Spelutveckling

med folk från TGJ

The Great Journey

Team Velocita

Karlstad Innovation Park
(och Stora Enso)

Spelutveckling

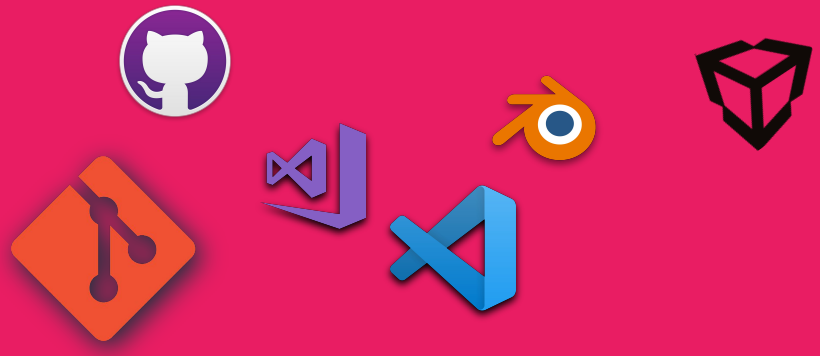
- Spelutvecklings-Community
- Håller evenemang i Kronoparken varje månad

- Spelutvecklare sitter där
- Folk- och Yrkeshögskola inom spel där också
- TheGreatJourney.se

Vad vi ska göra

idag och framtida tillfällen

- få se hur spelutveckling ser ut från början till slut
- få lära er allt ni (minimalt) behöver för att **arbeta** med, **samarbeta** med, och även **lansera** ert **eget spel**
- komma igång med spelutveckling i Unity



Verktyg

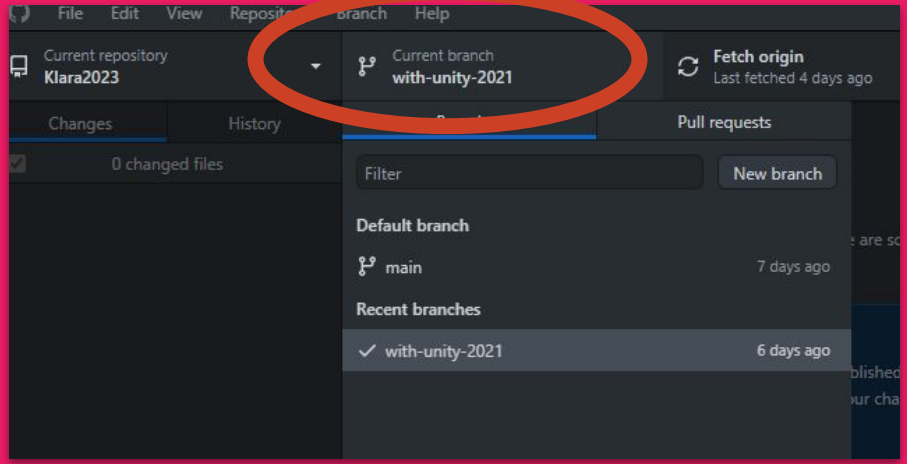
- Unity
- Visual Studio
  - eller Visual Studio Code, Rider, MonoDevelop, etc.
  - Helst med någon Unity plugin
- Blender
- Github Desktop
- Audacity eller Garage Band

Filerna igen:

https://github.com/ErikHogberg/Klara2023

OBS: Lägg inte filerna i OneDrive, Google Drive, etc.

Då går saker sönder i unity



Branching

Git
multi-dimensional
time travel

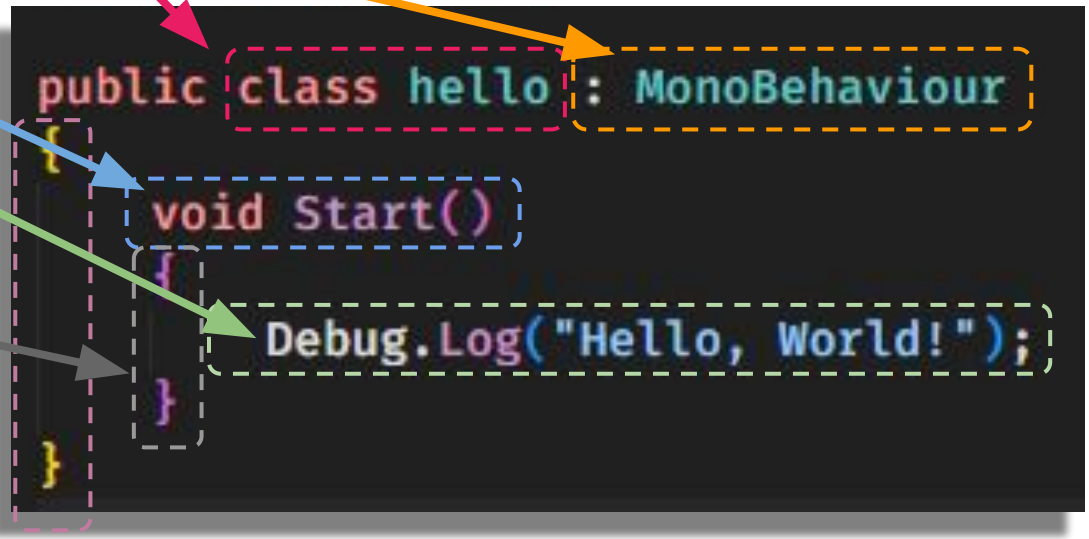
- Jag har gjort en ny “branch” med hur långt vi kom förra gången
- ni som är nya kan hoppa direkt in i den
  - eller om ni inte vill använda det ni tidigare gjort

Dagens uppgifter

- Få animationerna ni gjort att spela vid knapptrycking
- Få spelkaraktären att vända sig mot dit den går

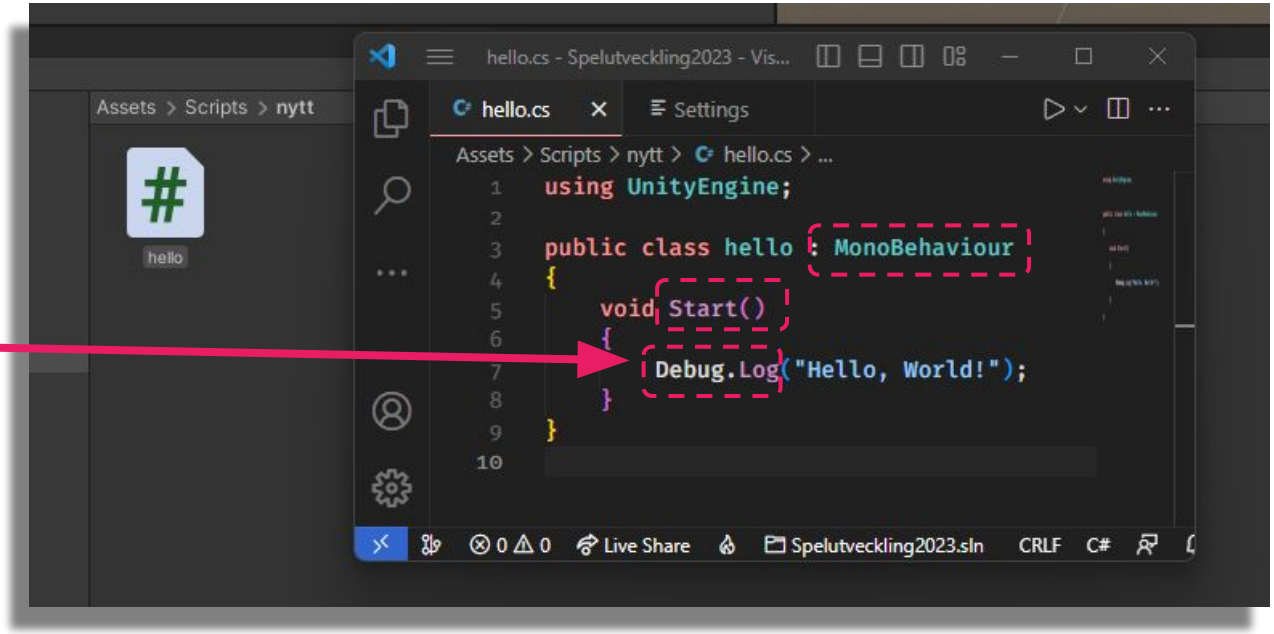
Lite ord (igen)

- **Klass/Class**
  - **Basklass/Base class**
- **Metod/Method**
  - en typ av **Funktion/Function**
- (Funktions/Metods-)**anrop**
- **Scope**



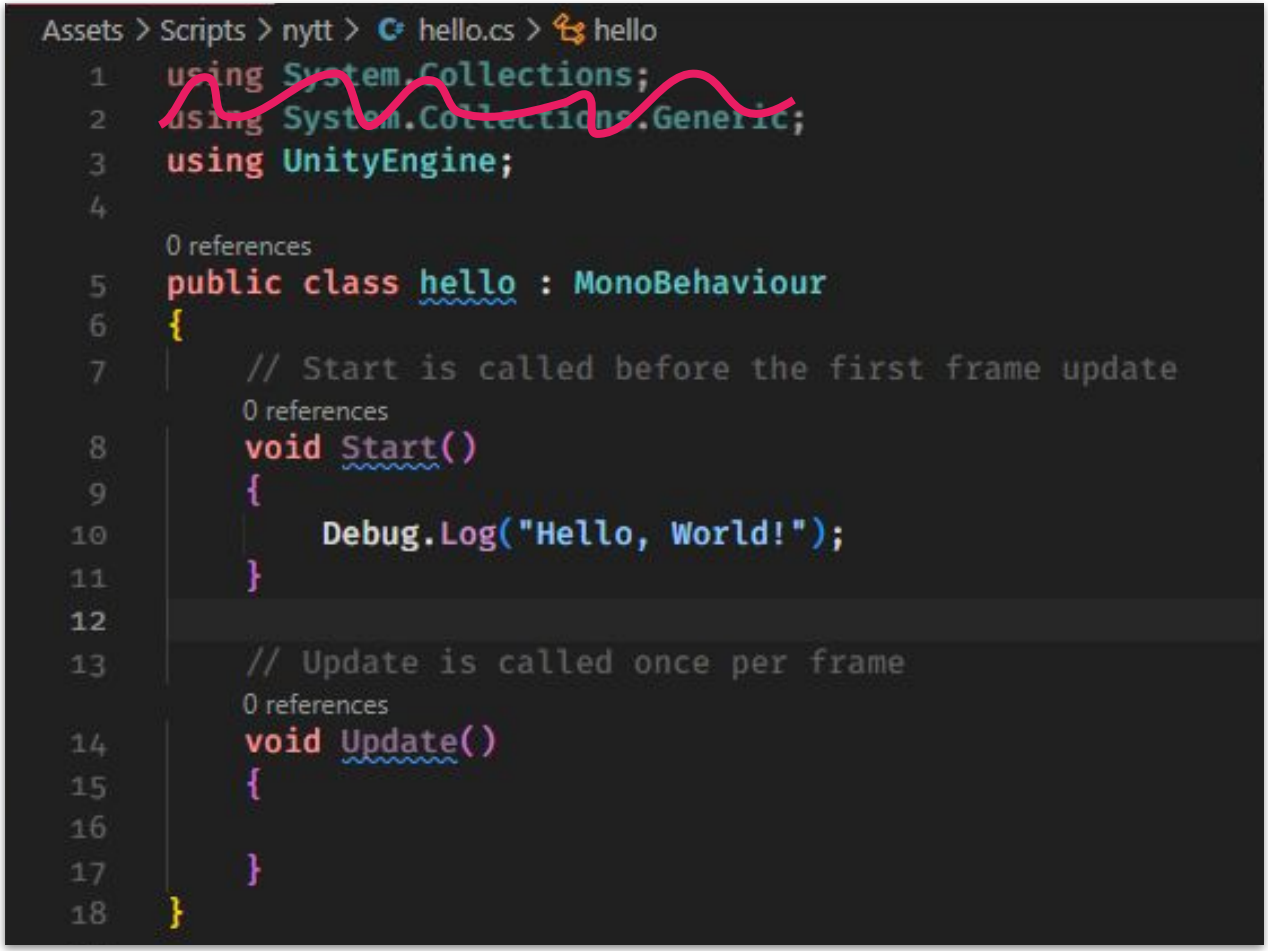
“Hello World”

Ett “MonoBevaviour”
i Unity med C#



“Hello World”

Ett nytt Unity-skript med “Hello, World!”, innan onödigt tagits bort



“Hello World”

Vid knapptryckning

med denna bit kod så sker “hello” när man trycker “e”

det går att byta “e” mot andra tangenter

```
public class hello : MonoBehaviour
{
    public int nummer;
    public float flyttal;
    public string text;
    public UnityEngine.Events.UnityEvent OnHello;

    void Start()
    {
        Hello();
    }

    public void Hello()
    {
        Debug.Log("Hello, World!");
        OnHello.Invoke();
    }

    private void Update()
    {
        if (Input.GetKeyDown("e"))
        {
            Hello();
        }
    }
}
```

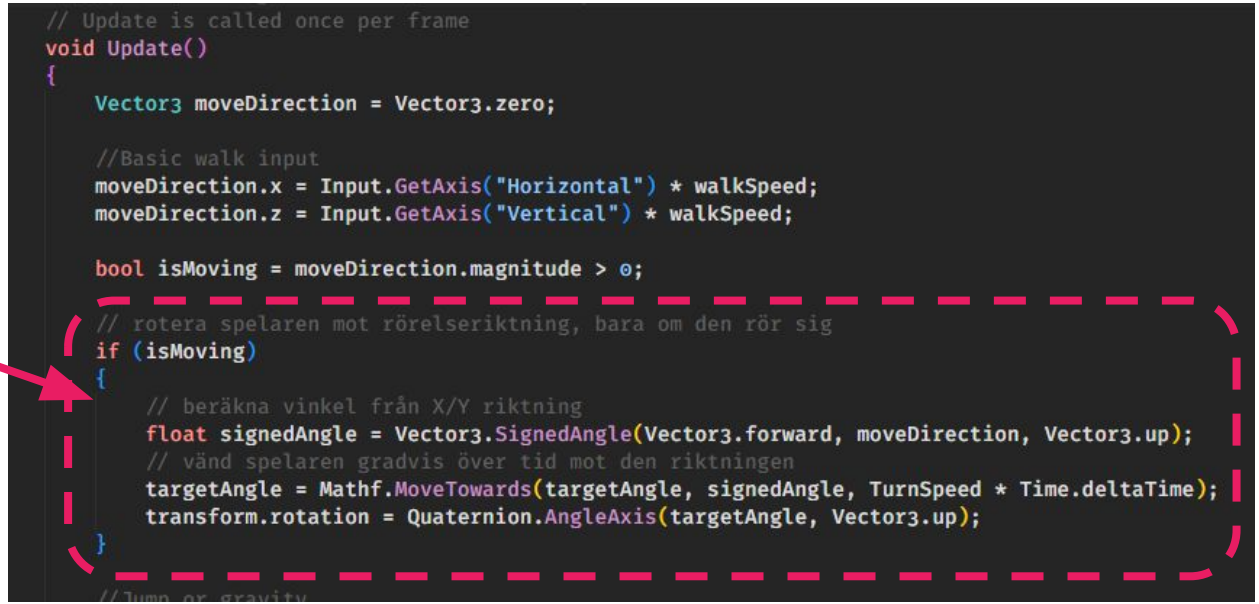
Rotera spelaren

Rotera spelaren

---

nu krävs matte!

öppna “PlayerInput.cs”



```
// Update is called once per frame
void Update()
{
    Vector3 moveDirection = Vector3.zero;

    //Basic walk input
    moveDirection.x = Input.GetAxis("Horizontal") * walkSpeed;
    moveDirection.z = Input.GetAxis("Vertical") * walkSpeed;

    bool isMoving = moveDirection.magnitude > 0;

    // rotera spelaren mot rörelseriktning, bara om den rör sig
    if (isMoving)
    {
        // beräkna vinkel från X/Y riktning
        float signedAngle = Vector3.SignedAngle(Vector3.forward, moveDirection, Vector3.up);
        // vänd spelaren gradvis över tid mot den riktningen
        targetAngle = Mathf.MoveTowards(targetAngle, signedAngle, TurnSpeed * Time.deltaTime);
        transform.rotation = Quaternion.AngleAxis(targetAngle, Vector3.up);
    }

    //Jump or gravity
```

Rotera spelaren

Först, lägg till några fält utanför “Update()”

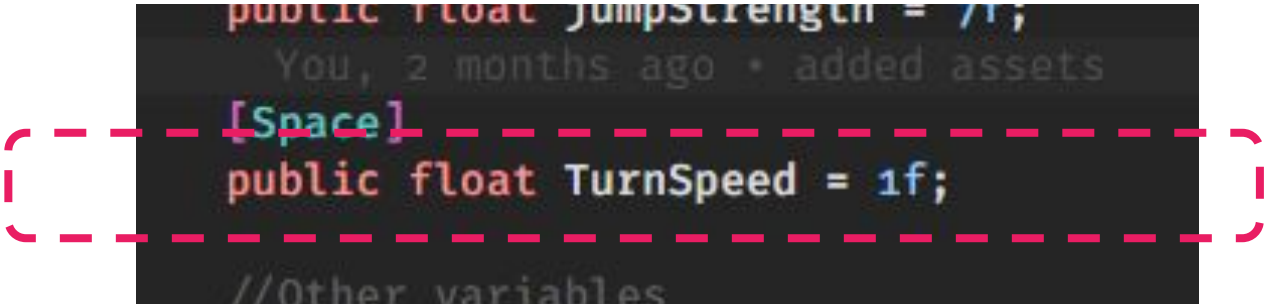
Dessa sparar var vi var förra gången vi uppdaterade

```
// värden för att spara data mellan varje uppdatering, för jämförelser och gradvisa förändringar
bool wasMoving = false;
bool wasGrounded = false;
float targetAngle = 0;
```

Rotera spelaren

Lägg också till en inställning (som är public) så att vi kan ändra hastigheten på vändningen i Unity

Även den utanför “Update()”, t.ex. uppe vid dom andra inställningarna



Rotera spelaren

---

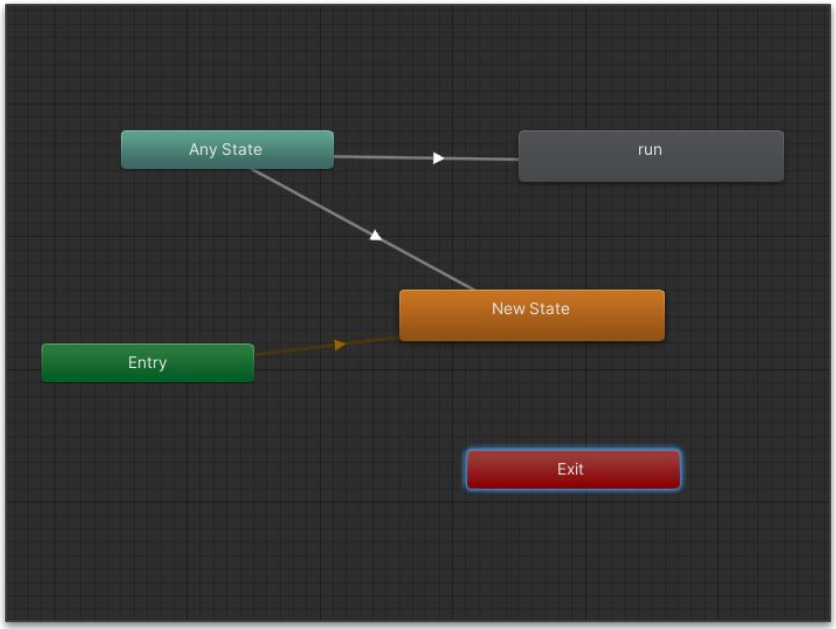
tillbaka till matten

```
// rotera spelaren mot rörelseriktning, bara om den rör sig
if (isMoving)
{
    // beräkna vinkel från X/Y riktning
    float signedAngle = Vector3.SignedAngle(Vector3.forward, moveDirection, Vector3.up);
    // vänd spelaren gradvis över tid mot den riktningen
    targetAngle = Mathf.MoveTowards(targetAngle, signedAngle, TurnSpeed * Time.deltaTime);
    transform.rotation = Quaternion.AngleAxis(targetAngle, Vector3.up);
}
```

Rotationer ändras i Unity genom en Transform

Efter rasten

Användning av animationer i skript



TGJ

The Great Journey

Karlstad Innovation Park
(och Stora Enso)

- Spelutvecklings-Community
- Håller evenemang i Kronoparken varje månad
- kalender i vår discord och web

- Spelutvecklare
- Även Folk- och Yrkeshögskola inom spel
- TheGreatJourney.se