

## Module 8 : Unity 1 (voor de student)

## Introductie

Tijdens deze module gaan we aan de slag met een database connectie tussen Unity en MySQL. Je gaat leren hoe je een naam en highscore kan sturen naar een database en je leert hoe je data uit een database kan ophalen en weergeven binnen een aangeleverde game. Voor deze module heb je de unity engine nodig. Op school werken we met de versie **2020.3.12** Het is belangrijk dat als je thuis werkt, je **DEZELFDE** unity versie thuis hebt. Anders komt het in conflict met elkaar. Je kan de juiste Unity versie via onderstaande link downloaden. **Zoek naar Unity 2020.3.12 LTS**

<https://unity3d.com/unity/qa/its-releases>

## De leerdoelen voor deze module zijn:

## Je kan

- de code van een bestaand spel (gemaakt in C#) interpreteren
- deze code aanpassen om een wijziging door te voeren
- vanuit Unity verbinding maken met een databaseserver
- de score van de speler in een database opslaan
- De score van een speler ophalen uit een database

## Inhoud

- Toepassing van de taal C# in Unity
- Het opzetten van een (eenvoudig) back-end systeem in C#
- Het maken van een database (enkele tabel voor score)
- Het maken van een verbinding met een database

## Tijdsplanning

<b>Week 1</b>			
BASIS2, les 1	2 uur	Instellen SQLite database en php code	
BASIS2, les 2	2 uur	Theorie Unity en data sturen naar DB	
BASIS3	4 uur	Zelfstandig aan de slag met data sturen naar een database	
<b>Week 2</b>			
BASIS2, les 3	2 uur	Theorie data ophalen uit via PHP	
BASIS2, les 4	2 uur	Theorie data ophalen en gebruiken in Unity	
BASIS3	4 uur	Zelfstandig aan de slag met data ophalen uit een database en verwerken in Unity	

# Les week 1

## THEORIE

We gaan deze module aan de slag met drie onderwerpen. Unity, MySQL en PHP.

Unity is een game engine. Dit betekent dat er met deze software games gemaakt kunnen worden. Je krijgt al een voorgemaakte game, maar je kan de game uitbreiden als je dat wilt.

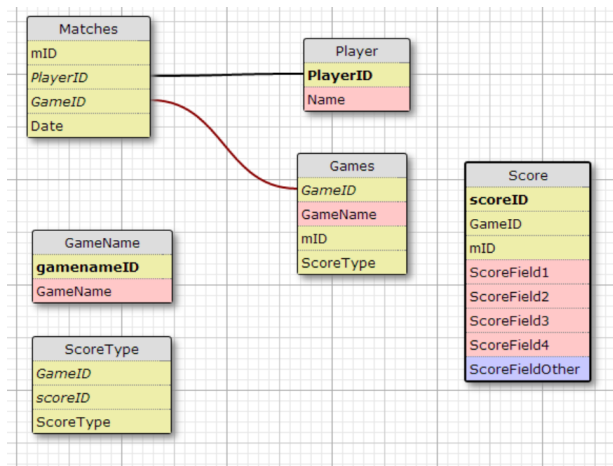
Tijdens deze module werken we ook met de programmeertaal PHP en MySQL. PHP staat voor Hypertext Preprocessor en het is een belangrijke taal voor het maken van dynamische webpagina's.

Wat PHP aan je webpagina toevoegt heeft te maken met de achterliggende server, PHP kan bijvoorbeeld documenten bewerken op de server. PHP kan eigenlijk alles wat een CGI programma kan, waarbij CGI staat voor Common Gateway Interface. Handig om te weten: CGI is eigenlijk de verbinding tussen webpagina's en databases. Hierboven wordt dus voornamelijk webpagina's genoemd. We gaan daarom ook via Unity een PHP webpagina ophalen om zo data in een database te zetten, maar ook om data uit een database te halen.



Hier zijn je score's opgeslagen

Als laatste werken we ook met MySQL. MySQL is een databasesysteem dat veel gebruikt wordt in combinatie met PHP. Een database is een digitaal opgeslagen archief gemaakt om makkelijk geraadpleegd en gebruikt te worden. SQL is een taal om vragen te stellen aan een database of om wijzigingen in een database aan te brengen. Via een SQL command (query) kan je dus code sturen naar de database om data weg te schrijven, op te halen of aanpassen. Als je hier meer over wilt weten, kan je de onderstaande bronnen bekijken.



Voorbeeld van een database structuur

## EXTERNE BRONNEN

Voor extra informatie kan je gebruik maken van de volgende links:

Wat is php? <https://www.master-it.nl/p/blog/wat-is-php>

Wat is MySQL? <https://kinsta.com/nl/kennisbank/wat-is-mysql/>

## Installeren van SQLite Database

Op digitale lesstof staat een bestand genaamd Demodb.db. Download dit bestand en **upload het naar je eigen webserver**. We hebben deze database al zo ingesteld dat er een ID, een naam en een score in opgeslagen kunnen worden. Dit gaan we doen via een PHP bestand.

## Wat is SQLite eigenlijk?

Zoals de naam al enigzins doet vermoeden is SQLite een databasesysteem dat gebruik maakt van Structured Query Language. Het mooie aan SQLite is dat het geen database server (aparte software) nodig heeft. Een SQLite database slaat al haar informatie in een bestand. Omdat je dit bestand op je schoolserver hebt gezet maken we een connectie naar de localhost.

## *Het versturen van data naar een database via PHP bestanden*

Je hebt een database op je eigen webserver gezet. Nu is het tijd om daadwerkelijk data te versturen via een PHP. Hiervoor gebruiken we losse PHP bestanden. Via deze bestanden gaan we SQL Queries sturen naar onze database. Deze week gaan we aan de slag met het versturen van data naar onze database. Volgende week gaan we data ophalen.

Maak een PHP bestand aan, noem het bestand "addscore.php" en open het bestand. Omdat jullie nog geen PHP hebben gehad krijg je de onderstaande code van ons. Neem het over en lees de code goed door. De docent zal de code verder uitleggen. **Upload dit bestand naar je webserver waar ook het database bestand staat.** Als je deze pagina aanpast en in plaats van \$name en \$score willekeurige waardes zet in de \$query zal deze data in je database komen wanneer je dit bestand opent. Bezoek de pagina niet te vaak want elke keer dat je de pagina opent komt er nieuwe data in de database. Je kan natuurlijk ook steeds de data uit je database verwijderen.

```
<?php
// Strings must be escaped to prevent SQL injection attack.
// Lees die input variabelen uit.
$name = SQLite3::escapeString($_POST["namePost"]);
$score = SQLite3::escapeString($_POST["scorePost"]);

// Maak verbinding met de database
$db = new SQLite3("DemoDB.db");
$db->busyTimeout(5000);

// Maak de SQL Query om het resultaat weg te schrijven.
$query = "INSERT INTO Highscores (ID, Naam, Score) VALUES (NULL, '$name', '$score')";

// Voer de query uit tegen de Database
$db->exec($query);

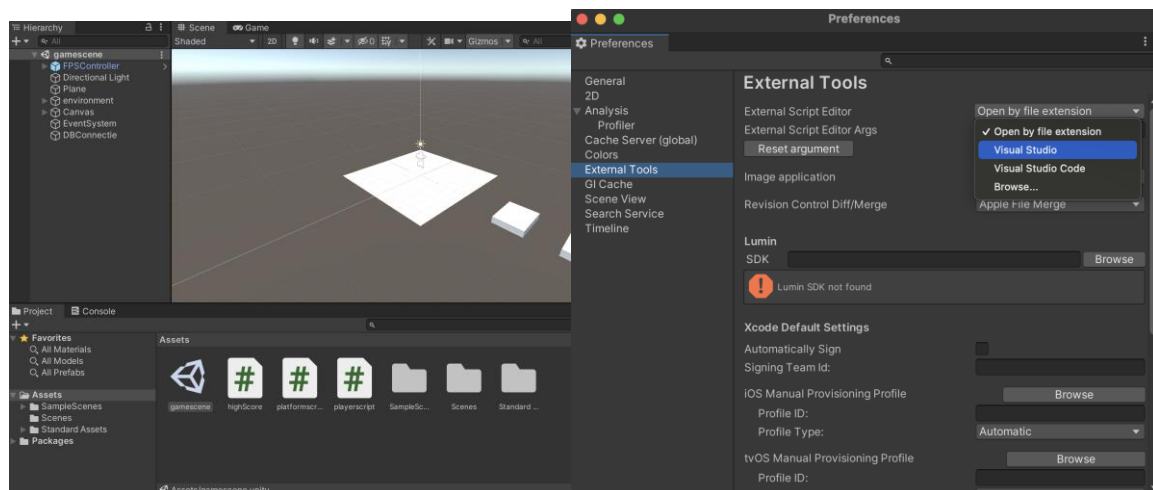
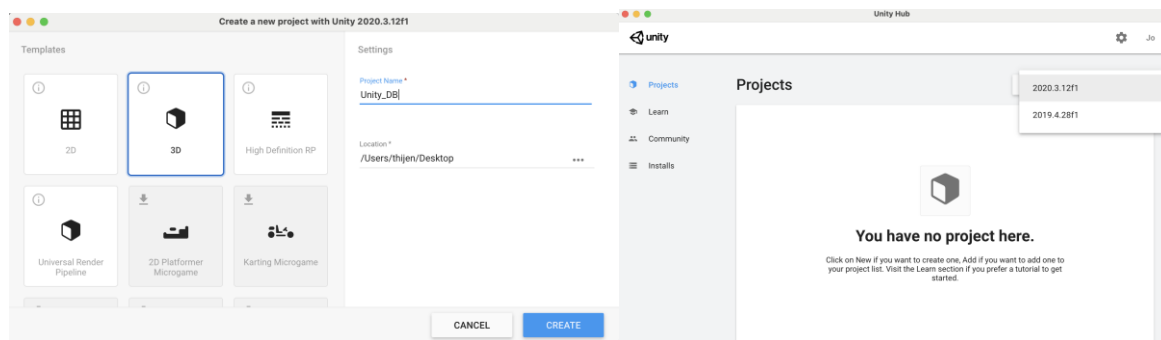
echo "Naam: ". $name . "<br>";
echo "Score: ". $score . "<br>";
```

## Het versturen van data naar een database via Unity

Nu gaan we via een Unity omgeving data naar een database sturen. Op digitaal lesstof staat een zip bestand met een game. Open dit bestand in Unity en speel het een paar keer. We gaan ervoor zorgen dat wanneer jij het einde hebt bereikt, je naam en score wordt gestuurd naar je database.

## Stappenplan aanmaken Unity3D project

- Open UnityHUB
- Maak een account aan of log in
- Maak een Unity3D project aan in versie 2020.3 en **sla het project op, op je desktop**
- Download het [GameProject.unitypackage.zip](#), unpack het bestand en dubbelklik er op. Dit zorgt ervoor dat je een simpele wereld hebt in unity.
- In het project venster zie je een bestand genaamd "Gamescreen" die open je.
- Bij Unity -> preferences -> external tools verander je dreamweaver naar Visual studio code.



### ***Data versturen naar een database via Unity***

Op dit moment werkt Unity met UnityWebRequest methodes. Met deze methodes kunnen we gebruik maken van twee belangrijke functies die we nodig hebben voor de koppeling met de database. Namelijk de UnityWebRequest.POST en UnityWebRequest.GET. Deze week behandelen we de UnityWebRequest.POST.

Met de UnityWebRequest.POST functie doen we een aanvraag naar een website URL (link) om iets in de database te zetten. In dit geval doen we een aanvraag naar het PHP bestand die we hierboven hebben gemaakt. Zorg dat het PHP bestand op je webserver staat en onthoud waar het bestand staat want dat is de URL die we gaan gebruiken binnen Unity.

### **Script aanmaken**

<https://docs.unity3d.com/ScriptReference/Networking.UnityWebRequest.Post.html>. Dit is een klein voorbeeld over hoe je een basis POST aanvraag doet binnen Unity.

Als je het project opent zie je een object in de hierachy genaamd "DBConnectie". Als de op dat object klikt zie je aan de rechterkant het highscore script. Dubbelklik op het script (bovenste veld) om het script te openen. Neem het onderstaande script over in het highscore script.

```
using UnityEngine;
using UnityEngine.Networking;
using System.Collections;

public class MyBehavior : MonoBehaviour
{
    void Start()
    {
        StartCoroutine(Upload());
    }

    IEnumerator Upload()
    {
        WWWForm form = new WWWForm();
        form.AddField("myField", "myData");

        using (UnityWebRequest www = UnityWebRequest.Post("https://www.my-server.com/myform", form))
        {
            yield return www.SendWebRequest();

            if (www.result != UnityWebRequest.Result.Success)
            {
                Debug.Log(www.error);
            }
            else
            {
                Debug.Log("Form upload complete!");
            }
        }
    }
}
```

In de start functie roepen we een StartCoroutine aan die de functie upload aanroept. Dit betekent dat er een functie wordt aangeroepen die pas iets uitvoert als er een reactie is. In dit geval doet de IEnumerator Upload functie het volgende:

- Maakt een leeg WWW formulier aan
- Voegt velden toe aan dit formulier met een waarde
- Stuurt het formulier naar de webserver
- Wacht op reactie
- Voert een console log uit met de tekst -> “Form upload is compleet” als het formulier goed verstuurd is en de data in de database komt.

### *Opdracht 1: Script compleet maken.*

De eerste opdracht van deze module is het script compleet maken. Jullie hebben klakkeloos een script overgenomen wat half werkt. Het is nu aan jullie om dit allemaal werkend te krijgen. Het volgende moet gebeuren:

- Form field moet aangepast worden. Dat is nu myField, maar in de database staat een andere benaming voor het veld. En er moet een echte naam meegestuurd worden
- Er moet een tweede field aangemaakt worden voor de score en de highscore variabele moet ook meegestuurd worden als waarde.
- De link moet aangepast worden naar jullie eigen webserver. Kijk goed naar de plek waar het phpbestand staat en neem de URL over en vervangt de url in het script.
- In het playerscript moeten regel 55 en 56 worden uncomment zodat de functie op de juiste tijd worden uitgevoerd. Wanneer de speler het einde bereikt. Je ziet meteen dat er een error komt omdat de functie niet bestaat. Verander de functie start naar “PostScore” en log de bug op die er ontstaat (de functie is niet public);

Wanneer de bovenstaande aanpassingen gedaan zijn kan je het spel testen. De scores worden pas verstuurd als je het einde van het parkour hebt bereikt (regel 55 en 56 van playerscript). In de volgende les gaat we de highscores ophalen en tonen op het scherm.



## Les week 2

Deze week gaan we aan de slag met het ophalen van scores uit de database. In de SQLite database is een tabel aangemaakt met verschillende waardes. Een ID, naam en score.

### SQLite Database

Naam	Score
<u>Johan</u>	1337
<u>Oskar</u>	1200
<u>Marien</u>	1000
<u>Adwin</u>	900

#### *Data ophalen uit een database via PHP*

Allereerst maken we een PHP bestand aan om de code te schrijven om data uit de database te halen. Maak een bestand aan met de naam displayScore.php en neem de onderstaande code over. De docent zal de code uitleggen.

```
<?php
// Maak verbinding met de database
$db = new SQLite3("DemoDB.db");
$db->busyTimeout(5000);

$query = "SELECT * FROM Highscores ORDER by Score DESC LIMIT 10";
$result=$db->query($query);

// Laat de resultaten zien net zo lang als dat er resultaten zijn
while($row=$result->fetchArray(SQLITE3_ASSOC)){
    echo " Name: ".$row['Naam']. " - Score: ".$row['Score']."\n";
}
```

Sla het bestand op en zet het op je webserver. Ook dit bestand moet in de map waar de database staat. Als je dit bestand opent zie je als het goed is de namen en scores in de browser.

---

Name: Oskar - Score: 200 Name: Johan - Score: 201 Name: Marien - Score: 178 Name: Adwin - Score: 1337

### *Data ophalen uit een database via Unity*

Om de scores op te halen nemen ongeveer dezelfde stappen als scores in de database te plaatsen. Om de scores op te halen maken we gebruik van de `UnityWebRequest.GET` methode. Bekijk de volgende link. <https://docs.unity3d.com/ScriptReference/Networking.UnityWebRequest.Get.html>

```
using UnityEngine;
using UnityEngine.Networking;
using System.Collections;

// UnityWebRequest.Get example

// Access a website and use UnityWebRequest.Get to download a page.
// Also try to download a non-existing page. Display the error.

public class Example : MonoBehaviour
{
    void Start()
    {
        // A correct website page.
        StartCoroutine(GetRequest("https://www.example.com"));

        // A non-existing page.
        StartCoroutine(GetRequest("https://error.html"));
    }

    IEnumerator GetRequest(string uri)
    {
        using (UnityWebRequest webRequest = UnityWebRequest.Get(uri))
        {
            // Request and wait for the desired page.
            yield return webRequest.SendWebRequest();

            string[] pages = uri.Split('/');
            int page = pages.Length - 1;

            switch (webRequest.result)
            {
                case UnityWebRequest.Result.ConnectionError:
                case UnityWebRequest.Result.DataProcessingError:
                    Debug.LogError(pages[page] + ": Error: " + webRequest.error);
                    break;
                case UnityWebRequest.Result.ProtocolError:
                    Debug.LogError(pages[page] + ": HTTP Error: " + webRequest.error);
                    break;
                case UnityWebRequest.Result.Success:
                    Debug.Log(pages[page] + ": \nReceived: " + webRequest.downloadHandler.text);
                    break;
            }
        }
    }
}
```

Neem **de eerste Startcoroutine functie(niet de non-existing)** uit de start functie en de **GetRequest functie** over in het highscore script. Ook om data op te halen uit een database maken we gebruik van een StartCouritine om een URL op te halen. De URL is de url naar het displayScore.php bestand. Wanneer deze URL wordt opgehaald krijgt Unity een stuk tekst terug nadat de request is gedaan. Daarom staat er een regel code “yield return webRequest.SendWebRequest. Dat betekend wacht tot dat de server heeft gereageerd en ga dan pas actie uitvoeren. Soms duurt het even voordat je antwoord krijgt van de server. De tekst die je terugkrijgt komt overeen met de tekst die je ziet als je het bestand in de browser opent. Deze data wordt in een object gestopt. Namelijk het object “webRequest.downloadHandler.text”. Eigenlijk krijg je een lap tekst terug die je kan gebruiken in de game (In de console).

### *Opdracht 2: Highscores in het spel laten zien.*

De tweede opdracht van deze module is het laten zien van de scores in de game. In het script staat een highscore tekst variabele. Dit is een tekst variabele die ook aan de linkerkant van het scherm te zien is als je het spel start. Zoek op het internet hoe je die tekst variabele kan vullen met de tekst die op hebt gehaald uit de database.

Daarbij mag de functie mag pas uitgevoerd worden als de speler het einde heeft behaald. Dus wanneer de score naar de database wordt gestuurd, moet de score ook opgehaald worden via de functie. Er mag niet meer gebruik gemaakt worden van de start functie.



## Eindopdracht

### Inleiding

Voor deze module geldt een eindopdracht. Aan deze eindopdracht werk je eigenlijk al tijdens de lessen. Aan het einde van de twee weken lever je je spel in waarbij er een naam en highscore wordt gestuurd naar de database. Daarbij is het mogelijk om de namen en highscores uit de database op te halen en te tonen binnen je spel.