Fejlett grafikai programok projektmunka, Billiárd

Csernai Kornél

2011. május 11.

1. Feladat

A csapat tagjai Bordé Sándor, Csernai Kornél és Ladányi Gergely. Csapatunk egy interaktív billiárdjáték megvalósítását tűzte ki célul. A tervek között szerepelt a fejlett kameramozgás (*kvaterniók*), játékmenet megvalósítása (interaktív szoftver, játékszabályok megkövetelése), modellezés és textúrázás (szoba: billiárdasztal, dákó, lámpák, padló, fal), és megközelítőleg realisztikus körülmények (billiárd golyók ütközése, visszapattanása, lyukba beleesés). A kitűzött célokat nagyrészt sikerült elérni.

A megoldásunk eredménye egy futtatható *OpenGL* program és a hozzá tartozó forráskód lett.

2. Munkakörnyezet

Az OpenGL programot *C++* nyelven objektum-orientált módon, a *GLUT* könyvtár és további szabad szoftverek segítségével készítettük el, *Linux* környezetben. Bizonyos modellek betöltésére a *3DS* könyvtárat és a *Bender* nevű modellező szoftvert használtuk. A kezdetekben teszt jelleggel a *Box2D* szimulációs szoftvert használtuk a játék fizikájának modellezésére (*ütközésdetektálás*).

Munkánkat nagyban segítette egy verziókövető rendszer, ami esetünkben a Subversion volt. Az egyes részfeladatokat külön-külön fájlokra bontottuk, így függetlenül tudtunk dolgozni.

3. Saját hozzátett munka

A keretrendszer kialakításában segítettem a csapat vezetőt. Ez alapvetően azt jelentette, hogy létrehoztam egy Subversion verziókövető lerakatot és időről-időre az elkészült kódot áttekintettem és objektum orientált környezetbe vittem át, a későbbi újrafelhasználás érdekében.

A játékmenet megvalósítása is részben közös munka volt, én konkrétan a fehér golyót szerepét, és a többi golyó lyukba esésének kezelését végeztem. Implementáltam a szokásos módon a 2D-s vektor osztályt, amely a számítások

során használatos műveleteket tudta (forgatás, eltolás, skálázás, hossz, normalizálás, belső szorzat, stb.).

A legfőbb kontribúcióm a golyók mozgásának szimulációja volt. Ide tartozik:

- golyók mozgatása súrlódás mellett
- golyók ütközése egymással (mozgó gömb-gömb ütközésdetektálás)
- golyók ütközése az asztal négy oldalával (gömb-sík ütközésdetektálás)
- golyók lyukba esése (mozgó gömb - statikus gömb ütközésdetektálás)
- állóhelyzetbe kerülés detektálása

A feladat megoldása során kettő algoritmust hasonlítottam össze.

Először a Box2D nevű könyvtárat integráltam, amellyel egyszerűen lehet szimulálni 2D-s testek interakcióját. Tapasztalataim alapján a megoldás elfogadható volt, azonban az egyes konstansok beállítása nehézkes volt, és a perdületet nem támogatta a rendszer. A precizitással nem volt gond, mert a rendszer folytonos szimulációt is tud végezni.

Második megoldást NeHe 30-as tutorial alapján készítettem el. A leírásból példaprogramjából integráltam a golyók egymással ütközését és a golyók fallal ütközését. Az elkészült kód átlátható lett és az algoritmus hasonlóan jól teljesít.

3.1. Technikai részletek

Az ütközés detektálás két részre bontható. Az első fázisban eldöntjük, hogy egyáltalán kell-e ütközéssel foglalkoznunk. Amennyiben nincs ütközés, a golyókat egyszerűen elmozgatjuk a sebességvektoruknak megfelelően. Különben az időben legközelebbi ütközést kell először megvizsgálni.

A szimuláció diszkrét módon történik, megadott lépésközzel. Az ütközés detektálásnál ezt kisebb időszakaszokra kell bontani.

Mivel a gömbök és a síkok száma kevés (16 és 4), így a naív módszer elegendően gyors eredményt ad, nincs szükség bináris partíciónálásra és egyéb módszerre.

Legyen adott az egyes golyókat reprezentáló gömbök pozíció helyvektora és sebesség vektora, a falakat reprezentáló sík normálvektora és a sík egy pontja.

3.2. Golyók és falak

A golyók és falak ütközése lényegében mozgó gömb és statikus sík ütközését jelenti. Az ütközés detektálása elvégezhető egy sugár segítségével.

Sugár egyenlete:

$$P = \overrightarrow{p_0} + t * \overrightarrow{d}$$

A sík egyenlete:

$$\overrightarrow{N} \cdot \overrightarrow{X} = c$$

A kettő együtt, t kifejezve:

$$t = \frac{\overrightarrow{N} \cdot (\overrightarrow{X} - \overrightarrow{p_0})}{\overrightarrow{N} \cdot \overrightarrow{d}}$$

Bizonyos esetekben kizárhatjuk az ütközést:

- $\overrightarrow{N} \cdot \overrightarrow{d} = 0$, a síkkal párhuzamos a sugár
- $t < 0$, az ütközés a sugár mögött történhetett korábban

Az ütközési pont megtalálása után az ütközés eredményét meg kell adni. Jelen esetben az ütközést az

$$\overrightarrow{R} = 2 * (-\overrightarrow{O} \cdot \overrightarrow{N}) * \overrightarrow{N} + \overrightarrow{O}$$

képlettel adhatjuk meg, ahol

- $\overrightarrow{R}$ az eredmény irányvektor,
- $\overrightarrow{O}$ a régi irányvektor,
- $\overrightarrow{N}$ az ütközési pontba eső normálvektor.

$\overrightarrow{O}$ és $\overrightarrow{N}$ egységvektoroknak kell lenniük, a procedúra végén $\overrightarrow{N}$ hosszával újra kell skálázni az eredményvektort.

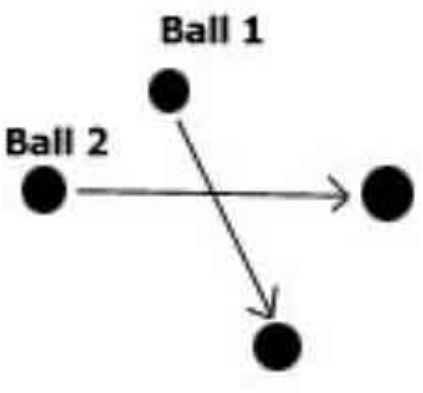
C++ kódrészletek:

```
bool TestIntersionPlane(const Plane& plane, const TVector& position,
const TVector& direction, double& lambda, TVector& pNormal) {
    double DotProduct = direction.dot(plane.normal);
    double l2;

    // determine if ray parallel to plane
    if (DotProduct < ZERO  && DotProduct > -ZERO){
        return false;
    }

    l2 = plane.normal.dot(plane.position - position) / DotProduct;

    if (l2 < -ZERO) {
        return false;
    }

    pNormal = plane.normal;
    lambda = l2;

    return true;
}
// ...
```

```
rt2 = movement[BallNr].mag();
movement[BallNr].unit();
movement[BallNr] = TVector::unit( (normal*
                   (2*normal.dot(-movement[BallNr])))
                   + movement[BallNr] );
movement[BallNr] = movement[BallNr]*rt2;
```

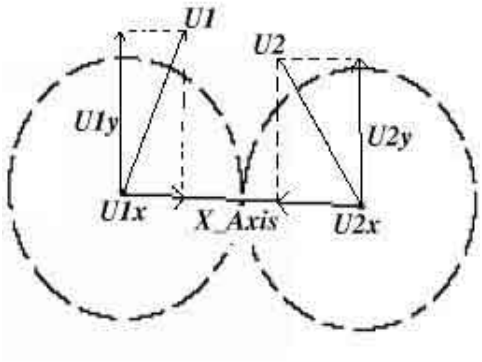
3.3. Mozgó golyók egymással ütközése

Két mozgó golyó esetében nem elegendő, hogy a két sebességvektor egymást metszi. További nehézség, hogy gyorsan mozgó golyók esetén az is előfordulhat, hogy a golyók egymáson áthaladnak. Ezért a diszkrét szimulációt kisebb lépésekre kell bontani és meg kell vizsgálni, hogy a két gömb középpontja hogyan viszonyul egymáshoz. Minél kisebb lépéseket használunk, annál pontosabb lesz a szimuláció.



1. ábra. Két mozgó golyó esete.

Ha megtaláltuk az ütközési pontot, az ütközés eredményét meg kell adni.



2. ábra. Két gömb ütközésének esete.

Legyen U_1 és U_2 a két golyó sebességvektora, vesszük ezeknek x és y komponenseit: $U_{1x}, U_{1y}, U_{2x}, U_{2y}$. Az x komponens a relatív pozíció, y erre merőleges.

V_1 és V_2 az eredmény sebességvektorok, ezeknek x és y komponensei: V_{1x}, V_{1y}, V_{2x}, V_{2y}. A két gömb tömege egyenlő.

A megoldás:

$$V_{1x} = \frac{U_{1x} + U_{2x} - (U_{1x} - U_{2x})}{2}$$

$$V_{2x} = \frac{U_{1x} + U_{2x} - (U_{2x} - U_{1x})}{2}$$

$$V_{1y} = U_{1y}$$

$$V_{2y} = U_{2y}$$