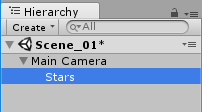
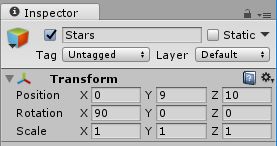
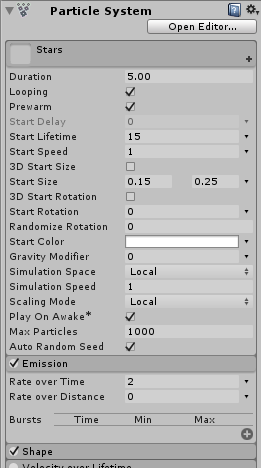
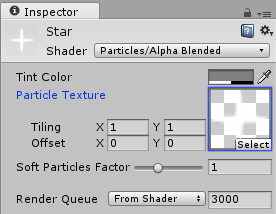
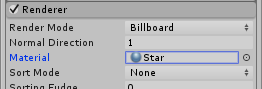
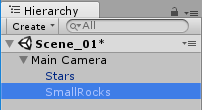
# CoderDojo Unity

Unity - <https://store.unity.com/download?ref=personal>

Into the space assets - <https://www.assetstore.unity3d.com/en/#!/content/20749>

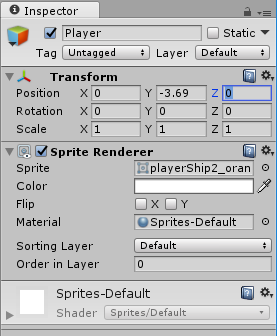
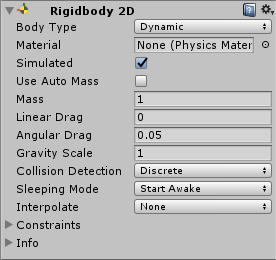
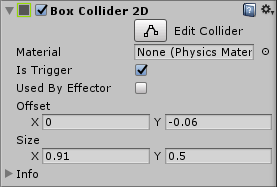
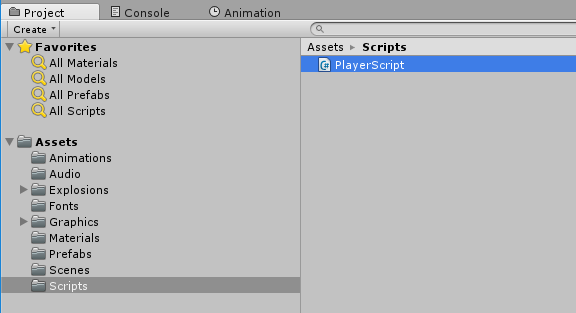
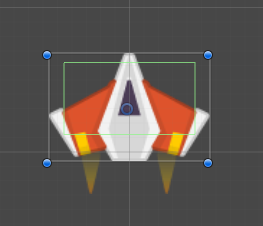
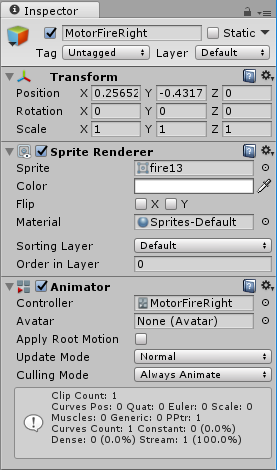
## Opdracht 1 – De achtergrond

We gaan het spel in verschillende stappen opbouwen en om te beginnen gaan we iets doen aan de achtergrond. Als je het spel nu zou uitproberen, dan krijgen we alleen maar een lelijke donkerpaarse achtergrond te zien. Dat is beter dan zwart, want dat is helemaal saai, maar veel stelt het nog niet voor. Laten we als eerste eens wat sterren toevoegen!

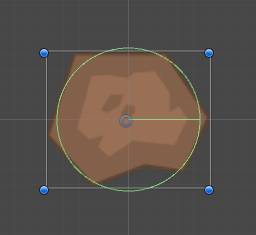
1. Maak een leeg GameObject aan onder de Main Camera en geef het de naam “Stars”, zoals je in het plaatje hiernaast ziet.
2. Stel de volgende waarden in voor de positie en rotatie:
3. Voeg een “Particle System” toe en pas de volgende waarden aan (de rest kan onveranderd blijven):  
     
   Prewarm: **aangevinkt**  
   Start Lifetime: **15**  
   Start Speed: **1**  
   Start Size: **0.15 – 0.25**  
   - Emission  
   Rate over Time: **2**  
   - Renderer  
   Order in Layer: **-100**   
   (niet afgebeeld)
4. Maak een nieuw “Material” aan in de “Materials” folder met de naam “Star” en de volgende waarden:  
     
   Shader: **Particles/Alpha Blended**  
   Particle Texture: **“star1”**   
   (uit de Graphics/Effects folder)
5. Koppel dit nieuwe materiaal aan het “Particle System” dat we hebben gemaakt onder “Renderer – Material”.
6. Sleep het GameObject “Stars” nu vanuit de “Hierarchy” naar de “Prefabs” folder. Dit zorgt ervoor dat we dit GameObject makkelijk kunnen hergebruiken.
7. Zoals je ziet hebben we al een GameObject in de “Prefabs” folder, genaamd “SmallRocks”. Dit doet bijna hetzelfde als ons eigen “Stars” maar nu moet stenen in plaats van sterren. Sleep “SmallRocks” naar de Main Camera zodat het onder “Stars” komt te staan.
8. Probeer het spel uit. Nu hebben we een mooie bewegende achtergrond, alleen kunnen we zelf nog niet veel doen. Dat wordt het onderwerp van de volgende opdracht.

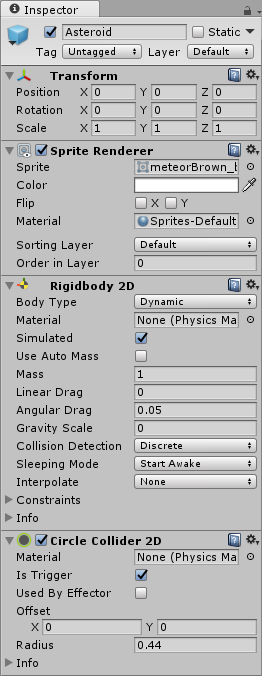
## Opdracht 2 – De speler

Als tweede stap gaan we het ruimteschip van de speler bouwen. Na deze opdracht kunnen we met ons eigen schip rondvliegen in de achtergrond die we al hebben.

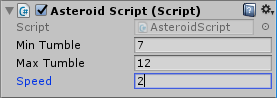
1. Zoek in de folder “Graphics\Player” het plaatje op met de naam “playerShip2\_orange” en sleep het naar de scene. Dit maakt er een GameObject van met een “Transform” en een “Sprite Renderer”. Pas de naam aan naar “Player” in vul de positie in zoals in het plaatje hiernaast.
2. Voeg een “RigidBody2D” toe. Dit is een component die er voor zorgt dat we de snelheid en acceleratie een GameObject kunnen regelen.
3. Start het spel nu, wat gebeurt er?
4. Naast snelheid regelt een “RigidBody2D” ook dingen als zwaartekracht voor ons. Dat hebben we in een ruimtespel niet nodig, dus zet “Gravity Scale” op 0.
5. Voeg ook een “BoxCollider2D” toe. Deze component (en alle andere “Collider” componenten) worden gebruikt om te bepalen of objecten in het spel elkaar raken.  
   Zorg dat “Is Trigger” is aangevinkt en dat de “Offset” en “Size” zijn ingevuld zoals in de afbeelding hiernaast.
6. Nu moeten we er voor gaan zorgen dat we het ruimteschip kunnen besturen. Om dat te doen hebben we een script nodig. Maak een nieuwe folder “Scripts” aan onder “Assets” en voeg daar een “C# script” aan toe. Noem dit script “PlayerScript”. Als alles goed is gegaan ziet je project er uit zoals in dit plaatje:
7. Dubbelklik op “PlayerScript”. Dit opent het script in de editor die in Unity is ingesteld. Vervang de inhoud die Unity automatisch heeft gemaakt door de volgende code: 
8. Koppel het script aan het “Player” GameObject. Nu kun je een nieuwe waarde invullen, namelijk “Speed”. Vul daar de waarde 6 in.
9. Probeer het spel uit. Nu kun je het ruimteschip besturen met de pijltjestoetsen.
10. Voordat we verder gaan met het maken van tegenstanders, gaan we ons ruimteschip nog een beetje verfraaien. Sleep het plaatje “fire13” vanuit de “Graphics\Effects” folder twee keer naar “Player” in de “Hierarchy”. Noem het eerste GameObject wat hierdoor aangemaakt wordt “MotorFireLeft” en de tweede “MotorFireRight”.
11. Stel de positie van de twee MotorFire-objecten als volgt in. Als het goed is ziet je ruimteschip er dan uit zoals het plaatje hiernaast.   
      
    MotorFireLeft:  
    X = -0.27 en Y = -0.431715  
    MotorFireRight:  
    X = 0.2565265 en Y = -0.431715
12. Koppel de “Animation Controller” “MotorFireLeft” uit de “Animations” folder aan het GameObject “MotorFireLeft”. Doe hetzelfde voor “MotorFireRight”. Beide GameObjecten zien er daarna in de “Inspector” uit zoals hiernaast: (het plaatje geeft “MotorFireRight” weer, de waarden van “MotorFireLeft” zijn uiteraard iets anders.)  
      
    Dit animeert de motoruitlaten, door elke 0.1 seconden het plaatje te verwisselen.

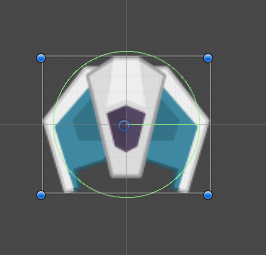
## Opdracht 3 – De tegenstanders



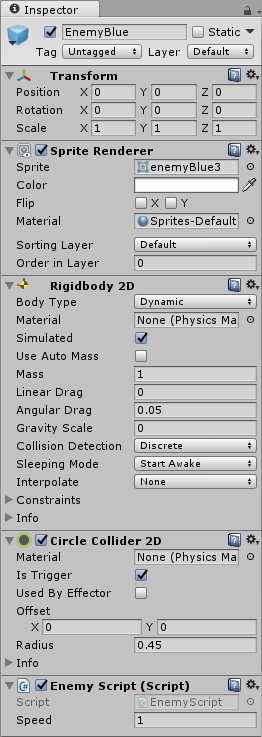


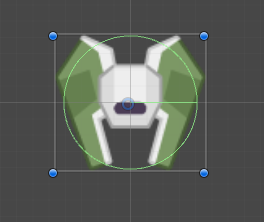


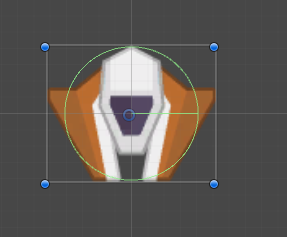


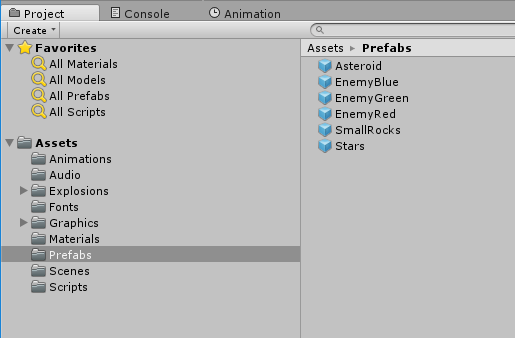




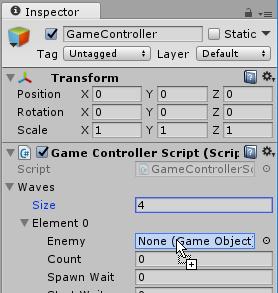




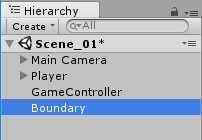




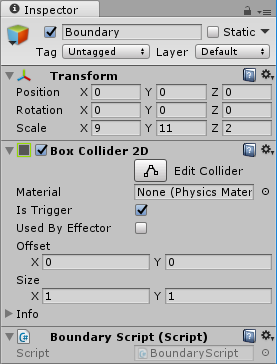




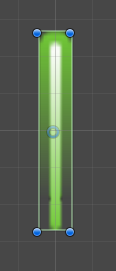




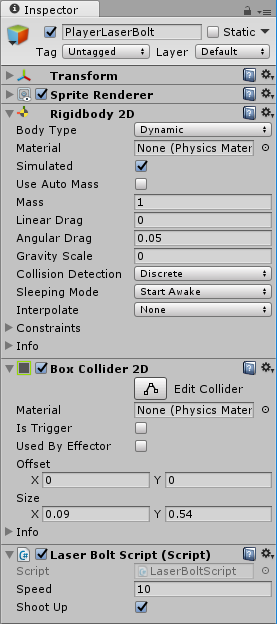


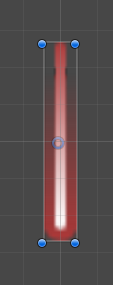


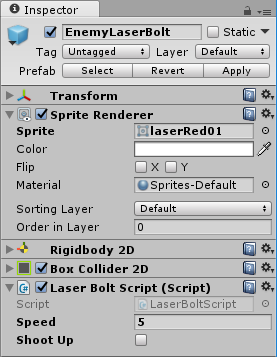
## Opdracht 4 – Lasers en explosies

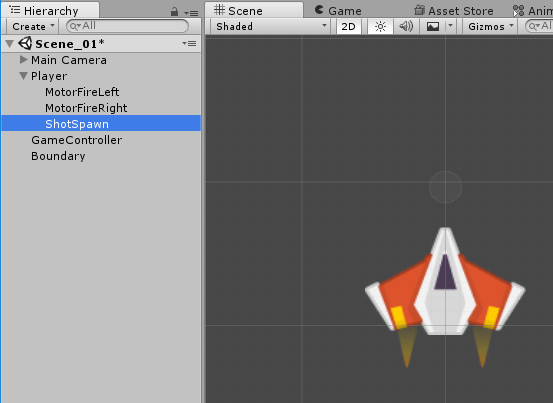


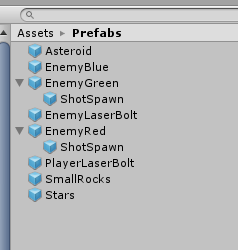




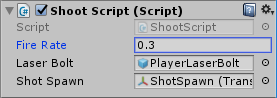


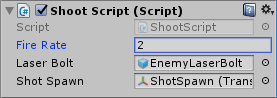


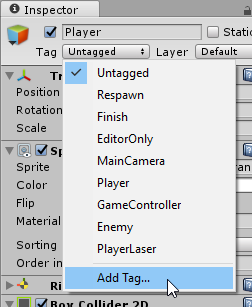




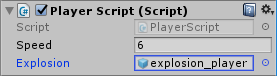


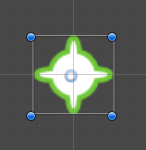


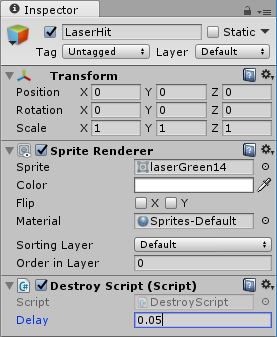








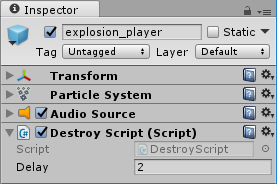












## Opdracht 5 – Losse eindjes





