

Projektmunka III

Lövedék detektálás és elkerülés gépi látás segítségével Unity 3D-ben

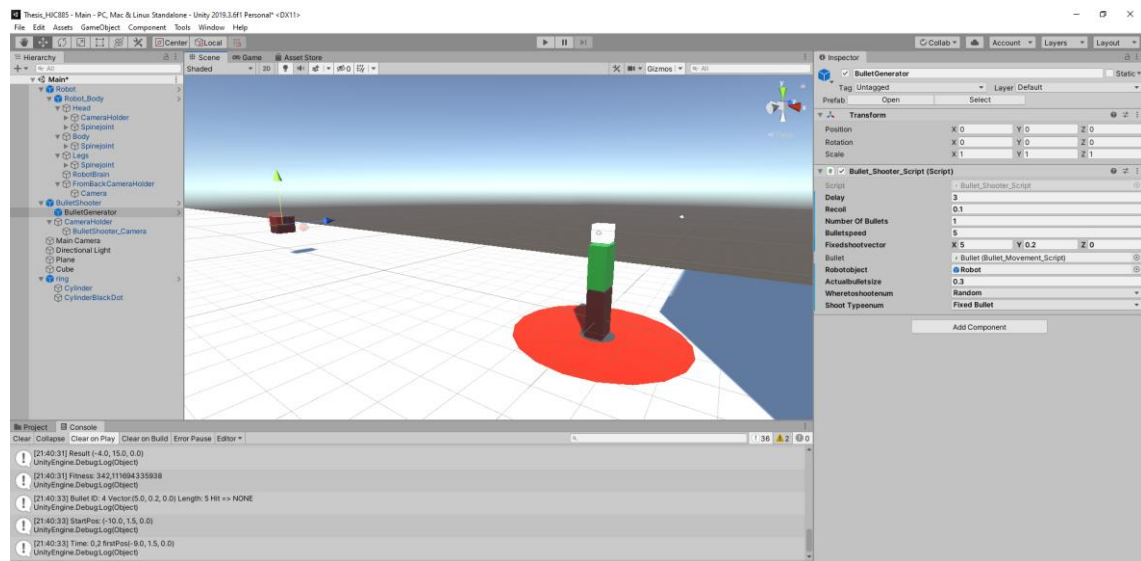
Váraljai Gábor HJC885

Tartalom

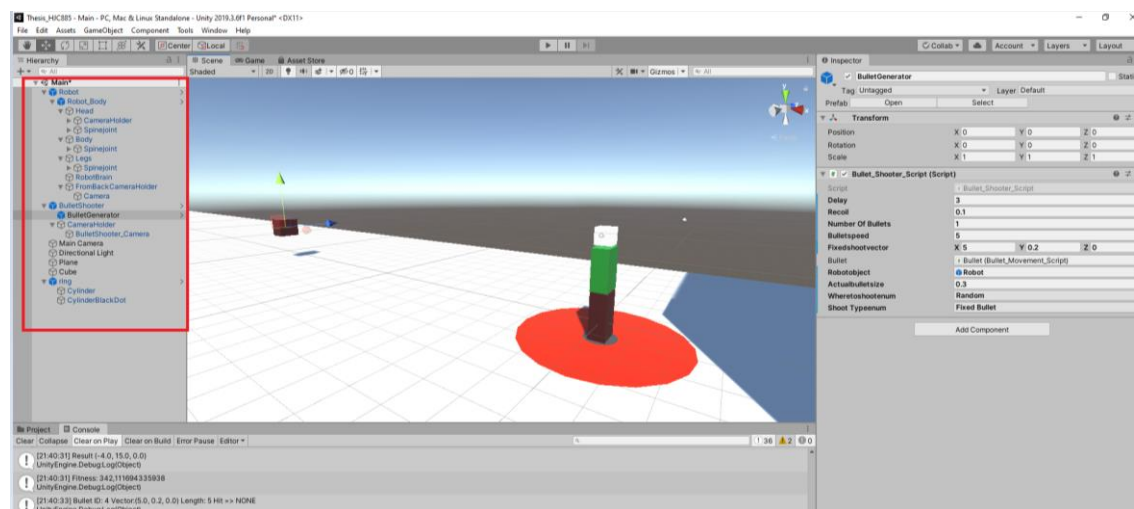
Program működése	2
User interface.....	3
Fixed Bullet:.....	4
Random:	5
Learned bullets:.....	5
Program futtatása	5
Kamera nézetek váltása:	6
Belső nézetes kameranézet:	6
Lövedék szemszögéből készült kameranézet:	6
Több szemszögű kameranézet:.....	7
Program futása alatt történő folyamatok:	7
Mintavételezés.....	7
Képfeldolgozás	9
differenciakép készítése.....	9
kép szürkeárnyalatossá alakítása	9
Küszöbölés után fekete-fehérré alakítás.....	10
Gauss szűrés a kör detektáláshoz	10
Körök detektálása	11
Vektorok átadása a mozdulatkezelő rendszernek.	11
legoptimálisabb mozdulatsor visszaadása, genetikus algoritmus segítségével.....	11
mozdulatsor végrehajtása.....	11
mozdulat elmentése	12
egy lövéshez tartozó adatok	12

