



Rull-en-ball

Rull-en-ball

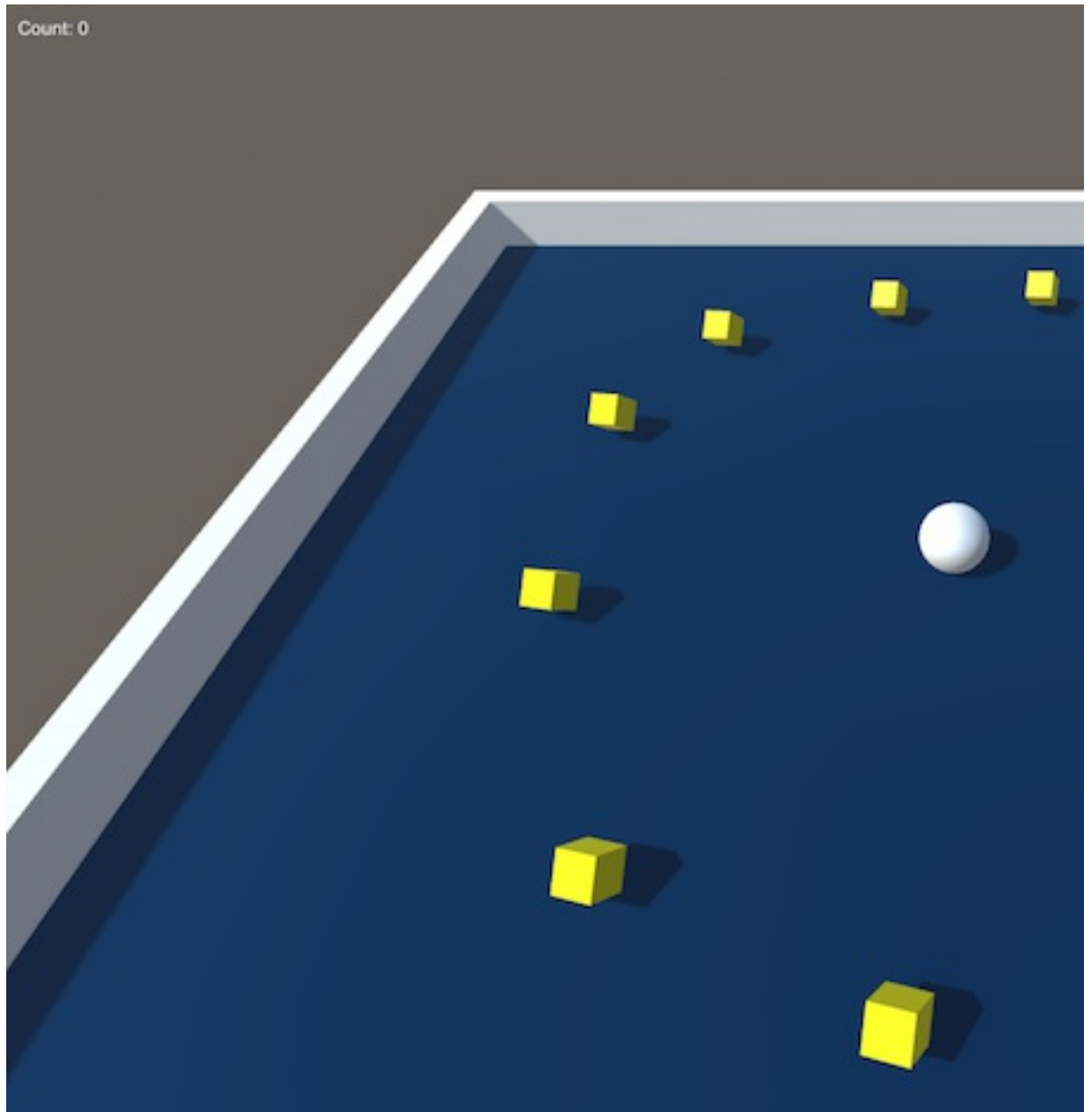
Denne uka skal vi lage vårt første spill! Spillet går ut på å være en ball man har samlet inn alle kubene.

