



**Eötvös Loránd Tudományegyetem**  
Informatikai Kar  
Térinformatikai és távérzékelési alkalmazások  
fejlesztése

---

## **Pontfelhő vizualizáció**

Nagy Richárd Tibor, Budapest, 2020

# Tartalomjegyzék

<b>Tartalomjegyzék</b>	<b>1</b>
<b>Bevezetés</b>	<b>2</b>
Felhasználói dokumentáció	<b>3</b>
Rendszerkövetelmények	3
Telepítés	3
Indítás	3
Főmenü	3
Kamera	3
Fejlesztői dokumentáció	<b>3</b>
Elemzés - pontfelhők megjelenítése Unity-ben	3
Fejlesztői környezet	3
Felhasználói esetek diagramja	4
Felhasználói esetek leírása	4
A komponensek diagramja	5
Modell	5
Perzisztencia	5
Nézet	5
Tesztelés	5
Egységtesztek	5
Végfelhasználói tesztek	5
Skálázhatóság	6

## 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

Telepítés

Indítás

Főmenü

Kamera

## 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 helyett 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

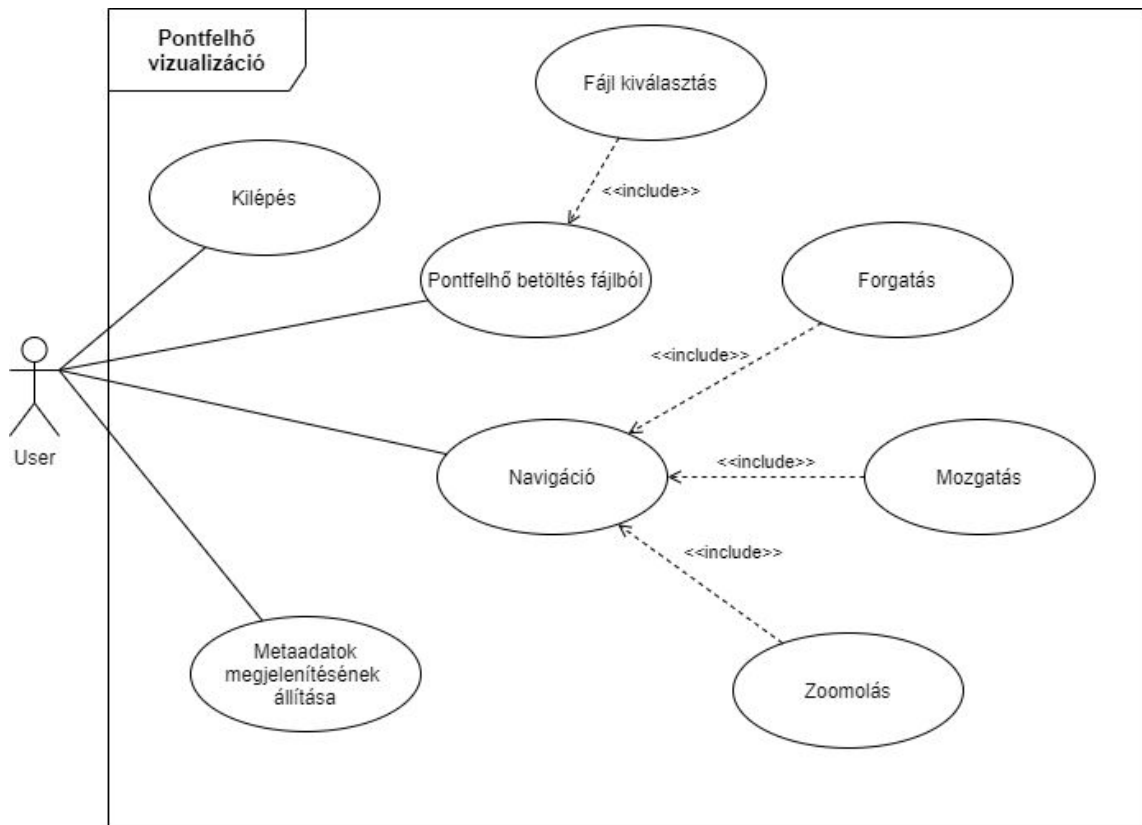
A beolvasott pontfelhőkből egy mesh-t generálunk. Ha lehetséges, ezzel a módszerrel lenne a leglassabb a betöltés, viszont maga a megjelenítés így lenne a legkisebb költségű.

## Fejlesztői környezet

A szoftver fejlesztése során az alábbi programokat használtam fel:

- Unity 2019.3
- Visual Studio 2019
- GitKraken
- GitHub

## Felhasználói esetek diagramja

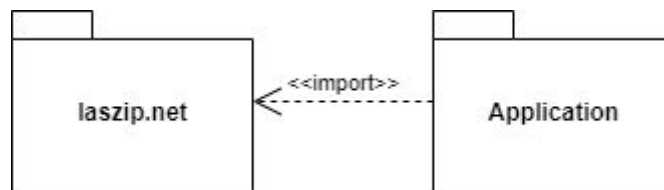


## Felhasználói esetek leírása

Felhasználói eset	Leírás	
Indítás	Given:	A fájlkezelőben van
	When:	Rákattint a futtatható állományra
	Then:	Elindul a program
Kilépés	Given:	Fut a program
	When:	Rákattint a kilépés gombra
	Then:	A programleáll
Betöltés	Given:	A főmenüben van
	When:	Rákattint a betöltés gombra
	Then:	Megjelenik a fájl választó menü
Betöltés	Given:	A fájl választó menüben van
	When:	Rákattint egy megfelelő fájlra
	Then:	A pontfelhő betöltődik

Navigáció	Given:	A pontfelhő betöltődött
	When:	Használja a navigációs inputokat
	Then:	A kamera elmozdul
Metaadatok megjelenítése	Given:	A pontfelhő betöltődött
	When:	Változtat a metaadat opciókon
	Then:	A megjelenő metaadatok megváltoznak

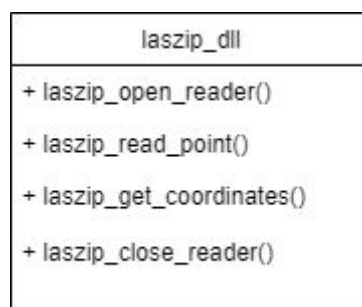
## A komponensek diagramja



## Modell

## Perzisztencia

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



## Nézet

## Tesztelés

### Egységtesztek

- 

### Végfelhasználói tesztek

- 

## Skálázhatóság