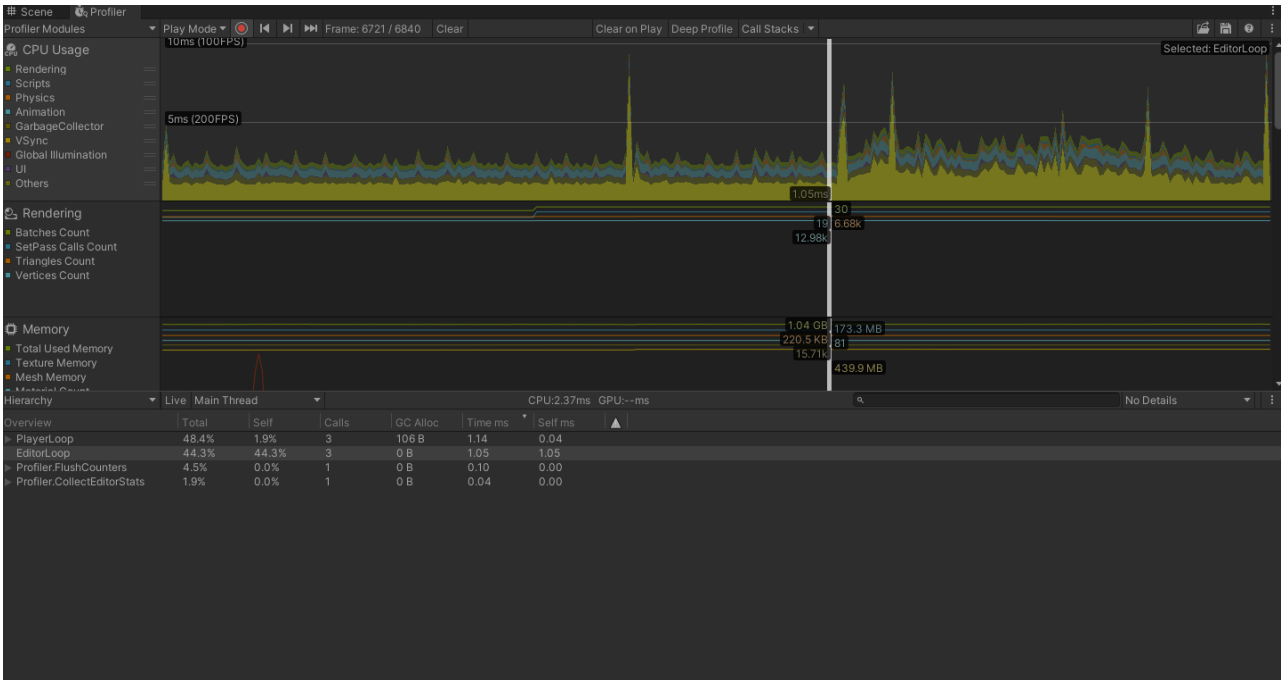
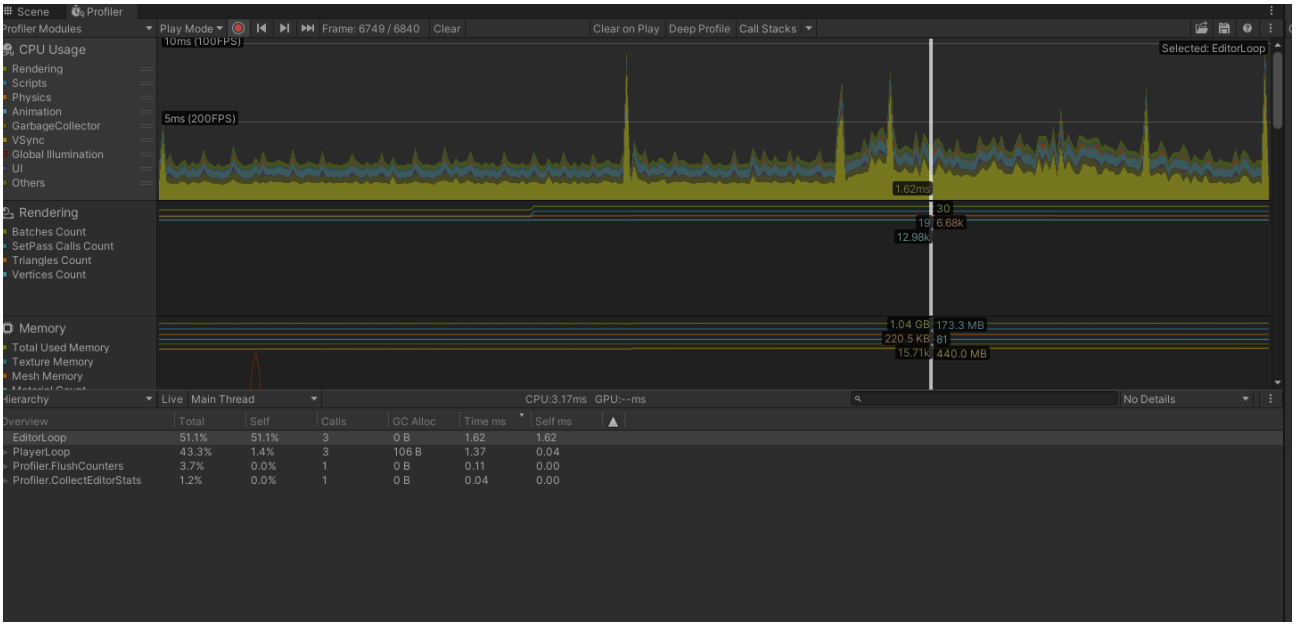
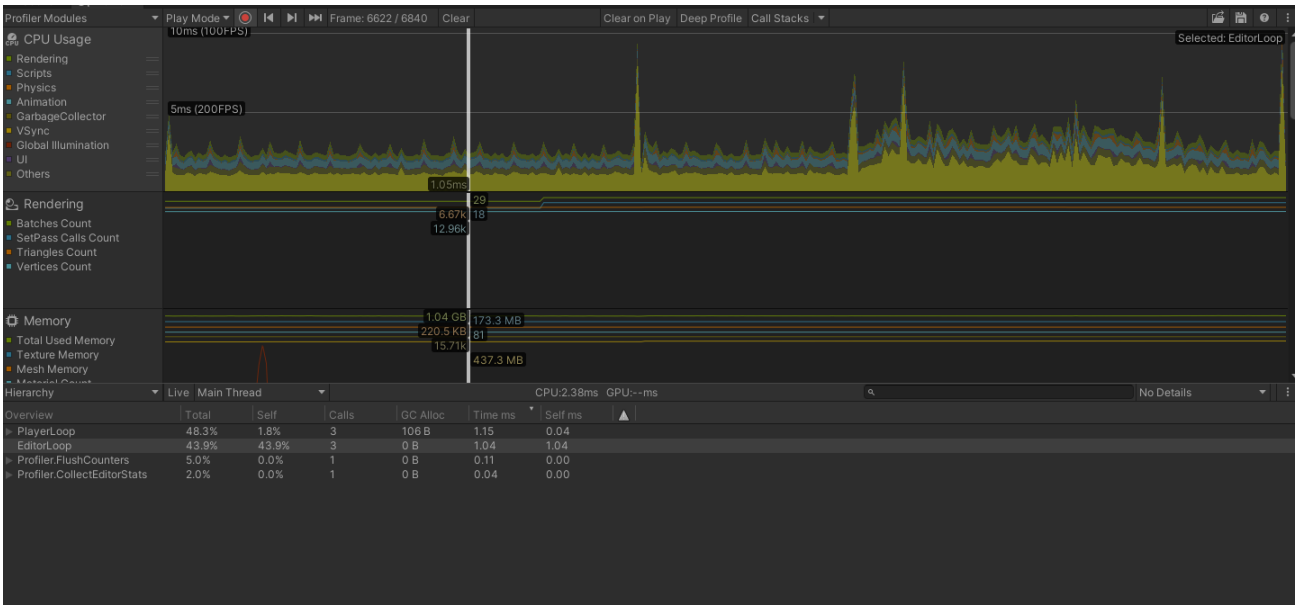
Testowanie gry podczas spoczynku:

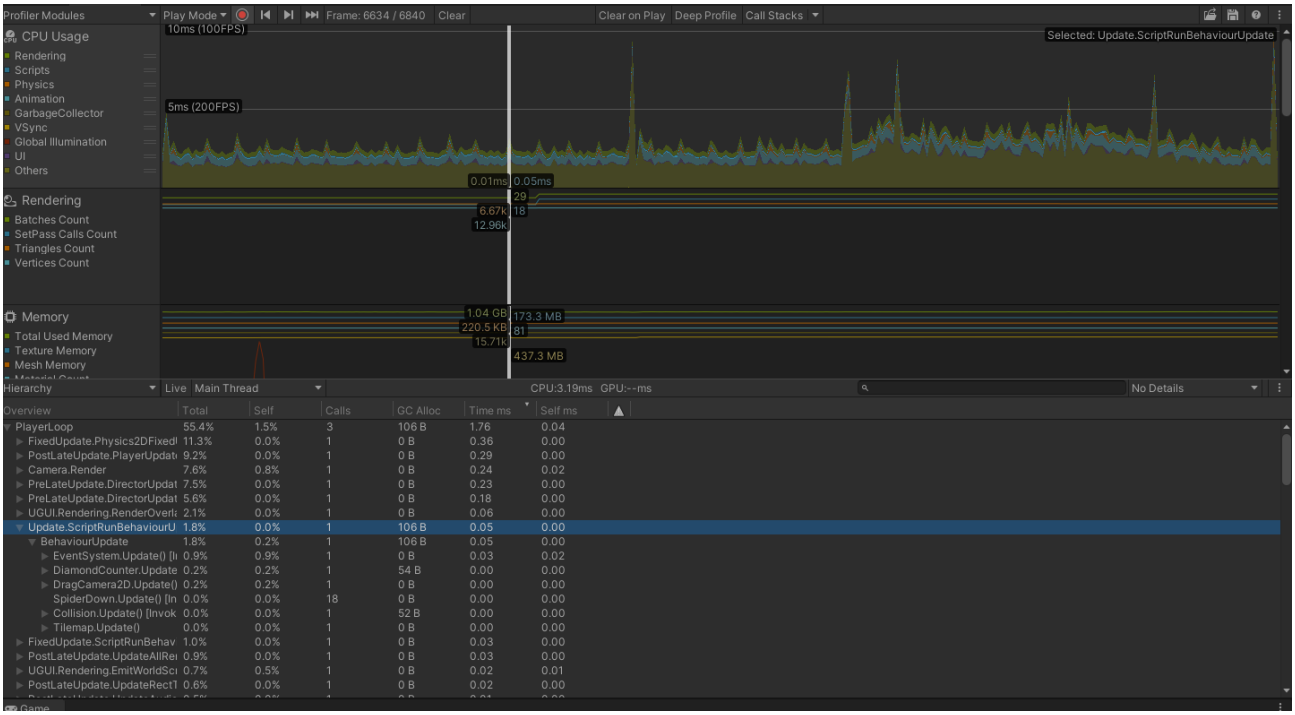
Co można zauważyć w większości klatek więcej procent zużycia procesora zajmuje PlayerLoop co jest złym syngałem.



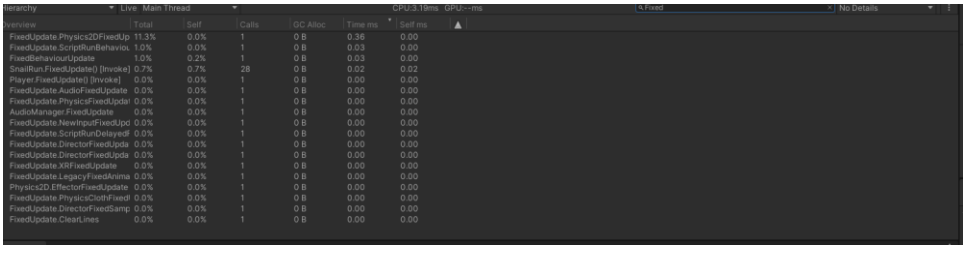


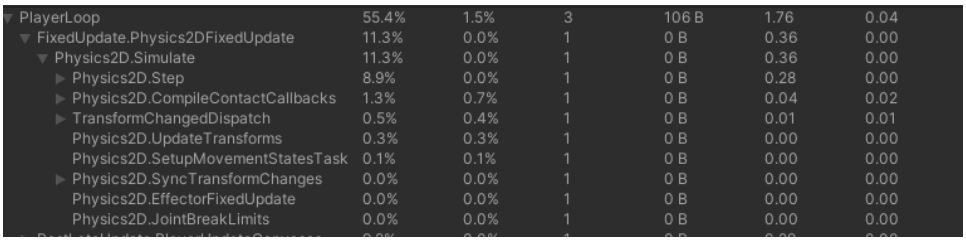


Analiza procesów w danej klatce:



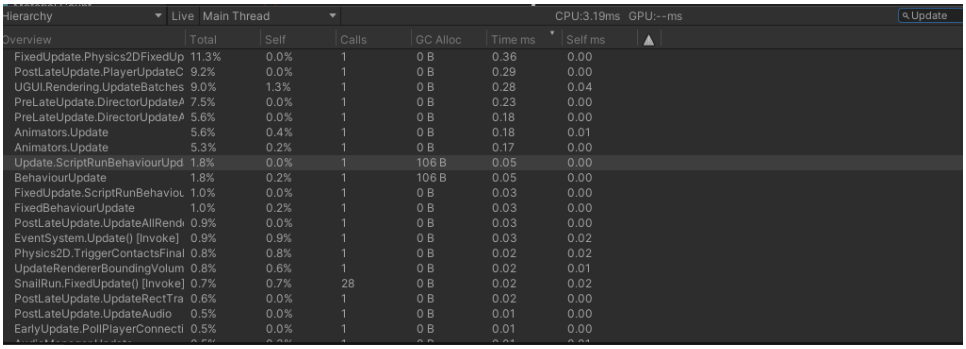
Fixed Update wynosi około 13% w tej klatce





PlayerLoop	55.4%	1.5%	3	106 B	1.76	0.04
FixedUpdate.Physics2DFixedUpdate	11.3%	0.0%	1	0 B	0.36	0.00
Physics2D.Simulate	11.3%	0.0%	1	0 B	0.36	0.00
Physics2D.Step	8.9%	0.0%	1	0 B	0.28	0.00
Physics2D.CompileContactCallbacks	1.3%	0.7%	1	0 B	0.04	0.02
TransformChangedDispatch	0.5%	0.4%	1	0 B	0.01	0.01
Physics2D.UpdateTransforms	0.3%	0.3%	1	0 B	0.00	0.00
Physics2D.SetupMovementStatesTask	0.1%	0.1%	1	0 B	0.00	0.00
Physics2D.SyncTransformChanges	0.0%	0.0%	1	0 B	0.00	0.00
Physics2D.EffectorFixedUpdate	0.0%	0.0%	1	0 B	0.00	0.00
Physics2D.JointBreakLimits	0.0%	0.0%	1	0 B	0.00	0.00

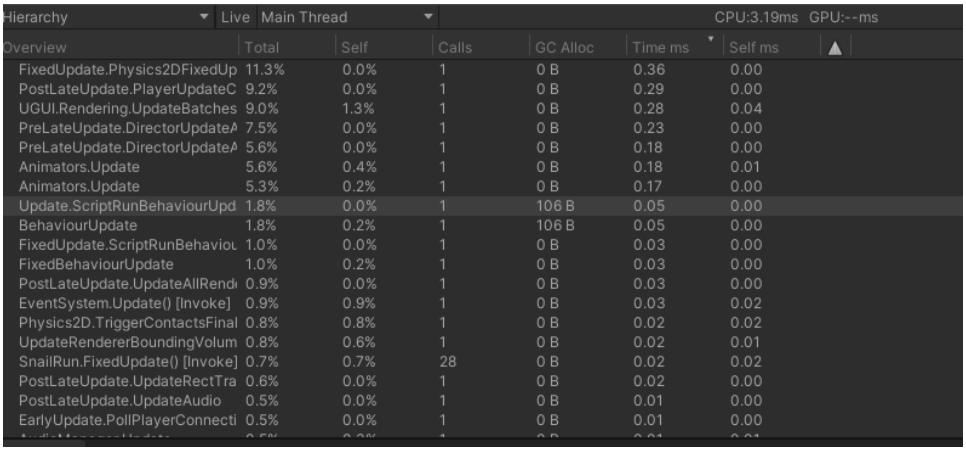
Update wynosi około 13%



Hierarchy | Live | Main Thread | CPU:3.19ms GPU:--ms | Update

Overview	Total	Self	Calls	GC Alloc	Time ms	Self ms
FixedUpdate.Physics2DFixedUp	11.3%	0.0%	1	0 B	0.36	0.00
PostLateUpdate.PlayerUpdateC	9.2%	0.0%	1	0 B	0.29	0.00
UGUI.Rendering.UpdateBatches	9.0%	1.3%	1	0 B	0.28	0.04
PreLateUpdate.DirectorUpdateA	7.5%	0.0%	1	0 B	0.23	0.00
PreLateUpdate.DirectorUpdateA	5.6%	0.0%	1	0 B	0.18	0.00
Animators.Update	5.6%	0.4%	1	0 B	0.18	0.01
Animators.Update	5.3%	0.2%	1	0 B	0.17	0.00
Update.ScriptRunBehaviourUpd	1.8%	0.0%	1	106 B	0.05	0.00
BehaviourUpdate	1.8%	0.2%	1	106 B	0.05	0.00
FixedUpdate.ScriptRunBehaviou	1.0%	0.0%	1	0 B	0.03	0.00
FixedBehaviourUpdate	1.0%	0.2%	1	0 B	0.03	0.00
PostLateUpdate.UpdateAllRend	0.9%	0.0%	1	0 B	0.03	0.00
EventSystem.Update() [Invoke]	0.9%	0.9%	1	0 B	0.03	0.02
Physics2D.TriggerContactsFinal	0.8%	0.8%	1	0 B	0.02	0.02
UpdateRendererBoundingVolum	0.8%	0.6%	1	0 B	0.02	0.01
SnailRun.FixedUpdate() [Invoke]	0.7%	0.7%	28	0 B	0.02	0.02
PostLateUpdate.UpdateRectTra	0.6%	0.0%	1	0 B	0.02	0.00
PostLateUpdate.UpdateAudio	0.5%	0.0%	1	0 B	0.01	0.00
EarlyUpdate.PollPlayerConnecti	0.5%	0.0%	1	0 B	0.01	0.00

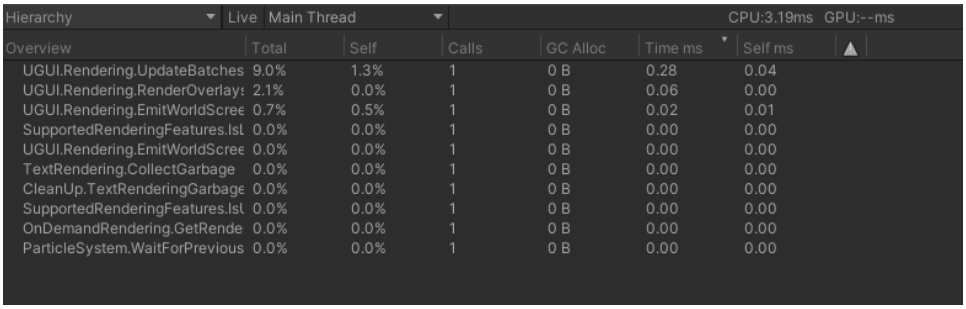
Late Update około 11%



Hierarchy | Live | Main Thread | CPU:3.19ms GPU:--ms

Overview	Total	Self	Calls	GC Alloc	Time ms	Self ms
FixedUpdate.Physics2DFixedUp	11.3%	0.0%	1	0 B	0.36	0.00
PostLateUpdate.PlayerUpdateC	9.2%	0.0%	1	0 B	0.29	0.00
UGUI.Rendering.UpdateBatches	9.0%	1.3%	1	0 B	0.28	0.04
PreLateUpdate.DirectorUpdateA	7.5%	0.0%	1	0 B	0.23	0.00
PreLateUpdate.DirectorUpdateA	5.6%	0.0%	1	0 B	0.18	0.00
Animators.Update	5.6%	0.4%	1	0 B	0.18	0.01
Animators.Update	5.3%	0.2%	1	0 B	0.17	0.00
Update.ScriptRunBehaviourUpd	1.8%	0.0%	1	106 B	0.05	0.00
BehaviourUpdate	1.8%	0.2%	1	106 B	0.05	0.00
FixedUpdate.ScriptRunBehaviou	1.0%	0.0%	1	0 B	0.03	0.00
FixedBehaviourUpdate	1.0%	0.2%	1	0 B	0.03	0.00
PostLateUpdate.UpdateAllRend	0.9%	0.0%	1	0 B	0.03	0.00
EventSystem.Update() [Invoke]	0.9%	0.9%	1	0 B	0.03	0.02
Physics2D.TriggerContactsFinal	0.8%	0.8%	1	0 B	0.02	0.02
UpdateRendererBoundingVolum	0.8%	0.6%	1	0 B	0.02	0.01
SnailRun.FixedUpdate() [Invoke]	0.7%	0.7%	28	0 B	0.02	0.02
PostLateUpdate.UpdateRectTra	0.6%	0.0%	1	0 B	0.02	0.00
PostLateUpdate.UpdateAudio	0.5%	0.0%	1	0 B	0.01	0.00
EarlyUpdate.PollPlayerConnecti	0.5%	0.0%	1	0 B	0.01	0.00

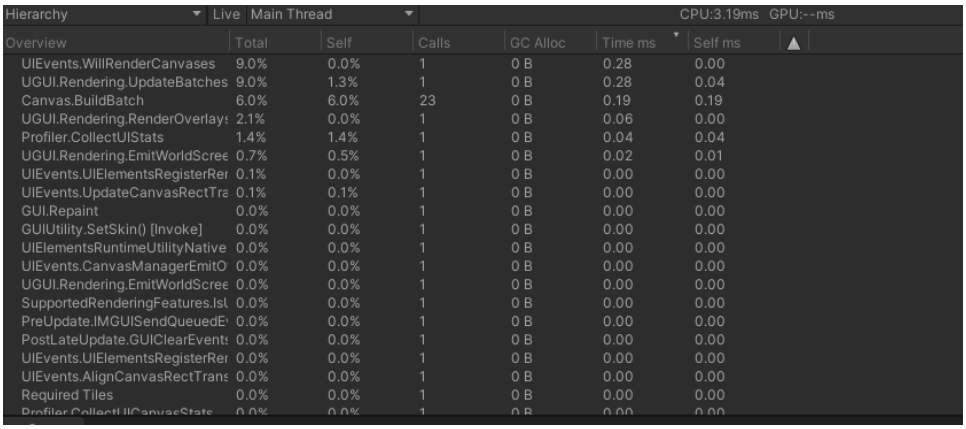
Rendering około 11%



Hierarchy | Live | Main Thread | CPU:3.19ms GPU:--ms

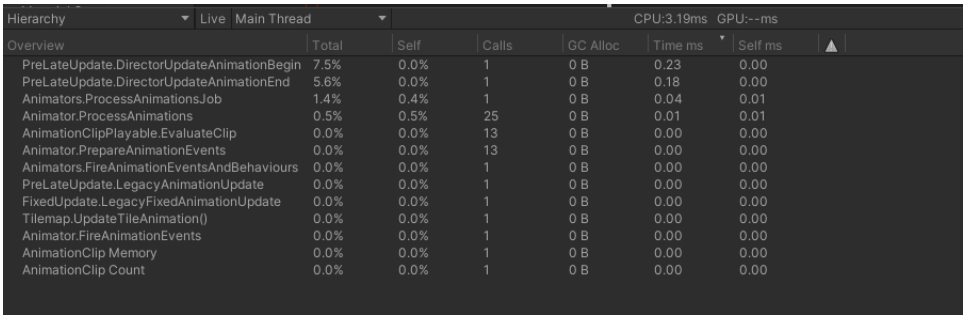
Overview	Total	Self	Calls	GC Alloc	Time ms	Self ms
UGUI.Rendering.UpdateBatches	9.0%	1.3%	1	0 B	0.28	0.04
UGUI.Rendering.RenderOverlays	2.1%	0.0%	1	0 B	0.06	0.00
UGUI.Rendering.EmitWorldScree	0.7%	0.5%	1	0 B	0.02	0.01
SupportedRenderingFeatures.IsL	0.0%	0.0%	1	0 B	0.00	0.00
UGUI.Rendering.EmitWorldScree	0.0%	0.0%	1	0 B	0.00	0.00
TextRendering.CollectGarbage	0.0%	0.0%	1	0 B	0.00	0.00
CleanUp.TextRenderingGarbage	0.0%	0.0%	1	0 B	0.00	0.00
SupportedRenderingFeatures.IsL	0.0%	0.0%	1	0 B	0.00	0.00
OnDemandRendering.GetRende	0.0%	0.0%	1	0 B	0.00	0.00
ParticleSystem.WaitForPrevious	0.0%	0.0%	1	0 B	0.00	0.00

Elementy UI około 25%



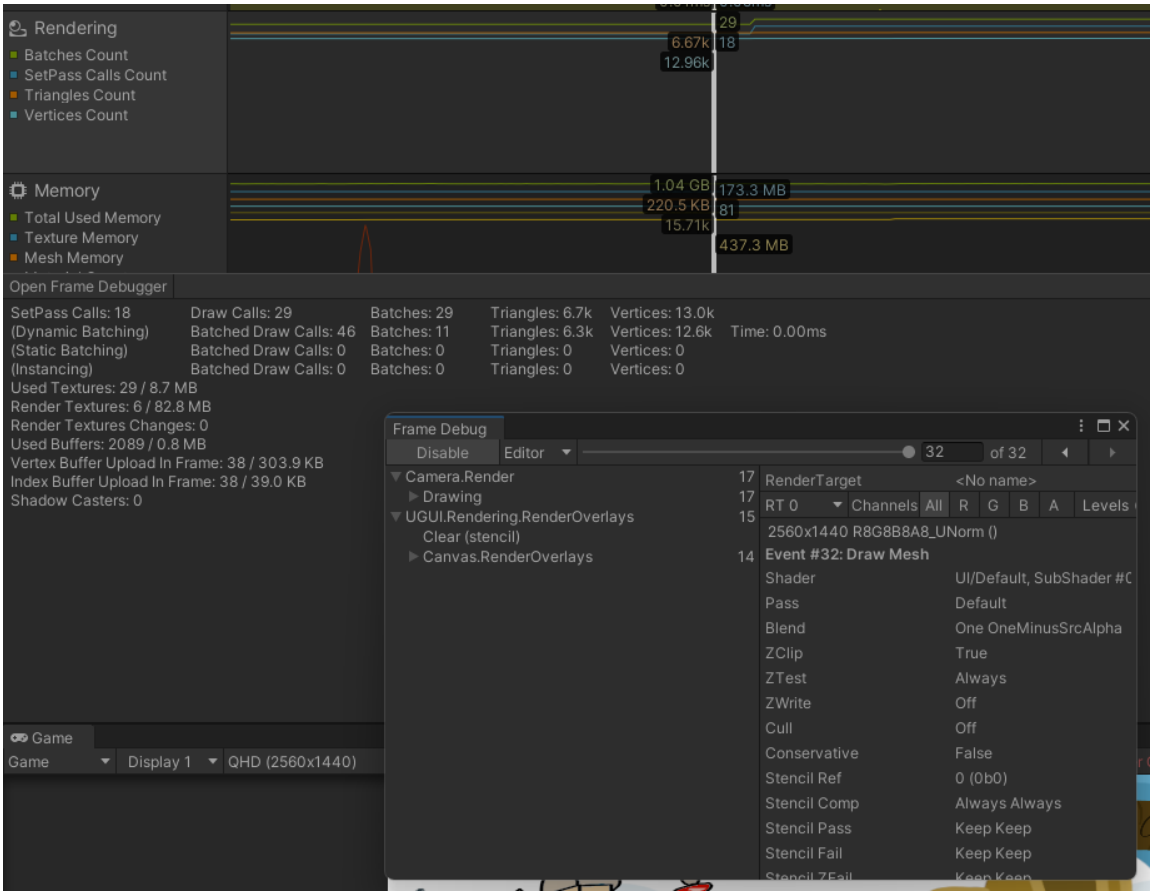
Overview	Total	Self	Calls	GC Alloc	Time ms	Self ms
UIEvents.WillRenderCanvases	9.0%	0.0%	1	0 B	0.28	0.00
UGUI.Rendering.UpdateBatches	9.0%	1.3%	1	0 B	0.28	0.04
Canvas.BuildBatch	6.0%	6.0%	23	0 B	0.19	0.19
UGUI.Rendering.RenderOverlays	2.1%	0.0%	1	0 B	0.06	0.00
Profiler.CollectUIStats	1.4%	1.4%	1	0 B	0.04	0.04
UGUI.Rendering.EmitWorldScree	0.7%	0.5%	1	0 B	0.02	0.01
UIEvents.UIElementsRegisterRe	0.1%	0.0%	1	0 B	0.00	0.00
UIEvents.UpdateCanvasRectTra	0.1%	0.1%	1	0 B	0.00	0.00
GUI.Repaint	0.0%	0.0%	1	0 B	0.00	0.00
GUIUtility.SetSkin() [Invoke]	0.0%	0.0%	1	0 B	0.00	0.00
UIElementsRuntimeUtilityNative	0.0%	0.0%	1	0 B	0.00	0.00
UIEvents.CanvasManagerEmitO	0.0%	0.0%	1	0 B	0.00	0.00
UGUI.Rendering.EmitWorldScree	0.0%	0.0%	1	0 B	0.00	0.00
SupportedRenderingFeatures.Isl	0.0%	0.0%	1	0 B	0.00	0.00
PreUpdate.IMGUISendQueuedE	0.0%	0.0%	1	0 B	0.00	0.00
PostLateUpdate.GUIClearEvents	0.0%	0.0%	1	0 B	0.00	0.00
UIEvents.UIElementsRegisterRe	0.0%	0.0%	1	0 B	0.00	0.00
UIEvents.AlignCanvasRectTrans	0.0%	0.0%	1	0 B	0.00	0.00
Required Tiles	0.0%	0.0%	1	0 B	0.00	0.00
Profiler.CollectUICanvasStats	0.0%	0.0%	1	0 B	0.00	0.00

Animacje około 12%

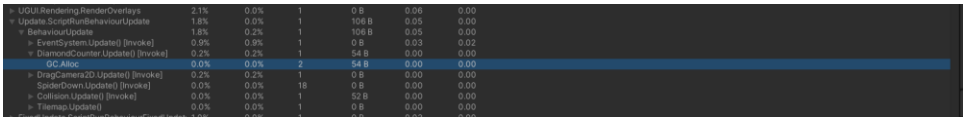


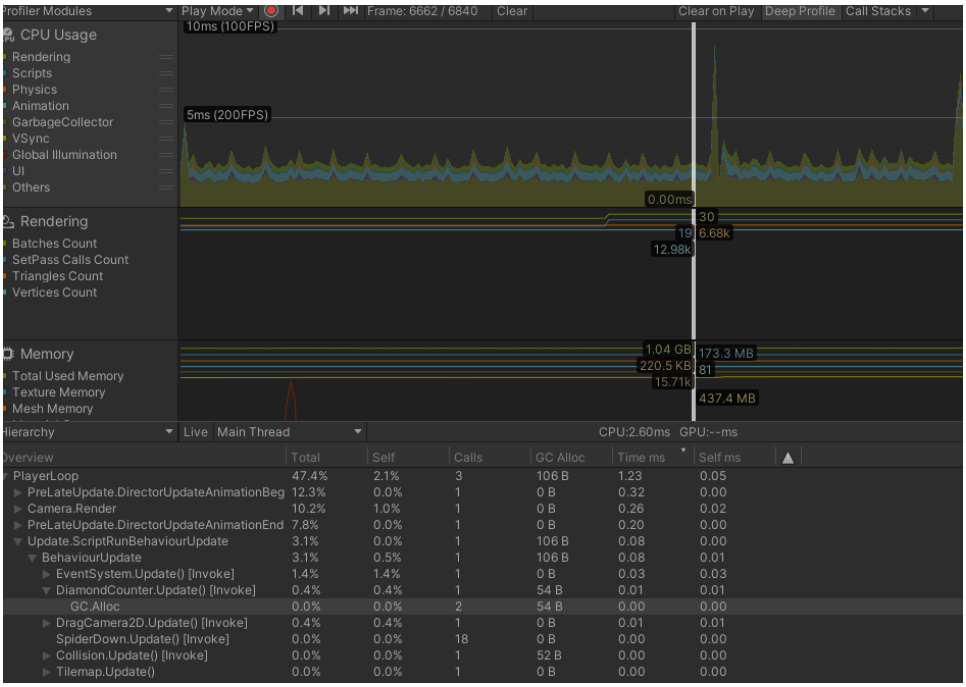
Overview	Total	Self	Calls	GC Alloc	Time ms	Self ms
PreLateUpdate.DirectorUpdateAnimationBegin	7.5%	0.0%	1	0 B	0.23	0.00
PreLateUpdate.DirectorUpdateAnimationEnd	5.6%	0.0%	1	0 B	0.18	0.00
Animators.ProcessAnimationsJob	1.4%	0.4%	1	0 B	0.04	0.01
Animator.ProcessAnimations	0.5%	0.5%	25	0 B	0.01	0.01
AnimationClipPlayable.EvaluateClip	0.0%	0.0%	13	0 B	0.00	0.00
Animator.PrepareAnimationEvents	0.0%	0.0%	13	0 B	0.00	0.00
Animators.FireAnimationEventsAndBehaviours	0.0%	0.0%	1	0 B	0.00	0.00
PreLateUpdate.LegacyAnimationUpdate	0.0%	0.0%	1	0 B	0.00	0.00
FixedUpdate.LegacyFixedAnimationUpdate	0.0%	0.0%	1	0 B	0.00	0.00
Tilemap.UpdateTileAnimation()	0.0%	0.0%	1	0 B	0.00	0.00
Animator.FireAnimationEvents	0.0%	0.0%	1	0 B	0.00	0.00
AnimationClip Memory	0.0%	0.0%	1	0 B	0.00	0.00
AnimationClip Count	0.0%	0.0%	1	0 B	0.00	0.00

Frame Debugger:



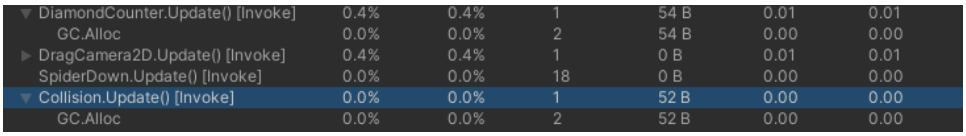
W metodze Update co klatkę jest wykorzystywany garbage collector do alokacji pamięci



