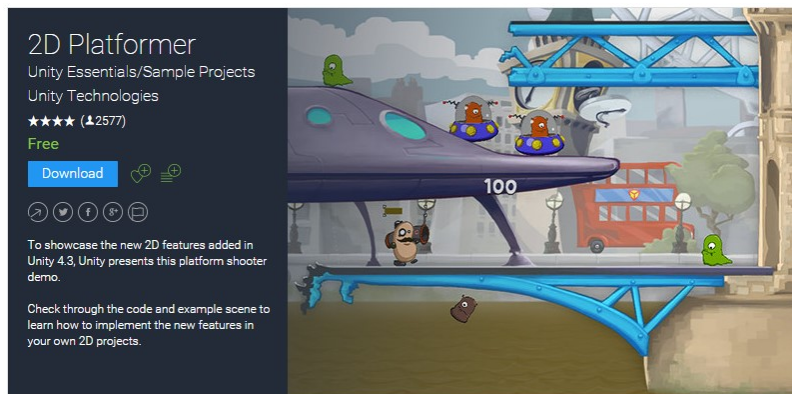


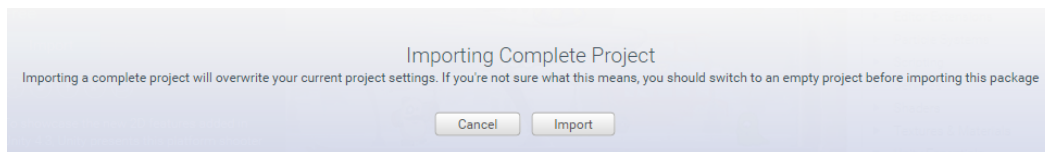
2D platformer

Uputa je temeljena na video vodiču za stvaranje osnovnog 2D platformera koji se nalazi ovdje: <https://unity3d.com/learn/tutorials/topics/2d-game-creation/creating-basic-platformer-game>

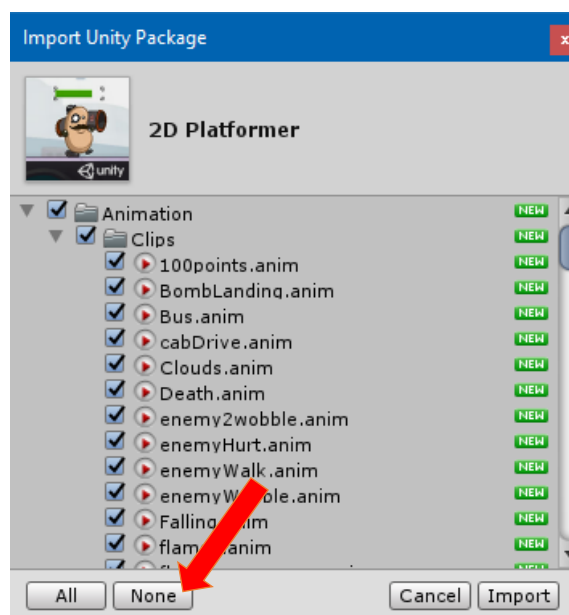
1. Stvoriti novi 2D projekt, naziva „2D Platformer“
2. Iz Asset Store-a dohvatiti resurse za „2D Platformer“
 - 2.1. Izbornik Window → Asset Store, podražiti „2D Platformer“ te ga uvesti u svoj projekt pritiskom na tipku „Download“



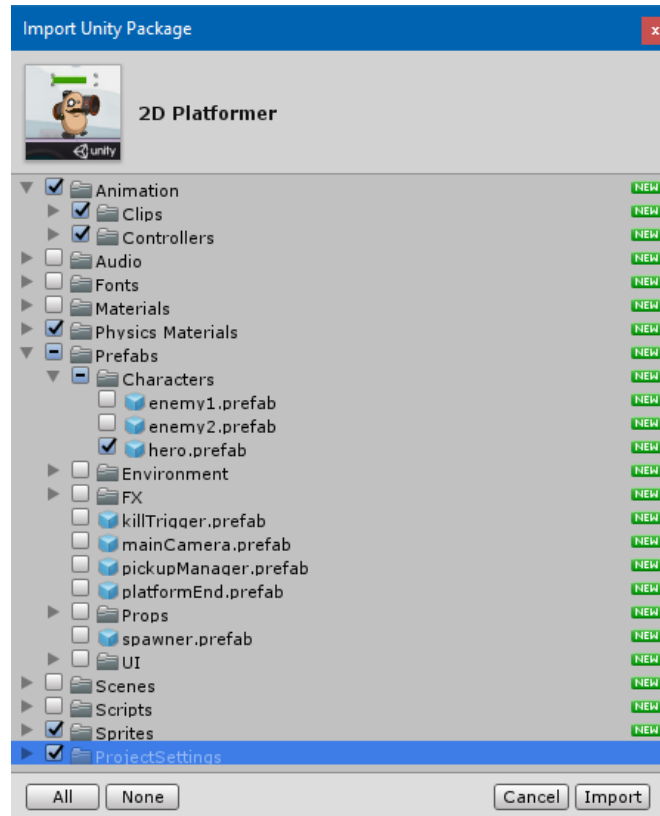
- 2.2. Potvrditi uvođenje u projekt, pritisnuti tipku **Import**.



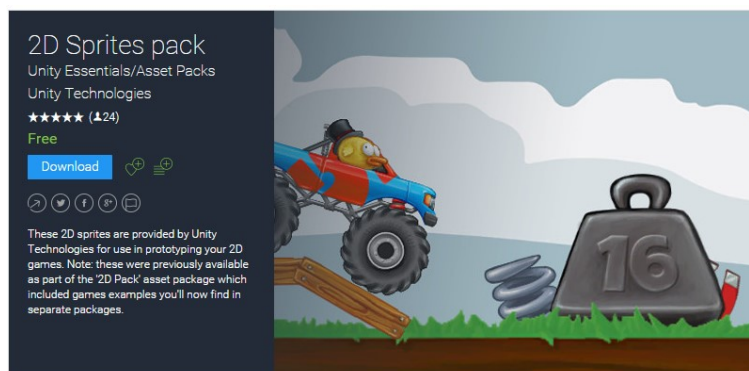
- 2.3. U prozoru Import Unity Package odznačiti sve komponente pritiskom na tipku **None**.



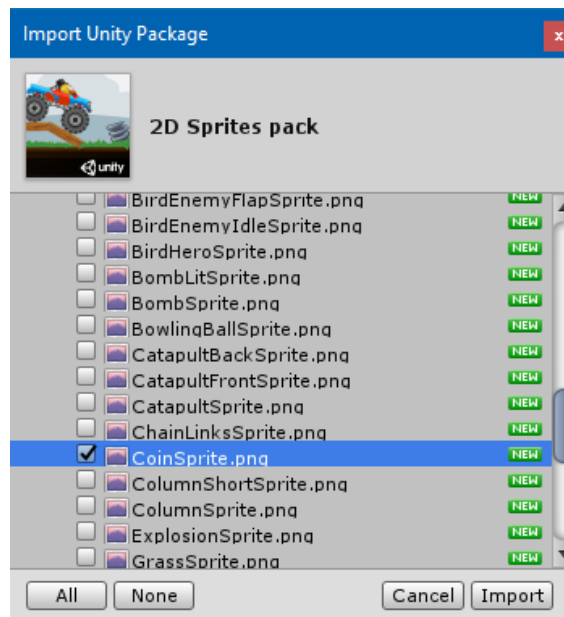
- 2.4. Odabrati pod Animation oboje Clips i Controllers, odabrati Physics Materials, iz Prefabs → Characters odabrati hero.prefab, odabrati Sprites i Project Settings kako pokazuje slika niže, te pritisnuti tipku **Import**.



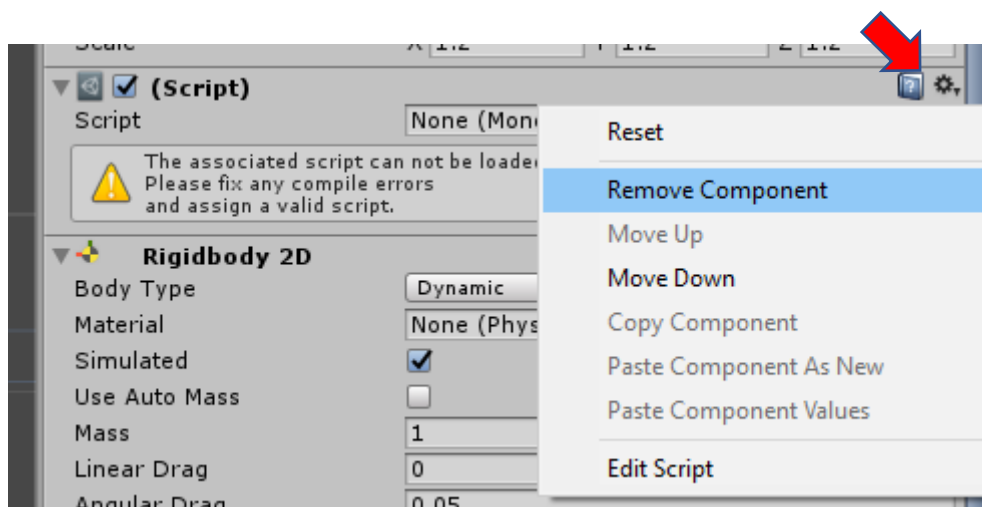
3. Ponovno se vratiti u Asset Store prozor i potražiti „2D Pack“. Ov ovog paketa jedino što je potrebno je CoinSprite.png. Potrebno je pronaći „2D Pack“ i preuzeti ga tipkom **Download**.



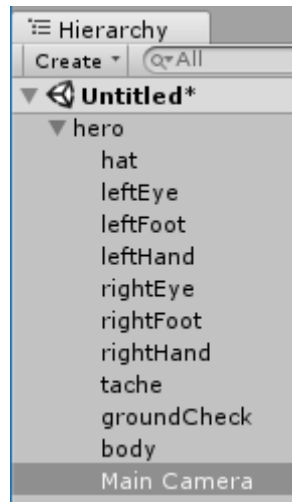
- 3.1. U prozoru Import Unity Package pritisnuti tipku **None**, pronaći i odabrati samo „CoinSprite.png“ te tada pritisnuti tipku **Import**. Sada se u projektu nalazi sve što je potrebno, svi potrebni resursi tj. „assets“ se nalaze u projektu.



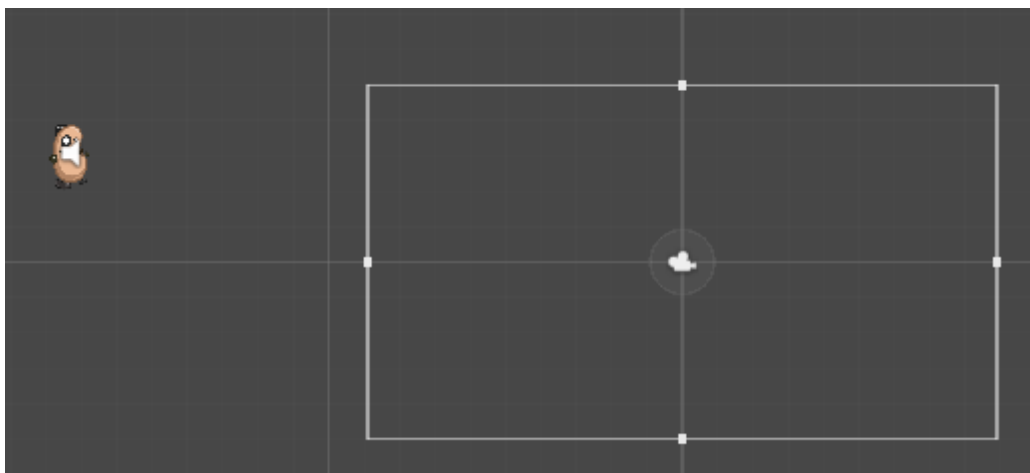
4. Iz Assets → Prefab → Characters odvući „hero“ u Hierarchy prozor te se pozicionirati na njega duplim pritiskom na tipku „hero“.
- 4.1. U prozoru Inspector vide se upozorenja da neke skripte nisu dostupne (nisu uvezene u projekt), te je potrebno te stavke ukloniti korištenjem opcije **Remove Component**.



