Gamotica

De zoektocht naar de juiste game engine

RoboScholars

Auteurs:

Cordell Stirling(s1154605)

Jelle Smelt(s1159072)

Jeremy de Groot(s1114990)

Rick Stoffer(s1162883)

Jacco Kalter(s1158783)

Jesper Kiers(s1156710)

**Versiebeheer**

|  |  |  |  |
| --- | --- | --- | --- |
| Versie | Datum | Omschrijving | Opmerkingen |
| 0.1 | 18-02-2021 | Opmaak en taakverdeling en meta informatie ingevuld |  |
| 0.2 | 19-02-2021 | Abstract en Introductie van bestand geschreven en methodes beschreven. Eerste deelvraag onderzocht en grootste delen uitgewerkt. |  |
| 0.3 | 24-02-2021 | Feedback opdrachtgever toegepast (deelvraag specificatie en volgorde) deelvraag 1 opnieuw geschreven |  |
| 0.4 | 16-02-2021 | Resultaten van deelvraag 4 en 5 |  |
| 0.5 | 19-03-2021 | Laatste componenten gecontroleerd voor engines, en de ingebouwde IDE en tutorials vergeleken van de engines |  |
| 0.6 | 23-03-2021 | Abstract aangevuld en prototyping deelvraag beantwoord. |  |
| 1.0 | 26-03-2021 | Kwaliteit controle |  |
| 1.1 | 19-04-2021 | Feedback verwerkt |  |

**Distributie**

|  |  |  |  |
| --- | --- | --- | --- |
| Naam | Rol | Datum | Versie |
| Cordell Stirling | Teammanager | 26/03/2021 | 1.0 |
| Jacco Kalter | Teamlid | 26/03/2021 | 1.0 |
| Jelle Smelt | Teamlid | 26/03/2021 | 1.0 |
| Jesper Kiers | Teamlid | 26/03/2021 | 1.0 |
| Rick Stoffer | Teamlid | 26/03/2021 | 1.0 |
| Jeremy de Groot | Teamlid | 26/03/2021 | 1.0 |

**Goedkeuring**

|  |  |  |  |
| --- | --- | --- | --- |
| Naam | Rol | Datum | Paraaf |
| Wille Conen | Opdrachtgever | 26/03/2021 |  |
|  |  |  |  |

Inhoudsopgave

[**1.** Abstract 1](#_Toc67669814)

[**2.** Introductie 2](#_Toc67669815)

[**2.1** Onderwerp 2](#_Toc67669816)

[**2.2** Context 2](#_Toc67669817)

[**2.3** Doelstelling 2](#_Toc67669818)

[**2.3.1** Hoofdvraag 2](#_Toc67669819)

[**2.3.2** Deelvragen 2](#_Toc67669820)

[**3.** Methoden 4](#_Toc67669821)

[**3.1** Literatuurstudie 4](#_Toc67669822)

[**3.1.1** zoekstrategie 4](#_Toc67669823)

[**3.1.2** betrouwbaarheid 4](#_Toc67669824)

[**3.2** Prototyping 4](#_Toc67669825)

[**4.** Resultaten 5](#_Toc67669826)

[**4.1** Wat is een game engine en waarom zou je die gebruiken? 5](#_Toc67669827)

[**4.2.** Welke engine componenten zijn allemaal nodig voor een Top-Down Shoot ’em up? 5](#_Toc67669828)

[**4.3.** Welke Java game engines zijn beschikbaar? 6](#_Toc67669829)

[**4.5**. Welke Java game engines hebben alle nodige componenten? 7](#_Toc67669830)

[**4.6**. Hoeveel kosten de licenties van elke Java game engine? 7](#_Toc67669831)

[**4.7.** Welke Java game engines zijn te gebruiken door HBO-ICT studenten? 8](#_Toc67669832)

[**4.7.1** Ardor3D 8](#_Toc67669833)

[**4.7.2** Jake2 8](#_Toc67669834)

[**4.7.3** Java 3D API 8](#_Toc67669835)

[**4.7.4** jMonkeyEngine 9](#_Toc67669836)

[**4.7.5** LibGDX 10](#_Toc67669837)

[**4.7.6** PlayN 10](#_Toc67669838)

[**4.7.7** DimensioneX 11](#_Toc67669839)

[**4.7.8** LITIENGINE 12](#_Toc67669840)

[**4.8.** Welke game engines zijn het meest positief beoordeeld door hun gebruikers? 13](#_Toc67669841)

[**4.8.1** Ardor3D 13](#_Toc67669842)

[**4.8.2** Jake2 13](#_Toc67669843)

[**4.8.3** Java 3D API 13](#_Toc67669844)

[**4.8.4** jMonkeyEngine 14](#_Toc67669845)

[**4.8.5** LibGDX 15](#_Toc67669846)

[**4.8.6** PlayN 16](#_Toc67669847)

[**4.8.7** DimensioneX 17](#_Toc67669848)

[**4.8.8** LITIENGINE 18](#_Toc67669849)

[**4.9.** Met welke engine konden wij het makkelijkst een redelijk demo ontwerpen? 18](#_Toc67669850)

[**4.9.1. jMonkeyEngine** 19](#_Toc67669851)

[**4.9.2. LibGDX** 20](#_Toc67669852)

[**4.9.3. Java 3D API** 20](#_Toc67669853)

[**5.** Discussie 21](#_Toc67669854)

[**6.** Literatuurlijst 22](#_Toc67669855)

# Abstract

In dit document bespreken wij al het onderzoek dat wij als projectgroep hebben uitgevoerd met betrekking tot de Java game engines en het gebruik hiervan in games. Het onderzoek begint met stellen van een hoofdvraag in de introductie. De hoofdvraag voor het onderzoek is “Welke Java game engine kun je het beste gebruiken tijdens het ontwikkelen van een Top-Down Shoot ’em up game?”. Om een antwoord te kunnen geven op de hoofdvraag hebben wij deelvragen opgesteld die ons helpen de hoofdvraag te beantwoorden. Om een goed onderzoek af te leveren is het belangrijk dat de informatie die we gebruiken betrouwbaar is. In het hoofdstuk “methodes” worden de strategieën beschreven die we daarvoor gebruiken. Een onderdeel hiervan is de literatuurstudie. Hierin beschrijven we hoe we tot zoekresultaten komen en leggen we vragen vast die ons helpen de informatie en bronnen te verifiëren. Bij het onderdeel prototyping testen we de gekozen engines op de werking en kunnen we dus zien welke engines wel of niet geschikt zijn voor het project. In hoofdstuk 4 geven we antwoord op de deelvragen aan de hand van de literatuurstudie die we beschreven hebben in de methode. De antwoorden op de deelvragen worden toegelicht in de discussie, waarin we met deze antwoorden een conclusie vormen over welke engine het meest geschikt is voor een Top-Down Shoot ‘em up game. Uit ons onderzoek blijkt dat de LibGDX engine de meeste voordelen heeft, en daarmee ook het meest geschikt is voor ons project.

# Introductie

## Onderwerp

In dit document wordt het onderzoek dat is uitgevoerd met betrekking tot Java game engines vastgelegd. We gaan eerst alle opties bekijken die beschikbaar zijn en vergelijken waarmee we uiteindelijk met behulp van deze vergelijking een conclusie trekken tot welke game engine voor ons project Gamotica het best past.

## Context

Eén van de eindproducten die opgeleverd moet worden aan onze opdrachtgever aan het eind van dit project is een simpele game die gebruik maakt van een Arduino of Raspberry Pi als gamecontroller. Hierbij is het de bedoeling dat we gebruik maken van een al bestaande game engine om de game tot werkelijkheid te brengen. Wel was één van de eisen dat er met Java gecodeerd werd dus moeten wij specifieker zijn in ons onderzoek en alleen game engines met als programmeertaal Java zoeken.

Om de beste keuze te maken over welke game engine wij gaan gebruiken is dit onderzoek opgesteld. Met behulp van dit onderzoek en de documentatie kunnen wij onze uiteindelijke keuze onderbouwen en vastleggen.

## Doelstelling

Met behulp van dit onderzoek willen wij de kennis die nodig is om een game engine te kiezen vastleggen. Hierbij gaan we de verschillende game engines vergelijken, met als uiteindelijke doel onze hoofdvraag te beantwoorden die in de volgende paragraaf beschreven wordt.

### Hoofdvraag

Voor de game die wij gaan ontwikkelen is het nodig om een game engine te gebruiken, om te bepalen welke wij gaan gebruiken moeten we een antwoord vinden op de vraag:

**Welke Java game engine kun je het beste gebruiken tijdens het ontwikkelen van een Top-Down Shoot ’em up game?**

### Deelvragen

Om tot een antwoord te komen op onze hoofdvraag gaan we die eerst verdelen onder een aantal deelvragen. De resultaten op deze deelvragen zouden alle informatie moeten bevatten die je nodig zou hebben om de hoofdvraag te kunnen beantwoorden.

