Assignment 3 – Performance measurements

Context of the assessment

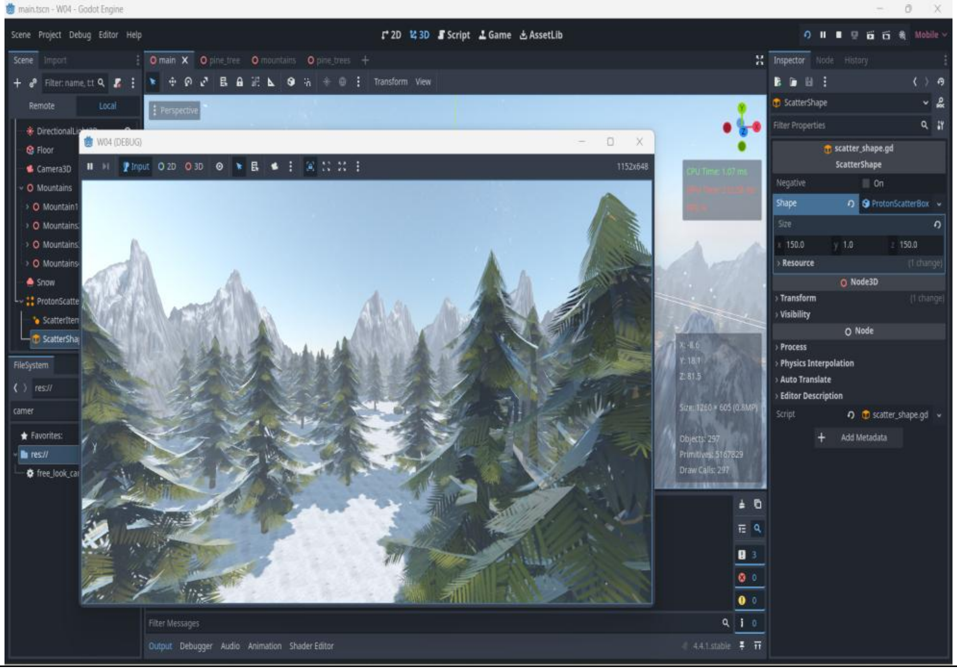
Write your PC specifications here:

Type	**Detailed Specs – be as specific as possible**
GPU (integrated)	Intel(R) Iris(R) Xe Graphics
GPU (dedicated) *specify your available memory*	NVIDIA GeForce MX350 Available memory: 2.0 GB
CPU	11th Gen Intel(R) Core (TM) i7-1165G7 @ 2.80GHz
Memory (RAM)	16 GB
Storage *Specify free space available*	476 GB SSD, 350 GB Free
Display resolution	1920 x 1080

You only have to fill this section once.

Performance assessment #1

Performance Metrics

Type	**Measured**
Viewpoint screenshot - Maximize your game - Take a full screenshot (print screen)	 main.tscn - W04 - Godot Engine Scene Project Debug Editor Help 2D 3D Script Game AssetLib W04 (DEBUG) 1152x648 CPU Time: 1.07 ms X: -8.6 Y: 18.1 Z: 81.5 Size: 1260 × 605 (0.8MP) Objects: 297 Primitives: 5167829 Draw Calls: 297 Inspector ScatterShape Output Debugger Audio Animation Shader Editor
Frames per second (FPS)	4
CPU Time (ms)	1.07
GPU Time (ms)	212.58
# objects	297
# primitives	5167829
# draw calls	297
# particle system visible	Snow
# light emitters in view	DirectionalLight3D with shadows enabled
# of physics active objects in scene (requires runtime)	
# of collision pairs in view in scene (requires runtime)	
Viewport dimensions (width x height, at runtime)	1152 × 648
Git commit hash	38e704984adaebc008b19d6b33c9c180566c1590

Save your current scene in version control	

Identification of the issue

Limiting resource CPU or GPU	**CPU**	**GPU**
Bottleneck issue identified Explain what you assessed as the main bottleneck and priority to solve	Shadows from DirectionalLight3D + very high draw calls from large tree count.	
Technique used to detect the bottleneck Ex: Profiler, Visual Profiler, Debugging, Binary search, Hypothesis testing, etc.	Visual Profiler (measured GPU time)	

Optimization technique

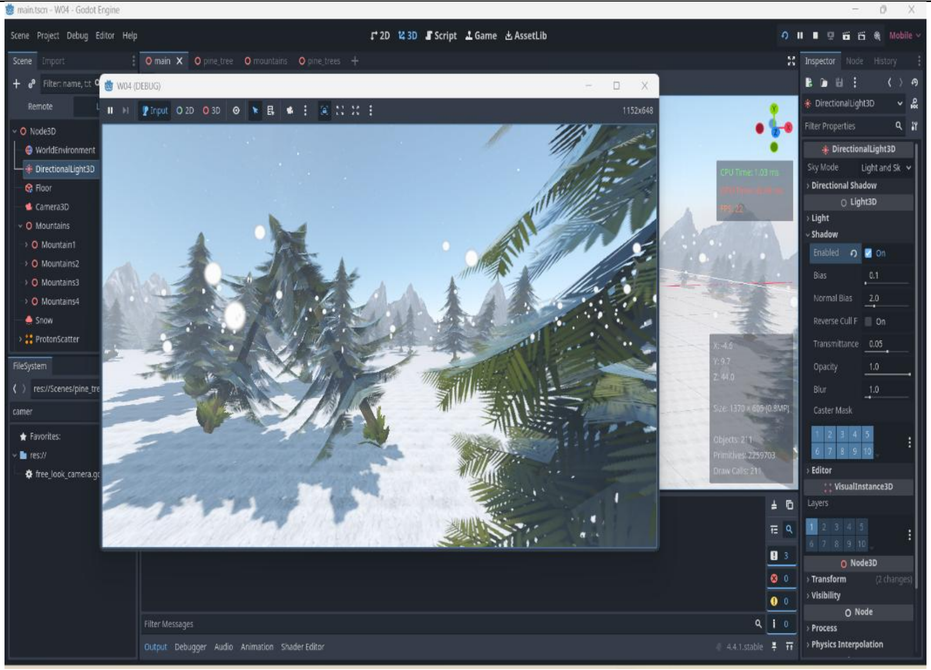
Optimization technique used to increase the performance
Ex: LOD, Visibility Ranges, Occlusion Culling, Lowering textures quality, Lowering mesh details, Disabling shadows, etc. Explain the technique used and how it was applied. Reduced the number of scattered trees (ProtonScatter). I had like 1000 and reduced it to 350

Gained performance

Explain briefly the increase in performance gained after the optimization was made. Lower primitive and draw calls, reduced GPU load. FPS went from 4 to 22 in later test.

Performance assessment #2

Performance Metrics

Type	**Measured**
Viewpoint screenshot - Maximize your game - Take a full screenshot (print screen)	 main.tscn - W04 - Godot Engine Scene Project Debug Editor Help 2D 3D Script Game AssetLib Mobile Scene Import main pine_tree mountains pine_trees Remote Node3D WorldEnvironment DirectionalLight3D Floor Camera3D Mountains Mountain1 Mountains2 Mountains3 Mountains4 Snow ProtonScatter FileSystem res://Scenes/pine_tre Favorites: res:// free_look_camera.gd W04 (DEBUG) Input 2D 3D 1152x648 CPU Time: 1.03 ms FPS: 22 X: -4.6 Y: 9.7 Z: 44.0 Size: 1370 x 605 (0.8MP) Objects: 211 Primitives: 2259703 Draw Calls: 211 Filter Messages Output Debugger Audio Animation Shader Editor 4.4.1.stable Inspector Node History DirectionalLight3D Filter Properties Sky Mode Light and Sk Directional Shadow Light3D Light Shadow Enabled On Bias 0.1 Normal Bias 2.0 Reverse Cull F On Transmittance 0.05 Opacity 1.0 Blur 1.0 Caster Mask Editor VisualInstance3D Layers Node3D Transform (2 changes) Visibility Node Process Physics Interpolation
Frames per second (FPS)	22
CPU Time (ms)	1.03
GPU Time (ms)	43.49
# objects	211
# primitives	2259703
# draw calls	211
# particle system visible	Snow
# light emitters in view	DirectionalLight3D with shadows OFF
# of physics active objects in scene (requires runtime)	
# of collision pairs in view in scene (requires runtime)	
Viewport dimensions (width x height, at runtime)	1152 × 648
Git commit hash	5066725a1db5ca7ac380be928bdaeef68af2b1da

Save your current scene in version control	

Identification of the issue

Limiting resource CPU or GPU	CPU	GPU
Bottleneck issue identified Explain what you assessed as the main bottleneck and priority to solve	Real-time shadows on DirectionalLight3D.	
Technique used to detect the bottleneck Ex: Profiler, Visual Profiler, Debugging, Binary search, Hypothesis testing, etc.	Visual Profiler (compared GPU ms before/after).	

Optimization technique

Optimization technique used to increase the performance
Ex: LOD, Visibility Ranges, Occlusion Culling, Lowering textures quality, Lowering mesh details, Disabling shadows, etc. Explain the technique used and how it was applied. Disabled shadows on DirectionalLight3D.

