

# Analisis Efektifitas Teknik Optimasi Occlusion Culling dan Level of Detail (LOD) pada Game VR Unity 'The Rise of Majapahit Kingdom'

Penulis pertama <sup>1\*</sup>, Penulis Kedua <sup>2\*\*</sup>, Penulis Ketiga <sup>3\*</sup>

\* Teknik Informatika, Politeknik Negeri Batam

\*\* Teknik Multimedia Jaringan, Politeknik Negeri Batam

[mail1@polibatam.ac.id](mailto:mail1@polibatam.ac.id) <sup>1</sup>, [mail2@polibatam.ac.id](mailto:mail2@polibatam.ac.id) <sup>2</sup>, [mail3@polibatam.ac.id](mailto:mail3@polibatam.ac.id) <sup>3</sup>

## Article Info

### Article history:

Received ...

Revised ...

Accepted ...

### Keyword:

*Virtual Reality,*  
*Optimization,*  
*Occlusion Culling,*  
*Level of Detail*

## ABSTRACT



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

## I. PENDAHULUAN

Perangkat *Virtual Reality (VR)* merupakan teknologi yang membuat pengguna bisa merasa masuk kedalam dunia maya dan bisa berinteraksi didalam dunia maya tersebut[1]. Agar pengguna bisa merasakan kenikmatan dalam bermain didalam dunia maya tersebut, permainan yang digunakan perlu dilakukan optimasi agar bisa dijalankan tanpa mengalami *stuttering* ataupun lagging.

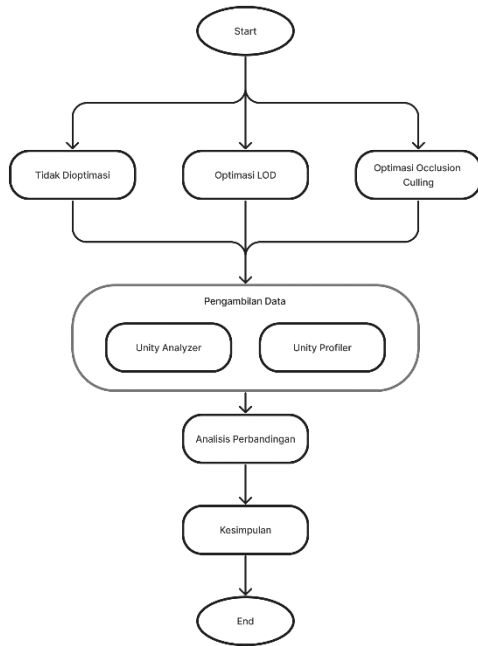
Teknik optimasi yang bisa dilakukan ada beberapa cara, tetapi ada 2 teknik yang sering digunakan yaitu *Occlusion Culling* dan *Level of Detail*. *Occlusion Culling* merupakan Teknik yang tidak menampilkan gameobject yang tidak terlihat oleh kamera. Lalu *Level of Detail* adalah Teknik yang membuat gameobject tidak memiliki detail yang akurat dari jarak tertentu.

Aplikasi permainan *Tower Defense: The Rise of Majapahit Kingdom* diangkat dari kisah sejarah kerajaan bercorak hindu – budha di Indonesia, yaitu runtuhnya Kerajaan Singasari dan bangkitnya Kerajaan Majapahit. Berdasarkan kisah sejarah tersebut, permainan akan dibagi menjadi dua *stage (hub)*. Namun yang berbeda ialah kita sebagai pemain yang akan berperan sebagai prajurit yang mempertahankan tower sesuai *role* dalam tiap *stage* tersebut, ketika berada di *stage* runtuhnya Kerajaan Singasari, kita berperan sebagai prajurit Singasari, dan ketika berada di *stage*

bangkitnya Kerajaan Majapahit kita menjadi prajurit Kediri. Tujuan utama dari penelitian ini ialah untuk melihat teknik mana yang paling optimal untuk aplikasi permainan ini.

## II. METODE PENELITIAN

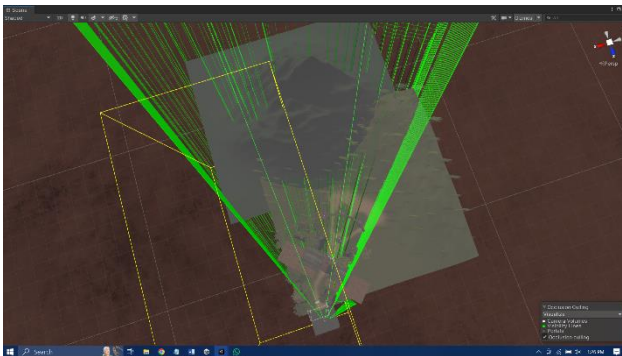
Metode penelitian yang digunakan untuk menganalisis efektifitas teknik optimasi *Unity* adalah metode kuantitatif dengan melakukan *Experimental Research* yaitu dengan mengontrol kondisi kelompok eksperimen dan kelompok kontrol[2]. Setelah itu hasil penelitian yang telah didapatkan dari hasil eksperimen dari 2 *game* yang memakai teknik optimasi yang berbeda sebagai kelompok eksperimen dan 1 *game* yang tidak dilakukan optimasi sebagai kelompok kontrol akan dibandingkan pada hasil akhir.



Gambar 1. Flowchart metode penelitian

Tahap awal pada penelitian ini adalah membuat *Game VR Unity* masing-masing sesuai dengan teknik optimasinya. Lalu ditahap berikutnya adalah mengumpulkan data dari masing-masing *Game VR Unity*, setelah itu dilakukan analisis perbandingan data-data tersebut hingga bisa diambil kesimpulan dari hasil perbandingan tersebut.

#### A. Occlusion Culling



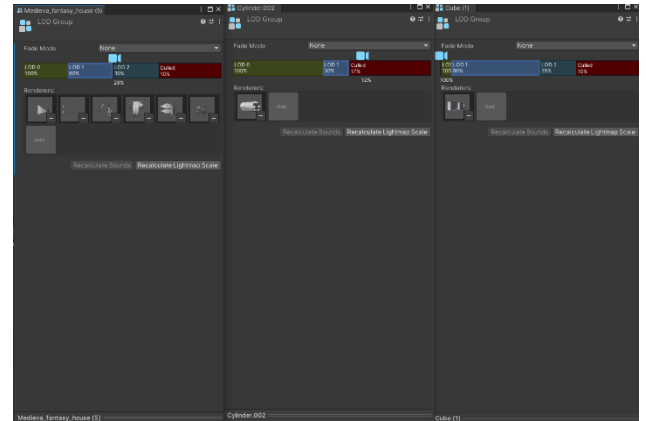
Gambar 2. Occlusion Culling pada unity.

*Occlusion Culling* adalah fitur yang menonaktifkan penggambaran *gameobject* ketika *gameobject* tersebut tidak terlihat oleh kamera karena terhalang (*occluded*) oleh *gameobject* lain. Hal ini tidak terjadi secara otomatis dalam grafika 3D komputer karena sebagian besar waktu objek yang paling jauh dari kamera ditarik terlebih dahulu

dan *gameobject* yang lebih dekat ditarik di atasnya (ini disebut "*overdraw*") [3].

Dengan teknik *occlusion culling*, *gameobject* yang tidak berada di dalam jarak pandang kamera atau *gameobject* yang terhalangi oleh *gameobject* lain tidak akan dirender oleh *unity*. Hal ini akan mengurangi proses pemanggilan *gameobject* dan akan meningkatkan kinerja *Unity* pada proses *runtime*.

#### B. Level of Detail (LoD)

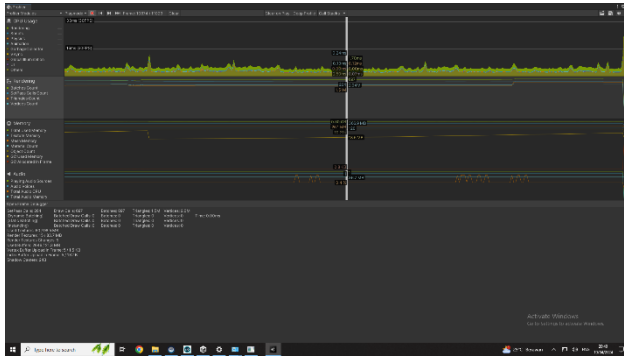


Gambar 3. Level of Detail pada unity

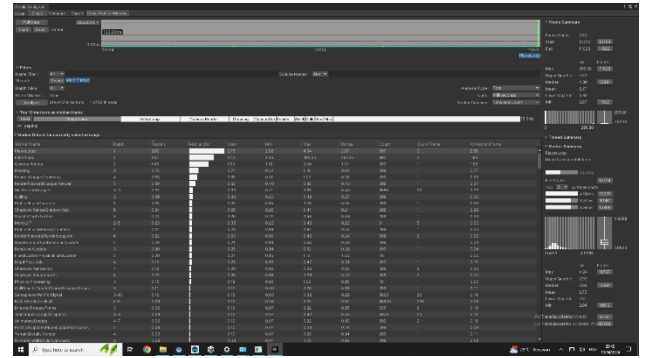
*Level of Detail (LoD)* adalah teknik yang mengurangi jumlah operasi *GPU* yang diperlukan oleh *Unity* untuk merender *mesh* yang jauh. Ketika sebuah *gameobject* dalam *scene* berada jauh dari kamera, terlihat detail yang lebih sedikit dibandingkan ketika *gameobject* berada dekat dengan kamera. Namun, secara *default*, *Unity* menggunakan jumlah segitiga yang sama untuk merender di kedua jarak tersebut. Hal ini dapat mengakibatkan operasi *GPU* yang terbuang percuma, yang dapat memengaruhi kinerja dalam *scene* [4].

Teknik *LoD* akan membuat *Unity* mengurangi jumlah segitiga yang akan dirender untuk *gameobject* berdasarkan jaraknya dari kamera. Penggunaannya, *gameobject* harus memiliki *mesh* yang tingkat detail yang akan berkurang secara bertahap tergantung jarak kameranya atau bisa disebut dengan *level*. Semakin jauh kamera dari *gameobject* maka *level* akan semakin tinggi dan *Unity* akan merender *gameobject* tersebut dengan detail yang rendah agar meningkatkan kinerja *hardware*.

#### C. Unity Profiler



Gambar 4. Unity Profiler



Gambar 5. Profiler Analyzer unity

Unity Profiler adalah tools yang dapat digunakan untuk mendapatkan informasi tentang kinerja Unity. Profiler mengumpulkan dan menampilkan data tentang kinerja Unity di berbagai area seperti CPU, memori, renderer, dan audio[5]. Profiler membantu untuk melihat data dan aspek-aspek apa saja yang tidak bisa dilihat ketika runtime. Aspek yang akan diambil untuk perbandingan adalah[6]:

- *SetPass Calls* = Jumlah berapa kali Unity mengubah penggunaan *shader pass* saat merender *gameobject* selama 1 *frame*.
- *Draw Calls* = *Total Draws Calls* yang diberikan oleh Unity selama 1 *frame* ketika merender *gameobject* ke *layer*.
- *Batches* = Jumlah *batches* yang di proses oleh Unity dalam 1 *frame*, jumlah ini meliputi *static* dan *dynamic batches*.
- *Triangles* = Jumlah dari segitiga yang diproses oleh Unity dalam 1 *frame*.
- *Vertices* = Jumlah dari *vertex* yang diproses oleh Unity dalam 1 *frame*.
- *Used Textures* = Jumlah dari tekstur yang diproses oleh Unity dalam 1 *frame*.
- *Render Textures* = Jumlah beberapa kali *Render Textures* Unity yang digunakan dalam 1 *frame*.
- *Render Textures Changes* = Jumlah beberapa kali unity merubah *Render Textures* sebagai target *render* dalam 1 *frame*.
- *Used Buffers* = Jumlah dari *GPU buffers* dan memori yang digunakan.

D. Unity Profile Analyzer

Unity Profile Analyzer adalah package dari Unity yang membantu mengumpulkan dan memvisualisasikan *Frame* dan *Marker* dari data Unity Profiler. Unity Analyzer melakukan analisis kinerja CPU pada beberapa *frame*[7]. Aspek yang akan dipakai untuk perbandingan akan berasal dari *Frame Summary*, yaitu[8]:

- *Max* = *Frame Time* terbesar atau maksimum dalam data pilihan.
- *Upper Quartile* = Menampilkan kuartil atas dari set data.
- *Median* = Menampilkan nilai median atau nilai tengah dari set data.
- *Mean* = Menampilkan nilai mean atau nilai rata-rata dari set data.
- *Lower Quartile* = Menampilkan kuartil bawah dari set data.
- *Min* = *Frame Time* terkecil atau minimum dalam data pilihan.

E. Frame Rate

$$FRR = F/T$$

- Where FRR is the Frame Rate (frames/second)
- #F is the number of frames
- T is the total time (s)

Gambar 6. Rumus menghitung frame rate

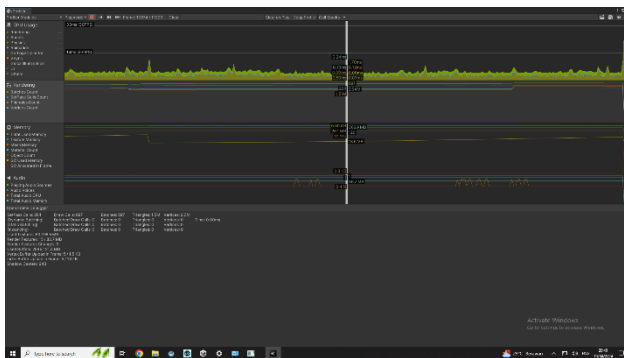
Frame rate adalah seberapa cepat gambar-gambar berurutan yang disebut *frame* yang ditampilkan dalam bentuk animasi, sering diukur dalam *frame per second (FPS)*[9]. *Frame rate* yang lebih tinggi akan meningkatkan pengalaman pemain, dikarenakan akan memberikan gerakan yang lebih halus dibandingkan dengan *frame rate* yang rendah. *Frame rate* yang rendah

akan memberikan gerakan patah-patah atau biasa disebut dengan *lag* atau *lagging*.

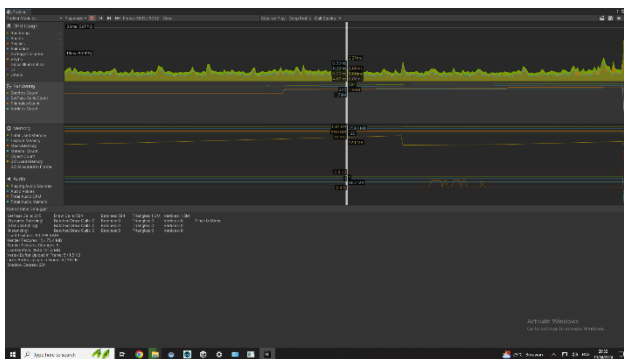
### III. HASIL DAN PEMBAHASAN

#### A. Profiler

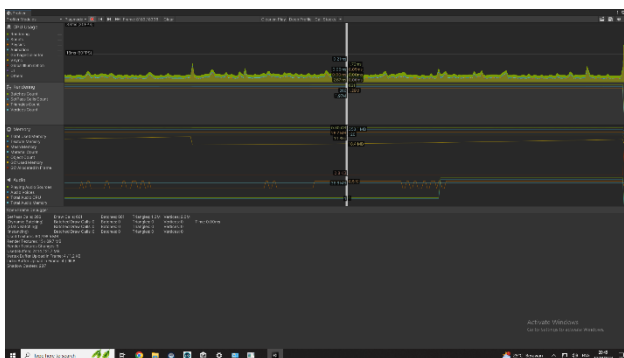
Gambar dibawah ini adalah hasil dari *Profiler* yang sudah didapatkan dari ketiga *game* yang telah diuji. Bisa dilihat bahwa *spike* grafik pada teknik optimasi *Occlusion Culling* dan *Level of Detail* tidak memiliki perubahan yang signifikan dengan yang tidak memakai teknik optimasi apapun.



Gambar 7. Unity Profiler data kontrol



Gambar 8. Unity Profiler data Occlusion Culling

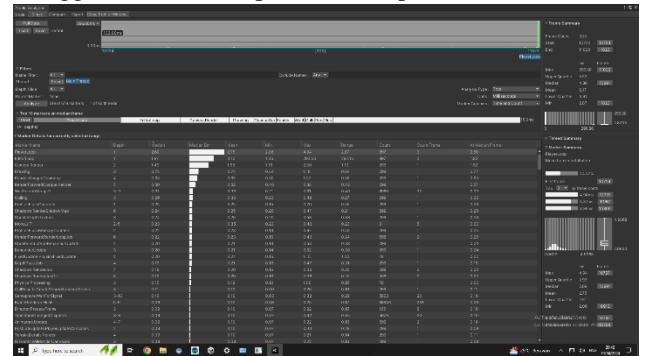


Gambar 9. Unity Profiler data Level of Detail

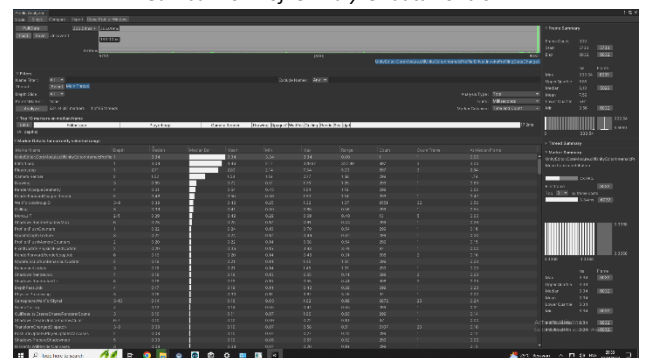
#### B. Profile Analyzer

Berdasarkan hasil dibawah, nilai rata-rata yang didapatkan dari hasil optimasi yang menggunakan *Occlusion Culling* ataupun *Level of Detail* tidak terlalu

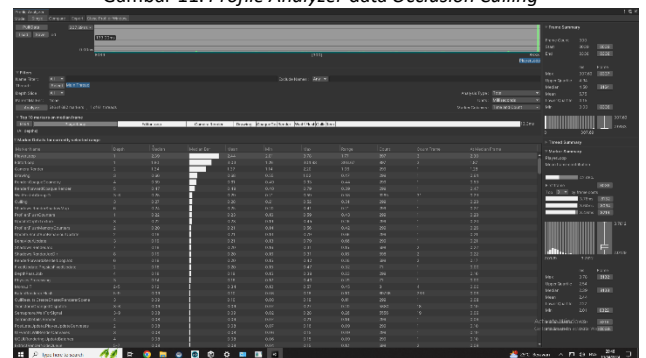
jauh berbeda dengan nilai rata-rata dari hasil yang tidak menggunakan teknik optimasi manapun.



Gambar 10. Profile Analyzer data kontrol



Gambar 11. Profile Analyzer data Occlusion Culling



Gambar 12. Profile Analyzer data Level of Detail

#### C. Perbandingan Data

TABEL I

PERBANDINGAN DATA

	Data Kontrol	Data Occlusion Culling	Data Level of Detail
<i>SetPass Cals</i>	264	245	266
<i>Draw Calls</i>	697	594	661
<i>Batches</i>	697	594	661

<i>Triangles</i>	1.5M	1.3M	1.3M
<i>Vertices</i>	2.0M	1.8M	2.0M
<i>Used Textured</i>	60 / 98.5 MB	60 / 98.5 MB	60 / 98.5 MB
<i>Render Textured</i>	15 / 83.7 MB	15 / 75.4 MB	15 / 69.7 MB
<i>Render Textured Changes</i>	5	5	5
<i>Used Buffers</i>	2646 / 21.3 MB	2645 / 21.3 MB	2414 / 21.7 MB
<i>Vertex Buffers Upload In Frames</i>	5 / 1.5 KB	5 / 1.5 KB	4 / 1.2 KB
<i>Index Buffer Upload In Frame</i>	5 / 132 B	5 / 132 B	4 / 96 B
<i>Shadow Caster</i>	243	224	237
<i>Max</i>	296.8	333.54	307.69
<i>Upper Quartile</i>	5.53	6.85	4.94
<i>Median</i>	4.86	6.1	4.5
<i>Mean</i>	6.17	7.59	5.75
<i>Lower Quartile</i>	4.61	5.61	4.15
<i>Min</i>	2.87	3.56	3
<i>Frame Rate</i>	205.76	163.93	222.22

Dari data yang sudah ditampilkan pada tabel diatas, bisa dilihat bahwa dari kedua teknik optimasi tersebut, teknik yang menggunakan *Level of Detail* memberikan *frame rate* yang lebih tinggi dibandingkan yang menggunakan *Occlusion Culling* ataupun tidak menggunakan teknik optimasi. Tetapi walaupun begitu kenaikan *frame rate* tersebut tidak signifikan, kenaikan tersebut sekitar 8% dibandingkan dengan yang tidak

menggunakan optimasi. Tetapi walaupun *frame rate* yang lebih tinggi, untuk aspek *SetPass Calls*, *Draw Calls*, *Batches*, *Triangles*, *Vertices* lebih rendah -1% sampai 13% dibandingkan dengan yang menggunakan *Occlusion Culling*.

Sedangkan teknik optimasi *Occlusion Culling* memiliki *frame rate* yang lebih rendah dibandingkan oleh yang tidak menggunakan teknik optimasi. *Frame rate* yang diberikan oleh *Occlusion Culling* menurun sebanyak sekitar 19% dibandingkan dengan yang tidak menggunakan teknik optimasi. Tetapi berbeda dengan menggunakan teknik *Level of Detail*, aspek *SetPass Calls*, *Draw Calls*, *Batches*, *Triangles*, *Vertices* lebih tinggi 7% sampai 15%.

#### IV. KESIMPULAN

Penggunaan teknik optimasi *Level of Detail* meningkatkan *frame rate* lebih tinggi dari penggunaan teknik *Occlusion Culling* dan yang tidak menggunakan teknik optimasi, tetapi tidak meningkat secara signifikan. Sebaliknya, teknik *Occlusion Culling* malah memberikan *frame rate* yang lebih rendah dibandingkan dengan yang tidak menggunakan teknik optimasi. Meskipun demikian, aspek *SetPass Calls*, *Draw Calls*, *Batches*, *Triangles*, *Vertices* dari teknik *Occlusion Culling* lebih tinggi dari teknik *Level of Detail*. Dengan menggunakan teknik optimasi *Level of Detail*, *frame rate* dalam game *VR Tower Defense: The Rise of Majapahit Kingdom* dapat ditingkatkan tetapi tidak terjadi peningkatan yang signifikan. Hal ini dapat mengindikasikan bahwa teknik optimasi *Level of Detail* dapat digunakan agar memberikan pengalaman bermain yang memuaskan.

## DAFTAR PUSTAKA

- [1] T. N. Fitria, "Augmented Reality (AR) and Virtual Reality (VR) Technology in Education: Media of Teaching and Learning: A Review," *International Journal of Computer and Information System (IJCIS)*, vol. 4, no. 1, pp. 14–25, February 2023, DOI: <https://doi.org/10.29040/ijcis.v4i1.102>.
- [2] R. Akbar, "Experimental Research Dalam Metodologi Pendidikan," Zenodo (CERN European Organization for Nuclear Research), Jan. 2023, DOI: <https://doi.org/10.5281/zenodo.7579001>.
- [3] "Unity - Manual: Occlusion Culling," docs.unity3d.com. <https://docs.unity3d.com/560/Documentation/Manual/OcclusionCulling.html>.
- [4] U. Technologies, "Unity - Manual: Level of Detail (LOD) for meshes," docs.unity3d.com. <https://docs.unity3d.com/Manual/LevelOfDetail.html>.
- [5] U. Technologies, "Unity - Manual: Profiler overview," docs.unity3d.com. <https://docs.unity3d.com/Manual/Profiler.html>.
- [6] U. Technologies, "Unity - Manual: The Profiler window," docs.unity3d.com. <https://docs.unity3d.com/Manual/ProfilerWindow.html>.
- [7] "Profile Analyzer window | Profile Analyzer | 1.2.2," docs.unity3d.com. <https://docs.unity3d.com/Packages/com.unity.performance.profile-analyzer@1.2/manual/profile-analyzer-window.html>.
- [8] "Frame Summary | Profile Analyzer | 1.2.2," docs.unity3d.com. <https://docs.unity3d.com/Packages/com.unity.performance.profile-analyzer@1.2/manual/frame-summary.html>.
- [9] L. H. Minh, T. B. Minh, L. V. Sang, and H. T. Khanh, "Object level frustum culling based frame rate acceleration method," *Journal of Military Science and Technology*, no. CSCE6, pp. 28–40, Dec. 2022, DOI: <https://doi.org/10.54939/1859-1043.j.mst.csce6.2022.28-40>.
- [10] "Frame Rate Calculator," Calculator Academy, Jul. 27, 2023. <https://calculator.academy/frame-rate-calculator/>.