



Eötvös Loránd Tudományegyetem

Informatikai Kar

Térinformatikai és távérzékelési
alkalmazások fejlesztése

Pontfelhő vizualizáció

Nagy Richárd Tibor, Budapest, 2020

Tartalomjegyzék

Tartalomjegyzék	1
Bevezetés	2
Felhasználói dokumentáció	3
Rendszerkövetelmények	3
Telepítés	3
Indítás	4
Főmenü	4
Kamera	4
Fejlesztői dokumentáció	4
Elemzés - pontfelhők megjelenítése Unity-ben	4
Fejlesztői környezet	5
Felhasználói esetek diagramja	6
Felhasználói esetek leírása	6
A komponensek diagramja	7
Modell	7
Perzisztencia	7
Nézet	8
Skálázhatóság	9

Bevezetés

A feladat egy olyan grafikus felületű alkalmazás megvalósítása, amely lehetővé teszi LiDAR (Light Detection and Ranging) pontfelhők interaktív 3 dimenziós vizualizációját. A felületnek támogatnia kell a szokásos megjelenítési funkciókat, úgy mint a navigálás, nagyítás, forgatás, metaadatok tematikus megjelenítése.

Felhasználói dokumentáció

Rendszerkövetelmények

Operating system	Windows	macOS	Linux
Operating system version	Windows 7 (SP1+) and Windows 10	Sierra 10.12+	Ubuntu 16.04 and Ubuntu 18.04
CPU	x86, x64 architecture with SSE2 instruction set support.	x64 architecture with SSE2.	x64 architecture with SSE2 instruction set support.
Graphics API	DX10, DX11, DX12 capable.	Metal capable Intel and AMD GPUs	OpenGL 3.2+, Vulkan capable.
Additional requirements	Hardware vendor officially supported drivers.	Apple officially supported drivers.	Gnome desktop environment running on top of X11 windowing system

Telepítés

A program külön telepítést nem igényel.

Indítás

A projekt betöltésével a Unity környezetbe, vagy lefordított verzió esetén a .exe fájl futtatásával.

Főmenü



A főmenüben kiválasztható a betöltésre szánt .laz állomány, valamint színezési módot is választhatunk.

Kamera

A kamera támogatja a szokásos megjelenítési funkciókat, a navigálást a WASDQE billentyűk, a forgatást az egér végzi, a kamera gyorsasága pedig a Shift gombbal növelhető.

Fejlesztői dokumentáció

Elemzés - pontfelhők megjelenítése Unity-ben

Minden pont egy Unity GameObject

<https://docs.unity3d.com/ScriptReference/GameObject.html>

A legkevésbé optimális módszer, jelentős valós idejű optimalizációra lenne szükség. (Pl.: távoli pontok elrejtése, nyolcadoló fával)

Minden pont egy particle a Unity ParticleSystem-ben

<https://docs.unity3d.com/ScriptReference/ParticleSystem.html>

Az alapoktól kezdve nagy mennyiségű elemek megjelenítésére lett optimalizálva, azonban az egyes ParticleEmitter-ek limitáltak.

Unity DOTs

<https://unity.com/dots>

OOP helyet Data Oriented Programming. Pl.: egy pont objektumokból álló lista helyett tároljunk 3 tömböt, az x, y, és z koordinátákkal. A memória olvasási overhead így drasztikusan csökkenthető.

Mesh generálás

A beolvasott pontfelhőkből egy mesh-t generálunk. Ha lehetséges, ezzel a módszerrel lenne a leglassabb a betöltés, viszont maga a megjelenítés így lenne a legkisebb költségű.

Választott módszer

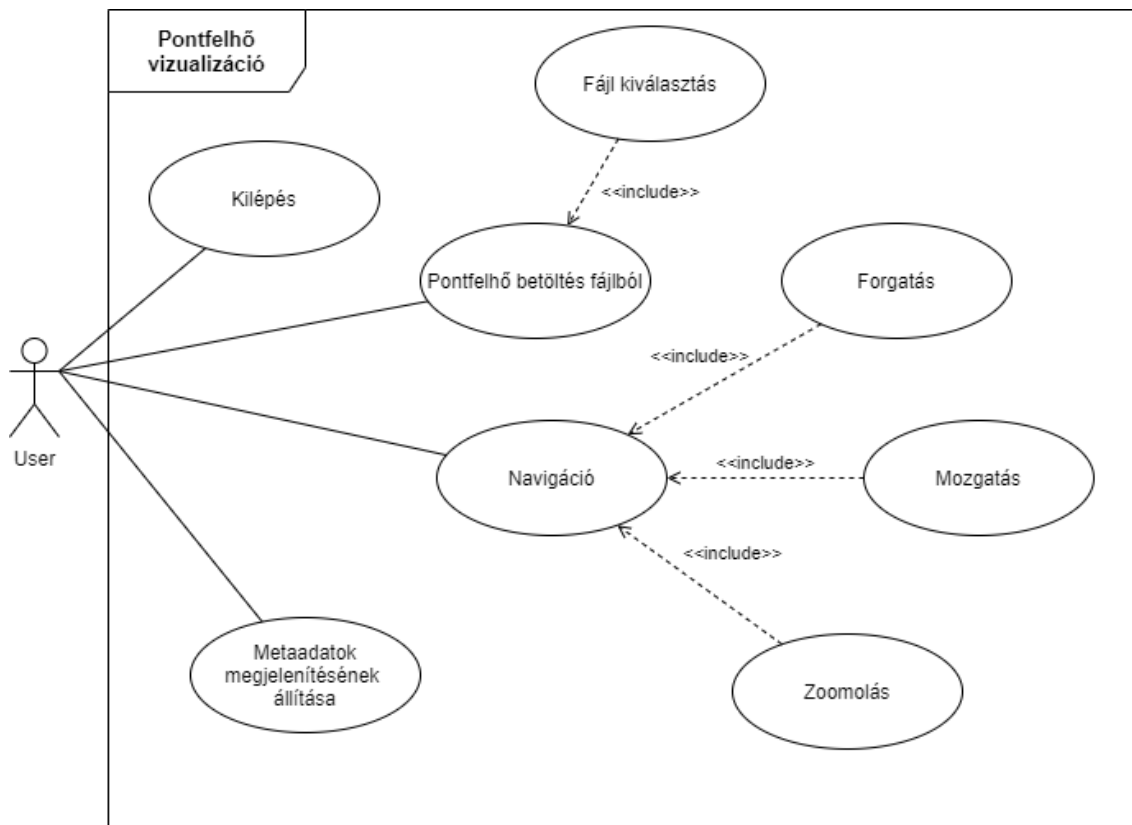
A legjobb módszernek a Unity DOTs és a Unity ParticleSystem ötvözete bizonyult.

Fejlesztői környezet

A szoftver fejlesztése során az alábbi programokat használtam fel:

- Unity 2019.3
- Visual Studio 2019
- GitKraken
- GitHub

Felhasználói esetek diagramja

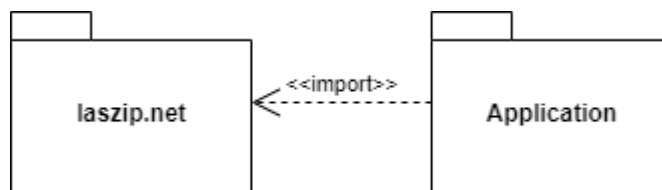


Felhasználói esetek leírása

Felhasználói eset	Leírás	
Indítás	Giv en:	A fájlkezelőben van
	Wh en:	Rákattint a futtatható állományra
	Th en:	Elindul a program
Kilépés	Giv en:	Fut a program
	Wh en:	Rákattint a kilépés gombra
	Th en:	A programleáll
Betöltés	Giv en:	A főmenüben van
	Wh en:	Rákattint a betöltés gombra
	Th en:	Megjelenik a fájl választó menü

Betöltés	Giv en:	A fájl választó menüben van
	Wh en:	Rákattint egy megfelelő fájlra
	Th en:	A pontfelhő betöltődik
Navigáció	Giv en:	A pontfelhő betöltődött
	Wh en:	Használja a navigációs inputokat
	Th en:	A kamera elmozdul
Metaadatok megjelenítése	Giv en:	A pontfelhő betöltődött
	Wh en:	Változtat a metaadat opciókon
	Th en:	A megjelenő metaadatok megváltoznak

A komponensek diagramja



Modell

A modell a Unity DOTs keretrendszerre épül. A pontok entitásként (Entity) vannak számon tartva, amik a következő komponensekkel rendelkeznek: PointPosition és PointColor.

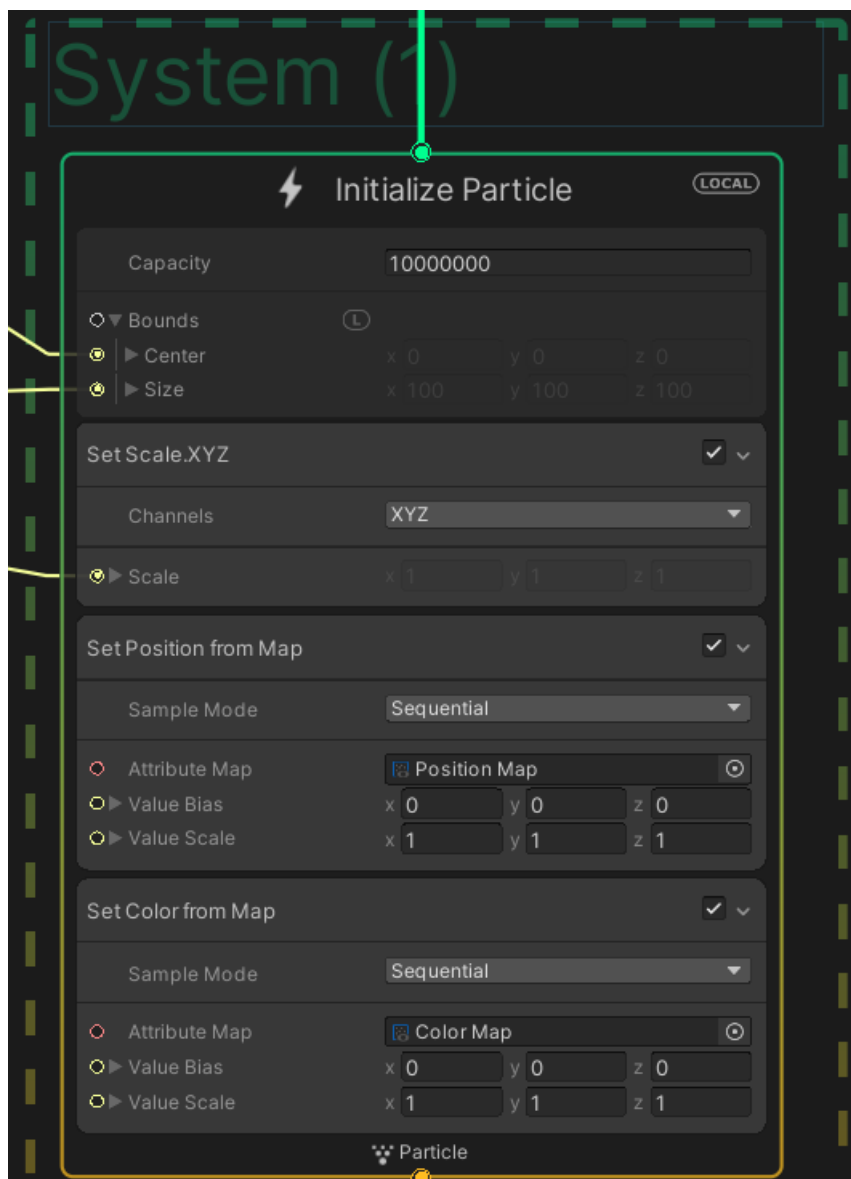
Perzisztencia

A bináris LAS fájlok beolvasását az integrált laszip.net könyvtár végzi. (<https://github.com/shintadono/laszip.net>)

laszip_dll
+ laszip_open_reader()
+ laszip_read_point()
+ laszip_get_coordinates()
+ laszip_close_reader()

Nézet

A megjelenítést a videokártya végzi, a Unity ParticleSystem rendszer felhasználásával. Az egyes pontok shader grafikonja a következő:



Skálázhatóság

Méréseim alapján egy 1 GB memóriával rendelkező videokártyán egyszerre 10 millió pont jeleníthető meg stabilan. Volt lehetőségem egy 2 GB-os videokártyán is tesztelni a programot, ott a 20 millió pont sem okozott akadályt. Emiatt úgy vélem a program lineárisan skálázható.