

Első feladat:

Pontok és egyenesek

(Vigyázat: Under construction)

Szirmay-Kalos László

BME IIT

2024

Revíziók

Dátum	Változások	Verzió
2024. január	Első változat	1.0

Tartalom

1. FELADATKIÍRÁS.....	3
2. ELŐFELTÉTELEK.....	5
3. ALAPOK.....	5
4. PROGRAMOZÁS.....	6
4.1. OBJECT OSZTÁLY	6
4.2. POINTCOLLECTION OSZTÁLY	6
4.3. LINE OSZTÁLY.....	6
4.4. LINECOLLECTION OSZTÁLY	6
4.5. ONKEYBOARD ESEMÉNYKEZELŐ.....	7
4.6. ONMOUSE ESEMÉNYKEZELŐ.....	7
4.7. ONMOUSEMOTION ESEMÉNYKEZELŐ	7

1. Feladatkiírás

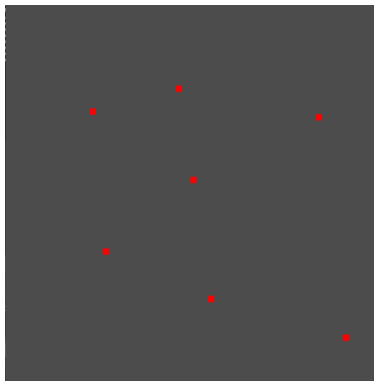
Az első feladat célja az alapvető koordinátagéometriai ismeretek felelevenítése és az OpenGL-lel való ismerkedés. Az elkészítendő programmal ízlésesen megválasztott szürke háttéren egérekattintásokkal pontokat vehetünk fel, a pontokra egyeneseket illeszthetünk, az egyeneseket tologathatjuk, és végül a kiválasztott egyenesek metszéspontjaiból újabb pontok hozhatók létre. A pontméret 10-es a vonalvastagság 3.

A világ-koordinátarendszer megegyezik a normalizált eszköz-koordinátarendszerrel, azaz a keletkező fénykép (**viewport**) a (-1,-1) és (1,1) sarokpontú négyzetnek felel meg. A fénykép pedig teljes mértékben lefedi a 600x600 felbontású alkalmazói ablakot.

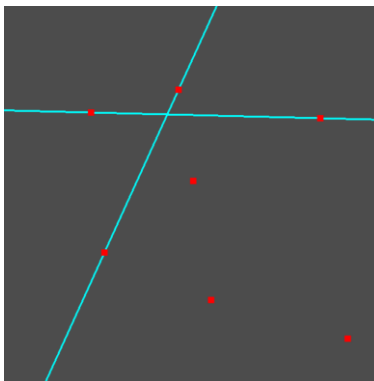
A CPU programban a 2D geometriát **külső szemlélettel**, azaz a 3D térbe ágyazva kell kezelni. Ez azt jelenti, hogy a 2D pontokhoz és vektorokhoz `vec3` típust rendelünk, pontokra a $z = 1$, vektorokra a $z = 0$ választással. Majd a csúcspont árnyalóban a 3D geometriát 4D-be ágyazzuk be $w = 1$ választással.

Felhasználói oldalról a programnak 4 állapota van, amelyeket rendre a p, l, m, i billentyűkkel lehet kiválasztani:

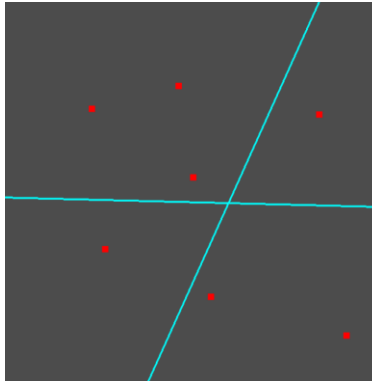
'p': Pont rajzolás, amely az egér bal gombjának lenyomásakor a kurzor helyére egy maximális intenzitású piros pontot tesz.



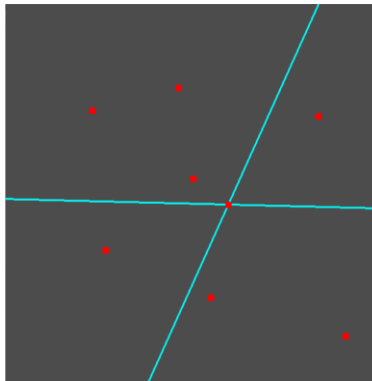
'l': Egyenes rajzolás, amelyhez két meglévő piros pontra kell rákattintani az egér bal gombjával. Az egyenes maximális intenzitású cián (türkisz) színben pompázik.



'm': Egyenes eltolás, amelyhez először az egyenest kell kiválasztani az bal egérgombbal, és ha ez sikeres, az egyenes követi a kurzort, azaz az egér lenyomott gomb melletti mozgását, mindaddig, amíg el nem engedjük az egérgombot.



'i': Metszéspont, amely két kiválasztott egyenes metszéspontjára (ha létezik) egy új piros pontot tesz.



A program a keletkező pontok Descartes koordinátáit, valamint a keletkező egyenesek implicit és parametrikus egyenleteit printf-fel a konzolra kiírja:

```
GL Vendor      : NVIDIA Corporation
GL Renderer    : GeForce GTX 960M/PCIe/SSE2
GL Version (string) : 3.3.0 NVIDIA 461.09
GL Version (integer) : 3.3
GLSL Version   : 3.30 NVIDIA via Cg compiler
Point -0.5, -0.3 added
Point -0.5, 0.4 added
Point -0.1, 0.6 added
Point 0.7, 0.4 added
Point -0.0, 0.1 added
Point 0.1, -0.5 added
Point 0.8, -0.8 added
Define lines
Line added
  Implicit: 0.9 x + -0.4 y + 0.3 = 0
  Parametric: r(t) = <-0.1, 0.6> + <-0.4, -0.9>t
Line added
  Implicit: 0.0 x + 1.0 y + -0.4 = 0
  Parametric: r(t) = <-0.5, 0.4> + <1.2, -0.0>t
Move
Intersect
Point 0.2, -0.1 added
```

2. Előfeltételek

Előfeltétel a Zöld háromszög és Konvex burok előadások megtekintése, különös tekintettel az alábbi fogalmak értelmezésére: VAO, VBO, csúcspont/vertex árnyaló, pixel/fragment árnyaló, uniform változó, GPU csúcspont árnyaló bemeneti regiszterek, csúcspont árnyaló táplálása a VBO tartalmával, egér események értelmezése.

Az ismereteket felfrissítendő, javasolt még „Klasszikus geometriák”, „Vektoralgebra” és „Analitikus euklideszi geometria” előadások meghallgatása, illetve a videók átfutása is, de az első feladathoz szükséges geometriai ismeretek nem haladják meg a középiskolában tanultakat.

3. Alapok

1. Írja fel a 2D egyenes paraméteres és implicit egyenleteit, és fejezze ki az implicit egyenlet paramétereit a parametrikus egyenlet paramétereiből és fordítva, valamint fejezze ki ezeket a paramétereket az egyenes két pontjának koordinátaiból (Erre az egyenesek felvételénél lesz szükség).
2. Vezesse le két egyenes metszéspontját meghatározó képletet (Erre meglepő módon akkor lesz szükség, amikor két egyenes metszéspontját kell kiszámolni):
 - a. Az implicit egyenletek alapján
 - b. A paraméteres egyenletek alapján
 - c. Az egyik egyenesre az implicit, a másikra a paraméteres egyenlet alapján
3. Adjon módszert egy egyenes két olyan pontjának előállítására, amelyek nem a $(-1, -1)$ és $(1, 1)$ sarokpontú négyzet belsejébe esnek (Ez azért kell, hogy az egyenes helyett szakaszt rajzolhassunk, az OpenGL ugyanis csak azt tud).
4. Vezesse le egy pont és egy egyenes távolságát meghatározó képletet mind az implicit egyenlet, mind pedig a paraméteres egyenlet alapján (Ez a mozgásra kiválasztáshoz kell).
5. Határozza meg azokat a lineáris képleteket, amely egy téglalapot a rajzlapnak feleltet meg. A téglalapot két sarkával adjuk meg, és azt várjuk, hogy a transzformáció az első sarkot a $(-1, -1)$, a másikat pedig a $(1, 1)$ pontra képezze le. A képletet írja át arra az esetre, amikor a kiindulási koordinátarendszer y tengelye felülről lefele, a cél koordinátarendszer y tengelye viszont alulról felfele áll (Ezzel a tudással fogjuk az operációs rendszer pixelkoordinátaiból átvárazsolni a pontjainkat normalizált eszköz koordinátarendszerbe, ahol a virtuális világ valójában létezik ebben a feladatban).

4. Programozás

4.1. Object osztály

Egy Object objektum csúcspontokat tárol a CPU-n, valamint a GPU-n egyetlen VAO/VBO-ban. A csúcspontok különálló pontokat, szakaszokat, töröttvonalat és konvex sokszögeket határozhatnak meg.

Az osztály műveletei:

1. A CPU-n tárolt gyűjtemény megváltoztatása.
2. A GPU reprezentáció szinkronba hozása az aktuális CPU reprezentációval
3. A GPU reprezentáció felrajzolása a megadott primitív típussal (pontok, szakaszok, háromszög legyező) és színnel.

4.2. PointCollection osztály

Egy PointCollection objektum pontokat tárol a CPU-n és a GPU-n összesen egyetlen VAO/VBO-ban.

Műveletei:

1. Új pont hozzávétele
2. Pont közelségi keresése
3. A pontok felrajzolása a kapott színnel.

Teszt: Az onInitialization() függvényben inicializálja az objektumot, és az onDisplay-ben rajzolja fel piros színnel. Tesztelje különböző koordinátákra, színekre és pont méretre.

4.3. Line osztály

A Line osztály egy egyenes geometriai tulajdonságait reprezentálja és az alábbi műveletekkel rendelkezik:

1. Konstruálás két pontból. A konstruáláshoz tartozik az implicit egyenlet és a paraméteres egyenlet paramétereinek a konzolra írása is.
2. Két egyenes metszéspontjának meghatározása.
3. Annak eldöntése, hogy egy pont rajta van-e (közel van-e, azaz normalizált eszköz-koordinátarendszerben a távolságuk 0.01-nél kisebb) az egyeneshez. Mivel normalizált eszköz-koordinátarendszerben a teljes fénykét 2 szélességű és magasságú, a 0.01 tényleg közelnek számít.
4. Az egyenes azon szakaszának meghatározása, amely a (-1,-1) és (1,1) sarokpontú négyzet belsejébe eső részt lefedi.
5. Az egyenes eltolása úgy, hogy a kapott ponton átmenjen.

4.4. LineCollection osztály

Egy LineCollection objektum egyeneseket tárol a CPU-n és a GPU-n összesen egyetlen VAO/VBO-ban.

Műveletei:

1. Új egyenes hozzáadása
2. Egy pont koordinátái alapján egy közeli kiválasztása
3. Egyenesek felrajzolása a kapott színnel.

4.5. onKeyboard eseménykezelő

A billentyű lenyomások a rendszerállapotot módosítják, az állapot pedig befolyásolja egy egéreseemény értelmezését. A 'p'=pont bevitel, 'l'=egyenes felvétel, 'm'=egyenes mozgatus, 'i'=egyenespár metszése billentyűeseményekre kell felkészülni.

4.6. onMouse eseménykezelő

Az onMouse függvényt valósítsa meg úgy, hogy a kapott operációs rendszerbeli pixelkoordinátákban adott kurzor pozíciót alakítsa át normalizált eszköz koordinátarendszerbeli ponttá, és az egérgomb lenyomásának hatására hajtsa végre az állapotfüggő reakciót, majd érvénytelenítse az ablakot (glutPostRedisplay).

Az operációs rendszer koordinátarendszerében az alkalmazói ablak bal felső sarka az origó, az y tengely lefelé mutat, az egység a pixel. Normalizált eszköz koordinátarendszerben pedig a viewport bal alsó pontja (-1,-1), a jobb felső pontja pedig (1,1). Implementálja a koordináta transzformációt feltételezve, hogy a viewport a teljes alkalmazói ablakot lefedi (megjegyzem azért érdemes kipróbálni azt az esetet is, amikor a viewport kisebb, mint az alkalmazói ablak).

Teszt: Egér gomb lenyomásakor a kurzor helyén vegyen fel egy új pontot és az ablakot érvénytelenítse glutPostRedisplay hívással, majd az onDisplay-ben rajzolja fel az módosított pontsorozatot.

4.7. onMouseMotion eseménykezelő

Ez az eseménykezelő az egér lenyomott gomb melletti mozgatusáról értesít, az operációs rendszer koordinátarendszerébe megadva, hogy a kurzor hova mozdult el. Ennek az eseménynek csak Egyenes mozgatus állapotnál van jelentősége, amikor az egyenest át kell helyezni úgy, hogy az új kurzor ponton átmenjen. Ez például az egyenes helyvektorának a kurzor helyére állításával valósítható meg, miután a kurzor helyét normalizált eszköz-koordinátarendszerbe transzformáltuk át.