Denne oppgaven tar utgangspunkt i [Roll-a-Ball på Unity sine sider](#). Tit på Unity sine sider der det er aktuelt. Du finner også lenker til alle vid [tutorial](#).

Steg 1: Omgivelser og spil

Introduksjon

Bildet nedenfor viser hvordan spillet ser ut når man er ferdig! I denne flytte spilobjekter med tastaturet, og kollisjon mellom objekter.



✓ Oppsett av spillet

- ☐ Start med å lage et nytt 3D-prosjekt.
- ☐ Lagre Scenen, **File > Save Scene**, legg den i en ny mappe **Ass**
- ☐ Lag brettet vi skal spille på. Dette er et 3D-plan. Du finner den i dette objektet **Ground**.
- ☐ Flytt brettet til origo, **(0, 0, 0)**, ved å trykke på tannhjulet i Tr

- ☐ Skru av Grid-lines på Scenen. Dette finner du under **Gizmos** > **Grid**.
- ☐ Lag bakken litt større. Dette kan du gjøre med å dra på pilene på Scale til **2** på **x**, **1** på **y** og **2** på **z**.
- ☐ Lag en spiller! Fra hierarkiet **Create** > **3D Object** > **Sphere** og på samme måte som vi gjorde med planet.
- ☐ Flytt ballen opp **0.5** enhet slik at vi får den til å ligge på planet posisjonen vi endrer med **0.5**.
- ☐ La oss bytte farge på brettet. Vi lager en ny mappe i Assets-mappe og lag et nytt material og kall den **Background**. Endre fargen på **Albedo**-feltet og velg en fin blåfarge. Du kan bruke RGB-verdier.
- ☐ Lag brettet blått ved å dra **Background**-materialet til brettet i Scenen.
- ☐ For å få bedre lys i spillet kan vi endre rotasjonen på lyset i hierarkiet.

✓ Bevege spilleren

- ☐ Siden vi vil at ballen skal rulle rundt på brettet, kollidere i vegger trenger vi litt fysikk. Dette får vi ved å legge til en Rigidbody-komponent når du har valgt spilleren og velg **Physics** > **Rigidbody**.
- ☐ Lag en ny mappe **Assets/Scripts** og et nytt C#-script som vi kaller det best å skrive selv.

Siden noen av dere jobber hjemmefra legger vi inn hele koden til

PlayerController.cs

```
using UnityEngine;
using System.Collections;

public class PlayerController : MonoBehaviour {
```

```
public float speed = 10.0f;

private Rigidbody rb;

void Start ()
{
    rb = GetComponent<Rigidbody>();
}

void FixedUpdate ()
{
    float moveHorizontal = Input.GetAxis ("Horizontal");
    float moveVertical = Input.GetAxis ("Vertical");

    Vector3 movement = new Vector3 (moveHorizontal,
    rb.AddForce (movement * speed);
}
}
```

- ☐ Siden vi vil at scriptet skal styre ballen må vi dra det inn på **Pla**



Test spillet

Klikk på Play-knappen øverst i Unity-editoren.

- ☐ Trykk på piltastene, flytter ballen seg?
- ☐ Hva skjer hvis du prøver å styre med tastene WASD?

Steg 2: Kamera og spilleor

✓ Flytte kameraet

- ☐ Flytt først kameraet opp 10 enheter (på y-aksen) og vipp det ne
- ☐ Flytt kameraet til et *barn* av **Player**-objektet, dette vil gjøre at programmeringsbegrep som betyr at noe er *under* noe annet, f. **Player**-objektet ved å dra det og slippe det på **Player**-objekte spilleren.
- ☐ Test spillet og se hva som skjer.
- ☐ Siden ikke alt gikk etter planen må vi lage et script for å flytte k slik at det ikke snurrer rundt lengre.
- ☐ Lag et nytt C#-script som du kaller **CameraController** og legg t

CameraController.cs

```
using UnityEngine;
using System.Collections;

public class CameraController : MonoBehaviour {

    public GameObject player;

    private Vector3 offset;

    void Start ()
    {
        offset = transform.position - player.transform.p
    }
}
```

```
void LateUpdate ()
{
    transform.position = player.transform.position +
}
}
```

- ☐ Dra scriptet over til kameraet og med kameraet valgt dra **Playe**



Test spillet

Klikk på **Play**-knappen øverst i Unity-editoren.

- ☐ Flytt ballen og se at kameraet følger etter den.



Sette opp spilleområdet

- ☐ La oss lage noen vegger! Lag et nytt tomt objekt som du kaller for alle veggene. Tilbakestill posisjonen til Walls til **(0, 0, 0)**.
- ☐ Lag første vegg. **Create > 3D Object > Cube** og tilbakestill posisjonen på kubens transform. Legg denne som et barn av Walls og kall den **West Wall**.
- ☐ Lag veggens lang og tykkelse. Under Scale setter vi **0.5**, **2** og **20.5**.
- ☐ Flytt veggens posisjon ved å sette **x** til **-10**.
- ☐ Dupliser veggens posisjon og kall den **East Wall**. Endre x-posisjonen

- ☐ Dupliser **East Wall** og kall den **North Wall**. Tilbakestill posisjon til **10** for å se at den kommer på riktig plass.
- ☐ Dupliser **North Wall** og kall den **South Wall**. Sett posisjon på z



Test spillet

Klikk på **Play**-knappen øverst i **Unity**-editoren.

- ☐ Prøv spillet og se at veggene er på plass.

Steg 3: Samle objekter og



Lag objekter vi kan samle inn

- ☐ Lag en kube vi kan samle inn. **Create** > **3D Object** > **Cube** og ballen havner inni denne kuben.
- ☐ Deaktiver **Player**-objektet.
- ☐ Flytt opp kuben med 0.5 enheter og sett scale til 0.5 i alle akser alle aksene.
- ☐ La oss få kuben til å snurre litt for å tiltrekke oppmerksomhet.
- ☐ Lag et nytt C#-script og kall det **Rotator**. Legg til koden:

Rotator.cs

```
using UnityEngine;
using System.Collections;

public class Rotator : MonoBehaviour {

    void Update ()
    {
        transform.Rotate (new Vector3 (15, 30, 45) * Time.deltaTime);
    }
}
```

- ☐ Start spillet og se at kuben snurrer rundt. Husk å legge til script
- ☐ Siden vi trenger mange kuber må vi gjøre **Pick Up**-objektet om et spillobjekt skal se ut og oppføre seg.
- ☐ Lag en ny mappe og kall den **Prefabs**. Dra **Pick Up**-objektet inn
- ☐ Lag en nytt tomt objekt for å organisere objektene våre. Kall den
- ☐ Endre synsvinkel til rett over spillebrettet og lag 10 nye kuber ved rundt omkring på brettet.
- ☐ Start spillet og se at alle kubene snurrer.
- ☐ La oss endre fargen på kubene! Dupliser **Background**-materiale Albedo og dra den ned på **Pick Up**-prefaben.



Test spillet

Klikk på **Play**-knappen øverst i **Unity**-editoren.

- ☐ Start spillet og se at alle kubene er gule og snurrer rundt!



Samle inn objekter

- ☐ Vis spilleren igjen.
- ☐ Legg merke til at **Player**-objektet har en Sphere Collider-komponent. Legg til **Pick Up**-objektene.
- ☐ Åpne **PlayerController**-scriptet igjen og legg til funksjonen under.

```
void OnTriggerEnter(Collider other)
{
    if (other.gameObject.CompareTag ("Pick Up"))
    {
        other.gameObject.SetActive (false);
    }
}
```

- ☐ Velg **Pick Up**-prefaben og legg til en ny **Tag** vi kaller **Pick Up**. **bokstaver på P og U!**
- ☐ Prøv spillet og se hva som skjer.
- ☐ For å hindre at Unity lager en kollisjon mellom ballen og kubene

komponenten til **Pick Up**-prefaben. Dette gjør at Unity registrer ikke på noen av objektene.

- ☐ Test spillet og se at alt er som det skal!
- ☐ (BONUS) For å gjøre en liten optimisering kan vi legge til en RigidBody. Se at alle kubene faller gjennom gulvet. Dette er fordi vi har satt `isTrigger` for å fikse dette. Les mer om dette på [tutorialen](#).