Program működése

User interface



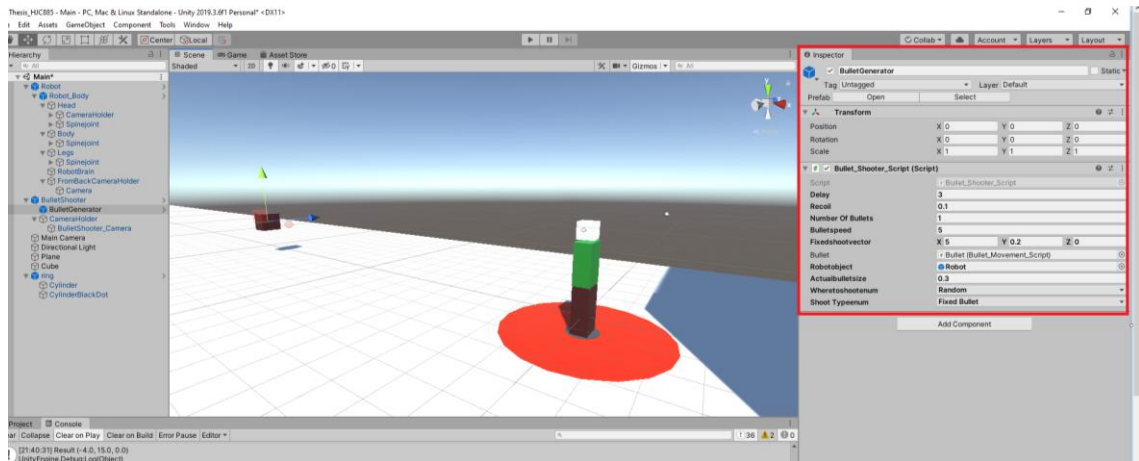
A fenti képen látható a szimuláció konfigurálásához, beállításához használható UI.



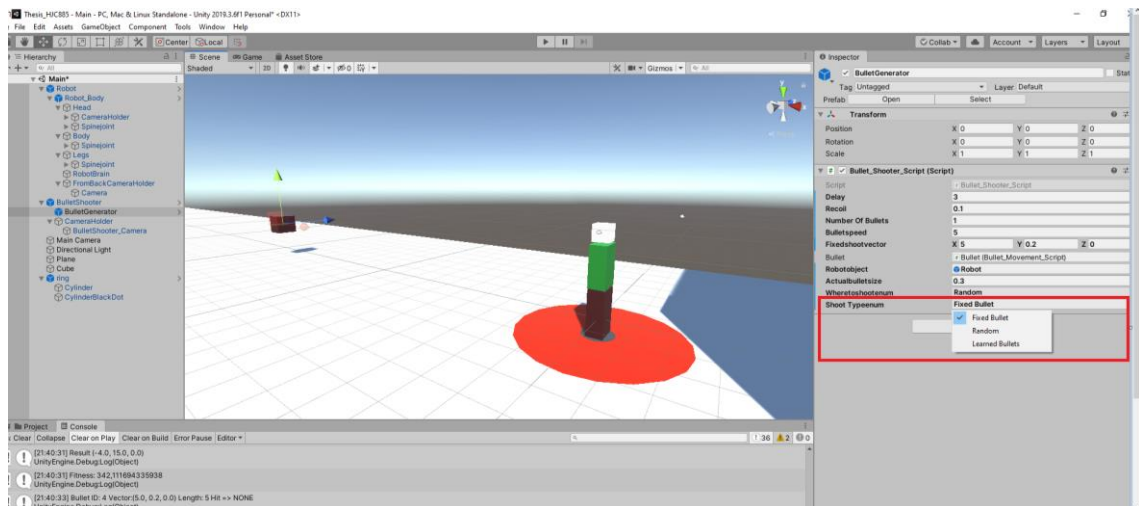
Bal oldalon található a Main scene, ahol a szimuláció történik, benne a 2 fő GameObject-tel, a Robot, és BulletShooter-rel.

A hierarchiából látható, hogy egy-egy GO (GameObject), tartalmazhat több objectet, jelen esetben a Robot Robot_Body-t, ami a „fizikai” megtestesülése a robotnak, és azon belül a RobotBrain-t, amely magát az üzleti logikát tartalmazza (Kép->vektor, vektor->mozdulat).

A Bulletshooter tartalmaz egy BulletGenerator objectet, ami segítségével lövedékeket tudunk generálni, és elindítani a robot felé.



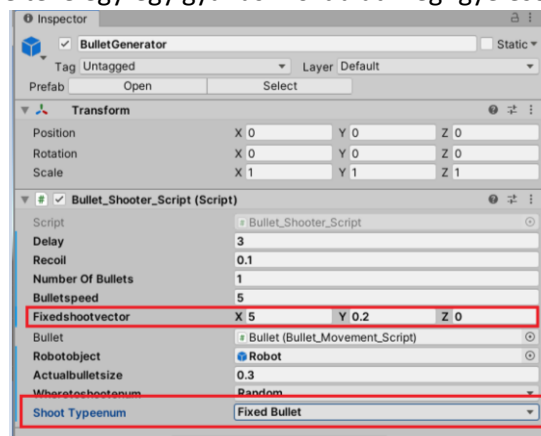
A jobb oldalon található a bal oldalon korábban kijelölt GameObject Inspectorra, ahol jelen esetben a generáláshoz szükséges paramétereket lehetséges beállítani.



Fontos kiemelni a „Shoot Typeenum” enumot, ahol az alábbi 3 funkciót lehetséges váltani.

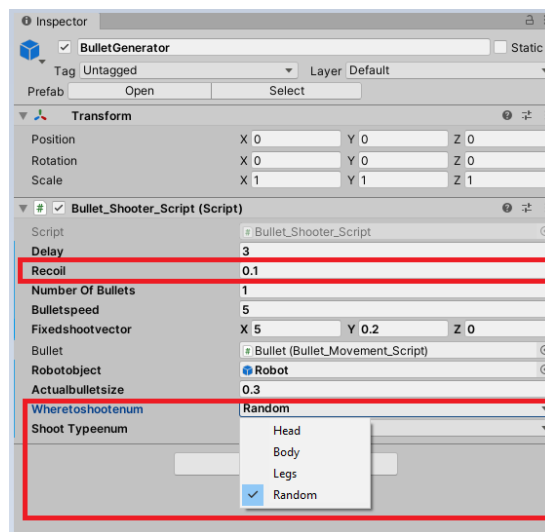
Fixed Bullet:

A generáló mindig a „Fixedshootvector” paraméterként megadott vektorra állítja a lövedék pályáját, ezzel lehetővé téve egy-egy gyanús mozdulat megfigyelését, a bugok észrevételét.



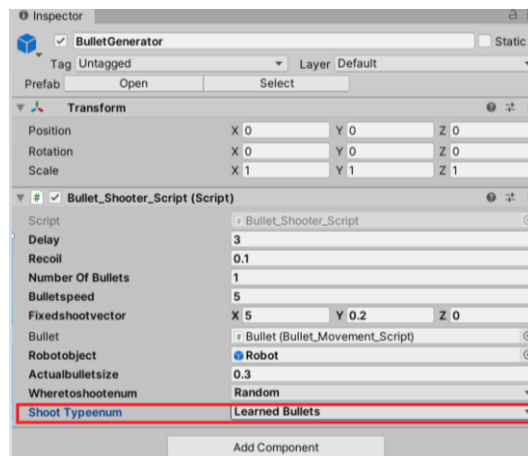
Random:

A generáló véletlenszerű, de duplikációmentes irányvektort állít elő, ami a „wheretoshootenum” által megadott testrésze lő „Recoil” mértékű szórással.



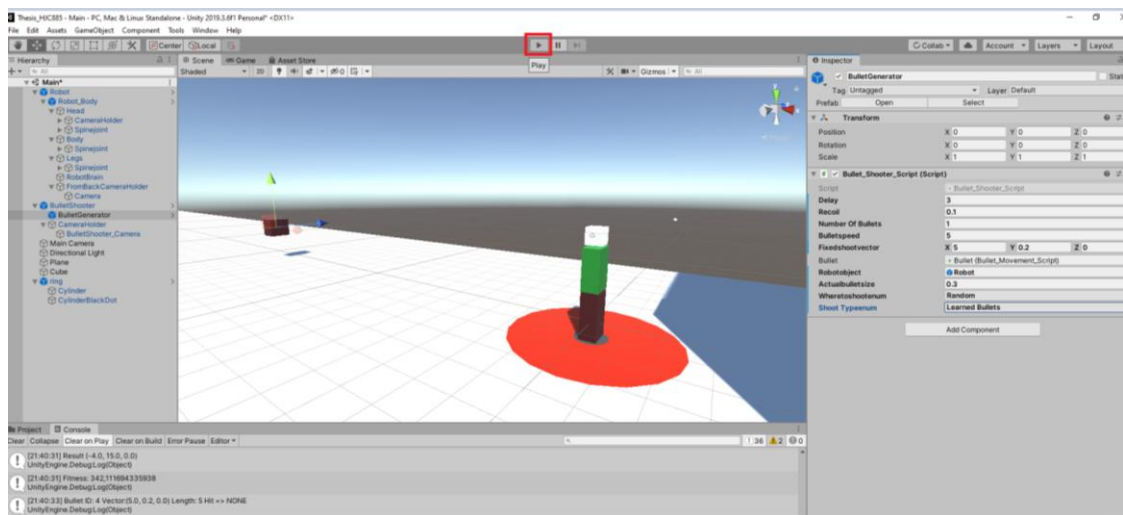
Learned bullets:

A generáló a már megtanult lövések pályáját adja át a lövedéknek, ezzel segítve a demonstrációt, illetve bugok észrevételét.

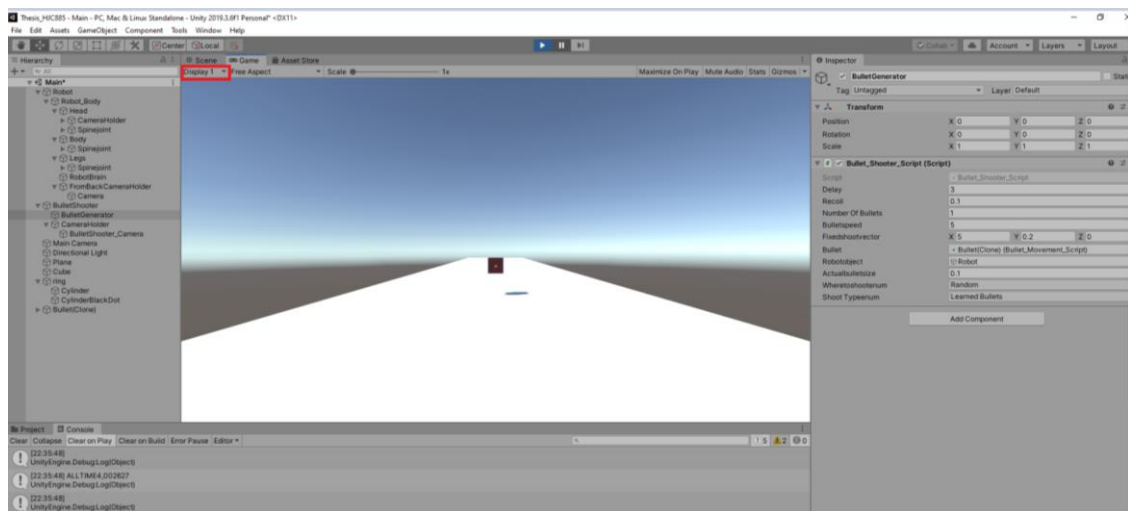


Program futtatása

A program indításához középen felül a „Play” gombra kell kattintani, ami elindítja a szimulációt.



Kamera nézetek váltása:



A szimulációhoz több kameranézet is elkészült, az átláthatóság érdekében.

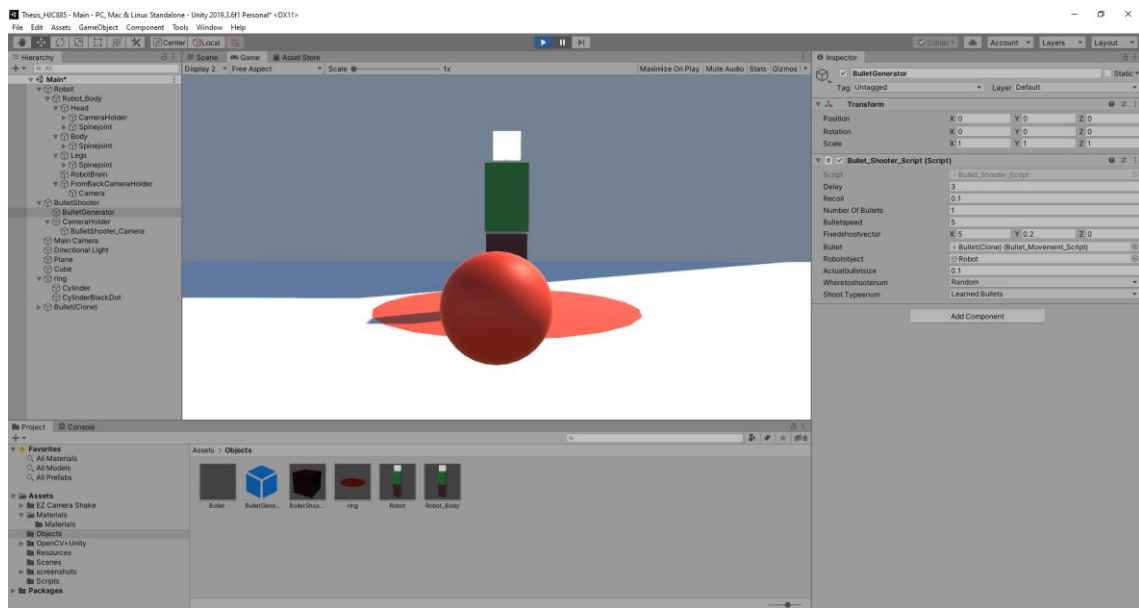
Jelenleg 3 kameranézet közül lehetséges választani.

Belső nézetes kameranézet:

Az első a fenti képen is látható Belső nézetes kameranézet, ahol a mintavételezés zajlik a RobotBrain képfeldolgozó algoritmusának bemeneteként.

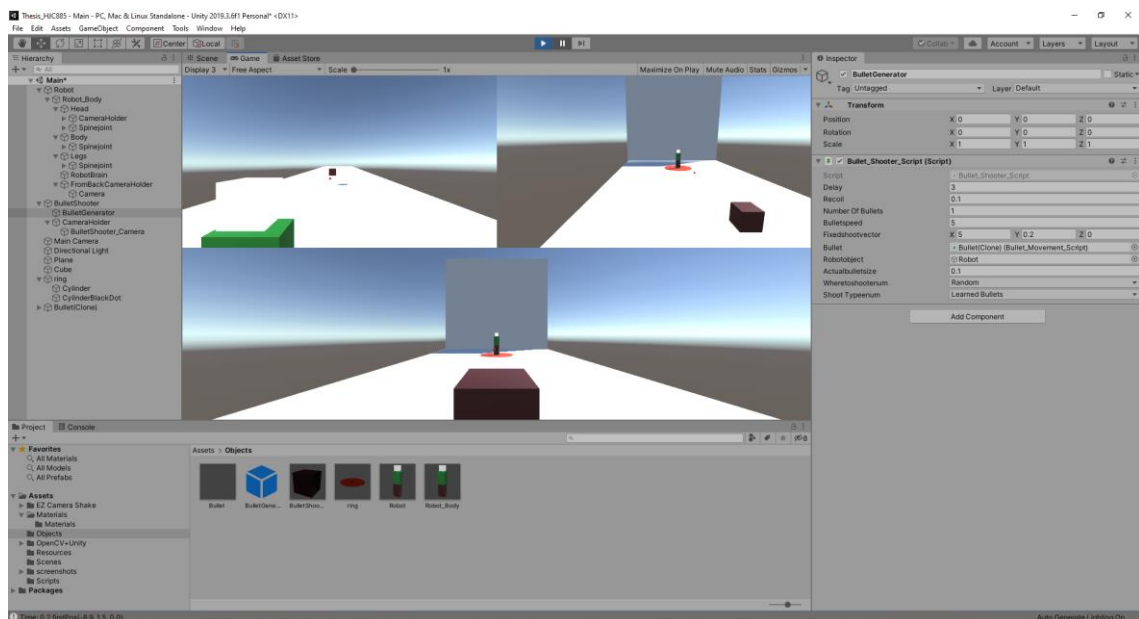
Lövedék szemszögéből készült kameranézet:

Itt lassan ráközelítve a lövedék pályájára meg lehet figyelni annak szemszögéből az esetleges ütközéseket, elkerüléseket.



Több szemszögű kameranézet:

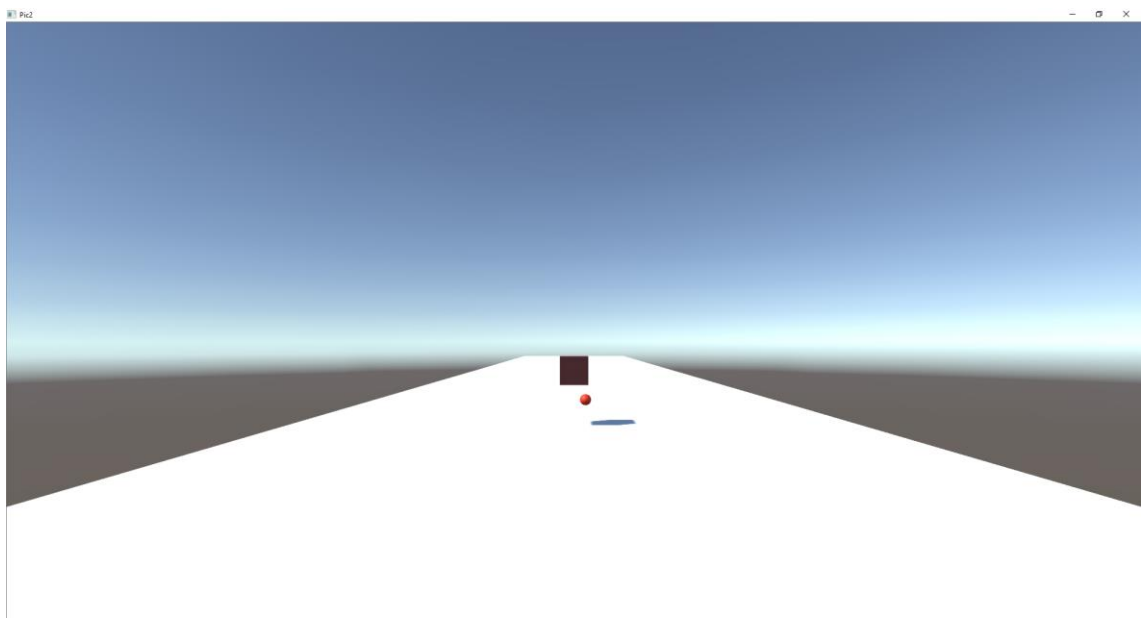
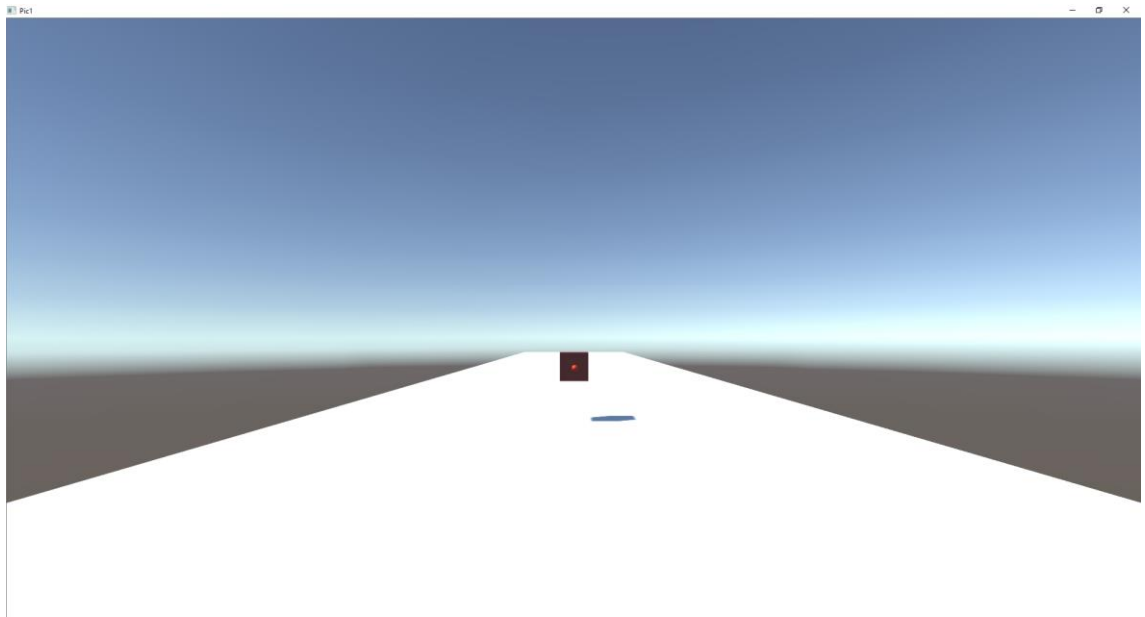
3 kamera együttes nézete (generáló, robot váll felőli, main kamera), ahol egyszerre meg lehet tekinteni több szögből a lövésre leadott mozdulatokat.



Program futása alatt történő folyamatok:

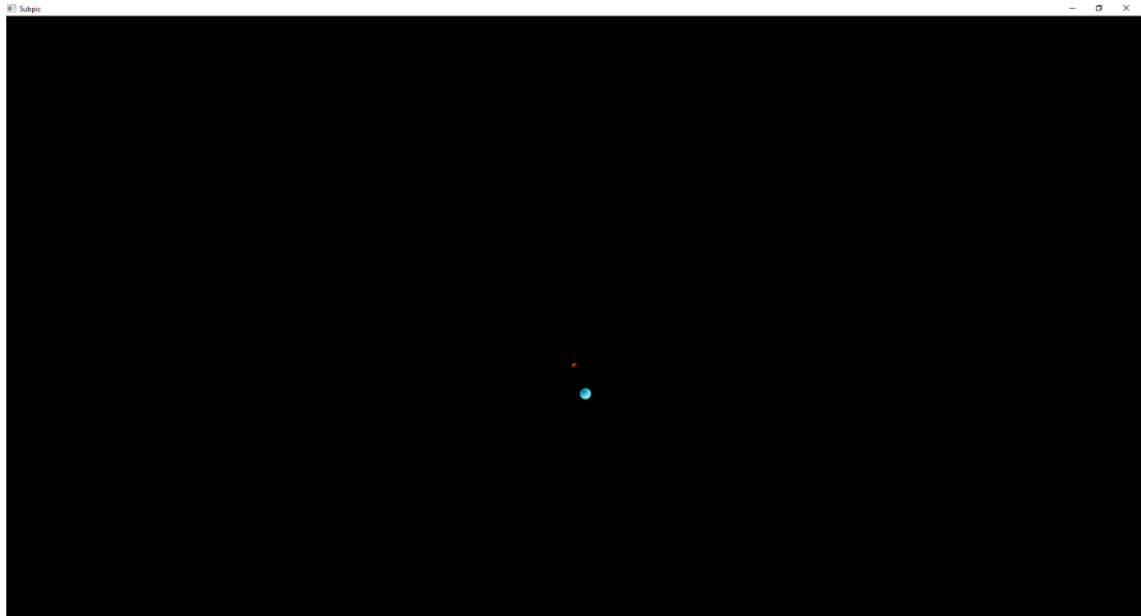
Mintavételezés

„RobotEyes” Robot belsőnézetes nézetéből képernyőörgzítéseket készít, melyeket átad a RobotBrain-nek.

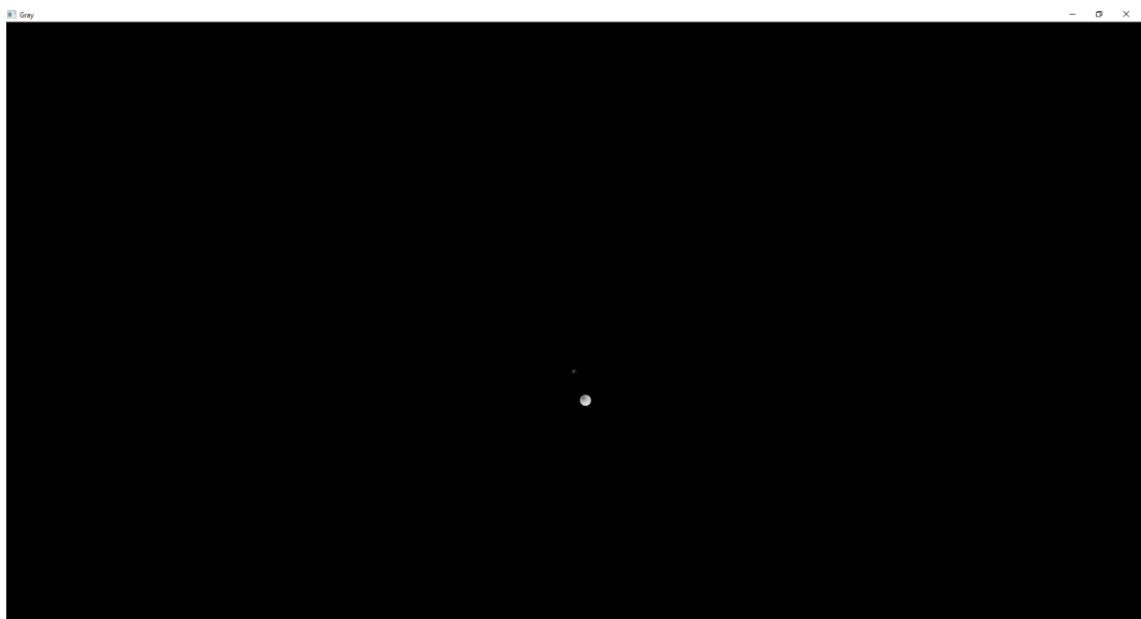


Képfeldolgozás

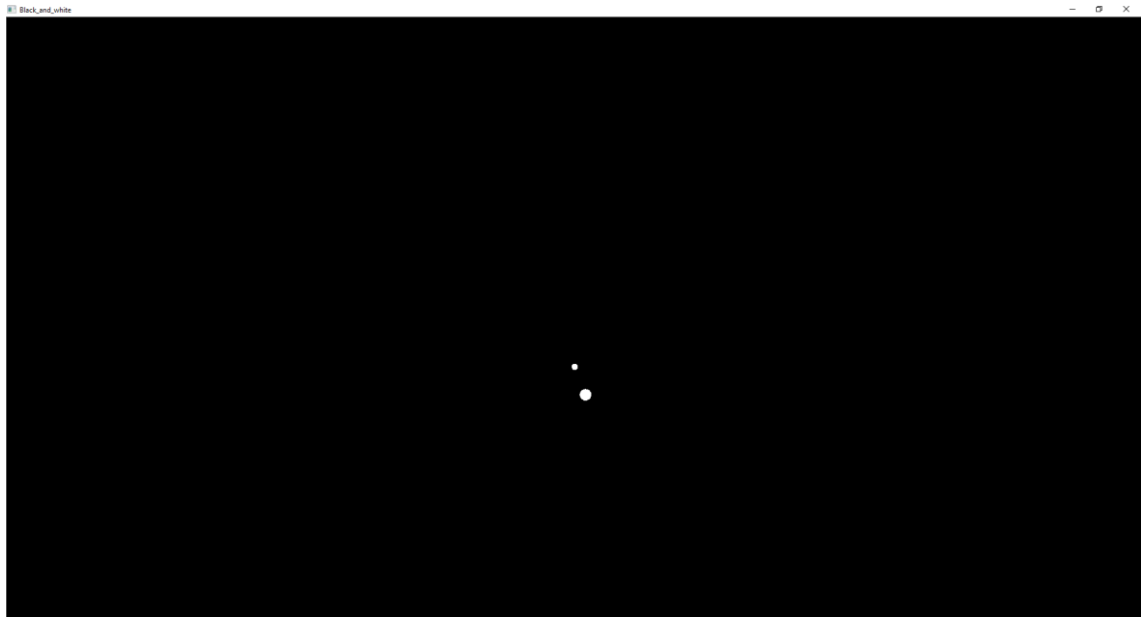
differentiálkép készítése



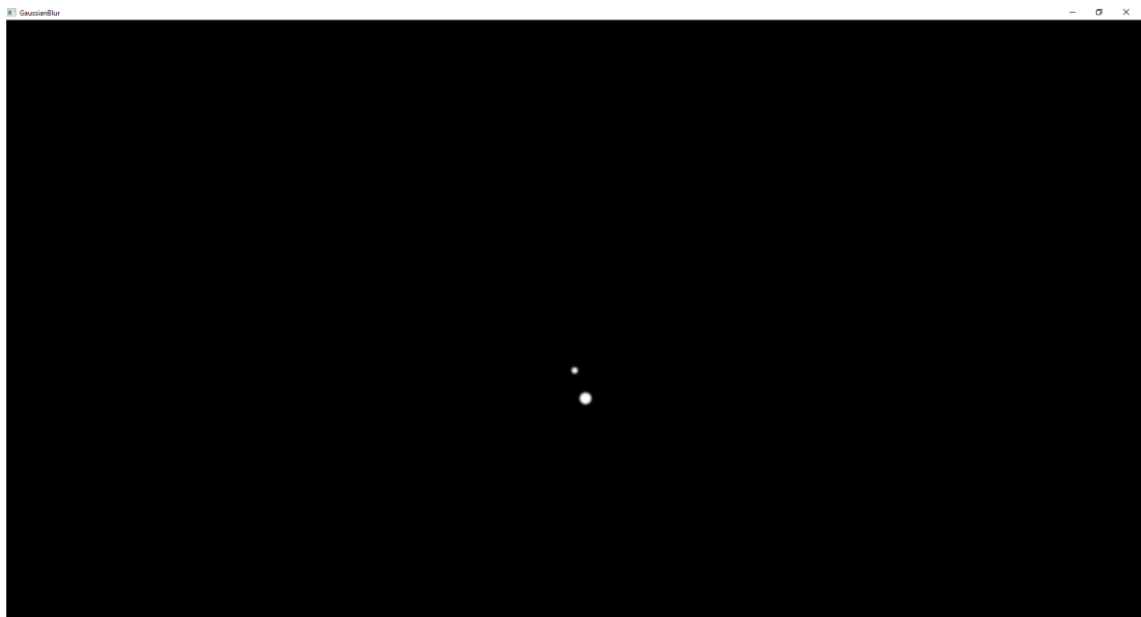
kép szürkeárnyaltossá alakítása



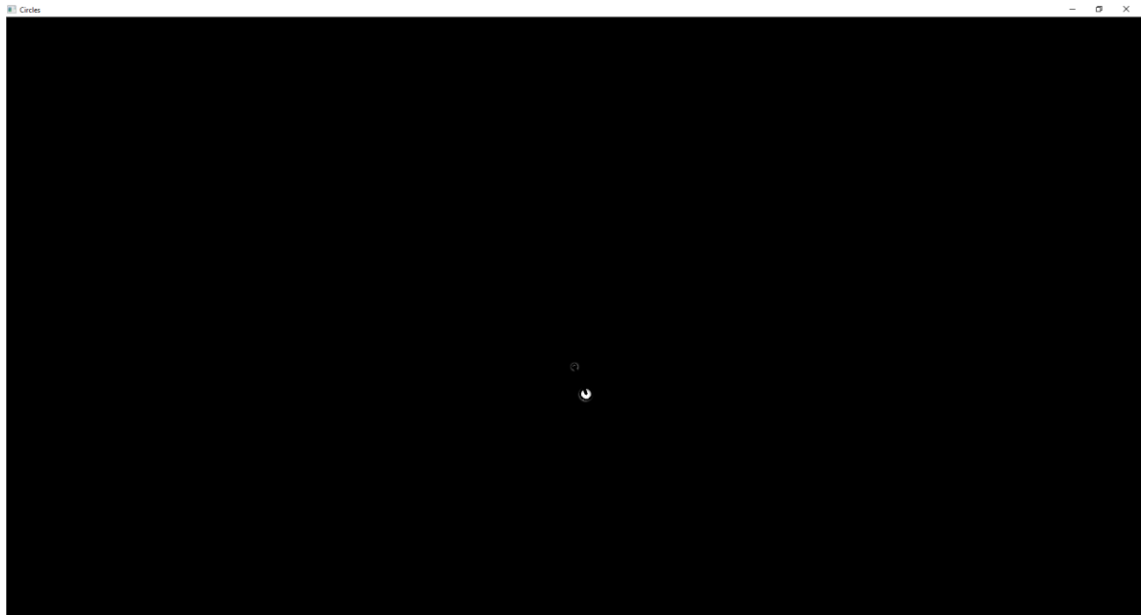
Küszöbölés után fekete-fehérré alakítás



Gauss szűrés a kör detektáláshoz



Körök detektálása



Vektorok átadása a mozdulatkezelő rendszernek.

```
FIRST: (980.5, 636.5, 0.0)
ie.Debug:Log(Object)
SECOND: (960.5, 590.5, 0.0)
ie.Debug:Log(Object)
Result (-20.0, -46.0, 0.0)
```

Mozdulatkezelő rendszer a kapott vektorról eldönti, hogy szerepelt-e már, ha igen végrehajtja a már megtanult mozdulatot, ha nem akkor elindítja a mozdulatkereső algoritmust a kapott vektor alapján.

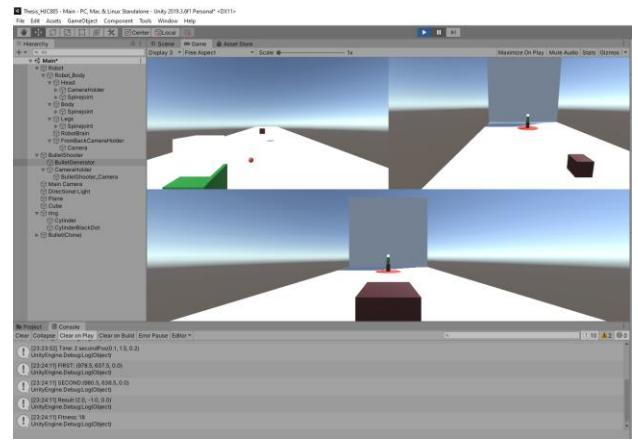
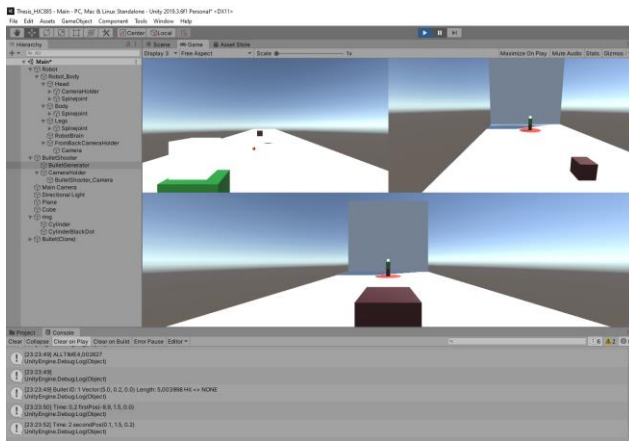
legoptimálisabb mozdulatsor visszaadása, genetikus algoritmus segítségével.

legoptimálisabb mozdulat tulajdonságai (fitness számítás szempontjai):

- a mozdulat során, egyik testrész sem ütközik, illetve tartalmazza egymást. (pl. fej nincs a testben)
- a mozdulat során a testrészek nem hagyják el egy bizonyos környezetüket. (pl.: megszorítások, gerinc bevezetése)
- a mozdulat után nincs egyetlen veszélyeztetett testrész sem (lövedék nem talál el semmilyen testrészt)
- a mozdulat a lehető legkevesebb energiába kerül (lehetőleg kevés egész testes mozdulat, inkább egy-egy testrész mozgatása, és leginkább csak forgatás)

mozdulatsor végrehajtása

RobotBrain átadja a mozdulat elvégzéséhez szükséges vektorokat, melyeket a robot elvégz, ezzel a testrészei pozícióját módosítva.



mozdulat elmentése

egy lövéshez tartozó adatok

1	(5.0, 0.2, 0.0)	(0.0, -11.0, 0.0)	(0.017, 0.099, 0.022)	(12.982, 2.930, -2.881)	(-0.092, -0.022, -0.052)	(-2.575, -6.523, -0.166)	(0.000, 0.000, 0.000)	(0.000, 0.000, 0.000)	(0.000, 0.000, -0.273)
---	-----------------	-------------------	-----------------------	-------------------------	--------------------------	--------------------------	-----------------------	-----------------------	------------------------

1. Lövedék azonosítása (1-2)
 - a. 1. vektor lövedék iránya
 - b. 2. vektor lövedék iránya képfeldolgozás által
2. Lövedék elkerüléséhez szükséges mozdulat (3-9)
 - a. 3.vektor fej mozgatója
 - b. 4.vektor fej forgatója
 - c. 5.vektor test mozgatója
 - d. 6.vektor test forgatója
 - e. 7.vektor láb mozgatója
 - f. 8.vektor láb forgatója
 - g. 9.vektor egész testes mozdulat