

Nu har vi bygget et spil vha. præfabrikerede byggesten både elementer (sprites, prefabs, lyde mm.) og scripts.
Nu vil vi kaste os ud i koden.

Dette foregår normal i visualstudio, men kan i princippet gøres i en vilkårlig texteditor.

Lad os tilføje den klassiske ufo der flyver øverst i skærmen.

Den kan vi lave med de sprites vi allerede har, og tilføje et spawnSaucer script til GameController objektet.

‘spawnSaucer’ scriptet kan laves ud fra elementer vi ‘hugger’ fra dropBombs scriptet:

De vil fungere ens i at de med jævne mellemrum skal skabe hhv. En bombe og en ufo.

dropBombs Scriptet, forklaret:

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 0 references
6 public class dropBombsScript : MonoBehaviour
7 {
8     public GameObject bombPrefab; // 'slot' til bombegrafik-prefab
9     private GameObject[] invaders; // array/liste til invaderne
10
11     public float bombFrequency = 2f; // one bomb every two seconds
12     private float bombDropTimer = 0f; // timer nulstillet
13
14     // Start is called before the first frame update
15     0 references
16     void Start()
17     {
18     }
19
20     // Update is called once per frame
21     0 references
22     void Update()
23     {
24         bombDropTimer += Time.deltaTime; // tæl timeren op med den tid der gik siden sidste frame
25         if (bombDropTimer > bombFrequency) // hvis der er gået lang nok tid til at vi skal kaste en bombe
26         {
27             invaders = GameObject.FindGameObjectsWithTag("Enemy"); // find alle invadere og prop dem ind i listen
28             int numVaders = invaders.Length; // hvor mange invadere er der?
29             if (numVaders > 0) // hvis der er mere end 0
30             {
31                 GameObject theChosenDropper = invaders[Random.Range(0, numVaders)]; // vælg en tilfældig invader
32                 var bomb = Instantiate(bombPrefab, theChosenDropper.transform.position - (transform.up / 2), theChosenDropper.transform.rotation); // 'instantier'/skab en bombe
33                 bomb.GetComponent<Rigidbody2D>().velocity = transform.up * (-3); // sæt den til at falde
34                 Destroy(bomb, 5f); // sæt bomben til at selvdestruere om 5 sekunder
35             }
36
37             bombDropTimer = 0f; // nulstil timeren
38         }
39     }
```

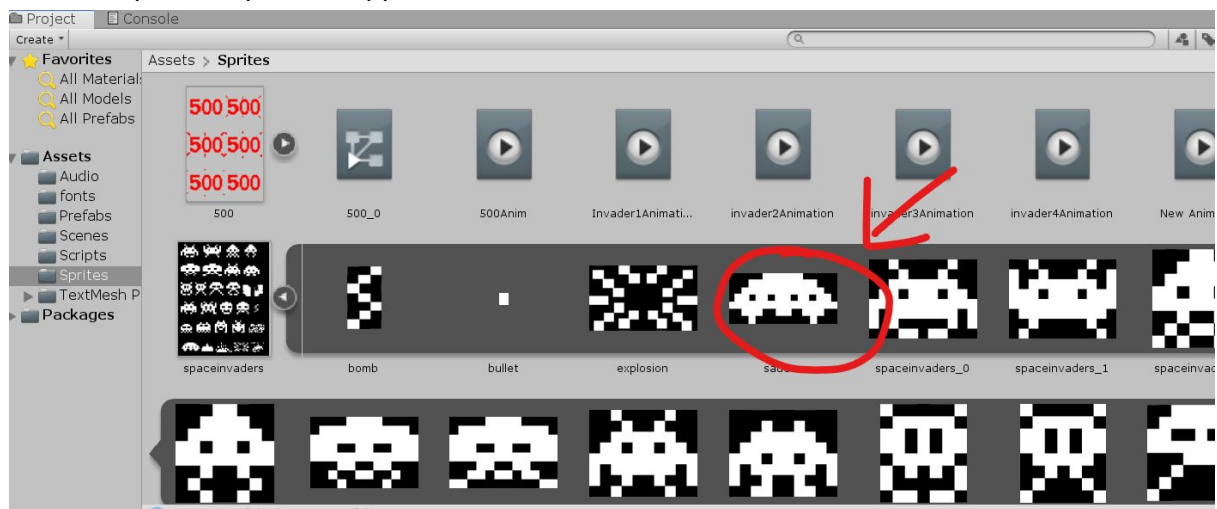
Så hvis vi laver et nyt script på GameController : klik ‘AddComponent’ og kald det ‘spawnSaucer’.

Scriptet kunne komme til at se sådan ud:

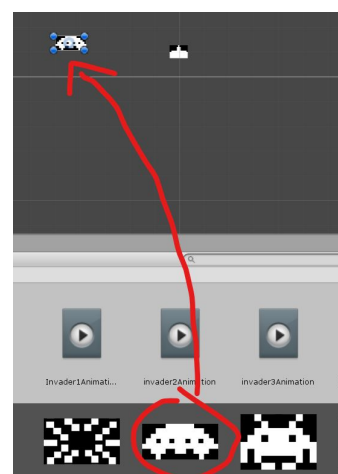
```
5 public class spawnSaucer : MonoBehaviour
6 {
7     private float spawnTimer = 0; // timer til at tælle tiden
8     public GameObject saucerPrefab; // slot til saucer prefabben
9     // Start is called before the first frame update
10    void Start()
11    {
12    }
13
14    // Update is called once per frame
15    void Update()
16    {
17        spawnTimer += Time.deltaTime; // tæl timeren op med den tid der gik siden sidste frame
18        if (spawnTimer > 10) // hvis der er gået mere end 10 sekunder
19        {
20            var saucer = Instantiate(saucerPrefab, new Vector2( 12, 4.5f), transform.rotation); // lav en saucer øverst til højre
21            saucer.GetComponent<Rigidbody2D>().velocity = transform.right * (-2.5f); // sæt den til køre mod venstre
22            Destroy(saucer, 5f); // sæt bomben til at selvdestruere om 5 sekunder
23            spawnTimer = 0; // nulstil timeren
24        }
25    }
26 }
27
```

Men før vi kan bruge det skal vi have lavet en saucerPrefab, altså et objekt vi kan 'instantiate':

Find ufo spriten i sprites mappen:

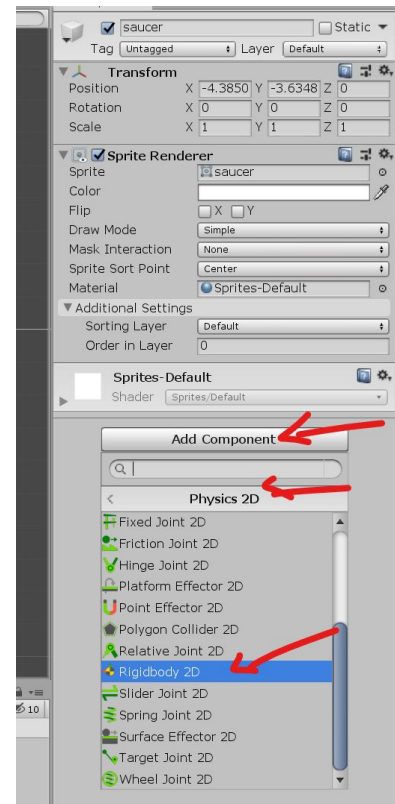
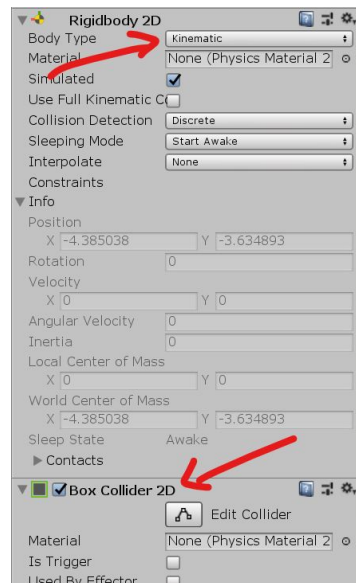


Træk den op på scenen:

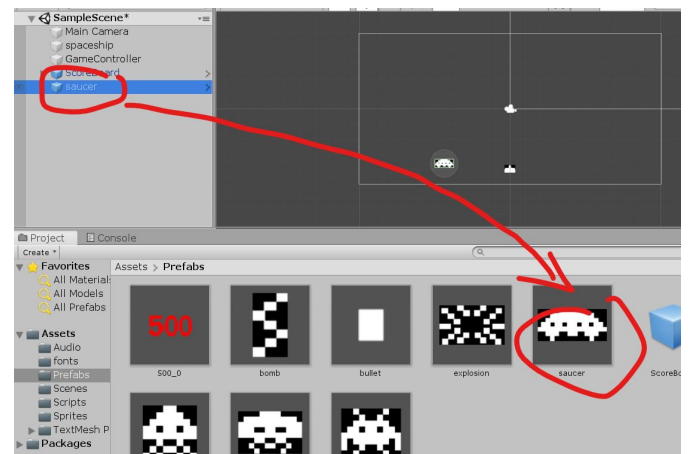


Tilføj nu en Rigidbody2D Component til sauceren/ufø
og sæt denne til 'Kinematic' istedet for 'Dynamic'

Tilføj derefter på samme måde en BoxCollider2D:

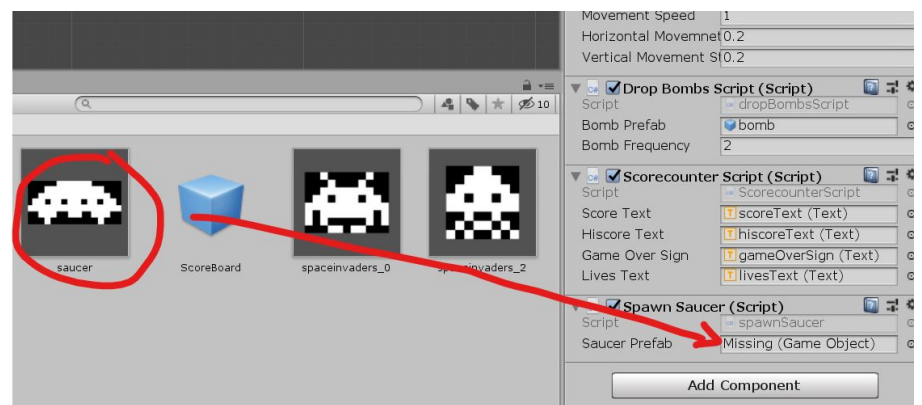


Nu er sauceren/ufoen klar, den mangler bare at blive konverteret til en Prefab.
Det gøres ved at tage saucer i 'hierarchy' vinduet og trække ned på 'prefab' mappen:



Slet nu sauceren fra scenen.

Vælg 'GameController' objektet og
træk suacer-prefaben over på 'saucer-prefab' (missing Gameobject):



Og test så spillet.

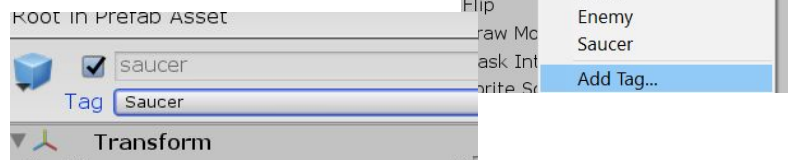
Nu kan det være svært at

ramme, men man kan jo 'snyde' ved at ændre shotDelay på rumskibet til 0.

Men der sker ikke noget når man rammer.

Det er fordi vores bullets kun kan ramme GameObjects der er tagget med 'Enemy', så hvis vi åbner vores Saucer prefab og giver den tagget Enemy, vil vi kunne skyde den.

MEN!! 'Enemy' tagget bruges også til at finde de gameobjects der skal flyttes (invaderne) så det bliver noget rod. I stedet laver vi en ny tag type "Saucer"



og så skal vi bare rette i vores bulletScript så den også kan ramme den type:

Tilføj `|| hit.tag == "Saucer"` det betyder 'eller objektet er tagget med 'saucer''

```
public class bulletScript : MonoBehaviour
{
    public GameObject explosionPrefab;
    0 references
    private void OnTriggerEnter2D(Collider2D collision)
    {
        var hit = collision.gameObject;

        if (hit.tag == "Enemy" || hit.tag == "Saucer")
        {
            var explosion = Instantiate(explosionPrefab, hit.transform.position);
            Destroy(hit);
            Destroy(explosion, 1f);
            Destroy(gameObject);
        }
    }
}
```

Test spillet igen.

Nu skulle det gerne virke, men den giver ingen point..

Det klarer vi let ved at trække 'scoreScript' over på vores saucer prefab, og sætte scoren til fx. 500:

