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

**Pontfelhő vizualizáció**

Nagy Richárd Tibor, Budapest, 2020

# Tartalomjegyzék

[**Tartalomjegyzék**](#_wahijp2tbo7l) **1**

[**Bevezetés**](#_vcxiyrhwgzvy) **2**

[Felhasználói dokumentáció](#_30j0zll) **3**

[Rendszerkövetelmények](#_1fob9te) 3

[Telepítés](#_3znysh7) 3

[Indítás](#_2et92p0) 3

[Főmenü](#_quq6rn86asya) 3

[Kamera](#_4d34og8) 3

[Fejlesztői dokumentáció](#_lnxbz9) **3**

[Elemzés - pontfelhők megjelenítése Unity-ben](#_35nkun2) 3

[Fejlesztői környezet](#_1ksv4uv) 3

[Felhasználói esetek diagramja](#_44sinio) 4

[Felhasználói esetek leírása](#_2jxsxqh) 4

[A komponensek diagramja](#_zgy635148j0x) 5

[Modell](#_3j2qqm3) 5

[Perzisztencia](#_41mghml) 5

[Nézet](#_2grqrue) 5

[Tesztelés](#_111kx3o) 5

[Egységtesztek](#_3l18frh) 5

[Végfelhasználói tesztek](#_206ipza) 5

[Skálázhatóság](#_2zbgiuw) 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 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

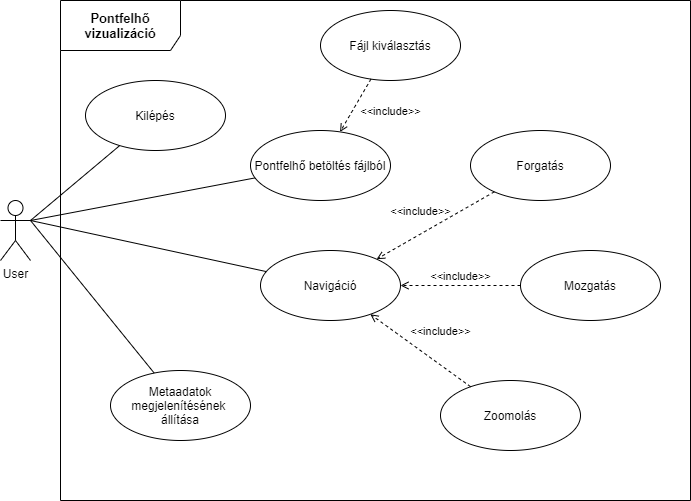
A beolvasott pontfelhőkből egy mesh-t generálunk. Ha lehetséges, ezzel a módszerrel lenne a leglassabb a betöltés, viszont maga a megjelenítés így lenne a legkisebb költségű.

## Fejlesztői környezet

A szoftver fejlesztése során az alábbi programokat használtam fel:

* Unity 2019.3
* Visual Studio 2019
* GitKraken
* GitHub

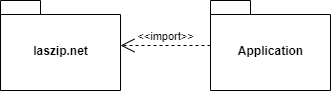
## Felhasználói esetek diagramja



## 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 | |

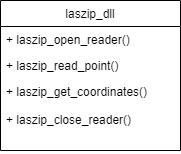
## A komponensek diagramja



## 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