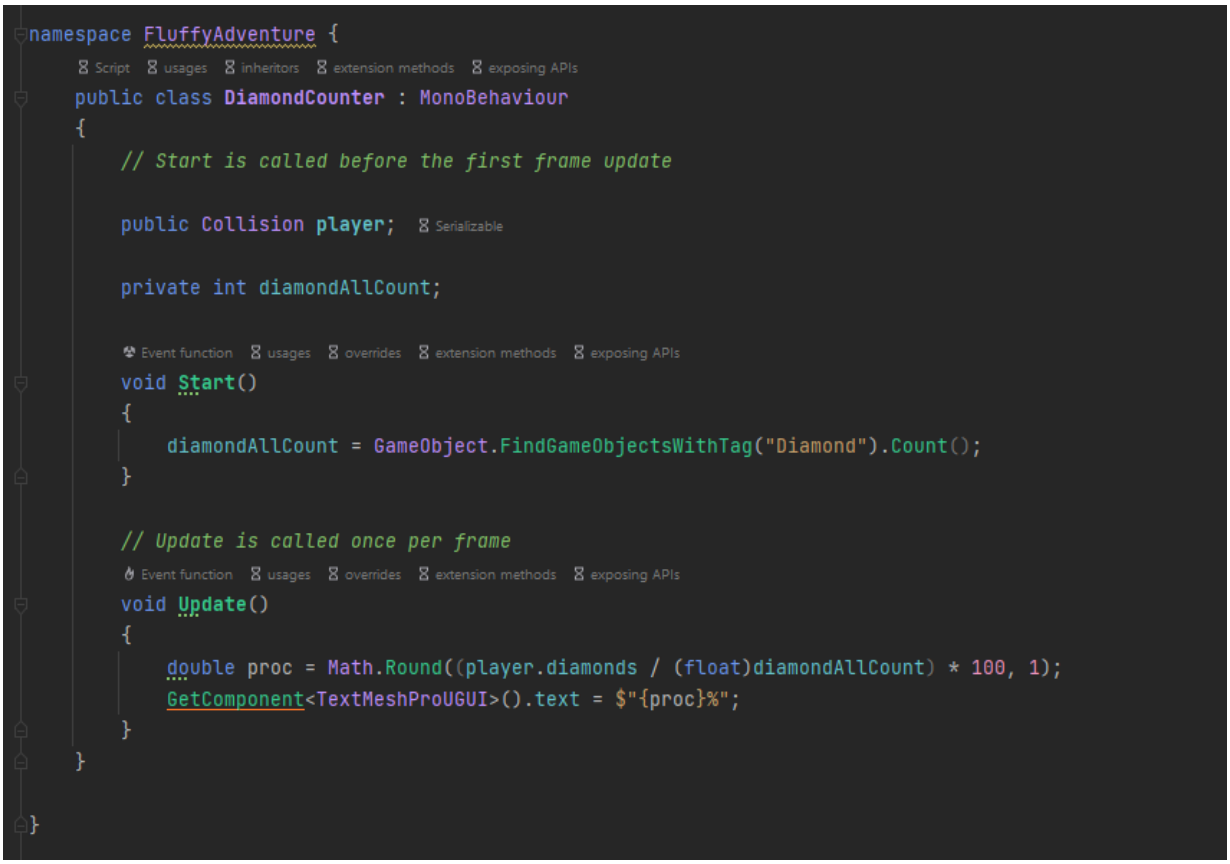


Zmieniliśmy klate ale wciąż jest 106B alokacji pamięci

Można Zauważyć że klasa DiamondCounter I collision powodują ten problem z alokacją.



Najpierw naprawmy DiamondCounter skrypt:



```
namespace FluffyAdventure {
    Script  usages  inheritors  extension methods  exposing APIs
    public class DiamondCounter : MonoBehaviour
    {
        // Start is called before the first frame update

        public Collision player;  Serializable

        private int diamondAllCount;

        Event function  usages  overrides  extension methods  exposing APIs
        void Start()
        {
            diamondAllCount = GameObject.FindGameObjectsWithTag("Diamond").Count();
        }

        // Update is called once per frame
        Event function  usages  overrides  extension methods  exposing APIs
        void Update()
        {
            double proc = Math.Round((player.diamonds / (float)diamondAllCount) * 100, 1);
            GetComponent<TextMeshProUGUI>().text = $"{proc}%";
        }
    }

}
```

Odrazu rzuca się w oczy metoda Update w której co klatkę pobieramy TextMeshProUGUI a także zmienna która tworzymy I alokujemy w pamięci na starcie I usuwamy na końcu metody. A także formatowanie stringa które jest zbędne.

Skrypt po refaktoryzacji:

```
namespace FluffyAdventure {
    2 asset usages
    public class DiamondCounter : MonoBehaviour
    {
        // Start is called before the first frame update

        public Collision player;  Changed in 1 asset
        public TextMeshProUGUI textCounter;  Unchanged

        private int diamondAllCount;

        Event function
        void Start()
        {
            diamondAllCount = GameObject.FindGameObjectsWithTag("Diamond").Count();
            textCounter = GetComponent<TextMeshProUGUI>();
        }

        // Update is called once per frame
        Event function
        void Update()
        {
            textCounter.text = (Math.Round((player.diamonds / (float)diamondAllCount) * 100, 1)).ToString();
        }
    }

}
```

Rezultaty:

Alokacją zmiejszyła się do 24B a także zużycie procesora.

PlayerLoop	50.3%	1.4%	3	76 B	2.06	0.05
Update.ScriptRunBehaviourUpdate	13.9%	0.0%	1	76 B	0.57	0.00
BehaviourUpdate	13.9%	0.2%	1	76 B	0.57	0.01
EventSystem.Update()	12.6%	0.0%	1	0 B	0.51	0.00
DiamondCounter.Update()	0.4%	0.0%	1	24 B	0.01	0.00
Double.ToString()	0.2%	0.0%	1	24 B	0.01	0.00
Number.FormatDouble()	0.2%	0.0%	1	24 B	0.00	0.00
NumberFormatInfo.get_CurrentInfo(	0.0%	0.0%	1	0 B	0.00	0.00

Skrypt Collision:

Collision.Update()	0.2%	0.0%	1	52 B	0.00	0.00
Text.set_text()	0.1%	0.0%	1	0 B	0.00	0.00
String.Concat()	0.0%	0.0%	1	28 B	0.00	0.00
String.FillStringChecked()	0.0%	0.0%	2	0 B	0.00	0.00
String.IsNullOrEmpty()	0.0%	0.0%	2	0 B	0.00	0.00
String.FastAllocateString()	0.0%	0.0%	1	28 B	0.00	0.00
Int32.ToString()	0.0%	0.0%	1	24 B	0.00	0.00
Number.FormatInt32()	0.0%	0.0%	1	24 B	0.00	0.00
ReadOnlySpan`1.op_Implicit()	0.0%	0.0%	1	0 B	0.00	0.00
String.memset()	0.0%	0.0%	1	0 B	0.00	0.00
Time.get_time()	0.0%	0.0%	1	0 B	0.00	0.00

Jak widać alokację powoduje konwersja na typ string co klatkę

```
// Update is called once per frame
Event function
void Update()
{
    pointsText.text = ": " + points.ToString();

    if (Time.time - curentTime > 0.5)
    {
        hp--;
        hpStrip.Value = hp;
        curentTime = Time.time;
        if(hp == 0)
        {
            gameManager.OnGameOver();
        }
    }
}
```

Pamięć: