SWE585 Term Project

Game Development: Armstrong Strongarm

Group Members:

Fatma Betül Güreş, Deniz Dikbıyık

Date: 05.06.2022

Instructor: Atay Özgövde

Introduction

The game is a <u>First Person Shooter</u> (FPS) game, where the astronaut must shoot alien enemies. <u>You as a player:</u> are an astronaut with a costume that offers you gravity control ability (under your feet there is a magnetic system) and you must withstand alien attacks to escape. <u>Them:</u> Element bender powerful aliens.

Description

Armstrong Strongarm is a fast-paced first person shooter. Set on Planet EFAW (Earth, Fire, Air, Water) outdoor environments specific to the element bender aliens. The world might be thought of as a parallel universe to Earth, but with four types of aliens who live in harmony as they need each other and each season (spring, summer, fall, winter) another species is ruling. They have some fancier technology and the ability to control elements.

The player must not only eliminate the attack of the aliens, but must strive to maintain battery level in the suit, which is rechargeable when you reach batteries.

The aliens don't understand your purpose and see you as a thief, so they attack you. They will be using the NavMesh agent and Unity AI elements to chase you. As you are an astronaut with a costume that offers you gravity control ability this power will offer flexibility during fights.

Game Mechanics

Movement of the Player

The player is moving around in the game environment, holding a series of guns and being able to change guns and zoom in or zoom out using the lense of the guns.

<u>Shooting</u>

There are different types of guns which include different numbers of ammos inside. To kill the enemies, shooting is done by the player.

Changing Weapon

The other types of guns are changed to get another charged one.

Collecting Battery

The batteries are spread around in the game environment, so the player is able to reach them and collect them to increase the ability to survive.

Collecting Ammo

The guns are filled with different types of ammos, so ammos around the game environment are collectible.

Controlling Gravity

The astronaut, who is the game player, is able to change the gravity in some situations in the game. If there is a platform too high to jump, the astronaut can change gravity values to get there.

Enemies are Chasing

The enemies which are defined above are coming towards the player, when they realize the player is near them (if within chase range) or if the player shoots them (and they are still alive to chase).

Enemies are Attacking

Enemies are attacking the game player to decrease survival ability. Blood is seen on the screen when there is an attack by the enemy and health points of the player decreases.

Asset List

Guns Batteries Ammo Sky Particles Enemies: Min Legion Rock Golem in Asset Store Game Environment: Fantasy_Environment Pack in Asset Store





Summary of the Implementation Process

First of all, we wanted to improve our knowledge about Unity and took this course on Udemy. Complete C# Unity Game Developer 3D (<u>https://www.udemy.com/course/unitycourse2/</u>). We have both watched the course, especially the section related to 1st Person Shooter development. The course was about a zombie shooter, but we have applied the learned skills to develop the game for the things we have proposed in Project Phase 1, Armstrong Strongarm.

We added red wireframe spheres **-gizmos-** to the enemies which show the chase range in scene view. Chase range is in fact the distance, if triggered by the player, the enemy starts walking towards the player.



We use a raycast system to shoot the enemies. Raycasting can be thought of an invisible ray, of a certain length, coming out of the camera (field of view of the player). This invisible ray is looking for collieders. In our case, if the shooting raycast hits the enemy collider, the enemy will get damage.

After shooting, particle animation is shown to make shooting more visible. One particle appears at the tip of the weapon, and the other particle is instantiated at where the raycast system hits. Players can pick up ammo and batteries which increase ammo number or the lightning.

Terrain: We have used Unity's terrain tools to define the boundaries of the game world. We didn't want our player to go out of the game-zone, therefore made use of high altitudes at the surroundings. We have created materials with colors matching our game world and have also added lowpoly trees (once again added with the terrain tool, not added gameobjects one by one).

Pro-Grids and Pro-Builder: Both of them are Unity packages used for prototyping and creating level design. Pro-builder is like a simplified modeling tool that can be accessed within the game engine. Progrids, as the name suggests, creates helper grids for the user, and can be used together with pro-grids. We have experimented with these tools when creating the level design. Also we have created simple assets, like different ammos that can be picked up to recharge our weapon.



So, the main game is based on:

- Enemies that attack when provoked
- Shooting function
- Flashlight mechanic with recharging
- Ammo count
- Ammo pickups
- Weapons we can cycle through
- Zoom In-Out with some weapons

Minimum Viable Product (MVP):

The core events of the game are listed below.

- First Person Camera movement
- Raycasting to shoot
- Enemies move and attack AI
- Health and damage system
- Death/game over
- Weapon switch
- Ammo pickup
- Flashlight pickup

Implementation Process

First Person Controller:

When we created the project, we downloaded Unity Standard Assets. We did not add sample scenes, 2D things, physics materials and vehicles. After that, we created one plane. From the standard assets, we added a rigidbody fps controller as a first person controller. The name is changed as the player. We deleted the main camera because the one under player was enough for us.

NavMesh Agent for AI:

This is a helpful AI feature of Unity. We are using this for the enemies to come towards the player and be able to walk around the obstacles. Using the parameters we define the platforms that the enemies are able to jump on and off. We started implementing NavMesh by creating a 3D capsule. It was

really useful to see the main functions. We made the capsule collider true and added a NavMesh agent. This is the enemy which moves around.

Chase Range:

We made the enemy prefab and worked on the chase range. Chase range is about following the player and the work we did here was about how close will the enemy be to the player. The player is the distance target for the enemy, so the values are important here. We used OnDrawGizmosSelected to make the surrounding chasing ability. The following method is in Enemy.cs.

void OnDrawGizmosSelected()

```
{
  Gizmos.color = Color.red;
  Gizmos.DrawWireSphere(transform.position, chaseRange);
}
```

Enemy AI to Attack If Provoked:

Enemies also follow the player when they are shot. They follow if the player is in the visible area to them, they follow and attack if close enough. So, the functionality is based on engaging target related to is provoked boolean and make is provoked boolean true if close enough, then follow. In the engage situation -meaning that the player has shot and provoked the enemy- the enemy starts chasing the player no matter the distance. Once close enough to attack and give damage, it enters the attack state. In the animator, attack animation starts to play.

Gun to the Player:

We have 3 guns and add them as child objects to the player's camera, by dragging. By changing the weapon model's scale, position, color and rotation, we made all weapons aim at the center of the screen, as if our first person character is holding the weapon in his hands. After obtaining the desired looks, we applied the changes to prefab. To see the target point reticle, we added a UI canvas and placed the gun reticle image at the center, by adding an image to the canvas and attaching the correct image to the sprite renderer.

Raycasting:

Raycasting is based on origin, ray and collision. We have a weapon script for it. By this functionality, we can focus on the target by a line and shoot.

Enemy Health and Damage:

We created an enemy health script for this and called on weapon to decrease the health of the enemy when it is shot. With the values we have attached to the enemy health and weapon's shooting damage, the enemy can die when 3 shots are done.

Muzzle Flash:

We added this by adding the particle system from effects option. This is for the shooting effect. It can be found as VFX under prefabs. Looping and play on awake should be made false to call this only if shooting is done. The simulation space is world, and we changed the duration, lifetime, speed, and rate over time. We dragged this to the gun. We triggered this on the weapon scripts and added process raycast and play muzzle flash methods. By adding the material to the code, we called this easily and finally made the visuals more realistic.

Shooting Effect:

We use explosion here from the standard assets. We made it prefab and designed later. The code related to it is process raycast. It is instantiated there with quaternion look rotation and destroy it after 1 second.

Animations:

There are animations and transitions. From animation component, we assign animations to the game objects through **animator** controller. From animator controller, we can arrange the animations and transitions (state machine).



There are Entry, Idle-Move (Walk)-Attack states and AnyState-Die. These states are called according to triggers and booleans. Die can be called from any state because it can occur any time. We added animator component to the enemy and created animator controller. By creating states, we added the ones mentioned above. Transitions are also added accordingly. The settings on the right side of the editor are important to make transitions smooth. We made exit times around 1 second. We added animator controller to the enemy. Normally, animations can be created by recording, but we had very useful animations coming with our enemy assets. To the animation states above, we have attached the animations of the enemy for: idle, walk, attack and die. All enemies use the same animator, as they have the exact same behavior (only the colors and materials are different among the element bender aliens). The transitioning among the animation states are controlled with the triggers and booleans. In our game logic, once the player enters the chase range, the move trigger is fired from code. Similar to triggers, we also have booleans "attack", which is set to true and false according to the desired gamelogic.

Enemy.Animator.SetTrigger("move");

Enemy.Animator.SetBool("attack", true);

Inside the attack animation there is an event called "**AttackHitEvent**". At this frame of the animation (when the enemy is punching the player) the player gets health damage.



Player Health:

This is also a script similar to enemy health. We call this on enemy attack code with component of target. Because it is hard to call it with a target, we called from player health with an object in the enemy attack.

User Interfaces:

To create user interfaces, we have made use of Unity Canvases. We used more than one and activated/deactivated them, depending on the need of the game. For instance, we created the game over the user interface and included a play again button. In order to make this button functional, we created a public method that reloads the game (by reloading the scene). Similarly, to exit the game, we created another public method and attached it to the quit button.



Weapon System:

We have developed a relatively complex weapon system for our game. We did not only attach one gun, but three, from which the user can choose what they want. To change among the weapons, the user can scroll the mouse wheel or press on 1, 2 or 3 from the keyboard. Visually, we have made differences regarding color and scale, however we wanted to have some functional differences as well. For instance we wrote a script (WeaponZoom) for zooming in and out- and only attached it to two weapons. By toggling (from script) the Field Of View (FOV) variable of the camera, we obtained a basic zoom effect. To develop this feature even further, we wanted to make the zoomed-in version less sensitive so that the player can smoothly observe the distance and make more precise shots.



Another complex feature regarding the weapon system is the fact that, for filling each gun, we require another type of bullet. In the picture above, we called our red weapon-Shotgun, the white one is **Carbine** and the black one is **Pistol**. To shoot with Shotgun, we need **Bullets**; for Carbine we need **Shells**, and for the pistol, we need **Rockets**. In order to differentiate these features, we have created an enum script (which is not of type MonoBehaviour).

```
public enum AmmoType
{
   Bullets,
   Shells,
   Rockets
}
```

Gravity:

To offer gravity control to our player, we created inputs, such that when the player presses "G" key, the Physics.gravity is being multiplied with -1 (therefore changing direction). To activate this gravity switch, the player must realize a jump- which will add force, and will let player move in the chosen gravity direction.

Technical Challenges

NavMesh:

Armstrong Strongarm is a 1st person shooter, where all the enemies are using the unity.ai library. These are moving on their own, with the component NavMesh agent. To see the objects that an enemy can move on, the **Bake** can be checked. Static should be selected for all the surface gameobjects that the player can move on. Changing agent radius and max slope are important to have more realistic movements. We have done fine tuning by playing with the values, until the baked area made sense for our game environment. At this point, we started to write our enemy AI script. We gave a destination which is set according to the target position that allows us to make enemies follow the player. On the main camera of the player, the head bob is closed to remove unnecessary stucks. There was a problem with physics material like a wall, we changed the numbers to 0 and added the player's capsule collider, then made the friction minimum. So, the problem is solved.



Raycast:

The Raycasting system works like an invisible ray of a certain length, that is shot from a starting point (First Person Camera) towards a certain direction (foreward). This is especially useful for shooting mechanics. If the invisible ray of the raycasting system collides with colliders in the game scene, certain actions will be done (damaging enemies). The following code can be found in Weapon.cs.

void ProcessRaycast()

```
{
```

RaycastHit hit;

if (Physics.Raycast(FPCamera.transform.position, FPCamera.transform.forward, out hit, range))

رعو....

}

```
{
   CreateHitImpact(hit);
   EnemyHealth target = hit.transform.GetComponent<EnemyHealth>();
   if (target == null) return;
   target.TakeDamage(damage);
}
else
{
   return;
}
```

Enemies Facing the Target When Moving:

When creating the chasing functionality of the enemies, first we have used placeholders (the capsule gameobject in unity). Therefore, we did not worry about the faces of the enemies to look towards the player. However, once we have placed the character assets, we have seen the necessity of this functionality. To solve this challenge, we have made use of vectors. When subtracting the position of the enemy from the position of the player, we have calculated the vector between them. To allocate a direction to the enemy, we do not need the entire vector- unit vector is enough. By using the quaternion function of Unity (which is a way of defining orientation in 3D space, by using calculation methods that imply an imaginary 4D space) we have totated the orientation of the enemy, until it was facing at the direction of the unit vector we calculated (described above).



Gravity:

An important feature for our game was the gravity control ability. This has differentiated our game from other 1st person shooter games. However, the gravity control implementation was a little bit of a challenging process. We have observed that the rigidbody2D is more flexible in this aspect in comparison to the rigidbody for 3D gameobjects. **Rigidbody2D.gravityScale** is only available for rigidbody2D.

A solution we have thought of was to go to player settings and play with gravity values from there. If we went for this approach, this time the gravity would have changed for all gameobjects and not only our player- which made us worry at first. However, the only gameobject that would get affected by gravity is our player, as the enemies are moving around as NavMeshAgents and the rest of the game objects are static game environment assets.

```
void Update(){
    if(Input.GetKeyDown(KeyCode.G))
    {
        Physics.gravity *= -1;
    }
```

Coroutine on Shooting:

For the smooth change between shootings, we use coroutines to make the player wait before the next shot. They are not shooting automatically. Instead, they need to give the input each time. In order to obtain a delay in time and not directly do the operations in the speed offered by unity frame, we made use of "Coroutine". When creating coroutines there are 3 main steps that are different from a standard method. Firstly, when defining the coroutine, we use the keyword IEnumerator. To obtain the delay in time that the game requires, we can type "yield return new WaitForSeconds" and then type in the time delay we want to achieve. Finally, when calling the coroutine, we must say StartCoroutine and type in the name of it within parentheses.

```
IEnumerator Shoot()
```

```
{ canShoot = false;
if (ammoSlot.GetCurrentAmmo(ammoType)>0)
{
     PlayMuzzleFlash();
     ProcessRaycast();
     ammoSlot.ReduceCurrentAmmo(ammoType);
   }
   yield return new WaitForSeconds (timeBetweenShots);
   canShoot = true;
  }
.....
if (Input.GetMouseButtonDown(0) && canShoot==true)
  {
     StartCoroutine(Shoot());
  }
```

Libraries

Text mesh pro:

For using the canvas system with text that is suitable with the game theme, we have visited the site: "dafont.com". Here we have chosen a font style that we like, and is suitable for the game style, among the fonts that were free for personal use. After this download, we went back to Unity and used the **Text Mesh Pro Package** such that we generated the font atlas. After that, we were able to select this front from the text we included in the canvases.

Post processing:

We used post processing as a feature that might affect the performance of our game and we wanted to test it with the profiler. To use this, we downloaded the **Post Processing Package** of Unity, and applied the effects we wanted, like a blurry background and some slight modifications in lighting and saturation.



Standard Asset Pack:

From the Unity asset store, we downloaded this package- which is very old but some features are still very useful. Our first person player is taken from here. Also we took some particle effects that we modified into weapon shooting effects.

Profile Analyzer:

Profile analyzer is added to Unity editor as a library to evaluate the performance with profiler. It has two abilities as a single view and comparison. So, we can compare our profiler results to increase our performance with the result of it. It helps visualizing frame, thread and marker data. It shows min, max, median, mean and lower/upper quartile results for the selected frame.

Probuilder:

We downloaded the **Probuilder Package** from Unity's Package Manager. Pro-builder is a tool of easy and quick modeling that can be done right inside unity without using external 3D modeling software. When creating the level design for our game, we wanted to have some walls that are impossible to be passed by only jumping, so that we would force our player to use the gravity switch game mechanic.

Progrids:

We downloaded the **Progrids Package** from Unity's Package Manager. Pro-grids is a package that is mostly used together with probuilder. It allows the user to generate grids at the density and distance they desire. When used together with probuilder we can easily create walls at the distance we want by dragging the corners and snipping.

Terrain Tools:

We downloaded the **Terrain Tools Package** from Unity's Package Manager, and used it to define the boundaries of the game world. Using this tool, we have created colorful materials matching our game world and have also added lowpoly trees.

We Used Also

Universal Render Pipeline (URP):

The Universal Render Pipeline (URP) is a prebuilt Scriptable Render Pipeline, made by Unity. We have chosen this render pipeline over the Lightweight Render Pipeline (LWRP), because the asset packages that we were using required URP, and they would appear as (pink) no material objects, if we didn't do so. In order to upgrade all existing materials into URP, we had to do the following procedure: Edit \rightarrow Render Pipeline \rightarrow Universal Render Pipeline \rightarrow Upgrade Project Materials to UniversalRP Materials



Performance - Profiler

We use Unity profiler and Profile Analyzer tools at most for the performance evaluation. They give us general and object detail results. They measure CPU usage, GPU usage, rendering performance, memory usage and physics. To see the script details, the codes below are added to the code blocks at the start and end.

UnityEngine.Profiling.Profiler.BeginSample("<text>");

UnityEngine.Profiling.Profiler.EndSample();

Lower number of milliseconds on the profiler means faster games. Garbage collector allocation is the amount of memory that is allocated and needs to be cleaned up in a garbage collection, we generally want it to be 0. We see the timing also and a deep profile that shows us detailed analysis. These all provide us a view to find the part that makes our game slow or decrease the performance.

Profile analyzer is a really useful library that we added to our project because we can have single and compared views there. It is also possible to import data to profile analyzer that is recorded on profiler.

We looked at the parameters below for the results seen on profiler:

- Open vs Close Terrain Object
- Box Collider vs Mesh Collider for Environment Objects
- Open vs Close Global Volume (Post Process)
- Normal vs Increased Enemy Count
- Stay Ground vs Fly with Gravity and Look Down to Affect Rendering
- Box Collider with Increased Enemy Count

0. Game Scene Default Performance

Computer Properties for the Graphs Below:

	System Summary
Operating System	macOS Big Sur (Mid 2014)
CPU / GPU	2,2 GHz Quad-Core Intel Core i7 / Intel Iris Pro 1536 MB
RAM	16 GB 1600 MHz DDR3
Unity Version	2020.3.30f1

First of all, we see that our scene's gameplay performance is affected by computer performance so much. The graph below shows us high milliseconds with high CPU usage that results with low FPS. Lower performance is also affected by profiler usage. Because our computer properties are not enough, handling both usage of the game and profiler caused performance to be low.

🖿 Project 🛛 🗟 Console	Animation	Ĉ₀ Profiler	:
Profiler Modules -	Playmode ▼ 🧕	► Frame: 721 / 721 Clear Clear on Play Deep Profile Call Stacks	ë() ▲
Rendering Scripts Physics Animation GarbageCollector VSync Global Illumination Ul Others	33ms (30FPS) 16ms (60FPS)		
💁 Rendering			110
 Batches Count SetPass Calls Count Triangles Count Vertices Count 			1
Timeline 🔻	Live	CPU:16,566.85ms GPU:ms	
Main Thread		0ms 10ms 20ms 30ms 40ms 50ms 60ms ProfilerDriver.InvokeProfilingStateChange() (145.66ms)	
		ProfilerWindowAnalytics.ProfilingStateChange() (145.65ms)	
		ProfilerWindowAnalytics.StopCapture() (145.11ms)	
		ProfilerWindowAnalytics.RegisterEvent() (143.76ms)	
		EditorAnalytics RegisterEventWithLimit() (143.74ms)	
Render Thread		Gfx.WaitForGfxCommandsFromMainThread (16566.73ms)	
		Semaphore.WaitForSignal (16566.73ms)	
▶ Job			•

The screenshot below is the details of the profile analyzer. According to the details, physics is the most part of CPU usage.

