
Team 12

Frédéric Blondeel
Dirk Vanbeveren
Bert Van den Bosch

Inhoudsopgave

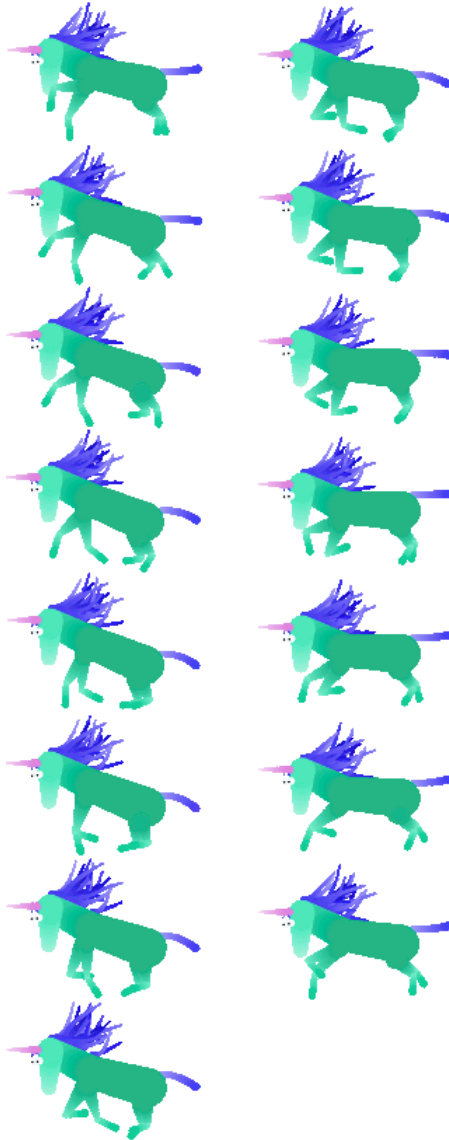
1	Introductie	2
2	Testen	2
2.1	HTTP request frames	2
3	Besluit	3

1 Introductie

2 Testen

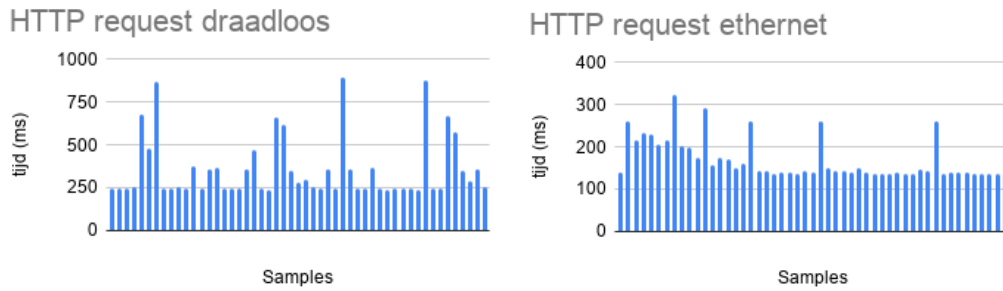
2.1 HTTP request frames

Voor deze test werd gebruik gemaakt van een sprite sheet met 15 frames op (figuur 1) en een animatie snelheid van 30 fps.



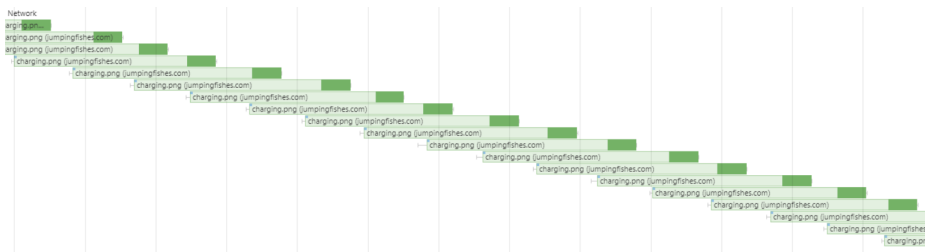
Figuur 1: Sprite sheet met 15 frames

Bij de eerste test wordt de hele sprite sheet per frame ingeladen. Dit is om het effect te creëren dat de server elke frame genereert en doorstuurt naar de client. Dit bleek zeer inefficiënt te zijn doordat de tijd om de frame op te halen met ethernet rond de 140 ms en met draadloos internet rond de 250 ms ligt (figuur 2a en figuur 2b).

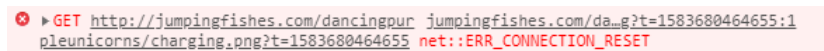


(a) HTTP request met draadloze verbinding (b) HTTP request met ethernet verbinding

Hoewel de frames asynchroon worden ingeladen (figuur 3) is de animatie alles behalve ideaal. Door lange wachttijden zijn er veel frame skips en door de vele http requests wordt de verbinding gereset om de server niet te overbelasten met 30 requests per seconde (figuur 4).

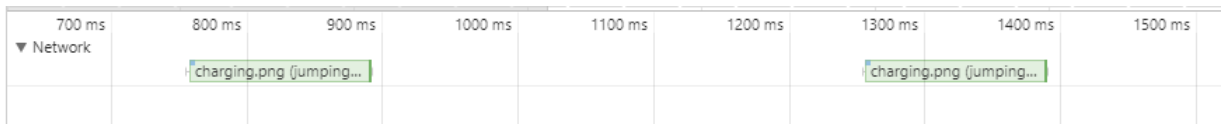


Figuur 3: Inladen van elke frame



Figuur 4: Connection reset door teveel requests

Bij de tweede test wordt gebruik gemaakt van een buffer. De sprite sheet wordt om de 0.5 seconden ingeladen zodat elke frame tijd had om te displayen ($30 \frac{\text{frames}}{\text{sec}} \frac{1}{15 \text{frames}} = \frac{0.5}{\text{sec}}$). Dit creëert het effect dat de server 15 frames genereerd en doorstuurt in 1 keer i.p.v. frame per frame. Dit is duidelijk een verbetering want de server krijgt 15 keer minder requests (figuur 5) en er zijn geen frame skips meer. Na het inladen van de sheet zal de client de 15 frames achter elkaar drawen zonder te hoeven wachten op de server. Daarna zal de volgende sheet (of buffer) gebruikt worden.



Figuur 5: Inladen van de buffer

3 Besluit

Referenties