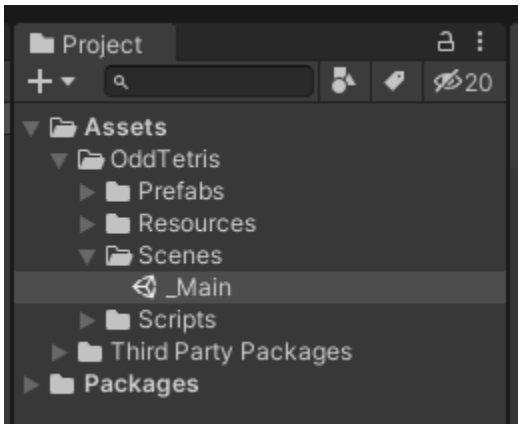
Project Report

Ramon Ferreira - Odd Tetris

All information and tests were made using:
Device: Samsung S10 SM-G973F / Android OS 12 / API-31
GFX Api: OpenGLES3
GPU: Mali-G76

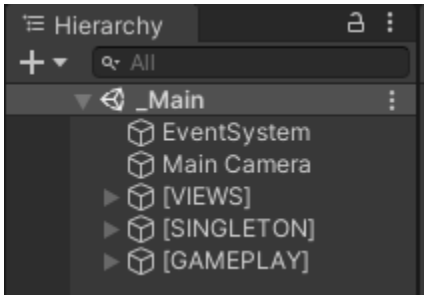
How to run it



Open scene **_Main** and run it. It's located at Assets/OddTetris/Scenes/_Main.

Play the scene

Scene _Main structure

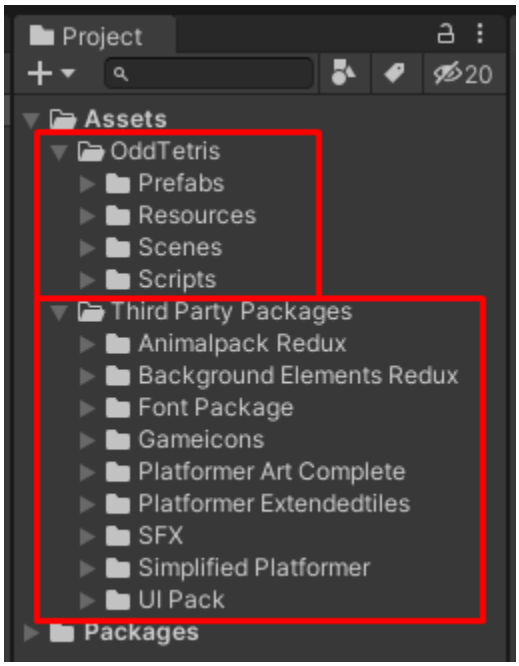


[VIEWS] - Every View from the project. Every is **view** is a piece of the screen, like a window.

[SINGLETON] - All singletons are created under this object. There are two pre-configured on editor time.

[GAMEPLAY] - Objects that are used inside gameplay like positioners, base, etc

Project Overall



OddTetris contains everything created for the project. (scripts, prefabs, scriptables, etc)

Third Party Packages contains every asset imported to be used (sprites, sfx, music, etc)

Memory Usage

Name	Memory	Ref count
Assets (783)	36.3 MB	
AudioClip (5)	30.7 MB	
Texture2D (27)	3.9 MB	
AudioManager (1)	1.1 MB	
MonoScript (586)	201.4 KB	
Shader (7)	156.5 KB	
Font (3)	108.0 KB	
MonoManager (1)	49.3 KB	
Sprite (24)	37.1 KB	
SpriteRenderer (24)	30.5 KB	
MonoBehaviour (11)	26.9 KB	
ScriptMapper (1)	11.4 KB	
Transform (32)	10.9 KB	
InputManager (1)	10.4 KB	
Material (4)	6.6 KB	
ResourceManager (1)	6.6 KB	
GameObject (32)	6.2 KB	
BoxCollider2D (11)	3.6 KB	
TagManager (1)	1.8 KB	
PlayerSettings (1)	1.5 KB	
QualitySettings (1)	1.1 KB	
TimeManager (1)	1.1 KB	
BuildSettings (1)	0.6 KB	
GraphicsSettings (1)	0.5 KB	
PreloadData (2)	448 B	
Physics2DSettings (1)	448 B	
Rigidbody2D (1)	320 B	
DelayedCallManager (1)	112 B	
RuntimeInitializeOnLoadManager (1)	104 B	
Other (91)	12.9 MB	
Scene Memory (2638)	1.6 MB	
Not Saved (50)	0.6 MB	
Builtin Resources	0 B	

Taking out the worst bgm file I could choose for Night Run Away - An Jone with 30.3MB we have a total of 6MB in assets being loaded. With a memory usage with the following data:

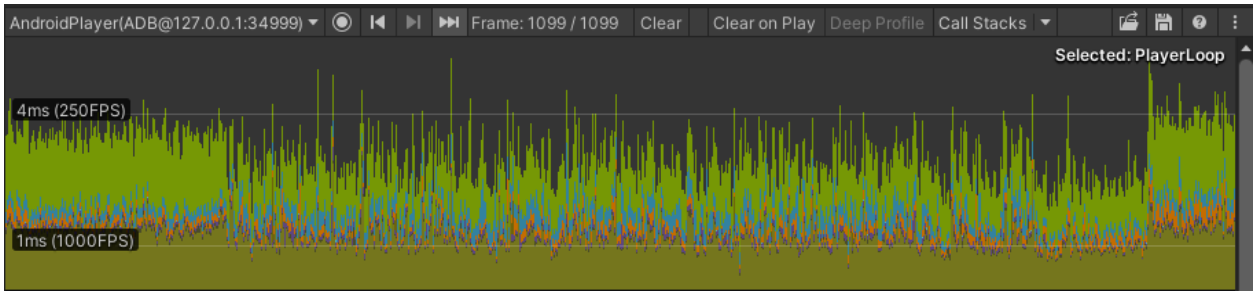
Total Used Memory: 106.7MB
GC: 0.6MB
Gfx: 4.2 MB
Audio: 31.8 MB
Video: 32 B
Profiler: 33.7 MB

Total Reserved Memory: 239.7 MB
GC: 0.8 MB
Gfx: 4.2 MB
Audio: 31.8 MB
Video: 32 B
Profiler: 48.0 MB
System Used Memory: 300.0 MB

Textures: 49 / 4.5 MB
Meshes: 1 / 16.1 KB
Materials: 12 / 18.3 KB
AnimationClips: 0 / 0 B
Asset Count: 801
Game Object Count: 880
Scene Object Count: 3038
Object Count: 3839
GC Allocation In Frame: 0 / 0 B

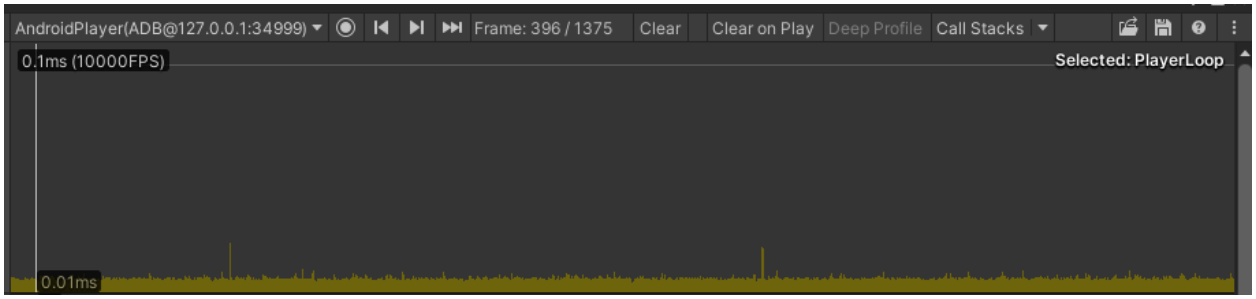
Framerate

This is the record of a match during the interactions. Taking off the VSync we are getting an average of 250Fps, with VSync turned on we are almost every frame clamped in 60Fps.

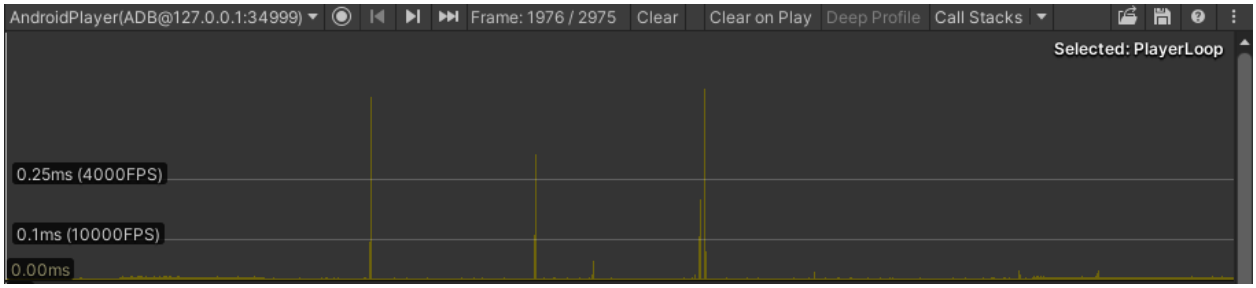


This is a variation trying to interact as most as I could but we are still getting around 4ms again.

Garbage Collector



This is what our garbage collector processing time looks like. We have one or two spikes of 0.02ms to process it, but it’s handling well.



This is another situation, spamming touches and killing a lot of pieces. We have “higher values” of 0.62ms because we are destroying pieces instead of returning them to pool, my bad on that. But shouldn’t be hard because the pool is already working.

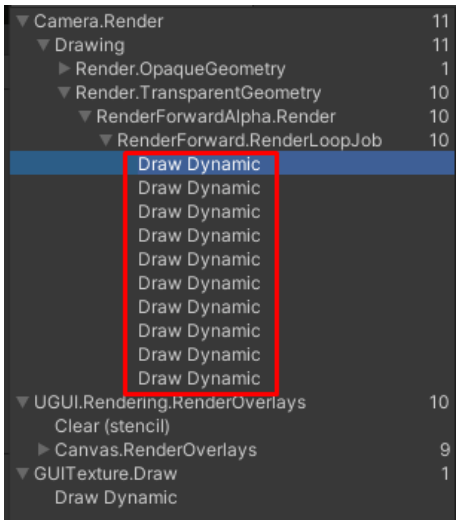
Binary Size

Binary file built for Armv7 results in a size of 14.2mb

Binary file built for Arm64 results in a size of 14.8mb

Binary file built for both results in a size of 21.3mb

Draw Calls



We could batch all those static images into 1 or 2 draw calls to reduce even more our count.

Next Steps

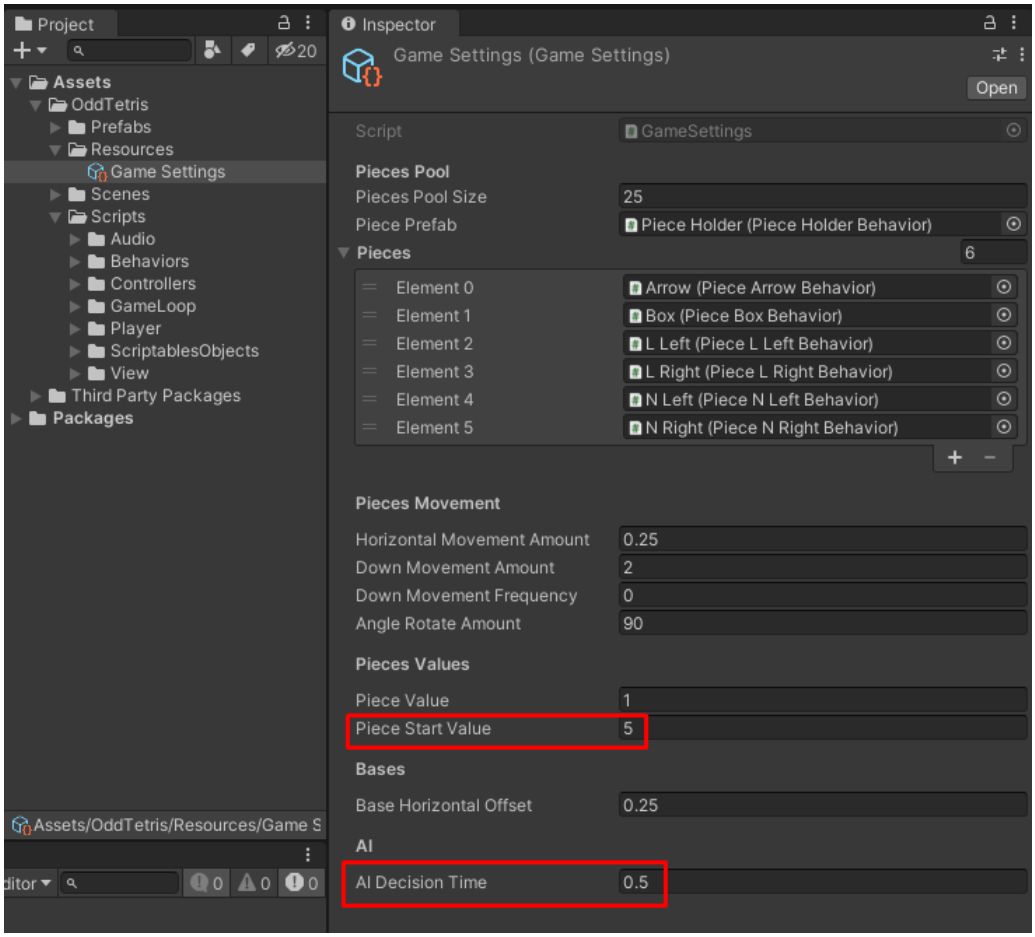
Here are some changes to make the project run better and technical debts I choose to postpone due time:

- Despawn pieces back to pool to improve even more our memory usage.
- Batch all sprites together to reduce drawcalls. Today we have 13 being generated that could be group together in 2.
- Load the game configurations and assets using Adressables instead of Resources. I choose resources just to speed up the project development.
- Preview the pieces that will be used and the AI movements.

Game Differences

- You and the AI can place your pieces in the “area” of the enemy to bother then.
- If you touch enemy pieces you will lose control of your piece two, so be careful trying to get too close to it.

Game Settings



Piece Start Value: how many pieces every player start and can “lose” during gameplay.

AI Decision Time: frequency for AI player to make a new decision