

Les 1

Ton:

USER BACKEND

Web Form \$_POST -> user

User frontend (Visual) (staat iets over "web statistics")

GAME BACKEND

Rest ("like") iets.

Je krijgt JSON binnen, in game vertaal je die naar locale variabelen.

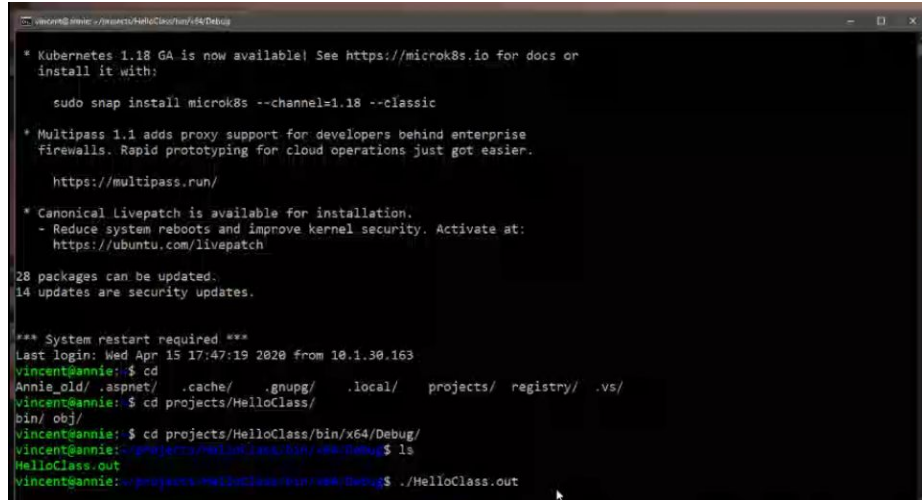
Ook info opsturen, met behulp van \$_GET (URL samenstellen)

Dat stuur je naar php file en die verwerkt t in de database.

Per les is Ton een minuutje bezig met de terminologie binnen security:

"Hashing, Salting, Hacking(Dictionary attacks, rainbow tables)"

Vincent:



```
vincent@annie: ~/projects/HelloClass/bin/x64/Debug
* Kubernetes 1.18 GA is now available! See https://microk8s.io for docs or
install it with:

    sudo snap install microk8s --channel=1.18 --classic

* Multipass 1.1 adds proxy support for developers behind enterprise
firewalls. Rapid prototyping for cloud operations just got easier.

    https://multipass.run/

* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
    https://ubuntu.com/livepatch

28 packages can be updated.
14 updates are security updates.

*** System restart required ***
Last login: Wed Apr 15 17:47:19 2020 from 10.1.30.163
vincent@annie:~$ cd
Annie_old/.aspnet/ .cache/ .gnupg/ .local/ projects/ registry/ .vs/
vincent@annie:~$ cd projects/HelloClass/
bin/ obj/
vincent@annie:~$ cd projects/HelloClass/bin/x64/Debug/
vincent@annie:~/projects/HelloClass/bin/x64/Debug$ ls
HelloClass.out
vincent@annie:~/projects/HelloClass/bin/x64/Debug$ ./HelloClass.out
```

Annie, een server van Vincent waar we tegen moeten praten

```
Microsoft Windows [Version 10.0.19608.1000]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\vince>cd Projects\HKU\HelloClassClient

C:\Users\vince\Projects\HKU\HelloClassClient>dir
Volume in drive C is Phao
Volume Serial Number is 44EC-5869

Directory of C:\Users\vince\Projects\HKU\HelloClassClient

14/04/2020  16:06    <DIR>          .
14/04/2020  16:06    <DIR>          ..
15/04/2020  17:28    <DIR>          HelloClassClient
14/04/2020  16:06                1,154 HelloClassClient.sln
               1 File(s)              1,154 bytes
               3 Dir(s)  40,909,119,488 bytes free

C:\Users\vince\Projects\HKU\HelloClassClient>cd HelloClassClient\bin\Debug\netcoreapp3.1

C:\Users\vince\Projects\HKU\HelloClassClient\HelloClassClient\bin\Debug\netcoreapp3.1>HelloClassClient.exe
Socket connected to 192.87.219.189:7331
Received: Annie says Hello!
```

Vincent zegt hoi tegen annie

Socket openen, tegen bepaald protocol praten. Als een stream naar socket al je data wegschrijven. Komt bij de server weer binnen. Annie stuurt alle ontvangen berichten door naar alle andere verbonden mensen,

Intro opdracht

- Pak je favoriete taal in je favoriete IDE
- Open een Socket
- Gebruik het TCP protocol
- Maak verbinding met annie.hku.nl
- Poort 7331
- Stuur tekstberichten
- Ontvang chatberichten

Synchronous Client Socket Example

Deze zoekt local host verbinding maar wij moeten een remote host zoeken. Wat je nu onder een socket kunt verstaan: Regelt voor jou verbinding tussen ene en andere computer. Je verbind met een andere socket eigenlijk.

Asynchroon functie Task kun je alsnog andere variabelen mee returnen door dit te doen:

```
0 references
static async Task<int> TaskAsync()
{
    return 3;
}
```

String interpolation:

\$"dit is een string en {Dit()} is een functie"

@"Dit is een multiline string en heet een literal, je kunt er tabs en enters aan toevoegen, super sick"

Ik raad je aan om te kijken naar GafferonGames? Begin met artikel "What every developer should know about nog wat..."

Bronnen:

[phpMyAdmin](#)

[Networking](#)

Les 2

Common Architectures:

Client. Server (dedicated server ergens anders (headless)) (veiligere omgeving)

Client-server, clients (server runt als een apart proces op de computer van de client (minecraft server op je pc)

Host, clients (client is de host, binnen de applicatie word het server runnen geregeld)

Three common approaches:

(Deterministic) Lock-step synchronization (slowest, least complicated)

Iedereen wacht tot de server zegt dat iedereen klaar is

Server verzamelt input van iedereen, als alles goed is zegt die "yo je mag 1 tick verder"

Erg traag want als 1 iemand kut ping heeft moet je op hem wachten.

Client/server(+ client-side prediction) (most common)

Server is baas over alles, maar client gaat proberen te voorspellen wat de server voor antwoord gaat geven. Bijv: client zegt ik ga schieten, client doet eerste bericht naar server of t mag, maar doet ook zelf gelijk al een check of t kan en wat er dan gebeurt.

Belangrijkste is dat server de baas is, hierdoor kan de client niet cheaten.

Distributed simulation + authority scheme (less common, more complicated)

Iedereen heeft owner ship maar het verwisselt wie t heeft. Sommige games word dit gebruikt

Synch strategies:

Authority vs Ownership

Bal overgooien, als speler moet je de baas zijn over de staat van dat object (controle op uitoefenen zodra je hem pakt. Als je m weggooit word t deel van simulatie krijgt server autoriteit. Als iemand anders pakt heb jij t weer. Er is een ding (server of clienthost) heeft **Ownership** en de **Authority** kan niet worden afgepakt. Ownership is ook vaak op dingen zoals een player avatar, want niemand kan die aanpassen behalve de speler.

Rule of thumb

Client is niet te vertrouwen, altijd iemand anders de baas laten zijn.

Socket/port

- Poort naar buitenwereld
- Socket word geopend op een "poort"
Wat binnen komt via poort gaat naar applicatie en komt binnen op een socket.
- Verstuur data packets door een socket
- Ontvang data packets door een socket.

Geen enkele poort is beter dan een andere poort, er zijn er veel omdat je computer moet weten naar welke applicatie de data verstuurd moet worden. Connectie is: connecten met ip adress voor computer > via een poort naar applicatie (zijn afspraken voor, https poort 434, zijn 1000 geregistreerde poort nummers waar je als applicatie niet aan mag komen (1000 tot 65000 poorten).

Als je 2 applicaties door zelfde poort runt zal de 2e niet toegang krijgen tot dat poort.

Port Forwarden: er zitten beveiligings lagen tussen je pc en internet. 1 daarvan is je router. Data van buitenwereld weet niet anders dan je publieke ip adres. Port Forward betekent dat je data gaat forwarden op basis van wat er binnen komt op een meegegeven poort.

Data packets:

Verzameling bytes.

- Verzender adres/port
- Bestemming adres/port
- Packet grootte
- Packet data

Kan onderweg opgepakt worden. (data sniffers draaien op je pc en pakken packets en checken die op hun content)

Protocolen

UDP(Unreliable) (connectie-loos) (User Datagram Protocol) (**data pakket is datagram**)

*Kent geen connectie, kent geen verbinding, verstuurt gwn pakketjes en boeit niet of t aankomt.
Kan dus vertraagd raken, onderschept worden. Mallicious of uberhaupt door ruis op de lijn.
Pakketjes verloren en in wellekeurige volgorde aankomen.*

Werkt t zelfde als t internet. Is daar bovenop gebouwt. Kan t op zn minst splitsen in pakketjes.

TCP (Reliable) (connectie-based) (Transmission Control Protocol)

Gaat ervan uit dat er een mutual verbinding is. Zorgt er voor dat ze in tact arriveren en in goeien volgorde. Klinkt fantastisch en super nuttig maar fakka we gebruiken UDP want zo werkt Unity en is gwn snel.

TCP handshake (Three - way handshake)

- (1) Synch
- (2) Synch / acknowledgement
- (3) Acknowledgement is afgehandeld

Duurt lang dus stuur nog een synch, blijven wachten op reactie. Als pakket 3 wel binnen is maar 2 nog niet en vorige was 1. Voordat 3 doorstuurt gaan we wachten op 2. Ookal is er nieuwe data gaan we wachten op de juiste volgorde. Na een bepaalde tijd niks meer binnen zegt die "aight er is geen verbinding meer"

*TCP is dus al slomer maar is ook meer data nodig (zit in header **SLIDE 11**)*

Sequence number en Acknowledgement number zijn de 2 belangrijkste extra

S = bericht nummer 237

A = afhankelijk van operating system (zou kunnen dat ze allebei ander nummer meekrijgen)

Besturings systeem maakt deze header dus hoef je niet zelf te maken.

Stel je wilt het half doen en TCP alleen voor belangrijke data kan dat niet want. Als je TCP blijft hangen heeft dat invloed op snelheid van UDP ontvangen en versturen en kan mogelijk ook blokkeren.

Het UDP gevaar moet je in de appllicatie zelf oppakken

(bijv. timestamp plakken op iedere message) (bijv. iedere beweging van character, verstuurt naar server met dit tijdstip, dan kan de server ze op basis van tijd gaan uitvoeren en als er dan een message binnen komt die ouder is kun je die skippen want je hebt al nieuwe data.)

Lock-step Age of Empires voorbeeld

Alleen mouseclick werd gestuurd en iedereen runt zn eigen simulatie.

Strategy game met 8 spelers en 200 population (1600 archers) ziek veel data wat je doorstuurt en ook op oude technologie dus dan moet je wel Lock-step gebruiken.

Nadelen:

Traag synchroon houden en server heeft niet absolute autoriteit. Game waarin dit gebeurt 100% deterministisch == dezelfde input geeft altijd dezelfde input.

Het hele spel ligt stil als iemand leaved omdat ze dan niet de approval van die speler krijgen.

Matchmaking

Slide 14 - Ronimo artikel

Voorbeeld van hoe t werkt:

- Spelers komen in een lobby server
- Lobby server verzamelt spelers (op basis van strategy (filter op land en console ofzo)
- Room word gecreerd
- Spelers joinen de room
- Spel word gestart

Waarom zo?

- Firewalls / router setup / nog iets kijk slide 16
- User friendly
- Centrale management is efficient
- Beveiliging

Unity networking in a nutshell

Unity heeft eigen dunne laag op UDP wat illusie van connectie maakt zonder TCP te gebruiken.

Network driver (UNITY CLASS)

Centrale uitgangspunt. Client en server maakt er 1.

Maak driver aan die verantwoordelijk is voor connectie en readers en writers.

Vormt de brug tussen applicatie en het netwerk. Soort van manager. Komen de connectie aanvragen binnen en word een connectie gemaakt.

Network connection (UNITY CLASS)

Cient maakt er 1 server maakt er 1 per client

Writers, Readers

Slide 19 write

Slide 20 read/client

Slide 21 read/server

Add = push

Pop = element

[Installing com.unity.transport | Unity Transport | 0.3.0-preview.6](#)

Doe deze dingen:

- **Workflows**
 - Creating a minimal client and server
 - Creating a jobified client and server

Voor volgende les: probeer hiermee een character te maken die je kunt bewegen op beide connecties.

Kijk uit. Oude systeem heet: UNITY TRANSPORT LAYER

Geen Photon want:

Hoop word geautomatiseerd, je ontwerpt meer dan t systeem maken. Deze les moeten we juist het systeem leren te begrijpen.

Les 3

https://home.hku.nl/~ton.markus/kern_gdv/

SQL = Structured Query Language

SQL(i), i = improved (toen ze dit maakten was sql dik 15 jaar oud)

Een bibliotheek is ook een database

Zorgen dat er geen verwarring is en dat alles zo efficiënt mogelijk is opgeslagen = hoekstenen van een database.

Soorten koppelingen:

One on one:

1 boek, 1 persoon

One on many:

1 schrijver, meerdere boeken geschreven

Many on one:

Meerdere mensen 1 boek geschreven

Many on many:

Meerdere mensen, meerdere boeken geschreven

Voor ons:

De speler kan meerdere scores hebben, maar de score kan maar 1 speler hebben.

Highscore opslaan is erg onnodig omdat je dat gewoon kunt berekenen van alle opgeslagen scores.

Datetime opslaan is vrijwel onmisbaar als je nuttige informatie uit je data wilt halen.

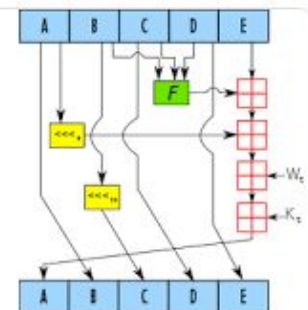
Beveiliging:

SHA

De **SHA**-familie (Secure Hash Algorithm) is een verzameling gerelateerde cryptografische hashfuncties ontworpen door de Amerikaanse National Security Agency en gepubliceerd door het Amerikaanse National Institute of Standards and Technology.

nl.wikipedia.org > wiki > SHA-familie ▾

[SHA-familie - Wikipedia](https://nl.wikipedia.org/wiki/SHA-familie)



AI = Auto Increment = hij telt automatisch op: bijv: 1, 2, 3, 4. Etc.

(Dit stuurde geoffrey door: <https://www.pixijs.com/>)

Les 4

Brute Force:

Dictionary Attack:

Inelligente broertje van brute force

MRMMM: meer resultaat met minder moeite

Besparing ten opzichte van Brute Force door alleen “goed te onthouden” woorden te testen.

Rainbow tables:

Slaat hash met password na berekenen op in een lijst en brute forced met deze table. Je moet hierin een balans vinden tussen opslagruimte en rekenkracht, maar tegenwoordig is opslag zo goedkoop dat het steeds een meer viable method word.

Salting:

Verdediging tegen zwakste schakel. Salting is het toevoegen van een random sequence aan het password voordat het gehasht wordt.

Salt wordt los opgeslagen samen met hash.

PHP regelt salting in de achtergrond zelf.

SQL voorbeelden:

```
SELECT score FROM scores WHERE score > 25 AND score < 50
```

```
SELECT score,date_time FROM scores WHERE user_id = 2 AND score > 25
```

```
SELECT * FROM users WHERE first_name = "mark"
```

Opdracht 3a

Scores onder 100:

```
SELECT score FROM scores WHERE score < 100
```

Alle scores onder de 100 van player 2

```
SELECT score FROM scores WHERE user_id = 2 AND score < 100
```

Alle scores boven de 100 van game 3

```
SELECT score FROM scores WHERE game_id = 3 AND score > 100
```

Hoogste score van game 5:

```
SELECT MAX(score) FROM scores WHERE game_id = 5
```

Hoogste score van game 5 van player 5

```
SELECT MAX(score) FROM scores WHERE game_id = 5 AND user_id = 5
```

Opdracht 4

Connect naar de database met php code. (als je het wilt includen moet het alsnog een .php bestandstype zijn, als je het een .inc maakt ziet iedereen je wachtwoord en gebruikersnaam!!!)

Fetch_assoc = associative array

Sessie: (zie afbeelding in mapje les 6)

Zie t als een hotel waar je 1 keer identificeert en een kamer sleutel krijgt, daarna kun je zonder identificeren naar je kamer met de sleutel (sleutel is sessie_id) en je kunt dus niet in andere "kamers". **(de database die wij gebruiken behoud de sessie id een half uur voordat die m weer weggooit (tenzij je hele tijd refreshed))**. In Unity moet je zelf de sessie id opslaan en meesturen (cookie).

Les 5

Layers

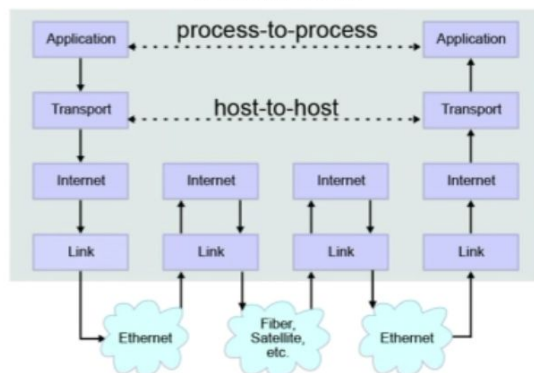
Netwerk functionaliteit in een computer bestaat uit een aantal losse "lagen" die aan elkaar gekoppeld (op elkaar gestapeld) worden

Losse lagen zorgen voor flexibiliteit

Applicaties (bovenst laag) kunnen gebruik maken van de lagen eronder zonder zich af te vragen hoe het werkt

Netwerk kaart vervangen wil zeggen dat alleen de driver voor de netwerk kaart (onderste laag) vervangen hoeft te worden

Data Flow



Transport (Transport Control Protocol) laag zorgt voor verpakking en verzending

Internet (Internet Protocol) laag zorgt voor adressering (het IP-adres van een specifieke computer).

TCP & IP

Login system

Login systeem bestaat uit:

- login pagina
- inschrijf/aanpas pagina
- (zorgen dat overige pagina's indien niet ingelogd doorverwijzen naar login pagina)



Sanitizen

Php functie "filter_var"

Er is een aparte filter voor email adressen

We gaan de password checken op alle vereisten (geen rare tekens, minimaal aantal tekens, etc.)

Login pagina moet maar 2 dingen kunnen terug sturen

1 voor "deze user bestaat"

0 voor "deze user bestaat niet"

https://home.hku.nl/~ton.markus/kern_gdv/index.php?les=7&content=3

Les 6? (iig van ton)

Public-Private key

1 public key voor encoden en 1 private key voor decoden. Deze hebben beide partijen. (Net als brieven naar elkaar sturen in 2 verschillende talen (hebreeuws en hongaars).