**3D-s hardveres gyorsítás nélküli grafikus motor**

**Varga Pál Márk – F1SIL0**

**Bevezetés:**  
 Ezen projekt egy Java nyelven megírt 3D-s grafikus kártyás gyorsítást nélkülöző grafikus motor. Ez lényegében OpenGL, vagy DirectX vagy hasonló keretrendszer használatát nélkülöző 3D-s OBJ típusú objektumok megjelenítésére alkalmas szoftver. A projektben a leginkább használatos, és leghatékonyabbnak tartott eljárásokat, és algoritmusokat használtam, amelyek minden keretrendszer alapját képezik, beleértve ebbe, az alap vonal- és háromszögrajzoló algoritmusokat, valamint az ezekre vonatkozó transzformációs, és mozgató eljárásokat.

**Az alapok és a források:**

Olyan alapvető osztályok mint a mátrixosztályok valamint a gradiensek csak az alapvető matematikai műveletek és struktúrák objektum orientált szemléletét hivatottak megvalósítani, ezek megvalósítása általában implementációtól független, azok matematikai sajátosságai miatt. Magát az objektumot (majomfej), és az ezt megfelelő programbeli struktúrákká való konvertáló két osztályt a Chris Hecker nevezetű blogger bejegyzései között találtam. Egyéb források amelyek a projektben felhasználásra kerültek: Dmitry V. Sokolov – Tiny Renderer című tutorialja, Jason Gregory – Game Engine Architecture c. könyve, David Rousset 3D Software Engine in C# c. tutorialja, valamint a reddit/r/gamedev nevezetű subreddit, és főkent az opengl-tutorial.org nevezetű weboldal és egyéb források.

**Osztályok:**

**Ablak.java:**

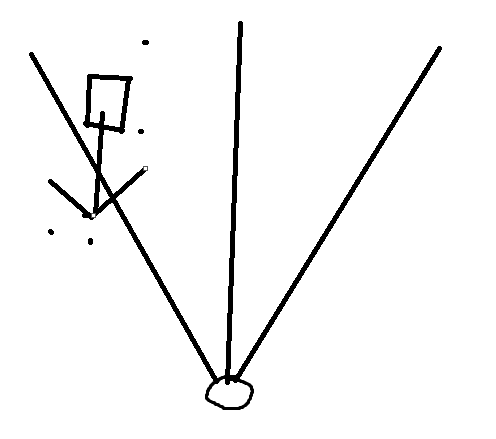
Az osztály implementálja a Canvas osztályt, eredetileg ennek azért lett volna nagy jelentősége, mert akartam User Inputot is rakni a programba, de odáig sajnos nem jutottam el. Egyébiránt az alkalmazást megjelenítő JFrame-t inicalizálja, és biztosítja a metódusokat a canvas bufferének a feltöltéséhez.

**Bitmap.java:**

Ez az osztály szolgál az egyes bitmapok létrehozásához, ez lényegében az egyes képek beolvasása, és azok pixeltömbökké alakításért felelős. Van továbbá két egy darab pixelt magában manipuláló metódus is.

**Csucs.java**

Alapvetően minden alakzat háromszögként, mint térbeli alakzatként jelenik meg, ezek egyszerű és szokványos reprezentációjának része, hogy a háromszögek elsődleges strukturális tárolása azok csúcsai által történik. A koordináták, és egyéb tulajdonságok tárolása vektorokon keresztül történik.

  
az fentebbi ábra lényegében az egyes koordináták kamerától való távolságával való leosztását jelenti, a jobb térérzet érdekében.

**Matrix4f.java**

Itt a grafikus keretrendszerekben meglévő mátrix osztályokhoz egy rendkívül hasonló osztály található. Az általában használt 4x4-es mátrix és az ahhoz tartozó elemi és szükséges mátrixműveletek találhatóak meg itt. Szorzás, egységmátrix, a képernyő inicializálásához szükséges mátrix, a tér 3 dimenziójának valamelyikében való elmozgatás stb.

**Vector4f.java**

Hasonlóan a mátrixhoz a vektor is „4 dimenziós”, ennek többszöri funkciója is van, a fejlettebb grafikus API-k quaternionokat is használnak, és ezekhez elengedhetetlen a 4-edik komponense, de egyébként is megkönnyebbíti a vektorok használatát, ha egyszerre akarunk skálázni, mozgatni stb.