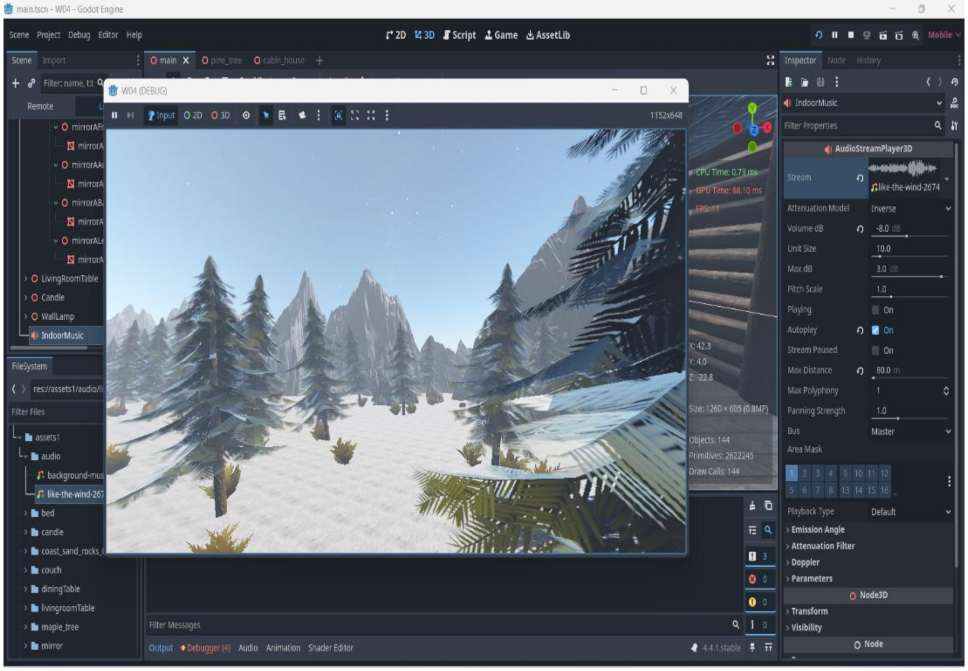
Gained performance

Explain briefly the increase in performance gained after the optimization was made. GPU time dropped from 212.58 ms to 43.49 ms. FPS doubled, from 22 to 54 and higher.

Performance assessment #3

Performance Metrics

Type	Measured

Viewpoint screenshot - Maximize your game - Take a full screenshot (print screen)	 main.tscn - W04 - Godot Engine W04 (DEBUG) 1152x648 CPU Time: 0.73 ms GPU Time: 88.10 ms FPS: 11 X: 42.3 Y: 4.0 Z: -22.8 Size: 1260 × 605 (0.8MP) Objects: 144 Primitives: 2622245 Draw Calls: 144 IndoorMusic AudioStreamPlayer3D Stream: like-the-wind-2674 Attenuation Model: Inverse Volume dB: -8.0 dB Unit Size: 10.0 Max dB: 3.0 dB Pitch Scale: 1.0 Playing: □ On Autoplay: ☑ On Stream Paused: □ On Max Distance: 80.0 m Max Polyphony: 1 Panning Strength: 1.0 Bus: Master Playback Type: Default
Frames per second (FPS)	11
CPU Time (ms)	0.73
GPU Time (ms)	88.10
# objects	144
# primitives	2622245
# draw calls	144
# particle system visible	Snow outside, Candle flame inside
# light emitters in view	Candle GPUParticles3D, Lamp OmniLight3D, with shadows initially enabled.
# of physics active objects in scene (requires runtime)	
# of collision pairs in view in scene (requires runtime)	
Viewport dimensions (width x height, at runtime)	1152 × 648
Git commit hash Save your current scene in version control	c3a4abf24dc2011c217eafd17c0a553bcc50a6f9

Identification of the issue

Limiting resource CPU or GPU	**CPU**	**GPU**
Bottleneck issue identified Explain what you assessed as the main bottleneck and priority to solve	OmniLight3D shadow, each light generates cube shadow maps, heavy.	
Technique used to detect the bottleneck Ex: Profiler, Visual Profiler, Debugging, Binary search, Hypothesis testing, etc.	Visual Profiler (saw GPU was high when indoor lights were on).	

Optimization technique

Optimization technique used to increase the performance
Ex: LOD, Visibility Ranges, Occlusion Culling, Lowering textures quality, Lowering mesh details, Disabling shadows, etc. Explain the technique used and how it was applied. Disabled OmniLight shadows and reduced OmniLight ranges.

Gained performance

Explain briefly the increase in performance gained after the optimization was made. GPU time reduced massively, FPS stabilized closer to 50 and more. Scene still realistic with glowing candle + lamp.