					Profile An	alyzer							
Profile Analyzer													
Mode: Single Compare Export Close Profi													
Pull Data 16566.85m v	3.00ms									▼ Frame Summa	ry		
426	6.00ms												
0.00ms												26572	
265								268	371			26871	
								PlayerLo	pop				
▼ Filters											ms 16567	Frame	
Name Filter : All 🔻				Exclude Names	Any 🔻					Linner Quartile	56.04	20071	
Thread : Select Main Thread										Median	44 94	26690	
Depth Slice : All							Analysis Type :			Mean		20000	
Parent Marker : None							Units :	Milliseconds	•	Lower Quartile	43.88		
Analyze 2124 of 2373 markers 1 of 85							Marker Columns :	Time and Count	•			26594	
,													
Top 10 markers on median frame												16	5567
26690 ctor() Playe	rLoop	PhysicsFixedU	pcProcessing	EditorLoop DoRe In	tern Rend R	tend Rend		212ms				19	
(All depths)													
Marker Details for currently selected range										▶ Thread Summ	ary		
Marker Name										= Marker Summ	-		
FixedUpdate.PhysicsFixedUpdate			24.14							Playerl oop	ary		
Physics.Processing			23.94							Mean frame cont	ribution		
EditorLoop			16.86										
mscorlib.dll!Mono.Globalization.Unicode::SimpleCo			8.56								32.37%		
mscorlib.dll!System::Char.ToString()										Firet framo		26572	
UnityEngine.CoreModule.dll!UnityEngine.Rendering			7.55							Top 2 w by		20372	
UnityEngine.CoreModule.dll!UnityEngine.Rendering			7.53							Top 3 • by	EE 0mc	100050	
Unity.RenderPipelines.Universal.Runtime.dll!UnityE			7.53								55.8ms	20032	
Unity.RenderPipelines.Universal.Runtime.dll!UnityE			7.44								54.2mg	20032	
mscorlib.dll!System::String.StartsWith()											04.3115	20051	
Unity.RenderPipelines.Universal.Runtime.dll!UnityE												T 55	
GC.Collect													
UnityEditor.CoreModule.dll!UnityEditor.Compilation	::C 8-9					0.00		0.00					
UnityEngine.CoreModule.dll!UnityEngine::Object.In:													
mscorlib.dll!System.Globalization::CompareInfo.IsPr													
Unity.RenderPipelines.Universal.Runtime.dll!UnityE										9 8289	55 925	9.	8289
mscorlib.dll!Mono.Globalization.Unicode::SimpleCo													
.mscorlib.dll!System::String.Concat()													

Other than that, the lowest millisecond result of gameplay is below. It has higher FPS.

Profiler Modules	•	Playmode 🔻 🧕	I I	► Frame: 1	860 / 2075	Clear Clear of	on Play Deep Profile Ca	III Stacks 🔻 🖆 🛗	€
🔑 CPU Usage						Selec	ted: Assembly-CSharp.	dll!::Enemy.Update()	
Rendering Scripts Physics Animation GarbageCollector		33ms (30FPS)							
 VSync 		16ms (60FPS)							
Global Illumination				ma - mar					
UIOthers			0.0	4ms					
💁 Rendering		BAAAAA		М				A AA AAAA	
 Batches Count SetPass Calls Count Triangles Count 		/*****		171 61.83k 65.72k					
Timeline	•	Live			CPU:1	1.84ms GPU:-	-ms		
Main Thread			yerLoop (1.	79 <mark>i</mark> orLoop (1.56	PlayerLoo	op (4.10ms)	EditorLoop (3.73ms)	erLoop (1.67.00p (1.	
					.Rendering::Re	nderPipelineMan			
					erPipeline.Inter	rnalRender() (3.9			U
					rsalRenderPipe	line.Render() (3.			
					ProfilingScope	ctor() (3.93ms)			
					der Pipeline.Re				
Dondor Throad		1							

In this time, we see that the most usage for performance is because of Unity Editor and editor loop, player loop are coming after them.



1. Closing Terrain object

To compare normal performance with terrain effect, we closed terrain object because we assumed that it requires much performance. It was our assumption. When we closed the terrain object, it showed higher FPS graphs.

Profiler Modules	•	Playmode 🔻 🦲	I	➡ Frame: 274	1/2787	Clear Clear on Play	Deep Profile Call	Stacks 🔻 🖆 🛱	16
🔗 CPU Usage						Selected: A	ssembly-CSharp.dll	!::Enemy.Update(b 🚖
 Rendering Scripts Physics Animation CorbaseCollector 		33ms (30FPS)							
 VSync Global Illumination UI Others 		16ms (60FPS) _					0.03n	ns	
💁 Rendering			< <u> </u>			~ ~		77	
 Batches Count SetPass Calls Count Triangles Count 							27.89	97 14.99K 9k	
Timeline		Live			CPU:	8.01ms GPU:ms			
Main Thread			Oms , erLoop (1	.0EditorLoop (1.66m	is) Pl ie.Rende derPipeli ersalRen Profilin inderPip	J5ms ayerLoop (3.06ms) ring::RenderPipelineManaç derPipeline.Render() (2.93 derPipeline.Render() (2.93 gScopector() (2.93ms) eline PonderCameraStack	EditorLoop (1.71ms)	ayerLoop (1.53m	
Dondor Throad									•

Here, the player loop is the most part of performance usage as it is seen in profile analyzer.



2. Box collider instead of mesh collider for the objects in the environment (not the player or enemies)

Box and mesh collider change did not affect the performance so much. Normally, we expected the milliseconds to be lower when the box collider is used instead of the mesh collider. It did not graph as our assumption. This may be because we do not have so much collision effect in our game. There are environment objects, but the player does not collide with them frequently. Our enemies are working with AI in the background.



The results of the profile analyzer is again below and it has the same results as above.

• • •			Profile Ana	alyzer							
Profile Analyzer											
Mode: Single Compare Export Close Profiler Window											
Pull Data 22765 95rr x (**********************************					_			Frame Summ	arv		
Load Save Unsaved 4											
8533.00ms								Frame Count			
								Start		15608	
0.00ms 15608			[300]				1590	7 End		15907	
							PlayerLo	p.			
▼ Filters										15907	
Name Filter : All		Exclude Names :	Any 🔻								
Thread : Select Main Thread								Median		15796	
Depth Slice : All 🔻						Analysis Type :	Total	Mean	90.73		
Parent Marker : None							Milliseconds	 Lower Quartile 	11.88		
Analyze 1828 of 2079 markers , 1 of 78 threads						Marker Columns :	Time and Count		9.88	15903	
Top 10 markers on median frame											22766
15796 ctor() PlayerLoop	EditorLoop DoRende InternalF Rend	ler() RenderC Rende					85.7ms				
(All depths)								0	22766		
Marker Details for currently selected range											
,								Thread Summ	lary		
Marker Name Depth	Median Median Bar Mea		Max	Range	Count	Count Frame		▼ Marker Summ	ary		
mscorlib.dli!Mono.Globalization.Unicode::SimpleColla 11-13	6.12 6.12			0.00	162		0.00	PlayerLoop			
EditorLoop I	5.09 7.83	3.68	/05.58	701.90	89/		5.53		tribution		
UnityEngine CoreModule dill InityEngine Rendering:5 2	4.45 4.45	2.94	7.20	3.49	200		4.34				
UnityEngine.CoreModule dIII InityEngine.Rendering::F 2	4.33 4.40	3.83	7 30	3.48	200		4.34		7.12%		
Unity RenderPipelines Universal Runtime dill InityEng 4	433 440	3.83	7.30	3.48	299		4.32			15608	
Unity RenderPipelines Universal Runtime dillUnityEng 6	4.28 4.34			3.46	299						
mscorlib.dll!System::Char.ToString() 8-9	3.87 3.87								9.96ms	15825	
Unity.RenderPipelines.Universal.Runtime.dll!UnityEng 8							3.49		9.61ms	15901	
UnityEditor.CoreModule.dll!UnityEditor.Compilation::C 8-9									9.51ms	15898	
mscorlib.dll!System.Globalization::CompareInfo.IsPref 10											
mscorlib.dll!Mono.Globalization.Unicode::SimpleColla 14										i i i i i i i i i i i i i i i i i i i	5.5337
mscorlib.dll!System::String.Concat() 8-11											
Unity.RenderPipelines.Universal.Runtime.dll!UnityEng 10											
Mono.JIT 3-14											
UnityEditor.CoreModule.dll!UnityEditor.Scripting.Scrip 10								5 0700			
Unity.RenderPipelines.Core.Runtime.dll!UnityEngine.F 10-11								5.3736			
Unity.RenderPipelines.Universal.Runtime.dll!UnityEng 13											

3. Closing global volume (post process) object

When we compared this change with default results of our scene, it shows us that post processing does not affect the performance so much. We expected it to result in lower milliseconds, but it did not change. This might be because we did not use so many features of post processing. Color adjustments might have affected the performance more.



The details of the profile analyzer is below. It shows the most performance usage is by player loop.

Profile Analyzer															
Mode: Single Compare Export Close Profile															
Pull Data 24682.57m -												▼ Frame Summa			
Load Save Unsaved 7	6.00ms														
8533	.00ms														
0.00ms														24074	
2407	4					[300]				24	373			24373	
										Player	Loop				
Tilters														24373	
Name Filter : All 🔻					ie Names :	Any 🔻									
Thread : Select Main Inread														24282	
Depth Slice : All 👻									Analysis Type :	Total					
Parent Marker: None										Milliseconds					
Analyze 2059 of 2288 markers , 1 of 85 t										Time and Count				24363	
Top 10 markers on median frame														-	24683
24282 ctor()	ditorl.oop	Playerl con	DoRendulIntern	alERenderiiR	ender(Ren	nde				66.2m					
(All depths)	Contorizoop	ridjenzoop	orienalintern	an pronact opt	enderenter								24602	1.1.1	10.139
													24003		
V Marker Details for currently selected range															
· marter betane for earenty encoted range												Thread Summa	ary		
Marker Name												Thread Summa Marker Summa	ary arv		
Marker Name PlayerLoop		Median 4.80	Median Bar M	Mean N 4.93 4	1in .09	Max 20.84	Range 16.74	Count 897		At Median Frame 4.59		 Thread Summa Marker Summa PlayerLoop 	ary ary		
Marker Name PlayerLoop Unity.TextMeshPro.dll!TMPro::TextMeshProUGUI.Re		Median 4.80 4.78	Median Bar M	Mean N 4.93 4 4.78 4	1in .09 .78	Max 20.84 4.78	Range 16.74 0.00	Count 897 6		At Median Frame 4.59 0.00		Thread Summa Marker Summa PlayerLoop Mean frame cont	ary ary ribution		
Marker Name PlayerLoop Unity.TextMeshPro.dllTMPro::TextMeshProUGUI.Re mscorlib.dll!Mono.Globalization.Unicode::SimpleColl		Median 4.80 4.78 4.77	Median Bar M 2 2 2	Mean N 4.93 4 4.78 4 4.77 4	1in .09 .78 .77	Max 20.84 4.78 4.77	Range 16.74 0.00 0.00	Count 897 6 162		At Median Frame 4.59 0.00 0.00	Î	Thread Summa Marker Summa PlayerLoop Mean frame cont	ary ary ribution		
Marker Vane PlayerLoop Unity TextMeshPro.dll1TMPro::TextMeshProUGUI.Rt msoriib.dllMono.Globalization.Unicode::SimpleColl Unity.TextMeshPro.dll1TMPro::TextMeshProUGUI.Or		Median 4.80 4.78 4.77 4.67	Median Bar M	Mean N 4.93 4 4.78 4 4.77 4 4.67 4	4in .09 .78 .77 .67	Max 20.84 4.78 4.77 4.67	Range 16.74 0.00 0.00 0.00	Count 897 6 162 3		At Median Frame 4.59 0.00 0.00 0.00	î	Thread Summa Marker Summa PlayerLoop Mean frame cont	iry iry ribution 5.10%		
Marker Name PlayerLoop Unity, TextMeshPro.dllTMPro:TextMeshProUGULR macorib adllMono Globalization Unicode-SimpleColl Unity, TextMeshPro.dllTMPro:TextMeshProUGUL0 Unity, TextMeshPro.dllTMPro:TMP.TextParseInput	Depth 1 at 10 at 11-13 af 11 T 12	Median 4.80 4.78 4.77 4.67 3.51	Median Bar A	Mean N 4.93 4 4.78 4 4.77 4 4.67 4 3.51 3	4in .09 .78 .77 .67 .51	Max 20.84 4.78 4.77 4.67 3.51	Range 16.74 0.00 0.00 0.00 0.00	Count 897 6 162 3 3		At Median Frame 4.59 0.00 0.00 0.00 0.00 0.00	î	 Thread Summa Marker Summa PlayerLoop Mean frame cont First frame 	ry ribution 5.10%	24074	
Marker Name PlayerLoop Unity, TextMeshPro.dllTMPro:TextMeshProUGUI.R mscorlib.dlliNono Globalization.Unicode:SimpleColl Unity, TextMeshPro.dllTMPro::TextMeshProUGUI.O Unity, TextMeshPro.dllTMPro::TMP_TextParseInput TMP Parse Text	Depth 1 11 11-13 11-13 11-13 11 11 12 13	Median 4.80 4.78 4.77 4.67 3.51 3.51	Median Bar A	Mean N 4.93 4 4.78 4 4.77 4 4.67 4 3.51 3 3.51 3	1in .09 .78 .77 .67 .51 .51	Max 20.84 4.78 4.77 4.67 3.51 3.51	Range 16.74 0.00 0.00 0.00 0.00 0.00	Count 897 6 162 3 3 3 3		At Median Frame 4.59 0.00 0.00 0.00 0.00 0.00 0.00	î	Thread Summa Marker Summa PlayerLoop Mean frame cont First frame Top 3 • by	ribution 5.10%	24074	
Marker Name RayerLoop Unity, TextMeshPro.dllTMPro:TextMeshProUGULR mscorii dilMono Globalization Unicode SimpleColl Unity, TextMeshPro.dllTMPcr:TMP_TextParseInput TMP Pane Text mscorii dilTSystem-String StartsWith()	Depth 1 t 10 a 11-13 lif 11 T 12 13 8-9	Median 4.80 4.78 4.77 4.67 3.51 3.51 3.42	Median Bar A	Mean N 4.93 4 4.78 4 4.77 4 4.67 4 3.51 3 3.51 3 3.42 3	4in .09 .78 .77 .67 .51 .51 .42	Max 20.84 4.78 4.77 4.67 3.51 3.51 3.42	Range 16.74 0.00 0.00 0.00 0.00 0.00 0.00	Count 897 6 162 3 3 3 3 108		At Median Frame 4.59 0.00 0.00 0.00 0.00 0.00 0.00 0.00	i	Thread Summa Marker Summa PlayerLoop Mean frame cont First frame Top 3 • by	iry ibution 5.10% frame cost	24074 Is 24279	
Marker Name Blayet dop Unity TextMeshPro.dllTMPro:TextMeshProUGUI.Rd unity TextMeshPro.dllTMPro:TextMeshProUGUI.O Unity TextMeshPro.dllTMPro:TextMeshProUGUI.O Unity TextMeshPro.dllTMPro:TMP.TextParseInput TMP Parae Text mescrift.dllSystem:String StarsWith) UnityEngine.CoreModule.dllUnityEngine.Rendering:	Depth 1 t 10 a 11-13 if 11 T 12 13 8-9 F 2	Median 4.80 4.78 4.77 4.67 3.51 3.51 3.51 3.42 3.26	Median Bar N	Mean N 4.93 4 4.78 4 4.77 4 4.67 4 3.51 3 3.51 3 3.42 3 3.29 3 3.29 3	4in .09 .78 .77 .67 .51 .51 .42 .00	Max 20.84 4.78 4.77 4.67 3.51 3.51 3.42 4.54	Range 16.74 0.00 0.00 0.00 0.00 0.00 0.00 1.54	Count 897 6 162 3 3 3 108 299		At Median Frame 4.59 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0	î	Thread Summa Marker Summa PlayerLoop Mean frame cont First frame Top 3 • by	ribution 5.10% frame cost 20.8ms 12.3ms	24074 Is 24279 24278	
Marker Name PlayerLoop Unity, TextMeshPro.dllTMPro::TextMeshProUGULR mscottib.dllNono.Globalization.Unicode:SimpleColl Unity, TextMeshPro.dllTMPro::TextMeshProUGULO Unity, TextMeshPro.dllTMPro::TMP_TextParsIenput TMP Parse Text mscottib.dllSystem::String.StarlsWth) UnityEngine.CoreModule.dllUnityEngine.Rendering: UnityEngine.CoreModule.dllUnityEngine.Rendering:	Depth 1 1 10 a 11-13 if 11 T 12 13 8-9 F 2 F 3	Median 4.80 4.78 4.77 4.67 3.51 3.51 3.51 3.42 3.26 3.25	Median Bar N	Mean N 4.93 4 4.78 4 4.77 4 4.67 4 3.51 3 3.51 3 3.42 3 3.29 3 3.29 3 3.28 2 2.75 2 5.75 2 5.	4in .09 .78 .77 .67 .51 .51 .42 .00 .98	Max 20.84 4.78 4.77 4.67 3.51 3.51 3.51 3.42 4.54 4.52	Range 16.74 0.00 0.00 0.00 0.00 0.00 0.00 1.54 1.54	Count 897 6 162 3 3 3 108 299 299		At Median Frame 4.59 0.00 0.00 0.00 0.00 0.00 0.00 0.00 3.34 3.33	i	Thread Summa Marker Summa PlayerLoop Mean frame cont First frame Top 3 • by	ry ribution 5.10% frame cost 20.8ms 12.3ms 9.80ms	24074 ts 24279 24278 24155	
Marker Name Blayer Loop Unity, TextMeshPro dllTMPro: TextMeshProUGULR unity, TextMeshPro dllTMPro: TextMeshProUGULO Unity, TextMeshPro dllTMPro: TextMeshProUGULO Unity, TextMeshPro dllTMPro: Thy Text Parsen maconth dllSystem: String StartsWith) Unity Engine. CoreModule dllUnityEngine. Reneting: Unity RenderPipelines. Universit Runtimo dllUnityEngi Unity RenderPipelines. Universit Runtimo dllUnityEngi	Depth 1 1 1 1 1 1 1 1 1 8-9 F 2 F 3 9 4	Median 4.80 4.78 4.77 4.67 3.51 3.51 3.51 3.42 3.26 3.25 3.25 3.25 3.25	Median Bar N	Mean N 4.93 4 4.78 4 4.77 4 4.67 4 4.67 4 3.51 3 3.51 3 3.29 3 3.29 3 3.28 2 3.28 2 3.28 2	4in .09 .78 .77 .67 .51 .51 .51 .42 .00 .98 .98	Max 20.84 4.78 4.77 4.67 3.51 3.51 3.51 3.42 4.54 4.52 4.52	Range 16.74 0.00 0.00 0.00 0.00 0.00 0.00 1.54 1.54 1.54	Count 897 6 162 3 3 3 108 299 299 299 299		At Median Frame 4.59 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0	î	Thread Summa Marker Summa Marker Summa PlayerLoop Mean frame cont First frame Top 3 • by	rry ribution 5.10% frame cost 20.8ms 12.3ms 9.80ms	24074 Is 24279 24278 24155	
Marker Name Blayet cop Unity. TextMeshPro.dllTMPro:TextMeshProUGUI.Re macorita.dllMono.Globalization.Unicode:SimpeCol Unity.TextMeshPro.dllTMPro:TextMeshProUGUI.O Unity.TextMeshPro.dllTMPro:TextMeshProUGUI.O Unity.TextMeshPro.dllTMPro:TextMeshProUGUI.O Unity.TextMeshPro.dllTMPro:TextMeshProUGUI.O Unity.Engine.ComModule.dllUnityEngine.Rendering: Unity.RenderPopelines.Universal.Runtime.dllUnityEngine.Unity.RenderPopelines.Universal.Runtime.dllUnityEngine.	Depth 1 1 1 1 1 1 1 1 1 1 8-9 F 2 F 3 9 4 9 6 0 0 0 0 0 0 0 0 0 0 0 0 0	Median 4.80 4.78 4.77 4.67 3.51 3.51 3.51 3.42 3.26 3.25 3.25 3.25 3.25 3.20	Median Bar	Mean N 4.93 4 4.78 4 4.77 4 4.67 4 3.51 3 3.42 3 3.28 2 3.28 2 3.28 2 3.28 2	4lin .009 .778 .77 .67 .51 .51 .51 .51 .51 .51 .00 .98 .98 .93 .93	Max 20.84 4.78 4.77 3.51 3.51 3.42 4.54 4.52 4.52 4.52 4.52	Range 16.74 0.00 0.00 0.00 0.00 0.00 0.00 1.54 1.54 1.54 1.54 1.52	Count 897 6 162 3 3 3 108 299 299 299 299 299		At Median Frame 4.59 0.00 0.00 0.00 0.00 0.00 0.00 0.00 3.34 3.33 3.33	î	► Thread Summa ► Marker Summa PlayerLoop Mean frame cont First frame Top 3 ▼ by	rry ribution 5.10% frame cost 20.8ms 12.3ms 9.80ms	24074 Is 24279 24278 24155	