Het eerste waar we naar moeten kijken is wat een game engine nou precies is en waar die voor gebruikt wordt, de vraag die wij daarvoor stellen is:

**Wat is een game engine en waarom zou je die gebruiken?**

Hierna moeten we weten welke componenten nodig/handig zijn tijdens het maken van een game in ons genre, wij stellen hiervoor de vraag:

**Welke engine componenten zijn nodig voor een Top-Down Shoot ’em up?**

Daarnaast moeten wij ook kijken welke engines allemaal beschikbaar zijn die gebruik maken van de programmeertaal Java, dit doen we door de vraag te stellen:

**Welke Java game engines zijn beschikbaar?**

Hierna gaan we kijken welke engines de componenten die wij nodig hebben beschikbaar hebben, om dit te kunnen beantwoorden vragen wij:

**Welke Java game engines hebben alle nodige componenten?**

Het volgende wat wij willen weten is welke Java game engine wij kunnen betalen met ons budget. We moeten dus de prijzen van de verschillende engines vergelijken met behulp van de vraag:

**Hoeveel kosten de licenties van elke Java game engine?**

Nog een belangrijk deel voor ons vanwege onze ervaring met game development is het gebruikersgemak van de game engine. We moeten ervoor zorgen dat wij genoeg kennis en ervaring hebben om de game engine daadwerkelijk te gebruiken tijdens het ontwerpen door deze vraag te stellen:

**Welke Java game engines zijn te gebruiken door programmeurs met de ervaring van HBO-ICT studenten.**

Iets wat wij kunnen bekijken om te bepalen welke keus wij moeten maken is welke game engine het meest positieve gebruikers heeft in de game industrie. Hiermee kunnen wij bepalen of een game engine in de werkelijkheid toepasbaar en gewild is. De beste manier om dit te bepalen is de vraag:

**Wat zijn de beste Java game engines met betrekking tot hoeveel gebruikers positief waren vergeleken met negatief?**

Als laatst gaan we met behulp van de vorige deelvragen een paar engines kiezen om een kleine demo te ontwerpen en bepalen welke engine het gemakkelijkst was om mee te werken.

**Met welke engine konden wij het makkelijkst een redelijke demo maken?**

# Methoden

## Literatuurstudie

Een van de methodes die wij gebruiken voor het uitvoeren van het onderzoek is de literatuurstudie. Hierbij kijken we naar welke zoekwoorden we gaan gebruiken (de zoekstrategie) en controleren de bronnen op de betrouwbaarheid van de inhoud.

### **3.1.1** zoekstrategie

Om een antwoord te vinden op de deelvragen is het belangrijk om goede zoektermen te gebruiken waardoor je goede informatiebronnen kan vinden. Voor de deelvragen hebben we gebruik gemaakt van de volgende zoektermen, waarbij we Google gebruikt hebben als search-engine:

* Game development
* Java
* game engines
* Etc.

Voor de zoektermen hebben wij vooral gebruik gemaakt van Engelse termen omdat er in die taal meer informatie beschikbaar is over de game engines.

### **3.1.2** betrouwbaarheid

Om de betrouwbaarheid van de verschillende bronnen te checken worden er een aantal vragen gesteld waar de bron aan moet voldoen om als een betrouwbare bron te worden beschouwd.

Deze vragen bestaan uit:

* Wat is het doel van de informatiebron?
* Bevat de bron feiten of meningen?
* Is de bron toepasbaar?
* Is de bron herzien?
* Voldoet de informatie aan de informatiebehoefte?
* Is de auteur een hedendaagse deskundige op het vakgebied?
* Heeft de auteur meer geschreven over het onderwerp?
* Is er een overzicht van gebruikte bronnen?
* Bevat de bron de full-tekst of is het een samenvatting of verwijzing?
* Hoe recent is de informatie?
* Is de informatie nog geldig of is deze verouderd?
* Zijn er inmiddels nieuwe inzichten over het onderwerp?

Wanneer er positief antwoord uit de vragen komen over een bron, word een bron als betrouwbaar beschouwen en word de informatie van de bron gebruikt in het onderzoek.

## Prototyping

Hiervoor hebben we besloten om de meest passende oplossingen die uit de deelvragen zijn gebleken te gebruiken om een kleine demo te ontwerpen. Dan gaan we bepalen welke demo het gemakkelijkst was om te ontwerpen of dat we überhaupt een demo kunnen ontwerpen met de gekozen game engine. Uiteindelijk komt hieruit onze definitieve keuze voor het volledige project.

# Resultaten

## Wat is een game engine en waarom zou je die gebruiken?

Het eerst wat wij hebben onderzocht met behulp van de literatuurstudie is wat een game engine is en waarom hij gebruikt wordt tijdens game development. De eerste zoekterm die wij hebben gebruikt is: “What is a game engine?”. Het eerste resultaat “how game engines work” door: (Baker, 2016) is een erg algemene uitleg over wat een game engine is en heeft een aantal voorbeelden van veel gebruikte engines. Hierin wordt verteld dat een game engine verschillende gereedschappen biedt aan game ontwikkelaars om het ontwikkelen van een game wat sneller en gemakkelijker te maken. Een game engine bestaat uit vijf componenten: de main programma die de game logica bevat; Een weergave engine die gebruikt kan worden om geanimeerde 3D graphics te generen; een audio engine die bestaat uit algoritmes met betrekking tot geluid; een physics engine die gebruikt wordt om natuurkundige wetten te implementeren; en als laatst een kunstmatige intelligentie.

Met betrekking tot vertrouwelijkheid komt dit artikel van een nieuws website met als auteur iemand die nog nooit in de game industrie heeft gewerkt. Dit betekent dat wij niet alles 100% kunnen vertrouwen wat er werd gezegd maar omdat het vrij veel in de buurt kwam met wat wij in ons gedachten hadden willen wij ook niet alles afschrijven. Ook zijn alle andere vragen over vertrouwelijkheid positief beantwoord dus is het grotendeels wel te vertrouwen maar er moet wel nog een bron gevonden worden die wel gemaakt is door iemand in de industrie.

Hierna hebben wij een andere pagina met dezelfde zoekterm gevonden en zijn we beland op de website van Gamescrye (gamescrye, 2019). Gamescrye is een organisatie die betrekking heeft tot de games industrie en zelf ook software produceren, de auteur is niet vernoemd bij het artikel maar vanwege de organisatie situatie zelf is deze bron wel wat vertrouwelijker dan die van (Baker, 2016). In het artikel van Gamescrye worden een aantal dingen wat helderder uitgelegd en waren een aantal aannames van (Baker, 2016) verkeerd bewezen. De eerste zijn de aantal componenten, voor elke engine is dit anders dus zijn er ook andere functies beschikbaar per verschillende engine, de enige component dat altijd aanwezig is in een engine is een game loop of op zijn minst een 2D/3D weergave engine. Ook werden er door Gamescrye een aantal voordelen voor het gebruik van een game engine vernoemd: sneller prototyping; sneller begrip krijgen op game ontwerp concepten; veel gebruikte geavanceerde technologieën worden voor je klaargezet; support voor meerdere platforms zonder gedoe.

## **4.2.** Welke engine componenten zijn allemaal nodig voor een Top-Down Shoot ’em up?

Om dit te onderzoeken hebben wij als eerste zoekterm “game engine components for shoot 'em up” gebruikt. Hiermee hebben wij met toeval een rapportage van (Abargues, 2019) gevonden, Zij zijn Spaanse universiteit studenten die voor een project een Shoot ‘em up hebben gemaakt. Voor deze deelvraag was er maar een klein onderdeel hieruit te gebruiken en dat was het feit dat ze Unity 3D engine gebruikten. Dit betekent dat wij ook niet 3D engines hoeven buiten te sluiten tijdens het kiezen van een engine. Dit rapport bevat veel informatie die van toepassing zijn voor ons project en zullen hierop dus hoogstwaarschijnlijk terugkomen. Omdat het project dat is uitgevoerd in ongeveer dezelfde scenario is als bij ons konden we erop vertrouwen dat zij hetzelfde niveau hadden als wij hebben.

Daarna hebben we nog een project gevonden van een universiteit student uit Spanje genaamd (Remeseiro, 2017). Deze gaat in hoofdstuk 3 verder in op de componenten van de engine die zij gebruikt hebben tijdens het maken van een 2D shoot ‘em up. Hieruit hebben wij een aantal componenten gehaald die wij ook nodig hebben tijdens het maken van onze shoot ‘em up: SpriteRenderer, Camera, Canvas, Scene graph, UI, Audio, ParticleSystem, Sprite, Texture, Effects engine, Collision detection en Transform. De net vernoemde onderdelen waren allemaal in Unity beschikbaar en werden gebruikt voor het maken van een shoot ‘em up door een student, het is dus te vertrouwen dat wij een eind vooruit kunnen komen als wij deze componenten hebben.

Als laatst hebben we de term “game engine components for 2D shoot 'em up” opgezocht en waren we beland op een artikel van een game maker die gebruik maakte van de stage3D component van flash (Kaitila, 2012). De schrijver heeft uitgebreide ervaring in het veld van games en documenteert hier het hele proces voor het maken van een 2D shoot ‘em up. Hierin gebruikt hij de stage3D rendering engine component beschikbaar in flash. Flash wordt niet meer ondersteund, maar we kunnen hieruit wel halen dat zelfs game makers met ervaring gebruik maken van een 3D rendering engine voor 2D games. We moeten dit dus goed in gedachten houden en geen 3D uitsluiten.

## **4.3.** Welke Java game engines zijn beschikbaar?

Hiervoor hebben wij simpel weg “Java game engines” gebruikt als zoekterm en dan zoveel mogelijk lijsten die tevoorschijn kwamen met elkaar vergeleken. Google geeft dan zelf bovenin 40 resultaten weer, na wat meer onderzoek van die 40 resultaten was eruit gebleken dat niet alle resultaten hiervan gebruik maken van Java.

Uiteindelijk zijn we beland op (moddb community, 2021), hieruit is gebleken dat er momenteel 713 game engines bestaan waarvan 485 al beschikbaar zijn op de markt. Dus hebben we hieruit alsnog geen lijst van game engines die gebruik maken van Java, en alle 485 engines onderzoeken om te bepalen welke wel of niet Java gebruiken viel buiten de scope van dit onderzoek.

Nadat wij erachter zijn gekomen dat de aantal game engines veel groter is dan wij eerst dachten en dat er niet echt een gemakkelijke lijst is met alle game engines die gebruik maken van Java hebben wij ervoor besloten om als begin punt de wikipedia pagina van game engines te gebruiken. (Wikipedia, 2021) Hieruit zijn acht engines gekomen met als hoofd programmeer taal Java. Hierna hebben we zoektermen zoals best Java Game engine gebruikt om nog game engines te vinden die niet voorkwamen in de wikipedia pagina. Dit zijn de Java game engines die wij als beschikbaar beschouwen voor de scope van dit project.

* [Renanse/Ardor3D: Ardor3D is a free Java based, professionally oriented, open source 3D graphics engine. (github.com)](https://github.com/Renanse/Ardor3D)
* [Jake2 download | SourceForge.net](https://sourceforge.net/projects/jake2/)
* [Java 3D API (oracle.com)](https://www.oracle.com/java/technologies/javase/java-3d.html)
* [jMonkeyEngine](https://jmonkeyengine.org/)
* [libGDX](https://libgdx.com/)
* [PlayN](http://playn.io/)
* [DimensioneX MMORPG Game Engine](https://www.dimensionex.net/)
* [LITIENGINE 🎮 Free and Open Source Java 2D Game Engine](https://litiengine.com/)

## **4.5**. Welke Java game engines hebben alle nodige componenten?

Hiervoor hebben we de componenten die wij hadden onderzocht tijdens paragraaf 4.2 in een tabel gezet tegenover de acht game engines die we in 4.3 hadden onderzocht. En dat voor elk engine bekeken of het wel of niet de component beschikbaar heeft.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Component | Game engine | Ardor3D | Jake2 | Java 3D API | jMonkeyEngine | libGDX | PlayN | DimensioneX | LITIENGINE |
| SpriteRenderer | | **X** | **X** | **X** | **X** | **X** | **X** |  | **X** |
| 2D Rendering engine | | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** |
| 3D Rendering engine | | **X** | **X** | **X** | **X** | **X** | **X** | **X** |  |
| Audio engine | |  | **X** | **X** | **X** | **X** | **X** | **X** | **X** |
| Scene graph | | **X** |  | **X** | **X** | **X** | **X** | **X** |  |
| Texture renderer | | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** |
| UI | | **X** |  | **X** | **X** | **X** |  | **X** | **X** |
| Canvas | | **X** |  | **X** | **X** | **X** | **X** | **X** | **X** |
| Camera | | **X** | **X** | **X** | **X** | **X** |  |  |  |
| Visual effects engine | | **X** |  | **X** | **X** | **X** | **X** |  |  |
| Collision detection | | **X** |  | **X** | **X** | **X** |  |  | **X** |

Om de componenten van de engines te vinden hebben we de website van de engine zelf bekeken en voor sommige engines hebben we de files bekeken om te bepalen wat ze wel/niet hebben. Hieruit is gebleken dat jMonkeyEngine en de Java 3D API alle componenten hebben die wij van tevoren hebben bepaald dat wij nodig hebben. We gaan de anderen niet buitensluiten voor nu, maar dit is wel een groot voordeel voor deze 3 engines en ook Ardor3D die alleen die audio mist waardoor het ook erg in aanmerking komt.

## **4.6**. Hoeveel kosten de licenties van elke Java game engine?

Om binnen de scope van ons project te blijven hebben wij alleen de prijzen en licentie kosten van de game engines die in 4.2 genoemd waren bekeken. Om dit te doen hebben wij simpelweg de website van elke game engine bekeken. Ook hebben wij gebruik gemaakt van de website (GNU, 2021), dit is een betrouwbare bron om alle licentie types te kunnen controleren.

Ardor3D is afgeleid van de game engine jMonkeyEngine en maakt gebruik van de zlib licentie, dit betekent dat het open-source is en dus gratis gebruikt mag worden.

Jake2 is vastgebonden aan een GPL-licentie, dit betekent weer dat er geen kosten aan vast zitten om er gebruik van te maken.

Java3D is vastgebonden aan een BSD-licentie, hierbij is weer geen kosten aangebonden.

jMonkeyEngine heeft dezelfde licentie regeling als Java3D engine.

LibGDX engine is gemaakt door apache en valt ook onder de Apache License versie 2.0. Aan deze license worden weer geen kosten gebonden.

PlayN is open-source en heeft dus geen licentie. Dit betekent natuurlijk ook dat er geen kosten bij komen voor het gebruik maken van deze engine.

DimensioneX is ook een open-source engine dus er zijn geen kosten aan vastgekoppeld bij het gebruik van de engine.

LitiEngine is net zoals de anderen ook open-source onder de MIT-licentie, dit houdt weer in dat er geen kosten aan vast zit tijdens het ontwerpen van een game met behulp van de engine.

Alle informatie die hier werd gevonden met betrekking tot licenties werd gevonden op websites van de makers van de engines zelf met wat hulp van de lijst die gebruikt werd in 4.2. (Wikipedia, 2021). De betrouwbaarheid van de resultaten hiervan zijn dus goed te vertrouwen omdat de informatie direct ontvangen was van de makers van de game engines.

## **4.7.** Welke Java game engines zijn te gebruiken door HBO-ICT studenten?

Om dit te bepalen hebben wij van alle 8 engines de ingebouwde IDE vergeleken en de beschikbaarheid van tutorials en uitleg bekeken.

### **4.7.1** Ardor3D



Voor Ardor3D is er geen ingebouwde IDE en dus veel moeilijker te gebruiken voor diegene met minder ervaring, ook zijn er weinig tutorials over te vinden. Dit komt waarschijnlijk doordat de originele maker er niet meer aan werkt en verschillende projecten eruit zijn gekomen waardoor het moeilijk te bepalen is welke engine je precies gebruikt. Er is 1 uitgebreide tutorial te vinden over de jogamps versie van Ardor3D. (goussej, 2014)

### **4.7.2** Jake2

Jake2 heeft geen ingebouwde IDE voor de engine, het maakt gebruik van code. Er is weinig informatie te vinden over het installeren en het gebruiken van Jake2, op internet zijn geen tuturials te vinden over de engine. Deze engine is dus niet geschikt voor (beginnende) studenten, omdat het enorm lastig is om Jake2 te installeren en te gebruiken.

### **4.7.3** Java 3D API

Voor Java 3D API is er niet echt een ingebouwde IDE. Het is meer in code die je dan met je bestaande IDEA, zoals ItelliJ IDEA dan kunt gebruiken door de bijbehorende libraries te importeren. Er zijn verder ook een redelijk aantal tutorials te vinden die het gebruik ervan uitleggen. Het installeren is makkelijk te vinden en de tutorials leggen de informatie wel duidelijk uit. Het is geschikt voor (beginnende) studenten die ook al een beetje de basis van Java weten.

### **4.7.4** jMonkeyEngine

JMonkeyEngine heeft zijn eigen SDK met een ingebouwde IDE. Hiermee kunt u op een simpele manier met uw code omgaan en de resultaten meteen bekijken. De IDE is vrij ingewikkeld met zijn user interface maar heeft niet veel verschillen vergelijken met ons meest bekende IDE IntelliJ. Ook is er uitgebreide documentatie over de game engine te vinden online zoals hieronder gezien kan worden.

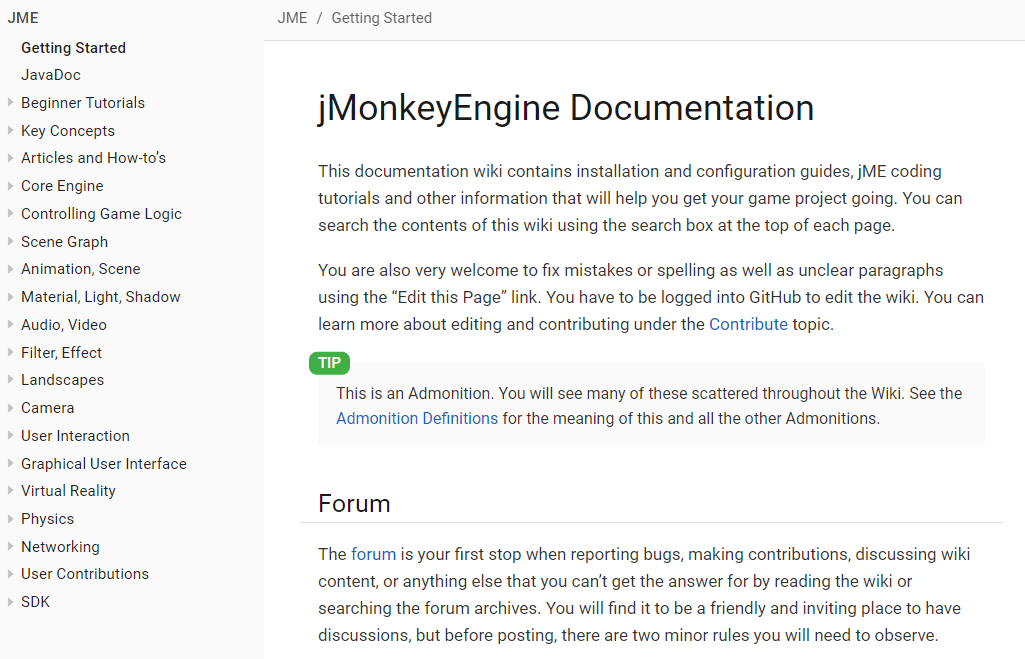


Figure https://wiki.jmonkeyengine.org/docs/3.3/documentation.html

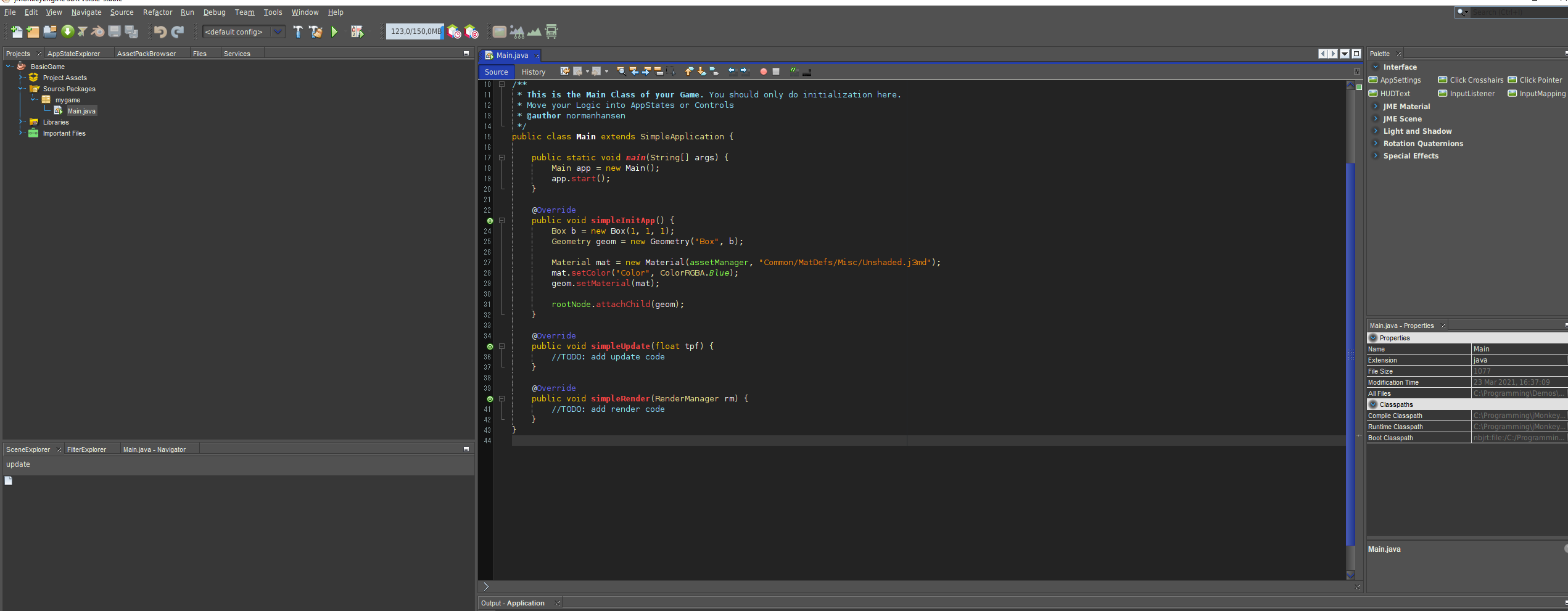
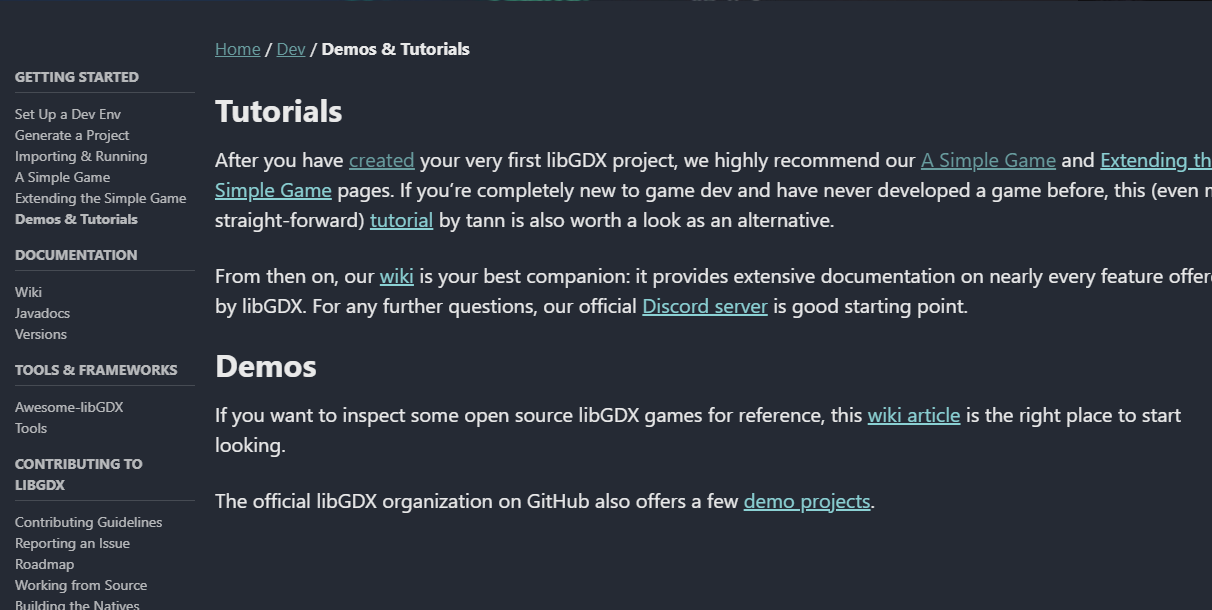


Figure 2 jMonkeyEngine SDK

### **4.7.5** LibGDX

LibGDX heeft ook geen ingebouwde IDE, de documentatie beschikbaar ervoor is aardig uitgebreid maar vergeleken met jMonkey mist er nog wat. Er zijn online wel aardig wat tutorials te vinden maar weer lang niet zo veel als bij sommige van de andere keuzes.

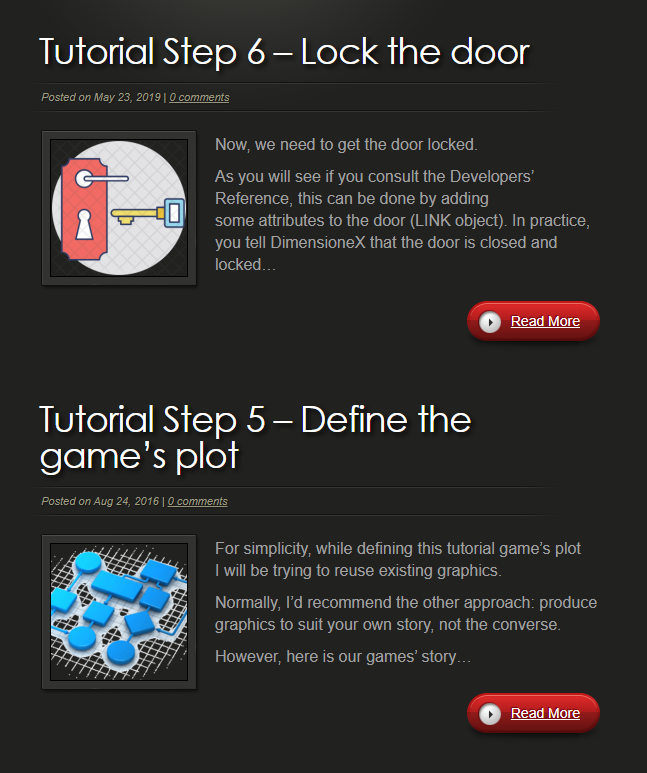


### **4.7.6** PlayN

PlayN heeft geen GUI en maakt gebruik van Maven of een Java IDE. Op de website van PlayN is een tutorial te vinden die je stap voor stap helpt met het maken van een game met PlayN. Daarnaast zijn er nog demo’s te vinden van games die gemaakt zijn met PlayN. Ook is er documentatie te vinden over hoe je de development omgeving opzet met Maven. Een instructie om het met een IDE zoals Eclipse of IntelliJ op te zetten wordt niet duidelijk gegeven. Over het algemeen is PlayN een prima engine die je ook als beginnende student zou kunnen gebruiken voor het maken van simpele games.

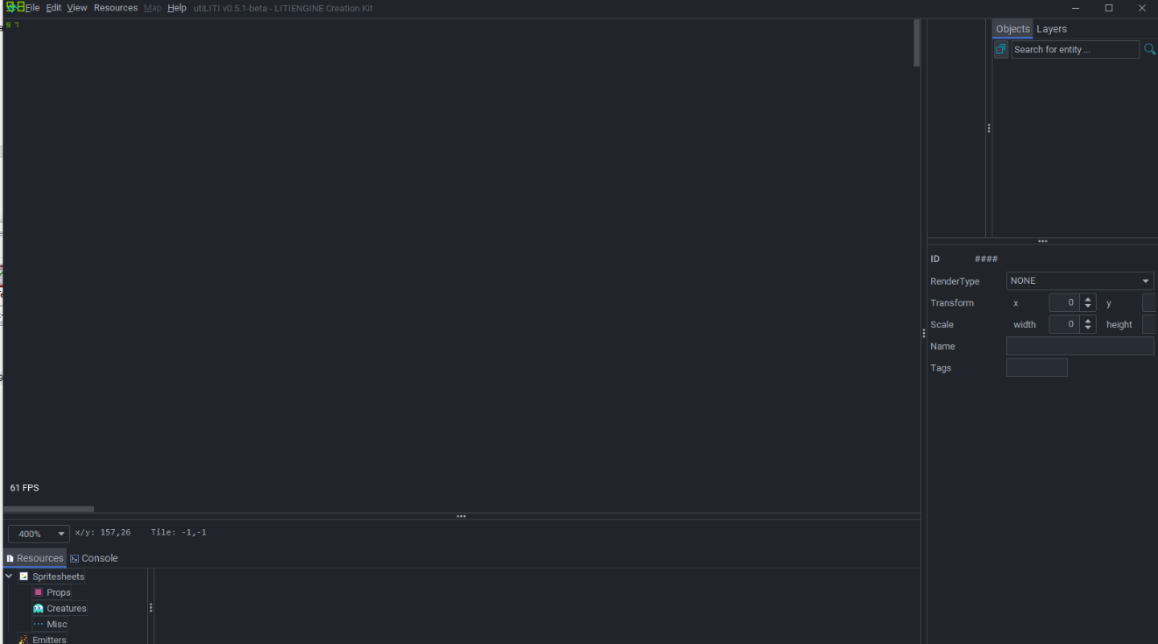
### **4.7.7** DimensioneX

Heeft net zoals de rest geen ingebouwde IDE, ook is er weinig te vinden in verband met uitleg en tutorials. Hieronder is te zien dat op de website van dimensioneX zelf maar 6 tutorials staan en online was er weinig te vinden over een tutorial voor installatie.

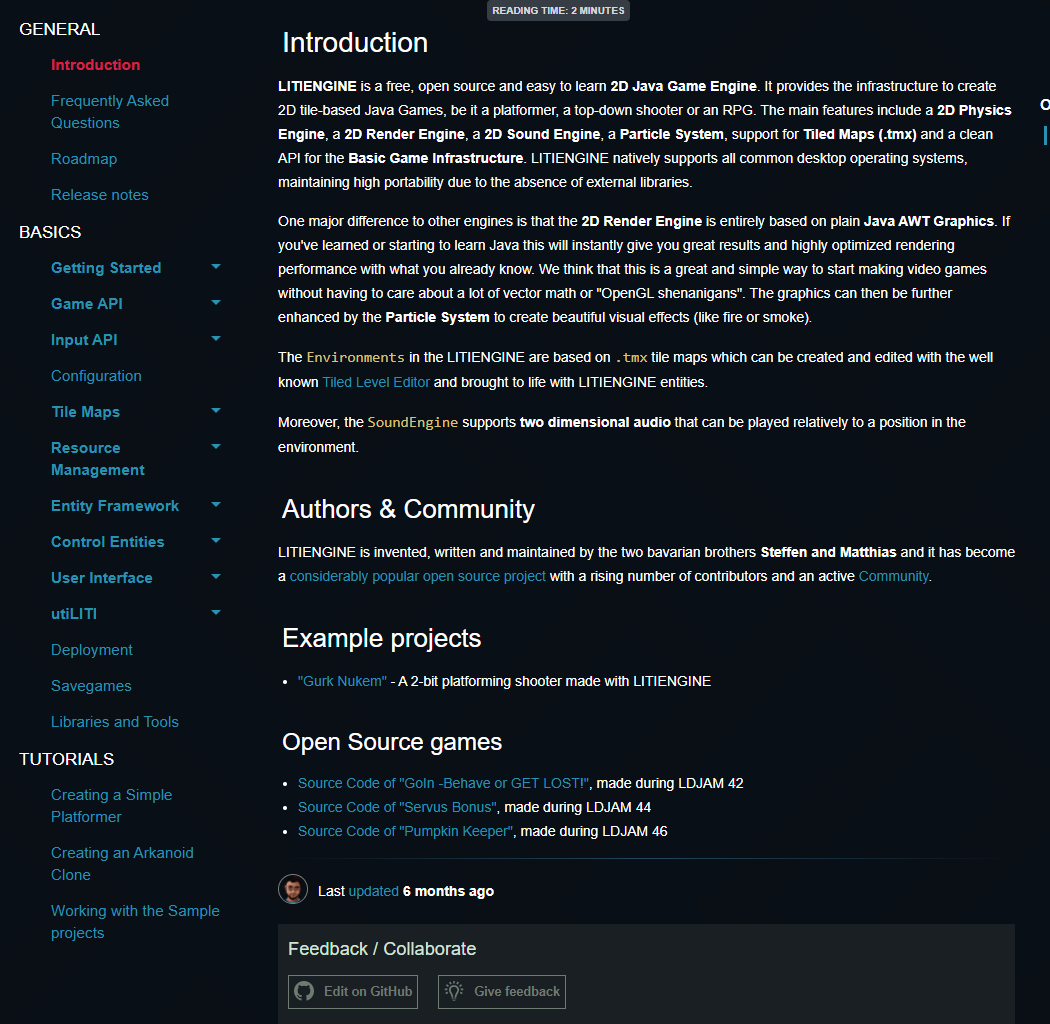


### **4.7.8** LITIENGINE

Zoals je hieronder kunt zien zijn er veel resources beschikbaar om te leren omgaan met de LITIENGINE en het heeft een redelijk overzichtelijk ingebouwde IDE die te gebruiken is door een student die nog leert. Er zijn ook tutorials beschikbaar op de website van LITIENGINE zelf er is dus zelfs meer te vinden waarschijnlijk. Hierdoor kun je de conclusie nemen dat LITIENGINE goed bruikbaar is voor Java programmeurs met weinig ervaring. Hieronder is het scherm te zien die er gebruikt zou moeten worden met de LITIENGINE. Ook is de extensieve documentatie daaronder te zien die wij kunnen gebruiken tijdens het gebruik van de engine.



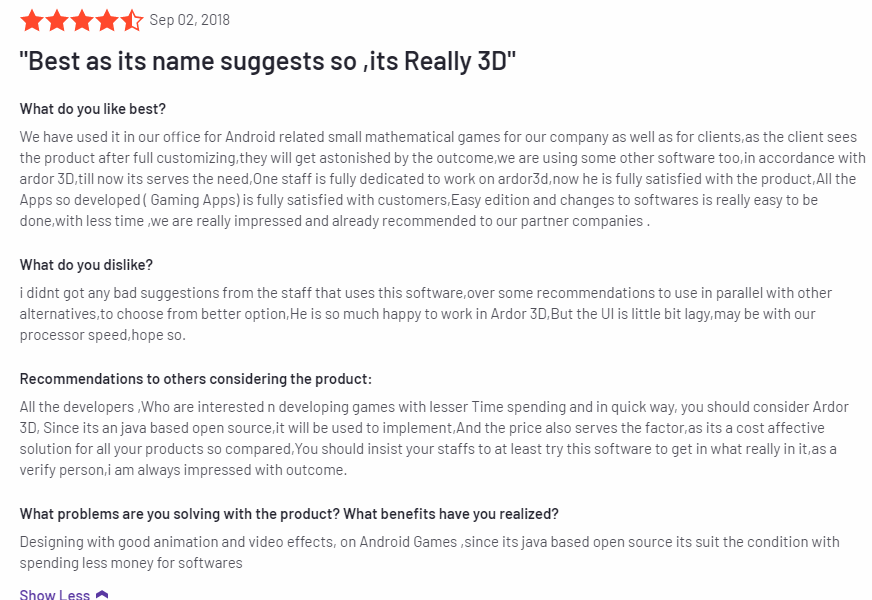
Figuur 1: afbeelding van het LITIENGINE IDE



Figuur 2: afbeelding van hun website waar je onderaan de tutorials ziet

## **4.8.** Welke game engines zijn het meest positief beoordeeld door hun gebruikers?

### **4.8.1** Ardor3D

Voor Ardor3D hebben wij 1 review gevonden die erg positief was, maar het probleem met Ardor3D is dat we niet weten welke branch is gereviewd. 

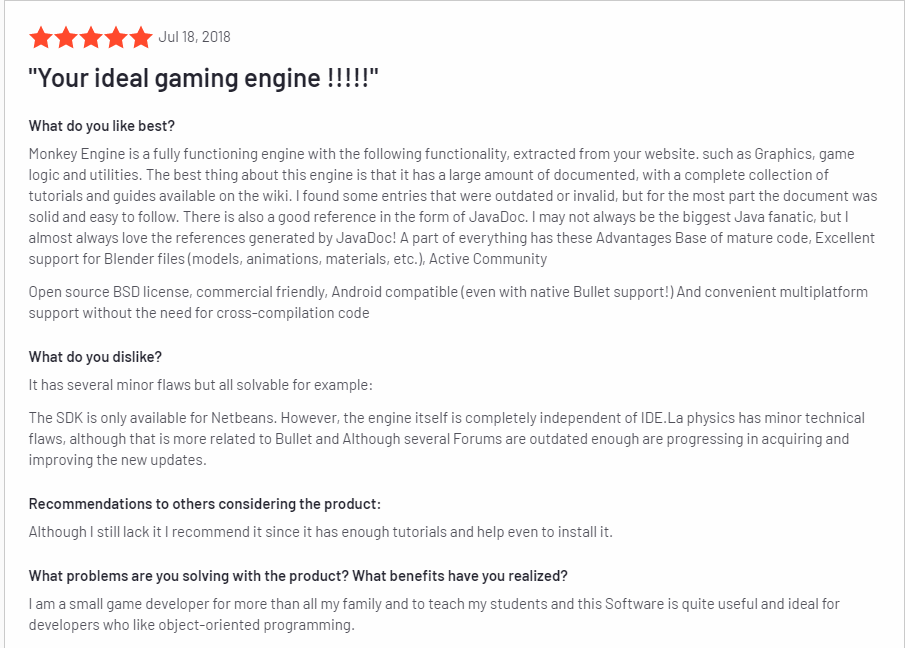
### **4.8.2** Jake2

Voor Jake2 konden wij geen reviews vinden, voor Jake2 in het algemeen is het erg lastig om informatie erover te vinden.

### **4.8.3** Java 3D API

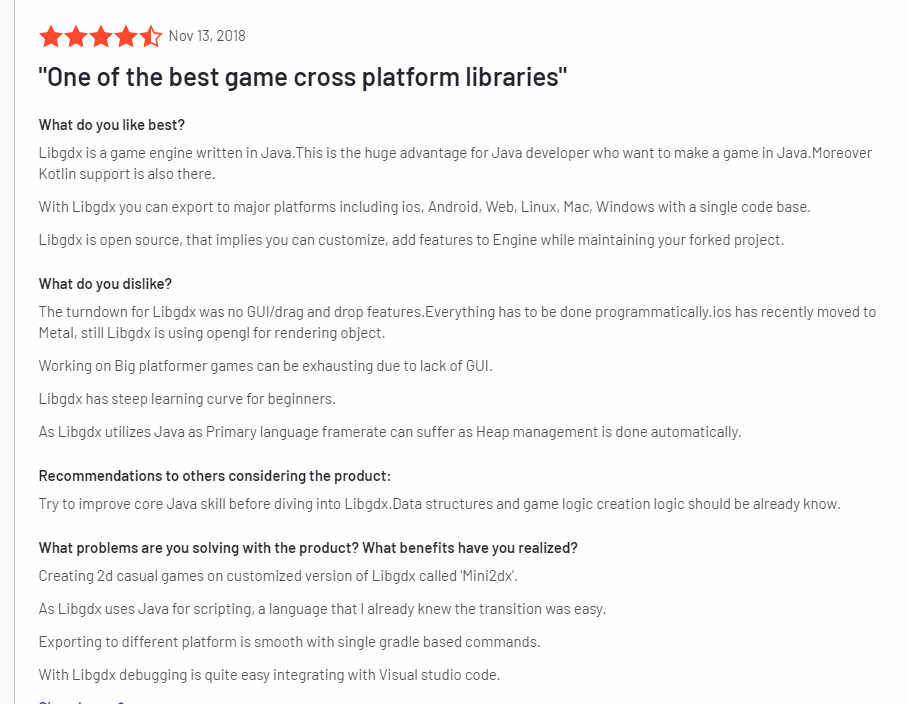
Voor Java 3D API zijn weinig tot geen reviews te vinden.

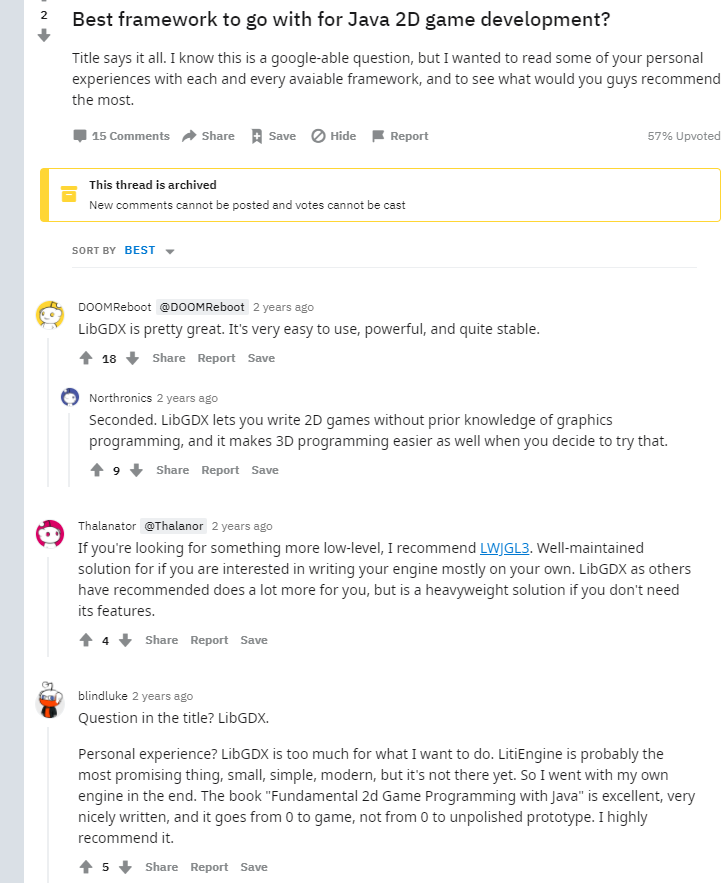
### **4.8.4** jMonkeyEngine

Voor jMonkeyEngine hebben we 3 reviews gevonden met als gemiddelde 4,7, er werden weinig negatiefs erover gezegd behalve kleine foutjes en dat het misschien een beetje ouder is dan modernere oplossingen. 

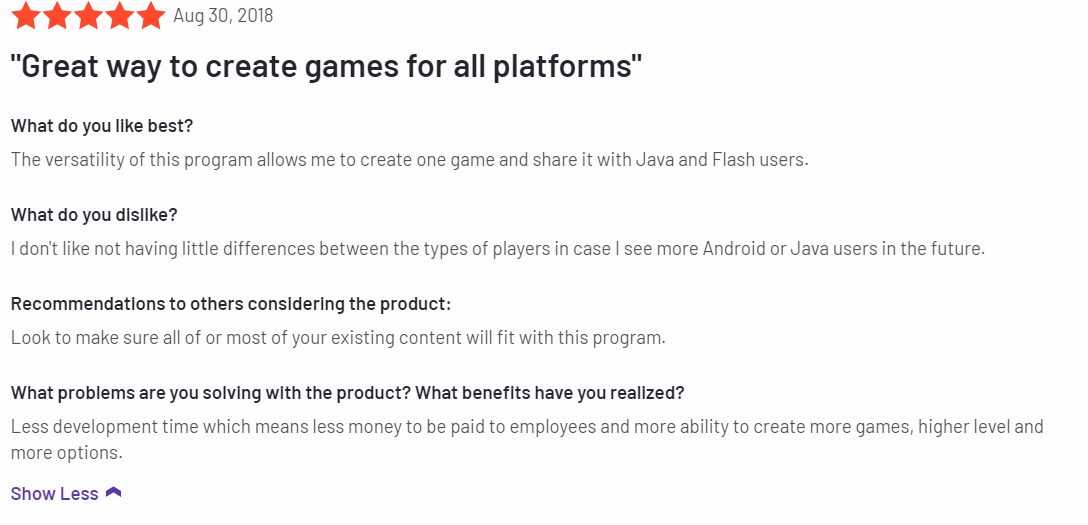
### **4.8.5** LibGDX

Over LibGDX is er veel meer te vinden vergeleken met de rest en over het algemeen zijn de meeste gebruikers er erg blij ermee. Online op chatforums wordt deze zelfs erg vaak aanbevolen voor beginner programmeurs, wat hieronder te lezen is.



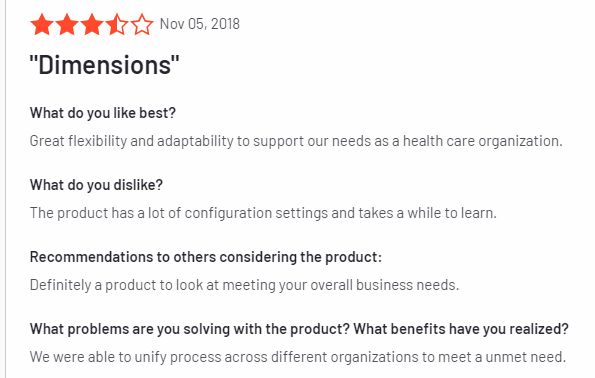


### **4.8.6** PlayN

Voor playN hebben we 1 review gevonden van 5 sterren, gedateerd uit 2018.

In deze review is de schrijver positief over de veelzijdigheid van de engine, omdat het voor meerdere platformen te gebruiken is. Daarbij raadt hij gebruikers wel aan om te kijken of hun game aansluit op de componenten van de engine.

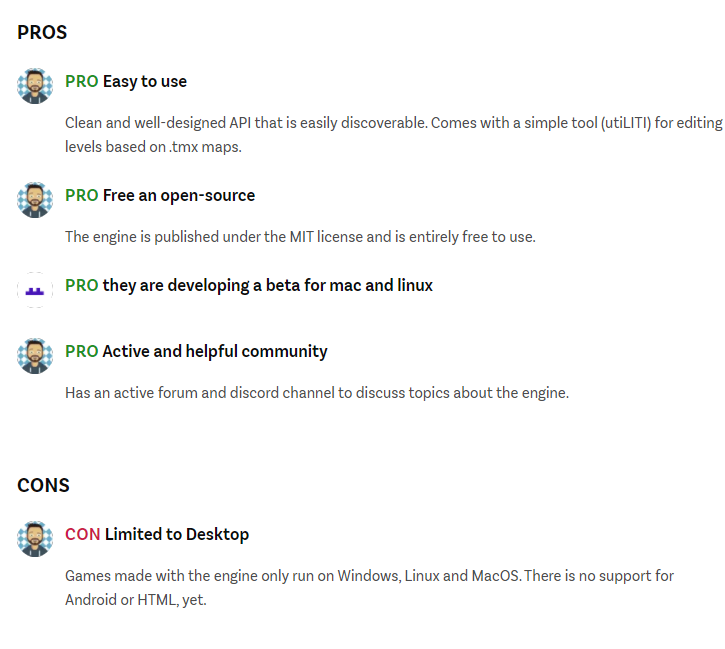
### **4.8.7** DimensioneX

Voor DimensioneX konden we online maar 1 review vinden van 3,5 sterren. 

Hier kun je zien dat de reviewer ook heeft gezegd dat het leren omgaan hiermee vrij moeilijk is waardoor het voor een ICT student lastig is om te gebruiken. Ook tijdens het downloaden van DimensioneX waren we via mediafire doorgelinkt en kwam er installaties tevoorschijn die wij niet verwachtte. Hierdoor is het minder betrouwbaar dan andere engines.

### **4.8.8** LITIENGINE

Voor LITIENGINE was er ook niet veel reviews over te vinden, dit komt waarschijnlijk doordat het een vrij nieuwe engine is. Maar wat we hebben gevonden kun je hieronder zien, hier wordt gezegd dat het gemakkelijk te gebruiken is en ook veel hulp van de gemeenschap te verkrijgen.



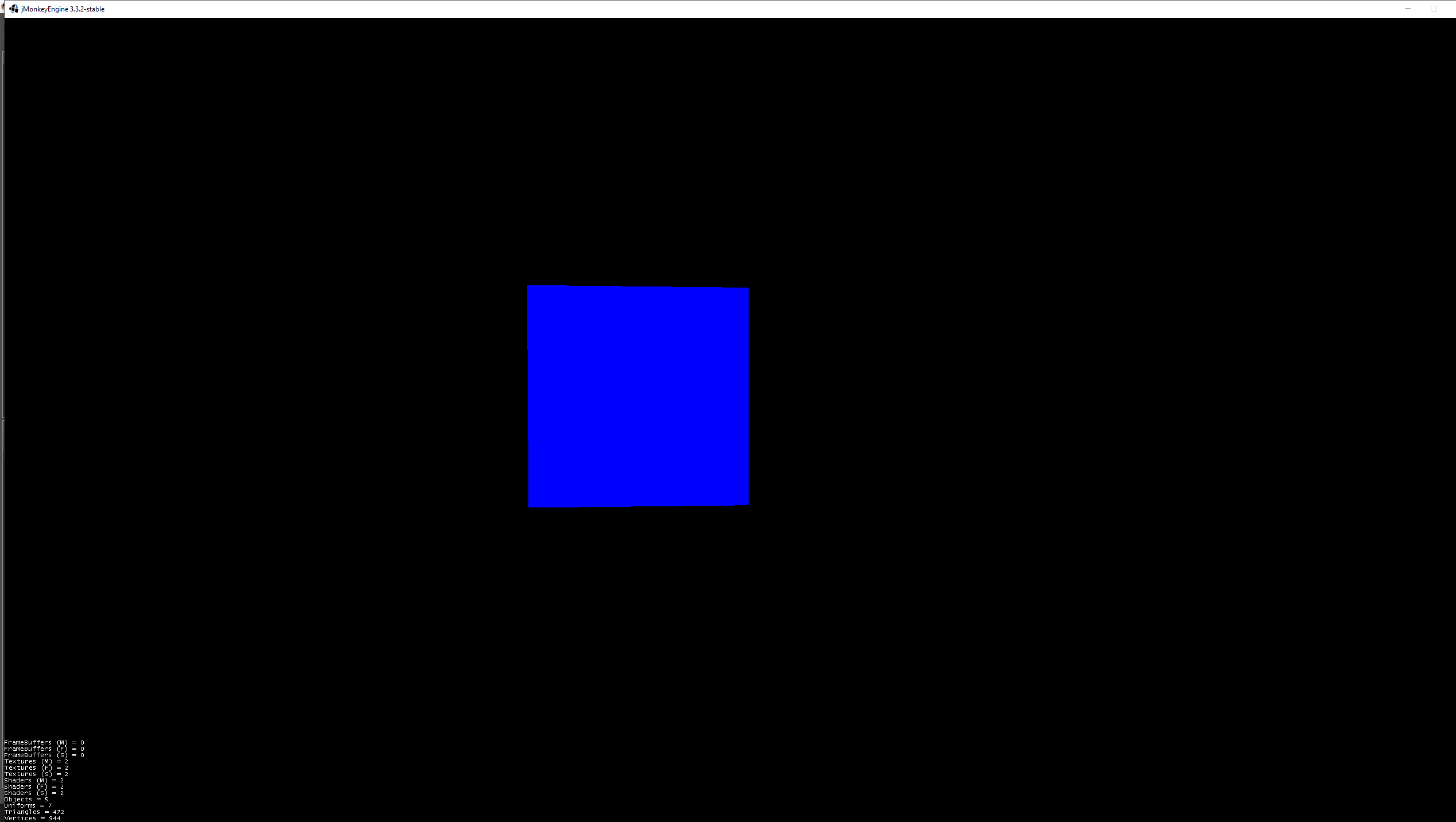
Hieronder is alles nog even samengevat.

|  |  |  |
| --- | --- | --- |
| Engines | Pros | Cons |
| Ardor3D | * Makkelijk verandering of aanpassingen te doen | * UI heeft soms lag |
| Jake2 | * N.v.t | * Lastig informatie over te vinden |
| Java 3D | * N.v.t | * N.v.t |
| JMonkeyEngine | * Veel informatie te vinden * Veel tutorials en guides te vinden | * Kleine foutjes * Beetje ouder dan modernere oplossingen. |
| LibGDX | * Gemakkelijk te gebruiken * Sterk en erg stabiel | * Geen GUI/drag en drop features |
| PlayN | * Engine is veelzijdig, voor meerdere platformen te gebruiken | * N.v.t |
| DimensioneX | * Goeie flexibiliteit en adaptability | * Vrij moeilijk te gebruiken * Niet betrouwbaar |
| LITIENGINE | * Gemakkelijk te gebruiken * Veel hulp van de gemeenschap | * Gemaakte games werken alleen op desktop |

## **4.9.** Met welke engine konden wij het makkelijkst een redelijk demo ontwerpen?

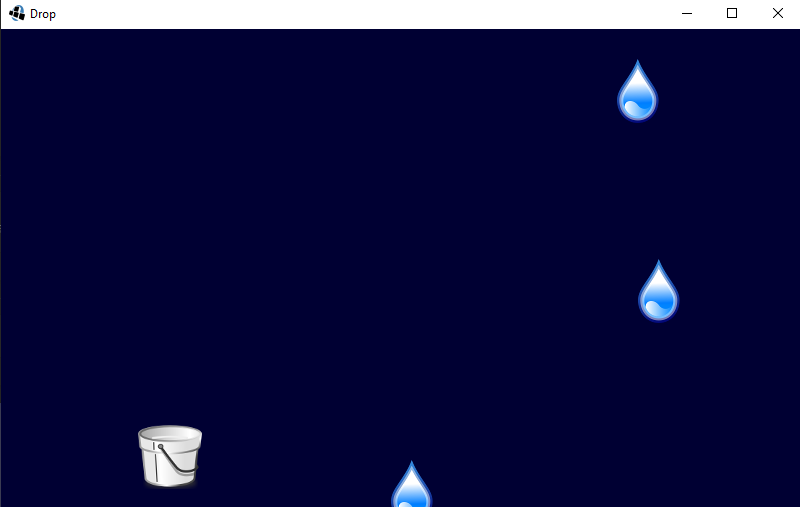
We hebben ervoor gekozen om voor 3 engines een kleine demo te ontwerpen, deze engines zijn; jMonkeyEngine, LibGDX en de Java 3D API. We gaan eerst de demo weergeven en dan bespreken we welke het beste uitkwam en welke het gemakkelijkst was om te gebruiken.

### **4.9.1. JMonkeyEngine**



We hebben hiervoor een basis ontwerp gebruikt die beschikbaar staat op de website van jMonkeyEngine. Het was ons niet gelukt om met IntelliJ de engine te integreren maar met behulp van de SDK van jMonkeyEngine was het ons wel gelukt om een kleine demo te maken met een blokje die je kunt bekijken van verschillende hoeken. We moeten hiervoor dus wel leren omgaan met een nieuwe IDE maar naar een paar minuten testen was het niet zo groot probleem.

### **4.9.2. LibGDX**



Voor LibGDX hebben we een klein spelletje kunnen demonstreren met behulp van een guide op de LibGDX website. Hiermee kun je druppeltjes opvangen en blijft alsmaar doorgaan, het is te gebruiken met behulp van de intellij IDE waar wij de meeste ervaring in hebben en het koste maar een aantal minuten om op te zetten. Hierdoor komt deze in verband met gemak van gebruik het meeste in aanmerking.

### **4.9.3. Java 3D API**

Het is ons niet gelukt om de Java 3D API te installeren, omdat dit niet duidelijk wordt aangegeven en niet duidelijk te vinden is hoe de API geïnstalleerd moet worden. Ook moest er een Oracleaccount aangemaakt worden voordat de engine gedownload kon worden. Dit maakt de engine niet heel geschikt voor gebruik door studenten, waardoor onze keuze niet valt op deze engine.

# Discussie

In dit hoofdstuk gaan we de resultaten bespreken en een definitieve keuze maken voor een game engine die wij gaan gebruiken tijdens het ontwerpen van onze 2d Top-Down shoot’ em up.

Het eerste wat wij wouden bespreken was dat wij tijdens het onderzoek erachter zijn gekomen dat er meer dan 700 game engines zijn waarvan 485 op de markt staan. Er is geen simpel overzicht van welke talen en componenten deze allemaal bevatten, om dus te zien of een engine in aanmerking kwam voor dit onderzoek hadden we elke game engine grondig moeten onderzoeken. In verband met tijd en beschikbaarheid van informatie hebben wij ervoor gekozen om een lijst van 8 Java game engines te gebruiken die vrij veel beschikbare informatie hadden op het internet.

Hieruit hadden we 3 engines gevonden die alle componenten hadden die overeenkwamen met onze benodigdheden en een aantal die bijna alles hadden. Deze componenten zijn gekozen op basis van 2 projecten die online zijn gevonden die allebei een 2d Top-Down shoot’ em up aan het ontwerpen waren. Veel van wat er in deze projecten voorkwam valt buiten de scope van onze project maar de basisinformatie was uitgebreid genoeg om ons een idee te geven van welke componenten wij nodig waren.

Met behulp van reviews en engine documentatie hadden we ervoor gekozen om de 3 engines met alle componenten uit te proberen met een kleine demo. Hierbij zijn we erachter gekomen dat het gebruik van 1 van deze engines omslachtig is en we konden het niet met onze tot nu toe behaalde kennis toepassen. De andere 2 engines jMonkeyEngine en LibGDX engine waren allebei wel te gebruiken door onze projectleden. Echter konden wij jMonkeyEngine alleen werkend krijgen op zijn eigen SDK waardoor er wat meer problemen zouden komen met het leren van een nieuwe IDE. Met LibGDX konden wij het gemakkelijk werkend krijgen met ons meest gebruikte IDE waardoor het beter toepasbaar is op onze groep. Ook waren de reviews van de engine erg positief, die van jMonkeyEngine ook. Echter waren er minder reviews te vinden dus heeft libGDX hier ook een voorsprong op.

Met al deze dingen in gedachten konden wij met volle vertrouwen een antwoord geven op welke Java game engine het beste was om te gebruiken voor een 2d Top-Down shoot’ em up, en dat is de LibGDX engine. Als er tijdens een toekomstig onderzoek een game engine gekozen moet worden zou het een goed idee zijn om elke engine grondig te onderzoeken en demo’s van maken. Omdat er voor veel van de engines niet genoeg informatie te vinden is online is het noodzakelijk om de files goed te bekijken en demo’s maken om te kunnen bepalen wat een engine kan doen.

# Literatuurlijst

Abargues, J. C. (2019, November 18). *Design and development of a game about*. Retrieved from Repositori Universitat Jaume I: <http://repositori.uji.es/xmlui/bitstream/handle/10234/179822/MEMORIA_CastellAlberguesJonathan.pdf?sequence=1&isAllowed=y>

Baker, M. (2016, November 2). *How game engines work*. Retrieved from Interesting Engineering: <https://interestingengineering.com/how-game-engines-work>

gamescrye. (2019, October 3). *what is a game engine?* Retrieved from Gamescrye: <https://gamescrye.com/blog/what-is-a-game-engine/>

GNU. (2021, January 31). *GNU*. Retrieved from license list: <http://www.gnu.org/licenses/license-list.en.html>

goussej. (2014, November 22). *Blog de julien Gouesse*. Retrieved from Goussej: <https://gouessej.wordpress.com/2014/11/22/ardor3d-est-mort-vive-jogamps-ardor3d-continuation-ardor3d-is-dead-long-life-to-jogamps-ardor3d-continuation/>

Kaitila, C. (2012, March 28). *Build a Stage3D Shoot-'Em-Up: Terrain, Enemy AI, and Level Data*. Retrieved from gamedevelopment tutsplus: <https://gamedevelopment.tutsplus.com/tutorials/build-a-stage3d-shoot-em-up-terrain-enemy-ai-and-level-data--active-11160>

moddb community. (2021, 2 24). *Engines*. Retrieved from moddb: <https://www.moddb.com/engines/latest>

Remeseiro, D. B. (2017, June 21). *Simple Space: 2D game design and development*. Retrieved from Dipòsit Digital de la Universitat de Barcelona: <http://diposit.ub.edu/dspace/bitstream/2445/119542/3/memoria.pdf>

Wikipedia. (2021, February 24). *List of Game Engines*. Retrieved from Wikipedia: <https://en.wikipedia.org/wiki/List_of_game_engines>