✓ Vis poengscore

- ☐ Vi trenger noen variabler for å holde styr på hvor mange objekter som er på skjermen.
- ☐ Åpne **PlayerController** og endre det til å være som følger:

PlayerController.cs

```
using UnityEngine;
using System.Collections;

public class PlayerController : MonoBehaviour {

    public float speed;

    private Rigidbody rb;
    public int count;

    void Start ()
    {
        rb = GetComponent<Rigidbody>();
        count = 0;
    }

    void FixedUpdate ()
    {

```

```

        float moveHorizontal = Input.GetAxis ("Horizontal");
        float moveVertical = Input.GetAxis ("Vertical");

        Vector3 movement = new Vector3 (moveHorizontal,
        moveVertical, 0f);
        rb.AddForce (movement * speed);
    }

    void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.CompareTag("Pick Up"))
        {
            other.gameObject.SetActive (false);
            count = count + 1;
        }
    }
}

```

- ☐ Sjekk at `count` endres når vi treffer kubene.
- ☐ Vi trenger noe på skjermen som viser poengene våre. Under hie Gi objektet navnet `CountText`.
- ☐ Endre farge på teksten til hvit og endre posisjon til `(0, 0, 0)`. Endre teksten som står der til `Count Text`.
- ☐ Flytt teksten øverst til venstre. Trykk på firkanten inne i Rect Tra **alt**.
- ☐ Endre Pos X og Pos Y til `10` og `-10`.
- ☐ Nå vil vi endre denne teksten med `PlayerController`-scriptet.

PlayerController.cs

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;

public class PlayerController : MonoBehaviour {

    public float speed;
    public Text countText;

    private Rigidbody rb;
    private int count;

    void Start ()
    {
        rb = GetComponent<Rigidbody>();
        count = 0;
        SetCountText ();
    }

    void FixedUpdate ()
    {
        float moveHorizontal = Input.GetAxis ("Horizontal");
        float moveVertical = Input.GetAxis ("Vertical");

        Vector3 movement = new Vector3 (moveHorizontal,

        rb.AddForce (movement * speed);
    }

    void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.CompareTag("Pick Up"))
        {
            other.gameObject.SetActive (false);
            count = count + 1;
            SetCountText ();
        }
    }
}
```

```
void SetCountText ()
{
    countText.text = "Count: " + count.ToString ();
}
}
```

- ☐ Dra CountText inn i Count Text-referansen til **Player**-objektet.
- ☐ Lagre og test spillet!
- ☐ Lag et nytt UI Text-objekt og kall det Win Text. Lag teksten hvit, s
Sett teksten til å være **Win Text** og flytt den litt opp. Pos X skal
- ☐ Lagre og gå tilbake til **PlayerController**-scriptet.
- ☐ Lag en ny variabel **winText**, initialiser den i **Start** og endre de
slik ut:

PlayerController.cs

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;

public class PlayerController : MonoBehaviour {

    public float speed;
    public Text countText;
    public Text winText;

    private Rigidbody rb;
    private int count;

    void Start ()
    {
```

```

        rb = GetComponent<Rigidbody>();
        count = 0;
        SetCountText ();
        winText.text = "";
    }

    void FixedUpdate ()
    {
        float moveHorizontal = Input.GetAxis ("Horizontal");
        float moveVertical = Input.GetAxis ("Vertical");

        Vector3 movement = new Vector3 (moveHorizontal,

        rb.AddForce (movement * speed);
    }

    void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.CompareTag("Pick Up"))
        {
            other.gameObject.SetActive (false);
            count = count + 1;
            SetCountText ();
        }
    }

    void SetCountText ()
    {
        countText.text = "Count: " + count.ToString ();
        if (count >= 12)
        {
            winText.text = "You Win!";
        }
    }
}

```



Test spillet

Klikk på Play-knappen øverst i Unity-editoren.



Prøv spillet og se hvor fort du klarer å samle inn alle kubene!

Utfordring 1: Bygg spillet slik at vennene dine

Akkurat nå er det jo kun du som kan spille spillet ditt. I [denne video](#) forskjellige plattformer, for eksempel Windows eller Mac. Se om du vennene dine!

Lisens: Oversatt fra unity3d.com **Forfatter:** Transkribert og over