Marker Name PlayerLoop Unity, TextMeshPro.dllTMPro:TextMeshProUGULR unity, TextMeshPro.dllTMPro:TextMeshProUGULR Unity, TextMeshPro.dllTMPro:TextMeshProUGULO Unity, TextMeshPro.dllTMPro:TextMeshProUGULO Unity, TextMeshPro.dllTMPro:TextMeshProUGULO Unity, FanderBrogene, CoreModule dllUnityEngine Rendering: Unity, RenderBrogenies, Universal: Runtime.dllUnityEn Unity, RenderBrogenies, Universal: Runtime.dllUnityEn Marker Statement (Statement)	Depth 1 1 10 1 11-13 11 11 12 13 8-9 F 2 F 2 F 3 g 4 g 6 8-9 - 9 - 9 - 9 - 9 - 9 - 9 - 9 -	Median 4.80 4.77 4.67 3.51 3.51 3.26 3.25 3.25 3.25 3.25 3.20 2.71	Median Bar M	Mean N 4.93 4 4.77 4 4.77 4 4.67 4 3.51 3 3.51 3 3.29 3 3.29 3 3.28 2 3.28 2 3.29 2 3.28 2 3.29 4 3.29 4 4.29 4 4.29 4 4.29 4 4.29 4 3.29 4 3.29 4 3.29 4 3.29 4 3.29 4 20 4 3.29 4 4.29 4 4.29 4 4.29 4 4.29 4 3.29	4in .009 .78 .77 .67 .51 .51 .51 .51 .00 .98 .98 .98 .93 .71	Max 20.84 4.78 4.67 3.51 3.51 3.42 4.54 4.52 4.52 4.52 4.45 2.71 2.67	Range 16.74 0.00 0.00 0.00 0.00 0.00 1.54 1.54 1.54 1.54 1.52 0.00	Count 897 6 162 3 3 3 3 108 299 299 299 299 299 299 200		At Modian Frame 4.59 0.00 0.00 0.00 0.00 0.00 0.00 0.00 3.34 3.33 3.33 3.28 0.00 0.21	î	► Thread Summa ► Marker Summa PlayerLoop Mean frame cont First frame Top 3 ▼ by	iry ribution 5.10% frame cost 20.8ms 12.3ms 9.80ms	24074 Is 24279 24278 24155	
Marker Name Biayett cop Unity, TextMeshPro,dIITMPro:TextMeshProUGUI Re macoliti.dIIMono Globalization.Unicode-SimpleColi Unity, TextMeshPro.dIITMPro:TextMeshProUGUI.Of Unity, TextMeshPro.dIITMPro:TextParsenput TMP Paras Text mocoliti.dIISystem:String StarsWith() Unity, Engline.CoreModule dIIUnityEngine.Rendering: Unity, RenderPipelines.Universal.Runtime.dIIUnityEngine.Rendering: Unity, RenderPipelines.Universal.Runtime.dIIUnityEngine.Rendering: Unity, RenderPipelines.Universal.Runtime.dIIUnityEngine.Rendering: Unity, RenderPipelines.Universal.Runtime.dIIUnityEngine.Rendering: Unity, RenderPipelines.Universal.Runtime.dIIUnityEngine.Rendering: Unity, RenderPipelines.Universal.Runtime.dIIUnityEngine.Rendering: Unity, RenderPipelines.Universal.Runtime.dIIUnityEngine.Rendering: Unity, RenderPipelines.Universal.Runtime.dIIUnityEngine.Rendering: Unity, RenderPipelines.Universal.Runtime.dIIUnityEngine.Rendering: Unity, RenderPipelines.Universal.Runtime.dIIUnityEngine.Runtime.dIIIUnityEngine.Runtime.dIIIUnityEngine.Runtime.dIIIUnityEngine.Runtime.dIIIIUnityEngine.Runtime.dIIIIInityEngine.Runtime.dIIIIInityEngine.Runtime.dIIIInityEngine.Runtime.dIIIInityEngine.Runtime.dIIIInityEngine.Runtime.dIIIInityEngine.Runtime.dIIIInityEngine.Runtime.dIIIInityEngine.Runtime.dIIIInityEngine.Runtime.dIIIInityEngine.Runtime.dIIIInityEngine.Runtime.dIIIInityEngine.Runtime.dIIIInityEngine.Runtime.dIIIInityEngine.Runtime.dIIInityEngine.Runtime.dIIIInityEngine.Runtime.dIIIInityEngine.Runtime.dIIIInityEngine.Runtime.dIIIInityEngine.Runtime.dIIIInityEngine.Runtime.dIIIInityEngine.Runtime.dIIIInityEngine.Runtime.dIIInityEngine.Runtime.dIIIInityEngine.Runtime.dIIIInityEngine.Runtime.dIIIIInityEngine.Runtime.dIIInityEngine.Runtime.dIIInityEngine.Runtime.dIIInityEngine.Runtime.dIIInityEngine.Runtime.dIIInityEngine.Runtime.dIIInityEngine.Runtime.dIIInityEngine.Runtime.dIIInityEngine.Runtime.dIIInityEngine.Runtime.dIIInityEngine.Runtime.dIIInityEngine.Runtime.dIIInityEngine.Runtime.dIIInityEngine.Runtime.dIIInityEngine.Runtime.dIIInityEngine.Runti	Depth 1 1 1 1 1 1 1 1 1 1 1 1 1	Median 4.80 4.78 4.77 4.67 3.51 3.51 3.51 3.26 3.25 3.25 3.20 2.71 2.64 2.62	Median Bar M 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 6 5 5 6 5 6	Mean N 44.93 4 4778 4 4.777 4 4.677 4 3.51 3 3.52 3 3.29 3 3.28 2 3.28 2 3.223 2 2.71 2 2.65 2 2.65 2	4in .009 .778 .77 .67 .51 .51 .51 .00 .98 .98 .98 .98 .93 .71 .38	Max 20.84 4.78 4.77 3.51 3.51 3.42 4.54 4.52 4.52 4.52 4.52 4.52 2.71 3.67 2.62	Range 16.74 0.00 0.00 0.00 0.00 0.00 0.00 1.54 1.54 1.54 1.54 1.54 1.52 0.00 1.29	Count 897 6 162 3 3 3 108 299 299 299 299 299 108 299		At Median Frame 4.59 0.00 0.00 0.00 0.00 0.00 0.00 3.34 3.33 3.33 3.28 0.000 0.00	i	► Thread Summa ► Marker Summa PlayerLoop Mean frame cont First frame Top 3 ▼ by	ribution 5.10% frame cost 20.8ms 12.3ms 9.80ms	24074 is 24279 24278 24155	
Marker Name Rayer Loop Unity, TextMeshPro.dllTMPro:TextMeshProUGULR mscorid allMono Globalization Unicode SimpleColl Unity, TextMeshPro.dllTMPc:TextMeshProUGUL Unity, TextMeshPro.dllTMPc:TMP_TextParseInput TMP Pane Text mscorid allSystem:String StartsWith() UnityEgnine.CoreModule dllUnityEngine.Rendering: Unity, RenderPipelines.Universal.Runtime.dllUnityEn Unity, RenderPipelines.Universal.Runtime.dllUnityEn Unity, RenderPipelines.Universal.Runtime.dllUnityEn Unity, RenderPipelines.Universal.Runtime.dllUnityEn Unity, RenderPipelines.Universal.Runtime.dllUnityEn unity, RenderPipelines.Universal.Runtime.dllUnityEn mscorid.dllUSystem:String.FastAllocateString()	Depth 1 t 10 a 11-13 if 11 T 12 13 8-9 F 2 F 3 g 4 g 6 8-9 g 8 9-9 g 8 9-97 C 0 2 0	Median 4.80 4.78 4.77 4.67 3.51 3.51 3.26 3.26 3.25 3.20 2.71 2.64 2.63 2.01	Median Bar M A A A A A A A A A A A A A A A A A A A	Mean N 4,93 4 4,77 4 4,77 4 4,77 4 4,67 4 3,51 3 3,51 3 3,51 3 3,51 3 3,52 3 3,28 2 3,28 2 3,28 2 3,28 2 3,28 2 3,27 1 2,265 2 2,265 2 2,265 2 2,265 2	Min .09 .78 .77 .51 .51 .51 .51 .51 .51 .00 .98 .98 .98 .98 .98 .93 .71 .38 .63 .24	Max 20.84 4.77 4.67 3.51 3.51 3.42 4.54 4.52 4.52 4.52 4.45 2.71 3.67 2.63 2.24	Range 16.74 0.00 0.00 0.00 0.00 0.00 1.54 1.54 1.54 1.54 1.52 0.00 1.29 0.00	Count 897 6 162 3 3 3 108 299 299 299 299 299 299 108 299 108		At Median Frame 4.59 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0	i	► Thread Summa ► Marker Summa PlayerLoop Mean frame cont First frame Top 3 ▼ by	ribution 5.10% frame cost 20.8ms 12.3ms 9.80ms	24074 Is 24279 24278 24155	
Marker Name Blayert oop Unity, TextMeshPro.dllTMPro:TextMeshProUGULR mscorib.dllMono Globalization Unicode-SimpleColl Unity, TextMeshPro.dllTMPro:TextMeshProUGULR Unity, TextMeshPro.dllTMPro:TextMeshProUGULR Unity, TextMeshPro.dllTMPro:TextProsten Unity, FenderStrondllTMProTextProgram Unity, BenderPropilines.Universal:Runtime.dllUnityEn Unity, RenderPropilines.Universal:Runtime.dllUnityEn Unity, RenderPropilines.Universal:Runtime.dllUnityEn Unity, RenderPropilines.Universal:Runtime.dllUnityEn Unity, RenderPropilines.Universal:Runtime.dllUnityEn Unity, RenderPropilines.Universal:Runtime.dllUnityEn Unity, RenderPropilines.Universal:Runtime.dllUnityEn Unity.RenderPropilines.Universal:Runtime.dllUnityEn Unity.RenderPropilines.Universal:Runtime.dllUnityEn UnityEnderPropilines.Universal:Runtime.dllUnityEn UnityEnderPropilines.Universal:Runtime.dllUnityEn UnityEnderDerDerDerDerDerDerDerDerDerDerDerDerDe	Depth 1 t 10 a 11-13 ii 11 T 12 13 8-9 F 2 F 3 g 4 g 6 8-9 g 8 9-17 C 8-9 D 7	Median 4.80 4.78 4.77 4.67 3.51 3.51 3.25 3.25 3.25 3.25 3.20 2.71 2.64 2.63 2.24 2.63 2.24	Median Bar N	Mean N 4.93 4 4.78 4 4.77 4 4.67 4 4.67 4 4.67 4 4.67 4 4.67 4 3.51 3 3.51 3 3.29 3 3.29 3 3.29 3 3.28 2 2.71 2 2.63 2 2.65 2 2.75 2 2.65 2 2.75 2 3.75 2 3.75 2 3.75 2 3.75 2 3.75 2 2.75 2 2.	Nin .09 .78 .77 .51 .51 .51 .51 .51 .51 .51 .51 .51 .51	Max 20.84 4.78 4.77 4.67 3.51 3.51 3.51 4.54 4.52 4.52 4.52 4.52 2.71 3.67 2.63 2.24	Range 16.74 0.00 0.00 0.00 0.00 0.00 1.54 1.54 1.54 1.52 0.00 1.29 0.00 0.00	Count 897 6 162 3 3 108 299 299 299 299 299 299 299 108 299 114 2		At Median Frame 4 59 0 00 0 00 0 00 0 00 0 00 0 00 3 34 3 33 3 28 3 28 0 00 2 81 0 00 0 000 0 00 0 00	i	■ Thread Summa Marker Summa PlayerLoop Mean frame cont First frame Top 3 ● by	rry ribution 5.10% frame cost 20.8ms 12.3ms 9.80ms	24074 is 24279 24278 24155	20.836
Marker Name Blayet dop Unity TextMeshPro.dllTMPro:TextMeshProUGUI.Re macorita dillhono Gitobalization Unicode:SimpleColl Unity, TextMeshPro.dllTMPro:TextMeshProUGUI.O Unity, TextMeshPro.dllTMPro:TextMeshProUGUI.O Unity, TextMeshPro.dllTMPro:TextMeshProUGUI.O Unity, TextMeshPro.dllTMPro:TextMeshProUGUI.O Unity, RenderProlimes.Universal Runtime.dllUnityEn macoritib.dllSystem:String.StartsWith() Unity, RenderProjetines.Universal Runtime.dllUnityEn macoritib.dllSystem:String.StartsMittor.dllUnityEn macoritib.dllSystem:String.StartsMittor.dllUnityEn macoritib.dllSystem:String.StartsMittor.dllUnityEn macoritib.dllSystem:String.StartsMittor.dllUnityEn macoritib.dllSystem:String.StartsOctor.Compation: Assembly-CSharp.dll:EnemyAttack.AttackHitEvent fC cclied	Depth 1 1 1 1 1 1 1 1 1 1 1 1 1	Median 4.80 4.78 4.77 4.67 3.51 3.51 3.42 3.26 3.25 3.20 2.71 2.63 2.26 2.63 2.24 2.22 2.19	Median Bar A	Mean N 4.93 4 4.78 4 4.77 4 4.77 4 4.77 4 4.77 4 4.77 4 3.51 3 3.51 3 3.20 3 3.20 3 3.28 2 2.71 2 2.65 2 2.65 2 2.24 2 2.21 2 2.22 2	4in 0.09 .77 .67 .51 .51 .51 .00 .98 .93 .93 .93 .63 .63 .24 .22 .21 .9	Max 20.84 4.78 4.77 4.67 3.51 3.51 3.51 3.51 4.54 4.52 4.52 4.52 2.71 3.67 2.63 2.24 2.22 2.19	Range 16.74 0.00 0.00 0.00 0.00 0.00 1.54 1.54 1.54 1.54 1.52 0.00 0.54 1.54 1.52 0.00	Count 8897 6 162 3 3 3 108 299 299 299 299 299 108 299 108 299 114 2 1		At Median Frame 4.59 0.00 0.00 0.00 0.00 0.00 0.00 0.00 3.34 3.33 3.33	i	■ Thread Summa ■ Marker Summa PlayerLoop Mean frame cont First frame Top 3 ● by 4.0039	ry ribution 5.10% frame cost 20.8ms 12.3ms 9.80ms	24074 Is 24279 24278 24155	

4. Increasing enemy count

Increased enemy count decreases the performance just a little. We multiplied the enemy count by 2 for our test case. The graphs below show that the FPS result does not change so much, but calling enemy script increases which decrease the performance just by a small amount.

Profiler Modules	▼ Playmode ▼ 🧲) 🔰 🕨 🍽 Frame: 206	67 / 2151 Clear Clear on Selected:	Play Deep Profile Call Stacks Assembly-CSharp.dll!::Enemy	▼ 🖆 🛗 Av .Update() ▲ Ap Update()	vatar oply Root M pdate Mode
 Scripts Physics Animation 	33ms (30FPS)					ulling Mode
 GarbageCollector VSync Global Illumination UI Others 	16ms (60FPS)			0 10ms	Ec Is M Ca	lit Collider Trigger aterial enter
Rendering Batches Count SetPass Calls Count Triangles Count				140 57	Ra He Di	idius eight rection
Timeline	▼ Live		CPU:16.42ms GPU:ms	S		Enem
Main Thread		Oms oop (1 EditorLoop (4.39ms)	ms 10ms PlayerLoop (5.61ms)	15ms EditorLoop (3.89ms)	prLoop (4.27 A	amage
			Rendering::RenderPipelineMan	horyC		Enem
			salRenderPipeline.Render() (5. ProfilingScopector() (5.36ms) derPipeline.RenderCameraSta		So Hi	cript t Poits
Render Thread	lerSingleCa ersalRende		iandsFromi mandsFromMnderS WaitForSig J.WaitForSigniversa	SingleCame disErc alRenderP ditEors ditEors	So CI	Finem Cript hase Range
▶ Job					- Tu	ırn Speed

Below we see the hierarchy details with the enemy script because we added profiler code to our enemy script.

Profiler Modules 🔹	Playmode •	7 🖲 I4 🕨	➡ Frame: 2	067 / 2151 Cle	ear Clear on I	Play Deep Prot	ïle Call Stacks 🔻 🖆 🛗	Avatar
CPU Usage ■ Rendering == ■ Scripts ==	22775 (20)				Selected:	Assembly-CSh	arp.dll!::Enemy.Update() 🔺	Apply Root M Update Mode Culling Mode
Physics Animation GarbageCollector	33ms (30)	-PS)						Edit Collider
VSync Global Illumination UI Others	16ms (60F	PS)				0.10ms		ls Trigger Material Center
💁 Rendering		11 1811	444 4					Radius
 Batches Count SetPass Calls Count Tring lag Opurt 						140 44.1 51.4 57		Direction
Hierarchy	Live Main	Thread		16.42ms_GDU:-	-me 0		Details v	🔍 📕 🖌 Enem
				10.42113 01 0.		* La		
Overview	Total	Self	Calls	GC Alloc	Time ms	Self ms		Damage
EditorLoop	52.2%	52.2%		0 B	8.58	8.58		Enor
 UnitvEngine CoreModule. 	diliU 32.7%	0.0%		0 B	5.38	0.00		
▼ Update.ScriptRunBehavio	ourU 2.9%	0.0%		28 B	0.49	0.00		
BehaviourUpdate	2.9%	0.1%		28 B	0.49	0.02		Hit Poits
EventSystem.Update	e() 1.1%	0.0%		0 B	0.18	0.00		🔻 🖽 🖌 Enem
Enemy.Update()	0.5%	0.0%	11	0 B	0.09	0.00		
Weapon.Update()	0.5%	0.0%		28 B	0.09	0.00		
DebugUpdater.Upda EirstPersonControlle	te() 0.2%	0.0%		0 B	0.03	0.00		Chase Range
► FlashLightSystem Lin	ndat 0.0%	0.0%		0.8	0.03	0.00		Turn Speed
D' I I D' I D'	0.070	0.0%		0.0	0.01	0.00	-	

Again profile analyzer results are below. The player loop is the most performance using criteria.

				Profile Ana	alyzer									
Profile Analyzer														
Mode: Single Compare Export Close Profiler Window														
Pull Data 39775.18rr -	_			_	_	_				▼ Frame Summa				
Load Save Unsaved 6														
17066.00ms														
0.00ms												19586		
19586				[300]				19	885			19885		
								Playert	oop					
												Frame		
▼ Filters												19885		
Name Filter: All -			Exclude Names :	Any 👻										
Inread : Select Main Thead										Median	12.01	19659		
Depth Slice : All 👻							Analysis Type :	Total		Mean	147.00			
Parent Marker : None								Milliseconds		Lower Quartile	11.87			
Analyze 1768 of 2015 markers , 1 of 83 threads							Marker Columns :	Time and Count			10.99	19796		
▼ Top 10 markers on median frame												-	39775	
19659 ctor() EditorLoop	PlayerLoop	DoRend Internal Ren	der(Render(Rende					71.9m						
(All depths)										0	39775	-		
Marker Details for currently selected range										-				
Marker Name Deoth											.,			
PlayerLoop 1		5.33								Marker Summa	iry			
mscorlib.dlllMono.Globalization.Unicode::SimpleColla 11-13		4.47								Moon fromo cont				
UnityEngine.CoreModule.dll!UnityEngine.Rendering::F 2		3.81												
UnityEngine.CoreModule.dll!UnityEngine.Rendering::F 3		3.80									3.62%			
Unity.RenderPipelines.Universal.Runtime.dll!UnityEng 4		3.80								Flant for me		10500		
Unity.RenderPipelines.Universal.Runtime.dll!UnityEng 6		3.74								First frame		19280		
mscorlib.dll!System::String.StartsWith() 8-9										Top 3 V by	rame cost	5		
Unity.RenderPipelines.Universal.Runtime.dll!UnityEng 8												19644		
mscorlib.dll!System::Char.ToString() 8-9											0.34ms	19643		
UnityEditor.CoreModule.dlllUnityEditor.Compilation::C 8-9											17.27ms	19798		
mscorlib.dll!System.Globalization::CompareInfo.IsPref 10												-		
Unity.RenderPipelines.Universal.Runtime.dll!UnityEng 10														
UnityEngine.UI.dll!UnityEngine.EventSystems::Execut 10-11														
mscorlib.dll!Mono.Globalization.Unicode::SimpleColla 14														
UnityEngine.UI.dll!UnityEngine.UI::Graphic.CrossFade 16-17														
GC.Collect 4										4.65	11.627		4.65	
mscorlib.dll!System::String.Concat() 8-11														
UnityEditor.CoreModule.dlllUnityEditor.Scripting.Scrip 10	1.06	1.06	1.06	1.06	0.00	1	1	0.00	•			_		

5. Flying with gravity and looking down which affects rendering

Flying with gravity does not affect performance so much, but it affects rendering. When flying is started and the player looks down, rendering increases about 2 times because the player sees so many objects from the top view. The area we see in the red box is the result of this. On the other hand, below it is seen the rendering results are lost when flying is continued to a much higher place because we do not see any object any more.

Profiler Modules	Playmode • • I I I Frame: 2990 / 2990 Clear Clear on Play Deep Profile Call Stacks • III 33ms (30FPS) Ifoms (60FPS) Ifoms (60FPS) Ifoms (60FPS) Ifoms (60FPS)	
Rendering Batches Count SetPass Calls Count Triangles Count Verices Count Timeline	Live CPU:48.94ms GPU:ms #	-
Main Thread	Oma JSms (Loop (455.9 Court UnityEditor,CoreModule.dl/UnityEditorInternats:ProfilerOrlver.InvolveProfilergStateChanget) (48.85ms) ProfilerWindowAnalytics.StepCepture() (48.64ms) ProfilerWindowAnalytics.RegisterEventWithLimit() (47.79ms) EditorAnalytics.RegisterEventWithLimit() (47.79ms) EditorAnalytics.RegisterEventWithLimit() (47.69ms)	Î

Profile analyzer details are given below.

Pridite Analyzer Under Sanze Land Expont Expont Expont Expont Expont Expont Expont Pridite Window Frame Summary Frame Summary Frame	• • •						F	Profile Ana	lyzer								
Mode Expont	Profile Analyzer																:
Pull Data 492.69ms 260.00m 2600 260	Mode: Single																
Save Unaved1 2000 3000 2001	Pull Data	492.69ms 🔻		_	_	_	_	_	_	_				▼ Frame Summar	v		
• Fitze	Load Save	Unsaved 1 266.00	Dms														
Andread of the second of th		133.0	Dms														
10000 2001 2000 2000 2001 2000 2000 0001 0001 0001 0001 0011 0001 0001 0001 0011 0001 0001 0001 0011 0001 0001 0001 0011 0001 0001 1 0011 0001 0001 1 0001 0011 0001 0001 1 0001 0011 0001 0001 1 0001 0011 0001 0001 1 0001 0011 0001 1 0001 0001 0011 0001 1 0001 0001 0011 0001 1 0001 0001 0011 0001 1 0001 0001 0011 0001 1 0001 0001 0011 0001 1 0001 0001 0011 0001 1 0001 0011 0011 0001 1 0001 0011 0011 0001 1 0001 0001 0011 0001 1 0001 0001 0011		0.00ms														2691	
		2691						[300]			-	29	990			2990	
 								UnityEd	tor.CoreModu	ile.dll!UnityE	ditor::EditorAnalyti	cs.RegisterEventWithLir	nit()				
Priefd Max 426.20 2880 Dump Titlet* All Exclude Names1: Arg Max 426.20 2880 Depth Siles: Seles: Main Titletad Max 426.20 2880 Depth Siles: All Seles: Main Titletad Main 8.03 2935 Parent Marker: None Units: MillineGonds 10.21 None 7.00 2006 Analysis Type: Total MillineGonds MillineGonds 10.21 None 7.00 2008 You Ponatewers on median frame 2008 Marker Columns: Time and Count 7.00 2008 Val depths Parent Marker Son median frame Marker Name Marker Marker Marker Marker Anne Depth Median Bar Mean Min Max Range Count Count Frame At Median Frame UnityEditor:CoewAdula: Dipph Median Bar Mean Min Max Range Count Count Frame At Median Frame UnityEditor:CoewAdula: Dipph Median Bar Mean Min Max Range A 88.0																Frame	
Name Price All Excluse Names: AdV Upper Quartie 8.1.4 Thread: Select: Mail Intread Gage Mail 0.21 Principle: All Intread: Select: Mail 0.21 Parent: Nois Units: Mailes: Intread: Select: Mail 0.21 Parent: Nois Units: Mailes: Units: Mailes: 0 7.00 Analysis Type: Total Mailes: Mailes: Mailes: Mailes: 0.21 Analysis Type: Total Mailes: Mailes: Mailes: Mailes: 7.00 Analysis Type: Total Mailes: Mailes: Mailes: 7.00 7.00 Analysis Type: Total Mailes: Mailes: Mailes: 7.00 9.0 4.92.69 7.0076 Valid estrin: Mailes: Mailes: Mailes: Mailes: 0.00 3.000 1.000 0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 <	▼ Filters														492.69	2989	
104 cm Select. Median 8.0 283 Depth Slice. Lott: Millineconds Median 8.0 283 Parent Marker: None Lott: Millineconds Median 8.0 283 Analyds Type: Total Units: Millineconds Median 8.0 283 Analyds Type: Total Units: Millineconds Median 8.0 283 Analyds Type: Total Millineconds Millineconds Median 8.0 283 Analyds Type: Parent Kersen Parent Kersen Median 492.09 2708 492.09 All depth Parent Kersen Depth Median Median Median 8.0 492.09 UnityEditor.CoendAdulae dillUnityEditor:EditorAnalytic S-7 95.40 95.40 95.40 0.00 3 3 0.00 UnityEditor.CoendAdulae dillUnityEditor:EditorAnalytic S-7 95.40 95.40 95.40 0.00 1 0.00 UnityEditor.CoendAdulae dillUnityEditor:EditorAnalytic S-7 95.40 95.40 95.40 1 0.00 UnityEditor.CoendAdulae dillUnityEditor:EditorAnalytic S-7 95.40 95.40 0.00 1 0.00 UnityEditor.Coend	Name Filter:	All *						Any 👻						Upper Quartile			
Light Siles: Analysis 1962: Main 0.21 Parent Marker: None Lower Countrie Source Lower Countrie Source Main 7.01 2708 * Too In markers on median frame 2265: Ector O Source Source Source 492.69 7.0876 * Zoor Countrently selected range Player Loop Editoric coel Source (Internalis Render) Render (Rende (kee) Source Source 492.69 7.0876 Marker Name Depth Median Marker Source Source 492.69 7.0876 Unity Editor CoreModule all/Unity Editor:EditorAnalytic F-7 95.40 95.40 95.40 0.0 3 3.00 Marker Name Marker Name Marker Source Marker Source<	Thread.	Select												Median		2935	
Parent Warker: None Units: Miniscendia: V Units: Miniscendia: V Units: Miniscendia: V Units: Voire Pushing V Analyze: Istic dr 2/06 markers, 1 of 66 threads Miniscendia: V V Miniscendia: V Miniscendia: V Miniscendia: V V Miniscendia: V Minii V Miniscendia: V <	Depth Slice :	All 👻									Analysis Type :	Total					
Analyze Tability Tybe markets Market Columns: Imme and Count Min 7.0 Z/208 * Top 10 markets on median frame 2285. Ecf:01 PlayerLoop EditorLoop/DoRend_InternalFRender() Render(Rende Exe 53.8ms 6 9.0 7.0978 7.09	Parent Marker :	None										Milliseconds		Lower Quartile	7.90	0700	
• To functers on median frame Playert.cop [Playert.cop [Playert.co	Analyze	1861 of 2106 markers , 1 of 66 th									Marker Columns :	Time and Count				2708	J I
2935 (bit depth) Playert.op [Editori.cog [Defende/]Render[]Render	Top 10 marker	s on median frame														T	492.69
(All depths) 0	2935 ctor()	1	PlayerLoop E	EditorLoop D	oRende Internall	FRender()	Render(Rend	e Exe				53.8ms					7 0976
Marker Details for currently selected range Median Median Median Marker Summer Count Count Frame Atheedan Frame Thread Summary Marker Name Dieph Median Median Median Marker Mane Marker Summer Atheedan Frame Atheedan Frame Marker Summary Mar	(All depths)													0	492.69	_	7.0370
Marker Name Depth Median Median Median Median Median Max Range Count Count Frame At Median Frame Tracka summary UnityEditor:CoreModule dilUnityEditor:EditorAndaly/editor:EditorAndaly/editor:EditorAndaly/editor.EditorAndaly/editorAndaly/editor.EditorAndaly/editor.EditorAndaly/editor.EditorAndaly/editor.	Marker Details	or currently selected range												h Thread Commen			
Marker Stammary Deput Mondania menungana menungan Marker Stammary Marker Stammary Marker Stammary UnityEditor.CoeModule allUnityEditor.EditorAnalytic 48.85 48.85 48.85 0.00 3 3 0.00 UnityEditor.CoeModule allUnityEditor.EditorAnalytic UnityEditor.CoeModule allUnityEditor.EditorAnalytic UnityEditor.CoeModule allUnityEditor.EditorAnalytic Marker Stammary UnityEditor.EditorAnalytic UnityEditor.EditorAnalyt	Markerbleme													■ Thread Summa	ry		
UnityEditor.CoeModule allUnityEditor.CoeModule a	Marker Name		5-7	95.40	ivieulan bai ivi	1640	95.40	95.40	nange	a	2			▼ Marker Summa	ry		
UnityEditor.CoreModule allUnityEditor.ProfilerWindo 2 48.64 48.84 48.84 0.00 1 0.00 Mean frame contribution UnityEditor.CoreModule allUnityEditor.ProfilerWindo 3 48.54 48.54 48.54 48.64 0.00 1 0.00 UnityEditor.CoreModule allUnityEditor.ProfilerWindo 3 48.54 48.54 48.54 0.00 1 1 0.00 UnityEditor.CoreModule allUnityEditor.ProfilerWindo 4 47.90 47.90 47.90 0.00 1 1 0.00 134.6% UnityEditor.CoreModule allUnityEditor.ProfilerWindo 3 48.54 42.43 47.38 0.00 1 1 0.00 134.6% UnityEditor.CoreModule allUnityEditor.ProfilerWindo 4 47.38 47.38 47.38 0.00 1 1 0.00 101 10.00 101 10.00 101 10.00 101 10.00 101 10.00 101 10.00 101 10.00 101 10.00 101 10.00 101 10.00 101 10.00 101 10.00	UnityEditor CoreM	Indule dill InityEditorInternal-Profile		48.85		8.85	48.85	48.85	0.00			0.00		UnityEditor.CoreN	lodule.dll!	JnityEdit	or::EditorAi
UnityEditor CoreModule dillUnityEditor:ProfilerWindo 3 48.54 48.54 48.54 48.54 0.00 1 0.00 934.6% UnityEditor CoreModule dillUnityEditor:ProfilerWindo 4 47.90 47.90 47.90 0.00 1 0.00 First frame 2990 UnityEditor CoreModule dillUnityEditor:ProfilerWindo 4 47.90 47.90 47.90 0.00 1 0.00 First frame 2990 UnityEditor CoreModule dillUnityEditor:PackageMans 8 42.43	UnityEditor.CoreN	lodule.dll!UnityEditor::ProfilerWindov			4												
UnityEditorCoreModule dllUnityEditor.ProfilerWindor 4 47.90 47.90 47.90 0.00 1 0.00 First frame 2990 UnityEditorCoreModule dllUnityEditor.PackageMane 7 47.38 47.38 47.38 0.00 1 0.00 First frame 2990 UnityEditorCoreModule dllUnityEditor.PackageMane 8 42.43 42.43 42.43 0.00 1 1 0.00 Top [3 • by frame costs UnityEditorCoreModule dllUnityEditorScreekodule dllUnityE	UnityEditor.CoreM	Iodule.dll!UnityEditor::ProfilerWindov			4										004.00		
UnityEditor.CoreModule dillUnityEditor.PackageMane 7 47.38 47.38 47.38 47.38 0.00 1 1 0.00 Furit frame 2990 UnityEditor.CoreModule dillUnityEditor.PackageMane 8 42.43 42.43 42.43 42.43 0.00 1 1 0.00 Top 8 by frame.costs This Packet@index.com 8 1742 58 123.3 95.4ms 2990	, UnityEditor.CoreM				4										934.0%		
UnityEditor Contrologiu al UnityEditor Control UnityEditor Contro	UnityEditor.CoreM	lodule.dll!UnityEditor.PackageMana			4									First frame		2990	
Unity RenderPinelines Core Runtime dIIII Inity Engine 5 5-20 12.64 12.85 12.17 15.65 3.48 17342 58 12.33 95.4ms 2990	UnityEditor.CoreM				4									lop 3 v by	rame cost	S	. 11
	Unity.RenderPipel				1									-	95.4ms	2990	
mscortib.dlllMono.Globalization.Unicode::SimpleColla 11-13 4.10 4.10 4.10 4.10 0.00 162 162 0.00	mscorlib.dll!Mono.				4												U
PlayerLoop 1 3.93 4.00 3.40 14.62 11.23 897 3 4.06	PlayerLoop				4												U
EditoriLoop 1 3.72 5.65 2.38 487.87 485.49 897 3 3.57 95.396	EditorLoop				5				485.49								95.396
mscorlib.dlllSystem::String.StartsWith() 8-9 2.95 2.95 2.95 2.95 0.00 108 108 0.00	mscorlib.dll!Syster																
UnityEngine.CoreModule.dlilUnityEngine.Rendering:# 2 2.74 2.78 2.62 3.38 0.77 299 1 2.68	UnityEngine.Corel	Module.dll!UnityEngine.Rendering::F						3.38									
UnityEngine.CoreModule.dllUnityEngine.Rendering:≱ 3 2.73 2.77 2.61 3.37 0.76 2.99 1 2.67	UnityEngine.Corel	Module.dll!UnityEngine.Rendering::F						3.37	0.76	299		2.67					
Unity HenderPipelines.Universal Runtime dtilUnityEng 4 2.73 2.77 2.61 3.37 0.76 299 1 2.67	Unity.RenderPipel	ines.Universal.Runtime.dll!UnityEng							0.76	299							05 206
Dnity Render/pelines Universal Runtime.dllUnityEng 6 2.69 2.73 2.57 3.31 0.75 2.99 1 2.61 95.396 95.396	Unity.RenderPipel	ines.Universal.Runtime.dll!UnityEng		2.69					0.75	299		2.61		95.396	95.396		
mscombanisystem.chai.rbstmgu 8-9 2.28 2.28 2.28 0.00 108 108 0.00	msconib.diilSyster	n::Char.ToString()		2.28			2.28	2.28	0.00	108							•

6. Box Collider with increased enemy count

As explained above in the box vs mesh collider section, box collider does not affect the performance so much because we do not have so many collisions. However, when we combine it with increased enemy count, they decrease the performance a little because enemy script calls are also increased.

Profiler Modules CPU Usage Rendering Scripts	Playmode Image: A state of the) I	1827 Clear Clear on Play Deep Profile Call Selected: Assembly-CSharp.dlll:	Stacks 🔻 🖆 🛗 Enemy.Update() 🔺	Avatar Apply Root Mo Update Mode Culling Mode				
 Physics Animation GarbageCollector VSync Global Illumination UI Others 	16ms (60FPS)	0.05ms			 Capsul Edit Collider Is Trigger Material Center 				
 Rendering Batches Count SetPass Calls Count Triangles Count 			5.57k		Radius Height Direction				
Timeline	▼ Live	Live CPU:11.66ms GPU:ms							
Main Thread		0ms /erLoop (1.70 EditorLoop (2.97ms) PlayerLoop (4.09ms) JitorLoop (2.10m	rLoop (1.2/Loop (3.9 🔺	Damage				
			erPipeline.InternalRender() (3.9 isalRenderPipeline.Render() (3.1 ProfilingScopector() (3.92ms) derPipeline.RenderCameraStad		 Enemy Script Hit Poits 				
Render Thread		tandsFrom WaitForSig	andsFrom handsFrom NenderSingleCameri VaitForSi WaitForSigrUniversalRenderPipt ueObject essing t	aitForSi	# Chase Range				
▶ Job				▼	Turn Speed				

The profile analyzer details are given below. The most usage is done by player loop.

Profile Analyzer														
Profile Analyzer														:
Mode: Single Compare Export Close Profi	ler Window													
Pull Data 10275 49n x					_	_	_				▼ Frame Summar	v		
Load Save Unsaved 5														
426	6.00ms													
													17435	
1743	35				[300]				177	34			17734	
									PlayerL	оор				
▼ Filters													17734	
Name Filter : All				Exclude Names :	Any 👻									
Inread : Select Main Thread											Median		17557	
Depth Slice : All 👻								Analysis Type :	Total			49.46		
Parent Marker : None								Units :	Milliseconds		Lower Quartile	11.87		
Analyze 2075 of 2320 markers , 1 of 81								Marker Columns :	Time and Count			11.15	17523	
Top 10 markers on median frame											275			
17557 ctor() P	layerLoop Edi	itorLoop DoRend	deInternalF Render()	RenderC Render					86.4ms					
(All depths)											0	10275		147
Marker Details for currently selected range											- Throad Summa			
Marker Name											P miead Summa			
Playerl oop		5.56				7.68	897				Marker Summa	ry		
mscorlib.dll!Mono.Globalization.Unicode::SimpleCol	- la: 11-13	4.34	4.34	4.34	4.34						PlayerLoop			
UnityEngine CoreModule dllUnityEngine Rendering: F 2		3.86									Mean frame contr			- U
UnityEngine.CoreModule.dll!UnityEngine.Rendering::F 3		3.84										11.014		- U
Unity.RenderPipelines.Universal.Runtime.dll!UnityE		3.84									-			- U
Unity.RenderPipelines.Universal.Runtime.dll!UnityE		3.80									First frame		1/435	- U
Unity.RenderPipelines.Universal.Runtime.dll!UnityE		3.13									Top 3 ▼ by1	rame cost	S	- U
mscorlib.dll!System::String.StartsWith()		3.11										12.4ms	17588	- U
UnityEditor.CoreModule.dll!UnityEditor.Compilation												12.0ms	17647	- U
mscorlib.dll!System::Char.ToString()											_	8.83ms	17656	- U
GC.Collect													T 12	392
UnityEngine.CoreModule.dlllUnityEngine::Object.Ins														
mscorlib.dlllSystem.Globalization::CompareInfo.IsPr														
mscorlib.dll!Mono.Globalization.Unicode::SimpleCol														
Unity.RenderPipelines.Universal.Runtime.dll!UnityE														
UnityEditor.CoreModule.dll!UnityEditor.Scripting.Sc											4 7112	12 202	4.7	'113
Mono.JIT														
mscorlib.dll!System::String.Concat()														

As seen below, the enemy script call is high when compared with default play. We see this result thanks to the code we added to the enemy script. We added the weapon script too, but did not use it for performance evaluation.

🖿 Project 🛛 🗟 Console	Animation	😨 Profiler						:
Profiler Modules 🔹	Playmode 🔻 🤘) 4 >	₩ Frame: 16	44 / 1827 Cle	ar Clear on Pla	y Deep Profile C	Call Stacks 🔻 🖆	
🤗 CPU Usage					Selected: As	ssembly-CSharp.	dll!::Enemy.Update	e0 🏛
Rendering Scripts Physics Animation GarbageCollector Global Illumination UI Others	33ms (30FPS) 16ms (60FPS)			i haddaanaad				
Decidering			0.05m	is j				
Rendering			FWF	45.57k		AP=WA	F3AAAI	
 Batches Count SetPass Calls Count Triangles Count 				53.33k ——		Y YY E		-
Hierarchy -	Live Main Thr	ead	▼ CPU:1	1.66ms GPU:	·ms ি	No Deta	ails 🔻	
Overview						Self ms	\mathbf{x}	
 PlayerLoop UnityEngine.CoreModule.d 	49.5% II!U 33.7%	0.9% 0.0%		108 B 0 B	5.78 3.93	0.10 0.00		Î
 Update.ScriptRunBehaviou BehaviourUpdate EventSystem.Update() 	IrU 3.5% 3.5%) 1.5%	0.0% 0.1% 0.0%		28 B 28 B 0 B	0.40 0.40 0.17	0.00 0.02 0.00		ľ
Enemy.Update()	0.3%	0.0%	9	0 B	0.08	0.00		
 DebugUpdater.Update RigidbodyFirstPerson(FlashLightSystem.Update FirstPersonController 	e() 0.2% Cor 0.2% dat 0.0%	0.0% 0.0% 0.0%	1 1 1 1	0 B 0 B 0 B 0 B	0.03 0.03 0.01 0.01	0.00 0.00 0.00 0.00		
		0.000		0.0	0.00	0.00		•

What To Do Next

We planned our game and designed accordingly, but there are still areas to be improved. For example, even if we collect batteries and increase lightning, the difference is not seen very well. We can change the screen colors to show this or we can add other collectable objects. We thought to have 4 different element stones that can be collected and if all are completed, the game ends as a win result. However, we did not add a win condition for now which is an improvement area for our game.