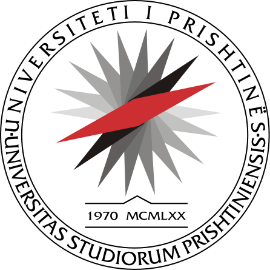
UNIVERSITETI I PRISHTINËS “HASAN PRISHTINA”

FAKULTETI I SHKENCAVE MATEMATIKO-NATYRORE

DEPARTAMENTI I MATEMATIKËS-“SHKENCA KOMPJUTERIKE”



Detyr ë Projekti

**Lënda:** Inteligjencë Artificiale

**Tema:** Genetic Algorithm - Car Game

**Profesori: Dr. sc Eliot Bytyqi Punuan: Din Dedaqi**

**Asistenti: Dr. sc. Astrit Hulaj Muhamet Sadiku**

**Hyrje**

Në këtë projekt ne kemi krijuar një program i cili do mësoj një veturë të vozitë me anë të algoritmit gjenetik. Algoritmi gjenetik ka disa faza të cilat i kemi implementuar në kod. Fillimisht kemi gjeneruar një popullsi fillestare të rastësishme e cila tenton ta arrijë qëllimin pa dalur nga rruga. Kjo popullsi fillestare vendoset në pozitë të caktuar të rrugës dhe fillon të lëvizë drejt qëllimit derisa të mos preken skajet e rrugës. Secilit individ pastaj i llogaritet distanca nga qëllimi për të caktur fitnesin. Sa më e vogël distanca nga qëllimi aq më i madhë është fitnesi. Pastaj në bazë të fitnesit zgjedhen prindërit të cilët përcaktojnë gjeneratën e re. Sa i madh është fitnesi i një individi aq i madh është probabiliteti që ai individ të përzgjedhet për kalimin e gjeneve në gjeneratën tjeter. Pas secilës gjeneratë vozitja e veturës shkon duke u përmirësuar dhe vetura shkon gjithnjë duke iu afruar qëllimit. Në gjeneratën e fundit kemi përmirësimin përfundimtar të vozitjes e cila pastaj shfaqet në kornizë(Frame).

**Shtjellimi dhe shpjegimi i kodit**

Kodi ka disa klasa që mirën me vizatimin dhe ekzekutimin e lojës, por pasi që punimi është në lidhje me AI nuk do të i shtjellojmë ato. Në lidhje me AI kemi dy klasa të rëndësishme. Ato janë LearnMain dhe CarMovement, LearnMain është menaxheri kurse CarMovement është një instancë e popullatës. Në LearnMain bëhen hapat e Algoritmit Gjenetik. Në fillim bëhet krijimi i popullatës fillestare. Një instancë CarMovement merr 500 instruksione për lëvizje këto instruksione e përbëjnë kodin gjenetik të instancës dhe nxjerrën në mënyrë të rastësishme. Pasi që bëhet gjenerimi fillestar e fitojmë gjeneratën tjetër me anë të një loop që ekzekutohet sa gjenerata dëshirojmë ti kemi. Me anë të asaj loop-e të thjeshtë mundemi që të krijojmë gjeneratën e ardhshme nga gjeneratën e kaluar.

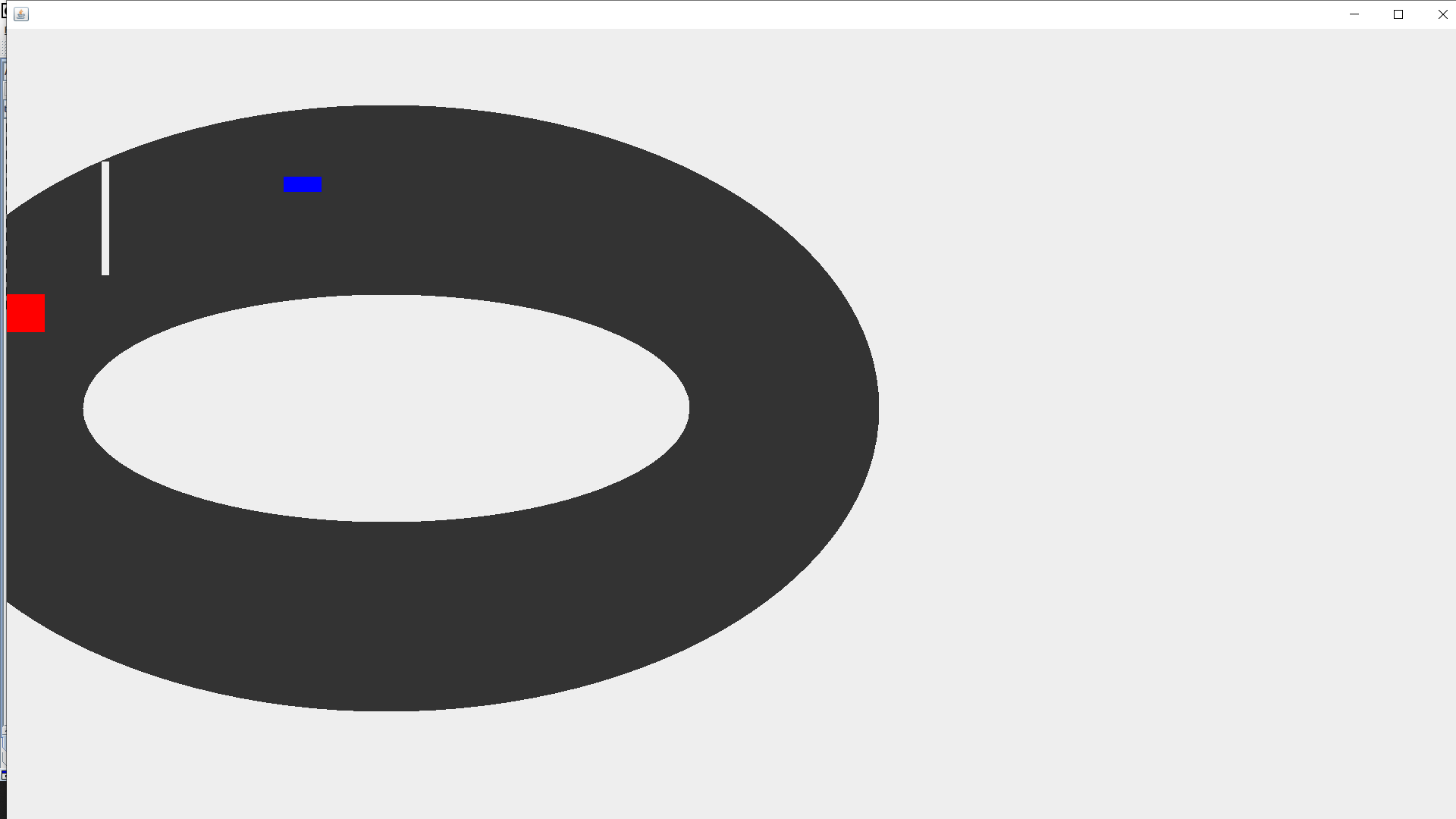
Metoda getNewGeneration() e kryen gjenerimin e gjeneratës së ardhshme. Në fillim ajo kalkulon fitnesin e çdo instance duke e thirrur metodën move. Pasi që kalkulimi merr shumë kohë më lehtë është ta bëjmë me shumë Thread-s. Kalkulimi bëhet thjeshtë duke thirrur metodën move tek çdo CarMovement. Pasi që gjejmë fitnesin për çdo instancë gjejmë probabilitetin për çdo instancë duke u bazuar në fitnesin e tyre. Kur e gjejmë fitnesit zgjedhim dy prindër në bazë të probabilitetit dhe këta prindër bashkohen tek gjysma e Array të “movements”. Pra merret gjysma e kodit gjenetik e njërit prind dhe gjysma tjetër merret nga prindi tjetër. Me këtë bashkim kemi fituar një instancë të re. Këta hapa përsëriten aq herë sa ka hapësirë popullsia dhe fitojmë gjeneratën e re.

Për konfigurimin e AI kemi disa ndryshore statike që mund ndërrohen si POPULATION, THREADS, MOVEMENTS, GENERATIONS.

CarMovement funksionon duke u bazuar në shkrimin e lojës. Detektimi i përplasjes së veturës me skajet e rrugës bëhet duke bashkuar Area të makinës dhe Area të rrugës dhe e kontrollon a është e njëjtë me Area të rrugës. Një udhëtim përfundon nëse një makinë përplaset me skajet e rrugës ose “movements” mbarojnë. Për arsye të testimit “movements” është 500 por shumë lehtë mundë të rritet më shumë. Makina ndalet çdo lëvizje për 17 mili sekonda (60 shfaqje për sekondë) dhe për shkak që dëshirojmë të ruajmë fitnesin si dhe ruajmë edhe pozitën e fundit të makinës. Makinën e ruajmë si vijë dhe fitnesin e kalkulojmë duke kontrolluar sa larg është vija me pikën e qëllimit. Ky numër është në proporcion të zhdrejtë me numrin e fitnesit. Me këtë mënyrë e gjejmë funksionin e fitnesit.

**Ekzekutimi i aplikacionit**

Pasi që të ekzekutohet kodi me anë të metodës main() programi fillon të gjejë rrugën më përshtatshme duke përmirësuar vozitjen e veturës për secilën gjeneratë dhe në fund e shfaq si animacion në kornizë.



**Fig. 1.** