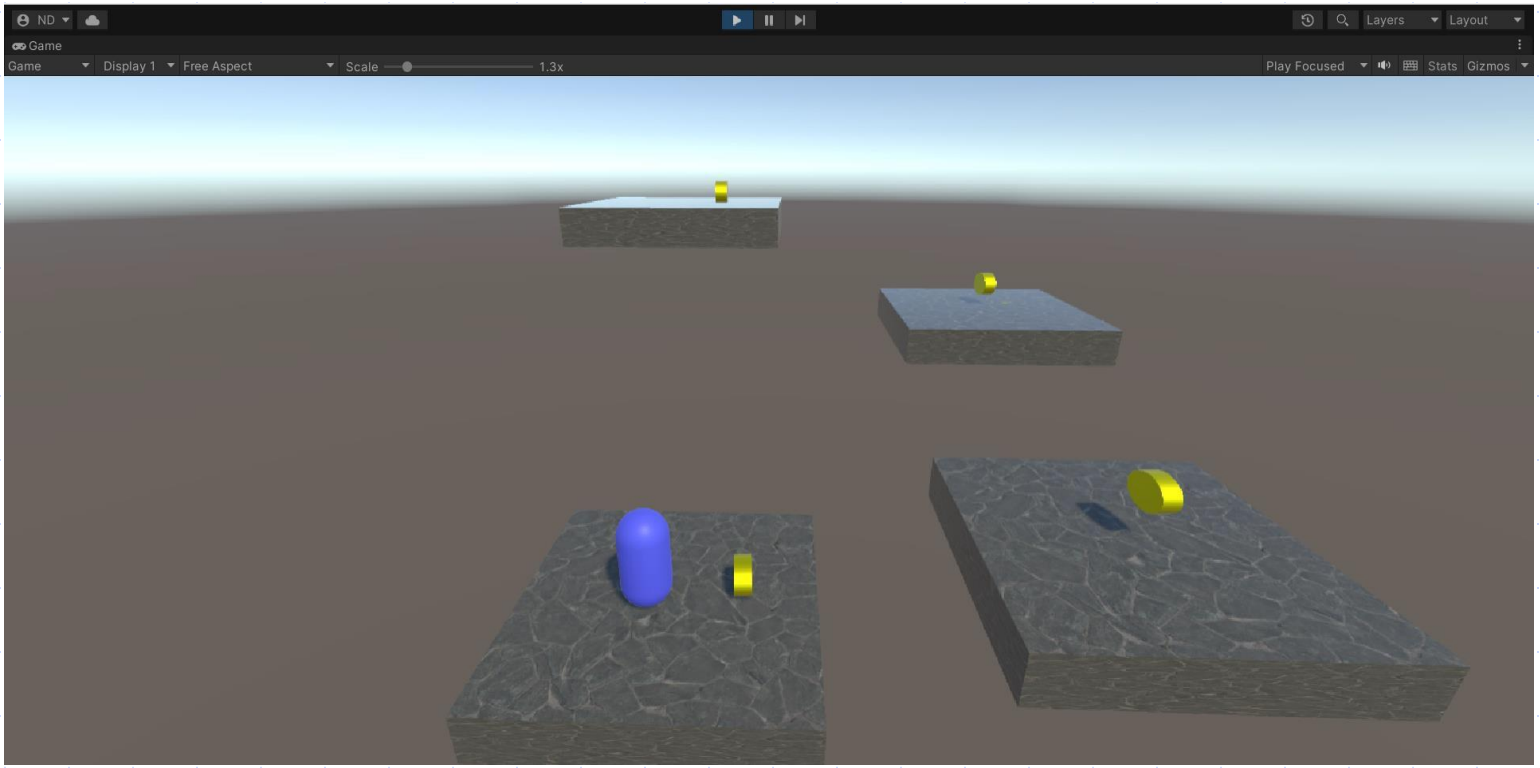
- Ukoliko želimo da kamera prati igrača, možemo objekat kamere prevući do objekta igrača. Na ovaj način, objekat igrača postaje roditeljski element kamere, te će kamera pratiti igrača

Zadatak za četvrtu laboratorijsku vežbu

- Potrebno je napraviti jednostavnu "igru" u okviru koje se igrač kreće i skače po platformama (može da se kreće duž x i z ose) i može da skače duž y ose
- Na platformama je potrebno postaviti objekte koje će igrač da sakuplja
- Platforme i objekte koje igrač sakuplja stilizovati materijalima (željenim teksturama koje se mogu naći na internetu)

Zadatak za četvrtu laboratorijsku vežbu

- Na slici je prikazan primer izgleda scene



Kretanje

- Za potrebe kretanja i skakanja moguće je kreirati C# skriptu
- Skripta se kreira tako što se nakon desnog klika u okviru Assest prozora (ili izborom *Assets* opcije iz toolbar-a) izabere opcija Cretate -> C# script
- Dvostrukim klikom na skriptu se skripta otvara u podrazumevanom editoru

Kretanje

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player : MonoBehaviour
{
    private float horizontalInput;
    private float verticalInput;
    private bool isGrounded;
    private bool spacePressed;
    private Rigidbody rigidbodyComponent;

    // Start is called before the first frame update
    void Start()
    {
        rigidbodyComponent = GetComponent<Rigidbody>();
    }
    // Update is called once per frame
    void Update()
    {
        if(Input.GetKeyDown(KeyCode.Space))
        {
            spacePressed = true;
        }
        horizontalInput = Input.GetAxis("Horizontal");
        verticalInput = Input.GetAxis("Vertical");
    }
```

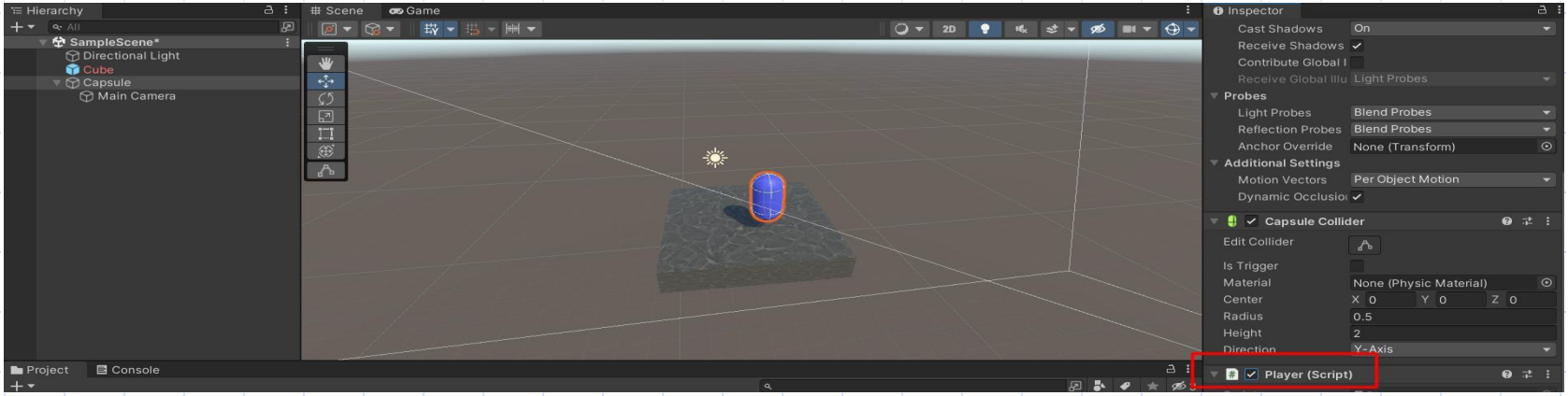
Kretanje

```
    private void FixedUpdate()
    {
        if(horizontalInput != 0f)
            rigidbodyComponent.velocity = new Vector3(horizontalInput * 2, rigidbodyComponent.velocity.y,
     rigidbodyComponent.velocity.z);
        if(verticalInput != 0f)
            rigidbodyComponent.velocity = new Vector3(rigidbodyComponent.velocity.x,
     rigidbodyComponent.velocity.y, verticalInput * 2);

        if(!isGrounded)
        {
            return;
        }
        if(spacePressed)
        {
            rigidbodyComponent.AddForce(Vector3.up * 10, ForceMode.VelocityChange);
            spacePressed = false;
        }
    }
    private void OnCollisionEnter(Collision collision)
    {
        isGrounded = true;
    }
    private void OnCollisionExit(Collision collision)
    {
        isGrounded = false;
    }
}
```

Dodavanje skripte objektu

- Skriptu pridružujemo objektu jednostavnim prevlačenjem do objekta u sceni (ili do objekta u Hierarchy prozoru)
- Ukoliko je skripta uspešno dodata objektu, biće prikazana u okviru Inspector prozora selektovanog objekta

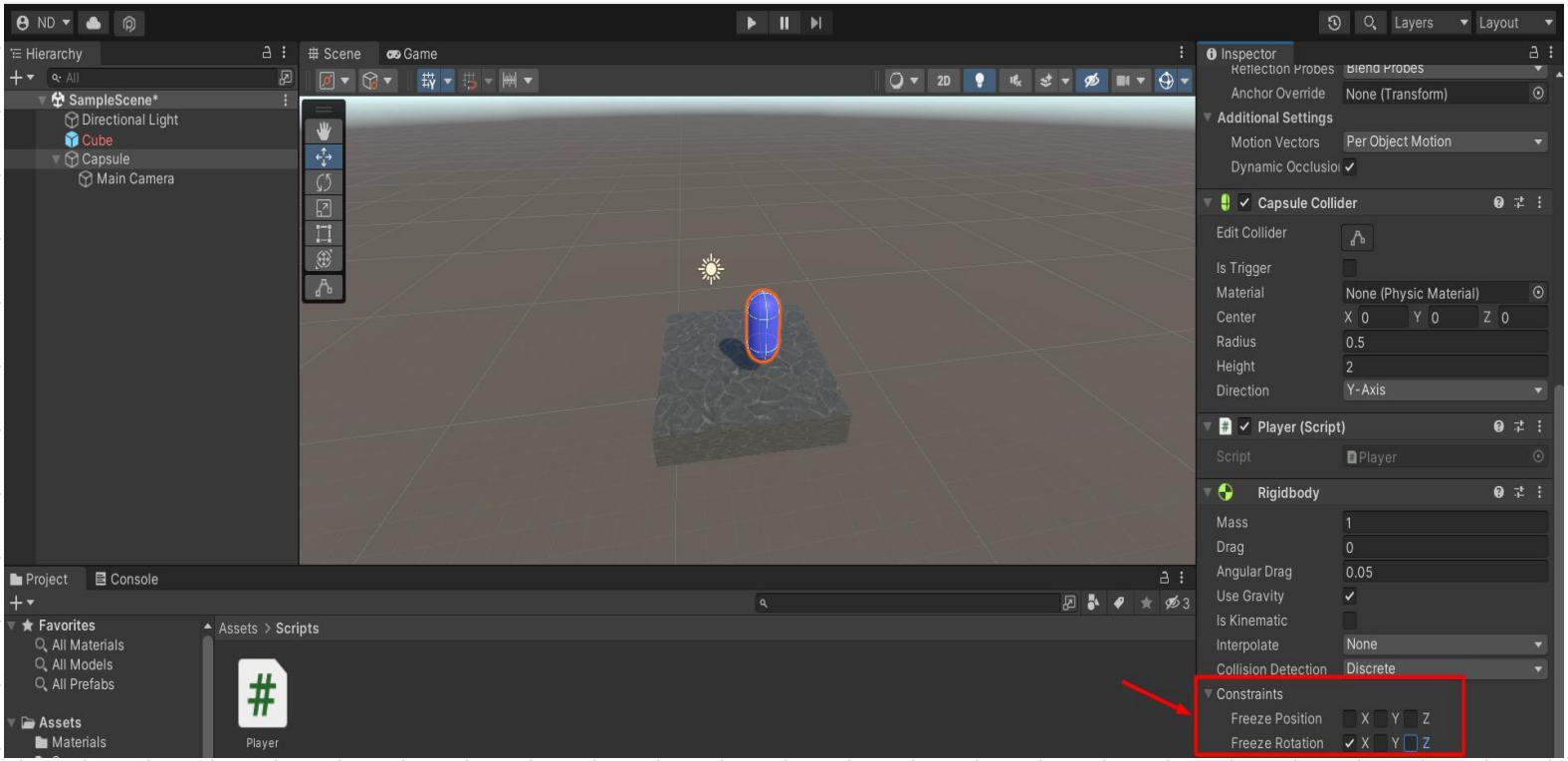


Constraints parametri Rigidbody-a

- U okviru Rigidbody komponente postoje Constraints parametri, pomoću kojih je moguće sprečiti pomeranje ili rotiranje objekta oko željenje ose
- Na primer, ukoliko je naš objekat kapsula koja se kreće po x osi, da bi kretanje bilo očekivano, bez rotiranja, potrebno je sprečiti rotaciju oko x ose

Constraints parametri Rigidbody-a

- Rotacija oko x ose se sprečava čekiranjem odgovarajućeg checkbox-a

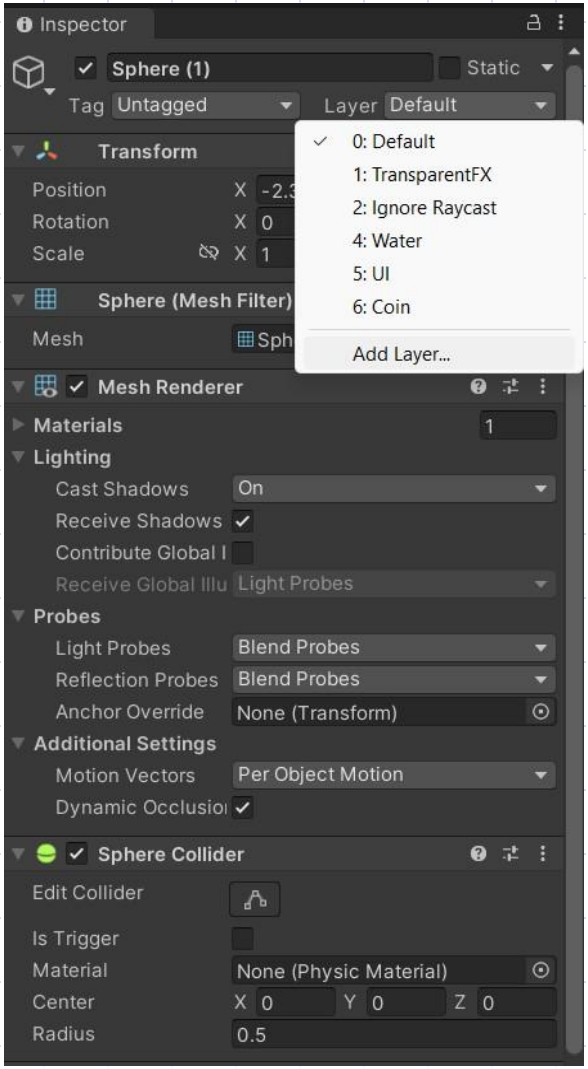


Susret dva objekta

- Ukoliko želimo da kada dođe do susreta našeg objekta koji predstavlja player-a, sa nekim drugim objektom, taj drugi objekat nestane, to možemo postići dodvanjem određene metode (biće data u nastavku) u skriptu player-a
- Pored skripte koju dodajemo, treba izvršiti i određena podešavanja za objekat koji želimo da nestane

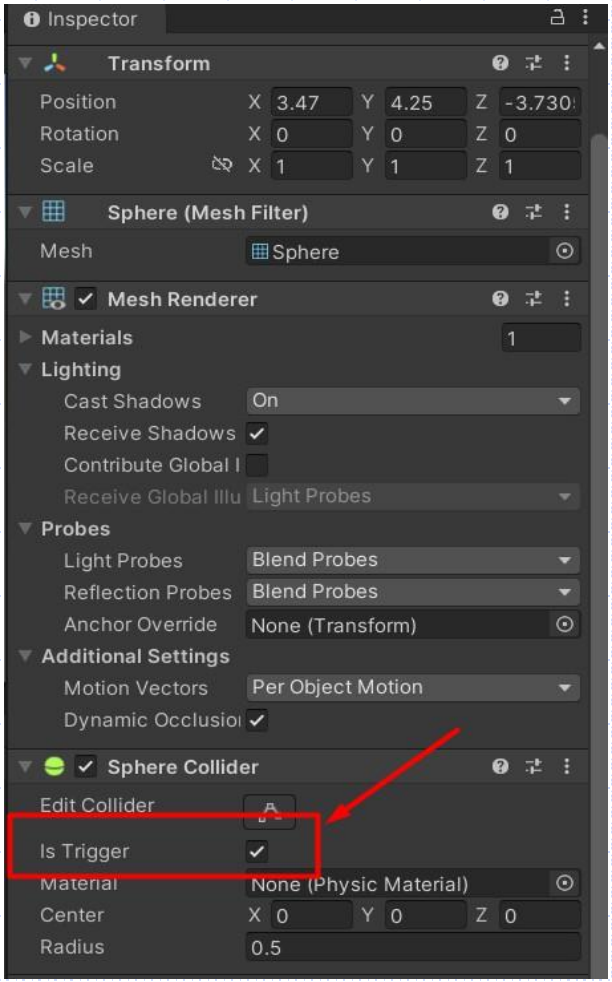
Susret dva objekta

- Postaviti objekat koji nestaje na nivo koji odgovara broju iz skripte (u ovom slučaju je to broj 6)
  - Potrebno je u Inspector prozoru selektovanog objekta dodati novi nivo (u ovom slučaju je to nivo 6) i nazvati ga proizovljno
  - Nakon što je nivo dodat, potrebno je postaviti objekat na taj nivo, izborom istog u okviru *Layer* opcije



Susret dva objekta

- Čekirati parametar
  Is Trigger u okviru Collider komponente objekta koji nestaje



Susret dva objekta

- Metoda koju je potrebno dodati u skriptu player objekta

```
private void OnTriggerEnter(Collider other)
{
        if(other.gameObject.layer == 6)
        {
            Destroy(other.gameObject);
        }
}
```

Susret dva objekta

- Metoda koju je potrebno dodati u skriptu player objekta

```
private void OnTriggerEnter(Collider other)
{
        if(other.gameObject.layer == 6)
        {
            Destroy(other.gameObject);
        }
}
```

- U if uslovu proveramo da li je nivo game object-a sa kojim se naš igrač susreo jednak 6 jer je 6 nivo na koji smo postavili taj game object

Šta je potrebno uraditi u lab. vežbi

- Kreirati objekat koji predstavlja igrača
- Stilizovati igrača korišćenjem materijala
- Kreirati objekte koji predstavljaju platforme
- Stilizovati platfrome korišćenjem materijala (i tekstura)
- Kreirati objekte koje predstavljaju novčiće
- Stilizovati novčiće korišćenjem materijala (i tekstura)
- Omogućiti kretanje igrača duz x i z ose
- Omogućiti skakanje igrača
- Obezbediti da igrač može da pokupi novčić (kada dođe do susreta igrača i novčića, novčić nestaje)