

# Vergelijkende studie van 3D capaciteiten bij studenten laptops met betrekking tot virtual reality

Brian Pinsard, Jovi De Croock, Thomas Vansevenant, Dieter Willems

## Samenvatting

Tekst...

Tekst....

## Keywords

Keyword1 — Keyword2 — Keyword3

## Inhoudsopgave

<b>1</b>	<b>Introductie</b>	<b>1</b>
<b>2</b>	<b>State-of-the-art</b>	<b>1</b>
<b>3</b>	<b>Methode van aanpak</b>	<b>1</b>
3.1	Keuze van software . . . . .	1
	Furmark • 3DMark vs Unigine • 3DMark	
3.2	3DMark: Sky Diver . . . . .	1
	Grafische testen • Physics test • Gecombineerde test	
3.3	Workflow . . . . .	2
	Benchmark via command line • Converteren van resultaten • Vergelijking van de resultaten	
<b>4</b>	<b>Resultaten</b>	<b>2</b>
<b>5</b>	<b>Conclusies</b>	<b>2</b>

## 1. Introductie

Tekst...

## 2. State-of-the-art

Tekst...

## 3. Methode van aanpak

### 3.1 Keuze van software

Na we onze onderzoeksvraag afgebakend hadden, zijn we ons onderzoek gestart met het zoeken van tools die beschikbaar zijn voor het testen van de visuele capaciteiten van een computer. Hierbij sprongen de volgende benchmarking tools uit: FurMark, Unigine en 3DMark.

#### 3.1.1 Furmark

Furmark is een zeer intensieve OpenGL benchmark die vooral als stabiliteit en stress test gebruikt wordt, hierbij wordt grafische kaart tot zijn uiterste geduwd en kan deze oververhitten. Dit was dus toch geen correcte tool voor onze doeleinden. (oZone3D, 2009)

### 3.1.2 3DMark vs Unigine

Tussen 3DMark en Unigine hebben we gekozen om 3DMark te gebruiken als benchmarking tool. De reden hiertoe is dat 3DMark een zeer bekende tool is en tegenwoordig er ook een gratis versie beschikbaar is via het Steam platform, dit heeft ervoor gezorgd dat ieder groepslid de zelfde versie van 3DMark geïnstalleerd staan had.

### 3.1.3 3DMark

3DMark bevat verscheidene benchmarks die elk gericht zijn op een specifieke klasse van hardware dit varieert van smartphones tot gaming computers. Deze benchmarks worden gebruikt door honderden hardware review websites en verscheidene van de hardware fabrikanten.

Uit de benchmarks die 3DMark beschikbaar stelt hebben we gekozen voor de benchmark “Sky Diver”, deze is geschikt voor gaming laptops tot en met mid-range computers. We hebben voor deze benchmark gekozen omdat dit overeen komt met ons standaard gebruik van de laptops die getest worden. De rendering resolutie staat bijvoorbeeld vast op 1920x1080 bij een benchmark geschikt voor een lagere klasse van apparaten staat deze resolutie vast op 1280x720, de resultaten van die benchmark zouden geen representatieve weergave tegenover daadwerkelijk gebruik van de laptop.

### 3.2 3DMark: Sky Diver

Binnen de benchmark “Sky Diver” zijn er vier testen en een demo die uitgevoerd kunnen worden, wij hebben gekozen om de vier testen uit te voeren omdat de demo niet relevant is tot de eindscore van deze benchmark.

#### 3.2.1 Grafische testen

De eerste grafische test maakt gebruik van de forward lighting method met één directioneel licht. Deze traditionele methode gaat elk object apart gaan oproepen en per lichtbron in de scene deze visualiseren terwijl de belichting berekend wordt, dit wil zeggen dat deze methode heel duur is qua performantie. Ook wordt er een depth of field effect toegepast als nabewerking. (Futuremark, 2016; De Vries, 2015)



De tweede grafische test focust zich op pixel processing en gebruikt een compute shader-based deferred tile lighting method. Het verschil met de eerste grafische test qua belichting ligt hem in de methode waarbij de objecten nu eerst allemaal gevisualiseerd zullen worden en pas daarna de belichting voor de volledige scene berekend zal worden. Bij deze test wordt er ook een lens reflectie effect toegepast als nabewerking. (Futuremark, 2016; De Vries, 2015)

### 3.2.2 Physics test

Terwijl de grafische testen vooral de GPU zullen belasten en testen, zal de physics test de CPU testen door 96 standbeelden als objecten neer te slaan met een hamer. Deze hamers hangen vast aan kettingen waardoor de CPU immens veel zal moeten bereken zoals hoe de hamers moeten zwaaien en wanneer de hamers een standbeeld raken hoe de stukken van het standbeeld moeten vliegen. Deze 96 standbeelden worden verdeeld over 4 levels, elk level wordt er significant meer standbeelden tegelijk gevisualiseerd en geraakt door één van de hamers, zodat de CPU naarmate de tijd verstrijkt meer zal moeten verwerken. (Futuremark, 2016)

### 3.2.3 Gecombineerde test

Deze test is een combinatie van de tweede grafische test en het derde level van de physics test. Er wordt gebruik gemaakt van een compute shader based deferred tiled lighting method zoals in de tweede grafische test en er worden 48 standbeelden geladen en omgezwaaid door hamers zoals in het derde level van de physics test. De test is ontworpen om een gebalanceerde belasting uit te voeren op zowel de GPU als de CPU. (Futuremark, 2016)

## 3.3 Workflow

We hebben gekozen om de bovenstaande vier testen op ieder groepslid zijn laptop 30 keer te laten draaien, het formaat waarin 3DMark standaard de resultaten exporteerd is helaas niet bruikbaar aangezien we de resultaten er niet kunnen uit halen. Hierdoor hebben we gebruik moeten maken van de command line tools om de resultaten te exporteren naar een bestand van het XML formaat, dit bestand was simpel te converteren met een zelf geschreven Java programma naar een bestand van het XLS formaat zodat we met de resultaten konden beginnen werken.

### 3.3.1 Benchmark via command line

Voor de benchmark via command line te laten draaien hebben we allereerst een custom XML setting bestand aangemaakt waarin de testen worden aangegeven die gedraaid moeten worden en er ook ingesteld wordt dat de benchmark systeem info moet verzamelen. Hiervoor hebben we onderstaand commando gebruikt:

---

```
3DMarkCmd.exe
--definition=skydiver_custom.3dmdef
--export=skydiver_naam.xml --loop=30
```

---

Het commando hebben we opgesteld gebaseerd op de commandos die te vinden zijn in de 3DMark Command Line Guide door Futuremark (2014).

### 3.3.2 Converteren van resultaten

Voor het converteren van het XML bestand naar een XLS bestand hebben we een eigen java programma geschreven dat gebruik maakt van de SAX Parser om het XML bestand om te zetten naar standaard Java objecten en daarna Apache POI om deze objecten weg te schrijven naar een XLS bestand. Het programma kan gebruikt worden via het volgende commando:

---

```
java -jar 3dmark_results_converter.jar
skydiver_naam.xml
```

---

Waar skydiver\_naam.xml het input bestand is, na dit commando uit te voeren zal het programma vragen om een output bestandsnaam te geven. Het XLS bestand met de resultaten wordt nadien gegenereerd.

### 3.3.3 Vergelijking van de resultaten

Bij ieder groepslid zijn resultaten worden de gemiddelden genomen van iedere test zijn 30 testresultaten en deze worden dan verder gebruikt voor...

## 4. Resultaten

Tekst...

## 5. Conclusies

Tekst...

## Referenties

- De Vries, J. (2015). Learn opengl: Deferred shading. verkregen op xx april, 2016 van <http://learnopengl.com/#!Advanced-Lighting/Deferred-Shading>.
- Futuremark (2014). 3dmark command line guide. verkregen op xx april, 2016 van <http://s3.amazonaws.com/download-aws.futuremark.com/3DMark-CommandLineGuide.pdf>.
- Futuremark (2016). 3dmark technical guide. verkregen op xx april, 2016 van [http://s3.amazonaws.com/download-aws.futuremark.com/3DMark\\_Technical\\_Guide.pdf](http://s3.amazonaws.com/download-aws.futuremark.com/3DMark_Technical_Guide.pdf).
- oZone3D (2009). Furmark: Gpu stress test and opengl benchmark. verkregen op xx april, 2016 van <http://www.ozone3d.net/benchmarks/fur/>.