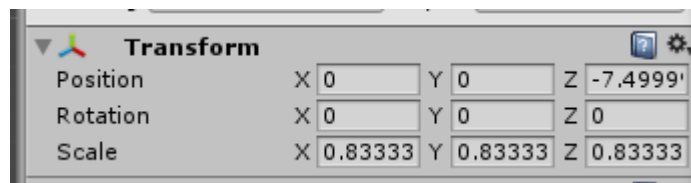
- 4.2. Skripte koje nisu učitane je potrebno ukloniti na tri mjesta. Proučiti prozor Inspector te na svako mjestu ukloniti skriptu gdje je upozorenje.
- 4.3. U projekt nisu ugrađeni zvukovi, te je u prozoru Inspector potrebno ukloniti i „Audio Source“.
- 4.4. U prozoru Hierarchy odabrati Hero, otvoriti stavku te ukloniti „Bazooka“ i „Gun“
5. Postaviti glavnu kameru („Main Camera“) pod čvor „hero“. U prozoru Hierarchy odvući „Main Camera“ pod „hero“ kako je pokazano na slici niže. Ovime će glavna kamera pratiti položaj lika u igri.



6. Položaj glavne kamere je sada relativan položaju lika „hero“ te joj je sada potrebno podesiti dimenzije tako da se glavna kamera nalazi točno na liku. Pozicija glavne kamere, vrijednosti X i Y su ostale kako su bile prije njezinog smještanja pod „hero“ te su sada ta dva objekta na različitim lokacijama.



- 6.1. U prozoru Hierarchy odabrati Main Camera, te za poziciju pod Transform postaviti X na nula i Y na nula.



- 6.2. Postavku Size kamere podesiti na 8 čime se kamera odmiče te dobiva nešto širi pogled s obzirom na početnu vrijednost od 5.
7. U narednom koraku je potrebno stvoriti skripte.
- 7.1. Pod Assets stvoriti novi direktorij naziva „Skripte“ u njemu stvoriti novu skriptu pod nazivom „SimplePlatformController“
- 7.2. Na vrhu nove skripte dodati programsko kôd:

```

5 public class SimplePlatformController : MonoBehaviour {
6
7     [HideInInspector] public bool ideDesno = true;
8     [HideInInspector] public bool skok = true;
9
10    public float silaPokreta = 365f;
11    public float najvecaBrzina = 5f;
12    public float silaSkoka = 1000f;
13    public Transform provjeraPoda;
14
15    private bool naPodu = false;
16    private Animator anim;
17    private Rigidbody2D rb2d;
18
19    |
20    /*
21    // Use this for initialization
22    void Start () {
23
24    }
25    */
26
27    void Awake () {
28        anim = GetComponent<Animator> ();
29        rb2d = GetComponent<Rigidbody2D> ();
30    }

```

7.3. Dodati programsko kôd u metodu Update:

```

32    // Update is called once per frame
33    void Update () {
34        naPodu = Physics2D.Linecast (transform.position, provjeraPoda.position,
35        1 << LayerMask.NameToLayer ("Ground"));
36
37        // nema skakanja u zraku
38        if (Input.GetButton ("Jump") && naPodu) {
39            skok = true;
40        }
41    }
--

```

7.4. Stvoriti metodu Okreni te dodati kod:

```

65    void Okreni() {
66        ideDesno = !ideDesno;
67
68        Vector3 theScale = transform.localScale;
69        theScale.x *= -1;
70        transform.localScale = theScale;
71    }

```

7.5. Stvoriti metodu FixedUpdate te dodati kôd:

```

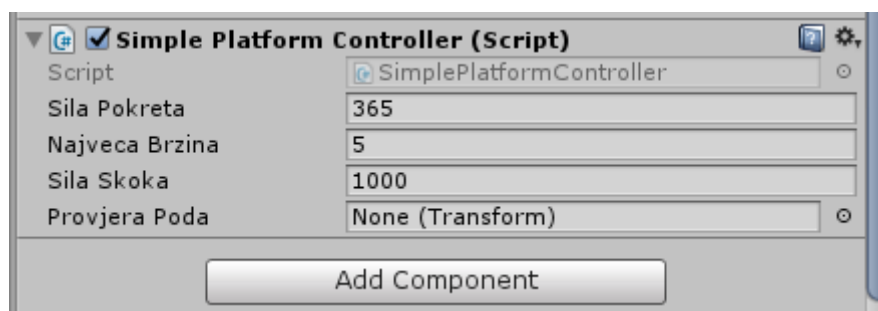
43 void FixedUpdate() {
44     float h = Input.GetAxis ("Horizontal");
45     anim.SetFloat ("Speed", Mathf.Abs (h));
46
47     if (h * rb2d.velocity.x < najvecaBrzina)
48         rb2d.AddForce (Vector2.right * h * silaPokreta);
49
50     if (Mathf.Abs (rb2d.velocity.x) > najvecaBrzina)
51         rb2d.velocity = new Vector2 (Mathf.Sign(rb2d.velocity.x) * najvecaBrzina, rb2d.velocity.y);
52
53     if (h > 0 && !ideDesno)
54         Okreni ();
55     else if (h < 0 && ideDesno)
56         Okreni;
57
58     if (skok) {
59         anim.SetTrigger ("Jump");
60         rb2d.AddForce (new Vector2 (0f, silaSkoka));
61         skok = false;
62     }
63 }

```

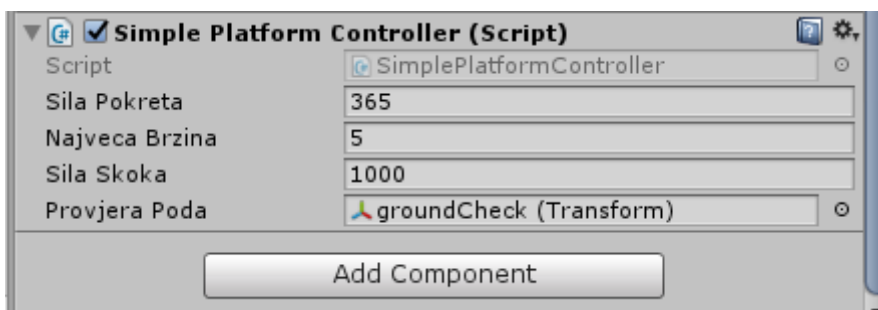
8. Skriptu je potrebno dodati liku za igranje.

8.1. U prozoru Hierarchy, odabrati „Hero“ te na dnu prozora Inspector pritisnuti na tipku **Add Component**.

8.2. Dodati skriptu „SimplePlatformController“. Ukoliko je skripta sintaksno ispravno napisana, parametri se pojavljuju u prozoru Inspector kako prikazuje slika:



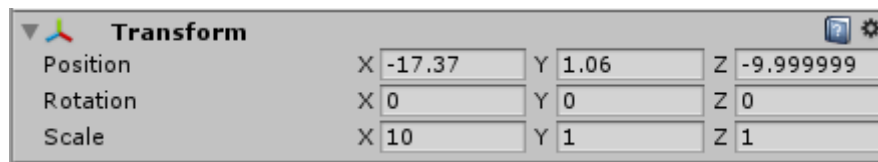
8.3. Parametar „Provjera Poda“ pod skriptom „Simple Platform Controller“ postaviti tako da se u polje parametra iz prozora Hierarchy odvuče objekt „groundCheck“. Učiniti to tako da je u prozoru Hierarchy „hero“ označen, sadržaj prozora Inspector podesiti tako da je vidljiv sadržaj sa prethodne slike te iz prozora Hierarchy odvući „groundCheck“ na polje parametra „Provjera Poda“. Rezultat je vidljiv na slici niže.



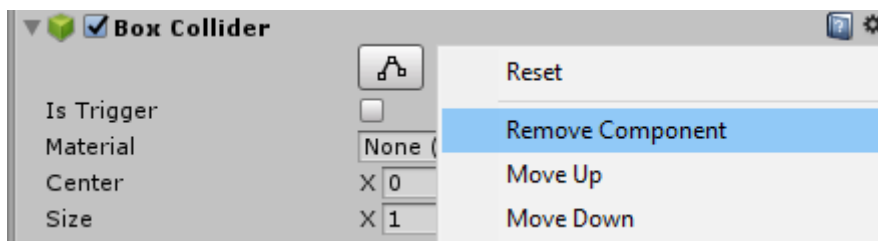
9. Objekt, lik s kojim se igra, treba platformu na kojoj može stajati.

9.1. Stvoriti platformu, odabrati opciju izbornika GameObject → 3D Object → Cube

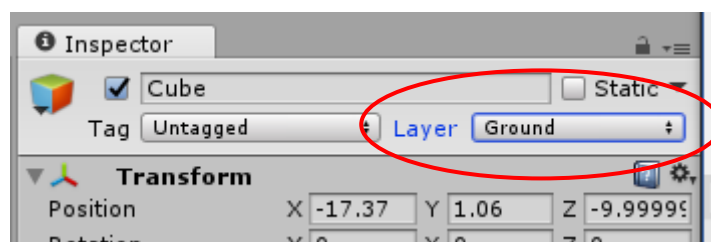
9.2. Platformu spustiti ispod lika te je prošiti, postaviti „Scale“ X koordinatu na 10



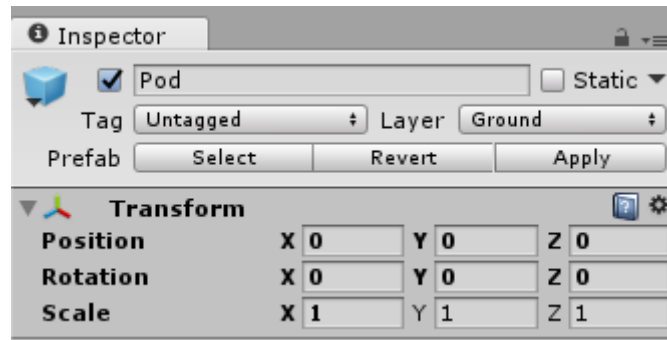
- 9.3. Kako se radi o 3D objektu koji mora funkcionirati u 2D okruženju, potrebno je ukloniti postojeći „Box Collider“ opcijom **Remove Component**.



- 9.4. Umjesto uklonjenog Box Collidera, potrebno je dodati novi. Uz označenu platformu koja u prozoru Hierarchy nosi naziv „Cube“ u prozoru Inspector pritisnuti tipku **Add Component**. U polje za pretragu upisati „Box Collider“ te odabrati „Box Collider 2D“.
- 9.5. U Inspector prozoru za dodanu platformu tj. objekt „Cube“ podesiti sloj (layer) na „Ground“.



- 9.6. Podesiti Transform poziciju koordinatu Z da bude jednaka transform poziciji objekta hero. Obje se mogu postaviti na nula.
- 9.7. Dodati svjetlo pomoću izbornika GameObject → Light → Directional light.
10. Pokrenuti izrađenu scenu i isprobati pomicanje i skakanje razmaknicom.
11. U daljnjim koracima se kreće na izradu platformi po kojima će se lik kretati.
- 11.1. Stvoriti novi prazan Game Object pomoću izbornika: GameObject → Create Empty te ga nazvati „StvaranjePlatformi“.
- 11.2. Resetirati poziciju za „StvaranjePlatformi“ kako pokazuje slika:



11.3. Početni objekt platforme trenutnog naziva „Cube“ preimenovati u „Pod“ te ga odvući iz prozora Hierarchy u prozor Assets pod „Prefabs“.

11.4. Odabrati direktorij Skripte i stvoriti novu skriptu te je nazvati „StvaranjePoda“

12. U skriptu StvaranjePoda dodati programski kôd:

```

5 public class StvaranjePoda : MonoBehaviour {
6
7     public int maxPlatforms = 20;
8     public GameObject platform;
9     public float horizontalMin = 7.5f;
10    public float horizontalMax = 14f;
11    public float verticalMin = -6f;
12    public float verticalMax = 6;
13
14    private Vector2 originPosition;
15
16    void Start () {
17        originPosition = transform.position;
18        Spawn ();
19    }
20
21

```

12.1. Dodati metodu Spawn te dodati programski kôd u tu metodu:

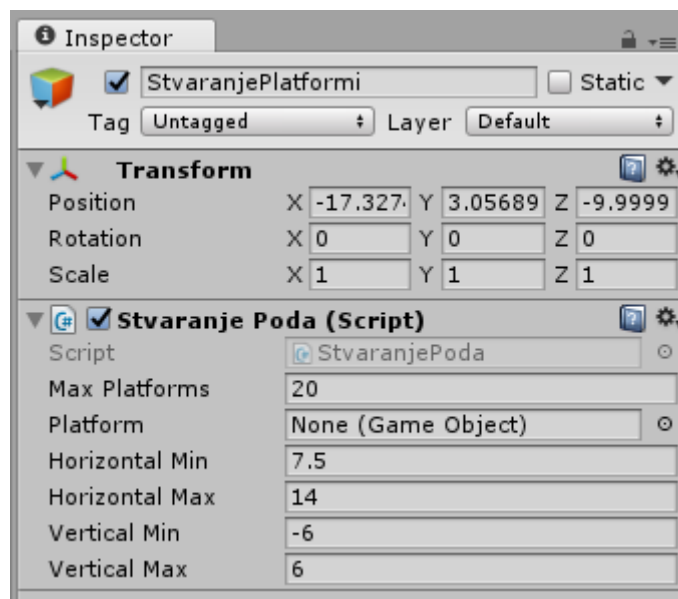
```

21    void Spawn()
22    {
23        for (int i = 0; i < maxPlatforms; i++)
24        {
25            Vector2 randomPosition = originPosition + new Vector2 (
26                Random.Range(horizontalMin, horizontalMax),
27                Random.Range (verticalMin, verticalMax));
28
29            Instantiate(platform, randomPosition, Quaternion.identity);
30            originPosition = randomPosition;
31        }
32    }
33
34    // Update is called once per frame
35    void Update () {
36
37    }

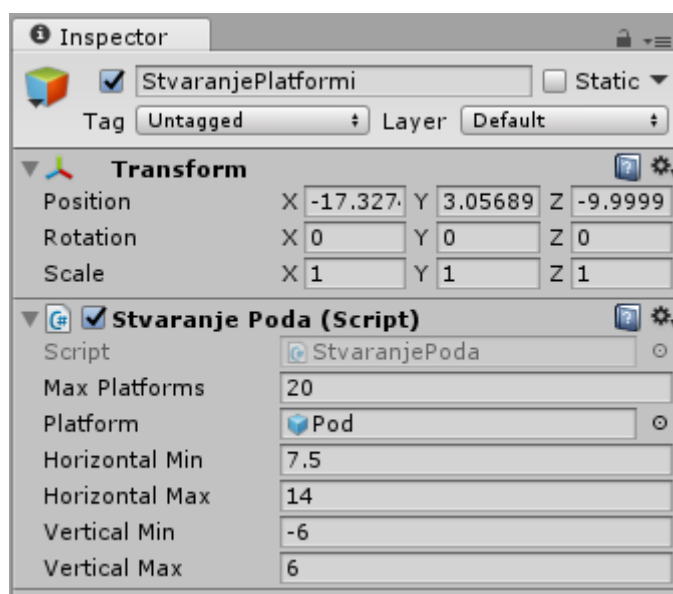
```

13. Skriptu je potrebno dodati pod StvaranjePlatformi u prozoru Hierarchy.

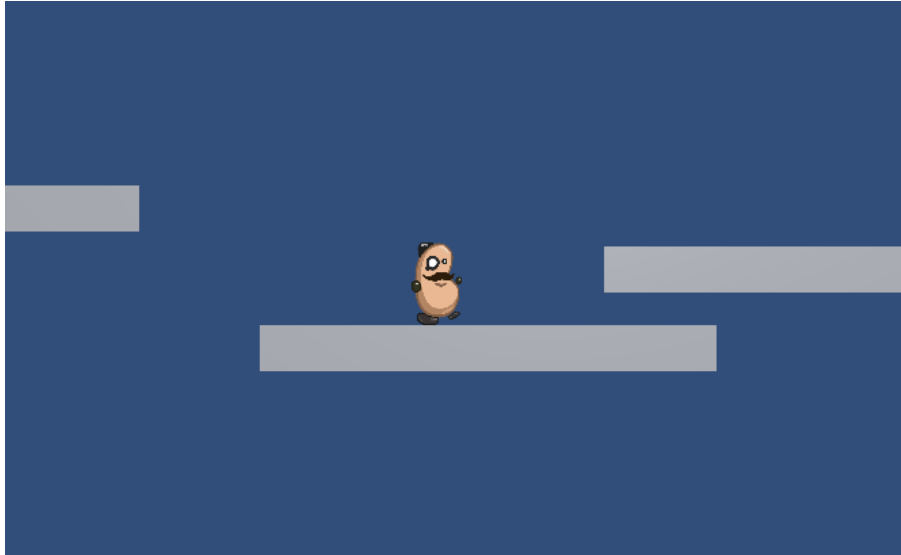
13.1. Odabrati StvaranjePlatformi u Hierarchy prozoru te dodati skriptu StvaranjePoda



- 13.2. Podesiti parametar „Platform“ tako da se iz Assets → Prefabs u okvir polja „Platform“ odvuče Pod. Rezultat je vidljiv na slici niže:



14. Provjeriti da li su pozicije lika („hero“), objekta „StvaranjePlatformi“ postavljene na iste lokacije, najlakše ih je postaviti na koordinate 0,0. Pokrenuti isprobati izrađeno.



15. Potrebno je izraditi skriptu koja se bavi padom s platforme. Pod Skripte izraditi novu skriptu naziva „PadPlatforme“.

15.1. Upisati programski kôd prema slici.

```

5 public class PadPlatforme : MonoBehaviour {
6
7     public float fallDelay = 1f;
8     private Rigidbody2D rb2d;
9
10    void Awake()
11    {
12        rb2d = GetComponent<Rigidbody2D>();
13    }
14
15    void OnCollisionEnter2D (Collision2D other)
16    {
17        if (other.gameObject.CompareTag("Player"))
18        {
19            Invoke ("Fall", fallDelay);
20        }
21    }
22
23    void Fall()
24    {
25        rb2d.isKinematic = false;
26    }
27 }

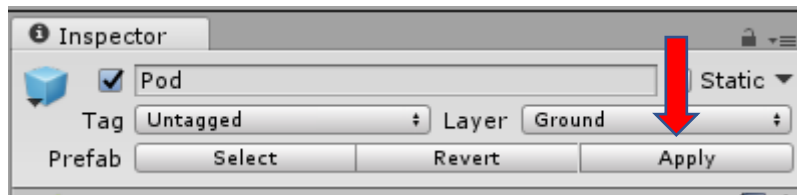
```

16. U prozoru Hierarchy označiti „Pod“ i dodati mu „RigidBody2D“. Pritisnuti na tipku **Add Component** u prozoru Inspector, odabrati Physics 2D te odabrati Rigidbody2D.

16.1. Promijeniti Body Type u „Kinematic“.

16.2. Objektu „Pod“ dodati skriptu „PadPlatforme“.

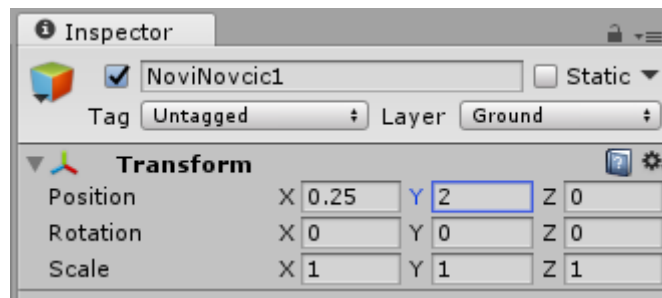
16.3. Ažurirati sve promjene za Prefab „Pod“ pritiskom na tipku **Apply** u prozoru Inspector.



17. Pokrenuti i ispitati rad stvorenog.
18. U ovoj fazi razvoja može se promijeniti boja pozadine i platforme po kojoj lik skače.
 - 18.1. Promjena boje pozadine može se izvršiti pomoću glavne kamere („Main camera“) promjenom njezina svojstva „Background“. Boja se može promijeniti na neku varijantu svijetloplave (4CBBFD00) .
 - 18.2. Boja platforme se mijenja tako da se izradi novi materijal koji se pridijeli platformi. Izraditi novi materijal desnim klikom na „Assets“ i odabirom opcije Create → Material. Materijal nazvati „Materijal platforme“. Boju materijala postaviti na zelenu ili tamno zelenu (00BA11FF). Materijal iz prozora „Assets“ odvući u prozor „Scene“ i spustiti ga na platformu. Primijeniti promjene na „Prefab“ tako da se pritisne tipka „Apply“ u prozoru Inspector za „Pod“.
19. Naredni korak je dodavanje novčića. Pronaći „CoinSprite“ u Assets→Sprites te ga odvući u Hierarchy prozor. Može ga se preimenovati u „Novcic“.
 - 19.1. Dodati mu „Circle Collider“. Pritisnuti na tipku **Add Component** te pod Physics 2D odabrati Circle Collider 2D.
 - 19.2. Postaviti „Is trigger“ opciju za „Circle 2D Collider“.
 - 19.3. Dodati novu skriptu pod Skripte te je nazvati „Novcic“.
 - 19.4. U skriptu „Novcic“ dodati programski kôd:


```

17  void OnTriggerEnter2D (Collider2D other)
18  {
19      if (other.gameObject.CompareTag("Player"))
20          Destroy(gameObject);
21  }
```
 - 19.5. Skriptu dodati objektu „Novcic“. Odabrati ga u prozoru Hierarchy, pritiskom na tipku **Add Component** dodati skriptu objektu Novcic. Ovime se postiže da novčić nestane kada ga igrač (tag: „Player“) dodirne.
20. Kako bi omogućili stvaranje novčića po cijeloj igri potrebno je dodati novu skriptu.
 - 20.1. Stvoriti novu skriptu pod „Skripte“ naziva „StvoriNovcice“.
 - 20.2. Skriptu „StvoriNovcice“ dodjeljujemo objektu „Pod“ tako da je odvučemo u prozor Hierarchy i ispustim na „Pod“
 - 20.3. Objektu „Pod“ dodaje se prazan Game Object tako da se u prozoru Hierarchy odabere „Pod“ te se u izborniku odabere opcija GameObject → Create Empty Child. Novo, objektu dodjeljuje se ime „NoviNovcic1“
 - 20.4. Poziciju za NoviNovcic1 postaviti tako daje X = 0.25 a Y = 2.



20.5. Dupliciranjem stvoriti još dva nova novčića, nazvati ih „NoviNovcic2“ i „NoviNovcic3“ te jednom postaviti X na 0 a drugom na -0.25.

20.6. U skriptu StvoriNovcice upisati programski kôd:

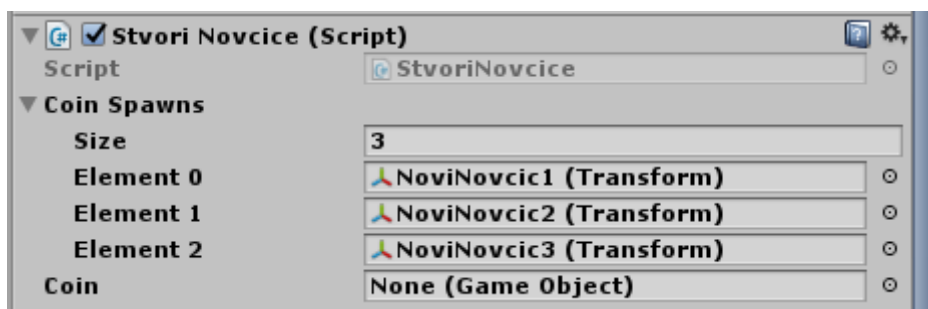
```

5 public class StvoriNovcice : MonoBehaviour {
6
7     public Transform[] coinSpawns;
8     public GameObject coin;
9
10    // Use this for initialization
11    void Start () {
12        Spawn();
13    }
14
15    void Spawn()
16    {
17        for (int i = 0; i < coinSpawns.Length; i++)
18        {
19            int coinFlip = Random.Range (0, 2);
20            if (coinFlip > 0)
21                Instantiate(coin, coinSpawns[i].position, Quaternion.identity);
22        }
23    }
24 }

```

20.7. U ovom koraku je potrebno pridijeliti vrijednost polju Transform[] coinSpawns iz 7. reda gornje skripte. Kako bi to mogli učiniti, prvo je potrebno u Hierarchy označiti „Pod“ te zatim zaključati Inspector pritiskom na ikonu otvorenog lokota u desnom gornjem kutu prozora Inspector.

20.8. Označiti u prozoru Hierarchy odjednom sva tri nova novčića tj. Označiti NoviNovcic1, NoviNovcic2 i NoviNovcic3 te ih sva tri označena odjednom odvući u prozor Inspector i ispustiti na Coin Spawns pod skriptom „Stvori Novcice“.

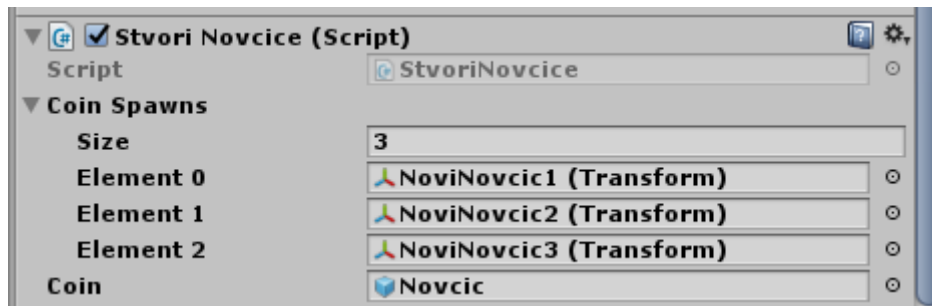


20.9. Otključati Inspector prozor, jednako kako je i zaključan u koraku 20.7.

20.10. Iz prozora Hierarchy odvući Novcic u Assets → Prefabs.

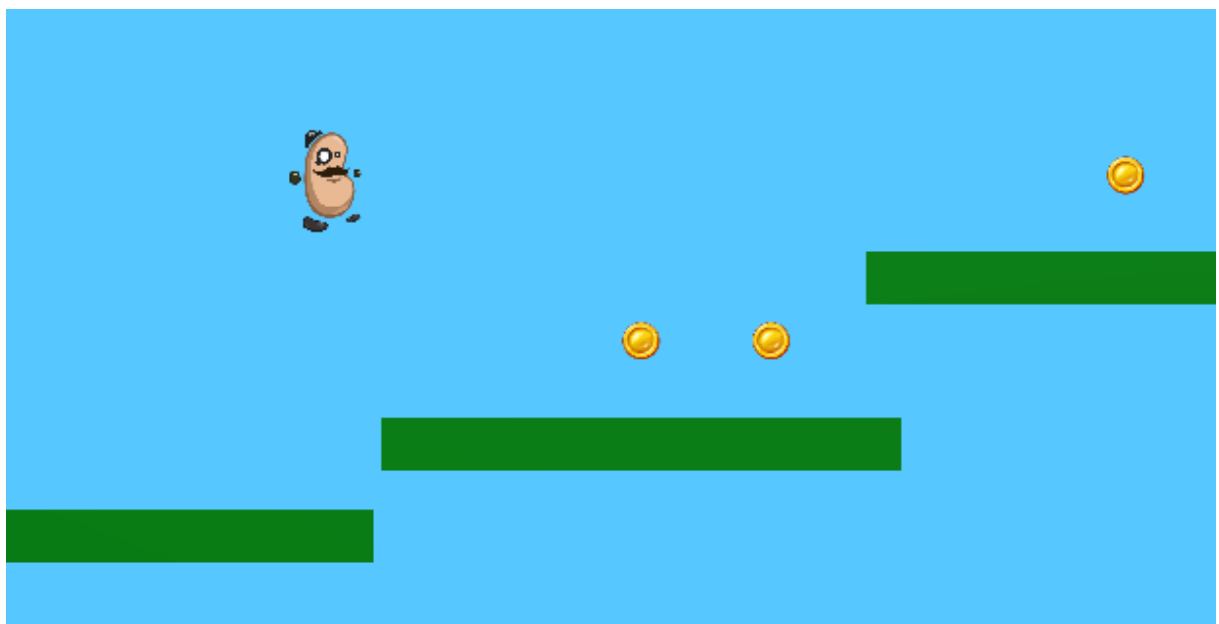
20.11. U prozoru Hierarch označiti (pritisnuti na) „Pod“

20.12. Novi Coin Prefab odvući iz prozora Assets → Prefabs u prozor Inspector i ispustiti ga na svojstvo Coin skripte „Stvori Novcice“. Rezultat je vidljiv na slici niže.



20.13. Promjene na objektu „Pod“ ažurirati za njegov „Prefab“ na način da se na vrhu Inspector prozora uz označen „Pod“ pritisne tipka **Apply**.

20.14. Pokrenuti i isprobati izrađeno.

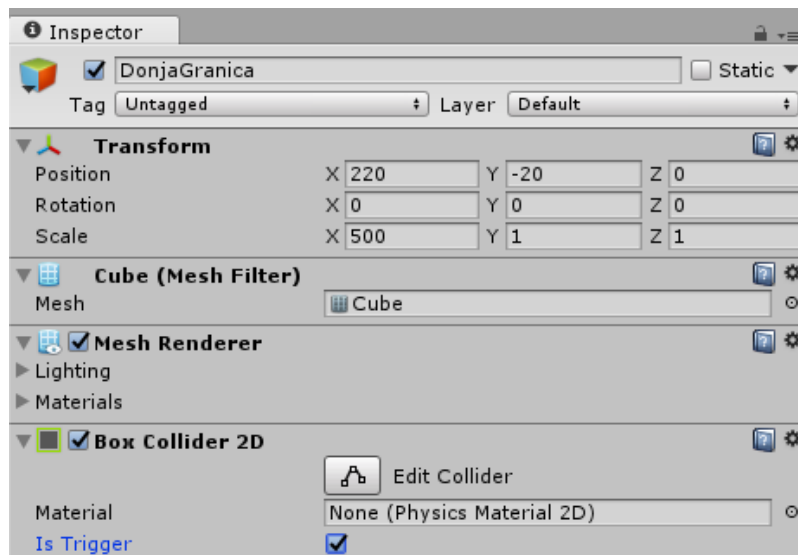


21. Ukoliko igrač ispadne s platforme, potrebno je detektirati pad.

21.1. U scenu se dodaje novi 3D objekt „Cube“ pomoću izbornika GameObject → 3D object → Cube te ga nazvati „DonjaGranica“

21.2. Objektu „DonjaGranica“ ukloniti postojeći Box Collider te umjesto njega dodati novi i to „Box Collider 2D“

21.3. Poziciju objekta „DonjaGranica“ postaviti tako da je X = 220 i Y = -20, a skaliranje po X koordinati na 500. Postaviti svojstvo „Is Trigger“. Vidjeti na slici niže.



21.4. U Skriptu stvoriti novu skriptu pod nazivom „PadNaGranicu“ te u skriptu dodati programski kôd i metodu kako pokazuje slika:

```

17 void OnTriggerEnter2D (Collider2D other)
18 {
19     if (other.gameObject.CompareTag("Player"))
20         Application.LoadLevel(Application.loadedLevel);
21 }

```

21.5. Dodati skriptu objektu „DonjaGranica“.

21.6. Objektu „DonjaGranica“ isključiti „Mesh Renderer“ kako ga se ne bi vidjelo.

21.7. Pkrenuti i isprobati izrađeno.

Pitanja – razumijevanje materije

1. Skripta SimplePlatformController koristi metodu MonoBehaviour.Awake. Po čemu se ta metoda razlikuje od MonoBehaviour.Start?
2. Za što se koristi metoda FixedUpdate? Zbog čega je potrebno koristiti baš tu metodu?
3. Druga linija metode Awake skripte SimplePlatformController sadrži ovaj programski kôd: „rb2d = GetComponent<Rigidbody2D> ();“. Za što se koristi metoda „GetComponent“? Što se ovom metodom dohvaća?
4. Što je to Rigidbody2D? Na koji način „hero“ ima pridijeljen Rigidbody2D?
5. Za što se koristi Rigidbody2D.velocity? Koji je tip podatka Rigidbody2D.velocity?
6. Na koji način je implementirano pomicanje lika („hero“) u skripti SimplePlatformController? Koja linija koda se koristi za dohvaćanje ulaza od korisnika? Za što se koristi Input.GetAxis?
7. Za što se koristi Rigidbody2D.AddForce? Po kojem pravilu se odrađuje akceleracija tijela uz primijenjenu silu?
8. Za što se koristi Physics2D.Linecast? Koja je primjena te metode u skripti SimplePlatformController u metodi Update?
9. Skripta StvaranjePoda koristi metodu MonoBehaviour.Instantiate. Za što se koristi ta metoda?

Zadatak

Nadogradite igru na način da ima početni i završni zaslon te pri tome koristiti scene iz prethodnih projekata. Omogućiti brojenje skupljenih novčića i prikazati rezultat.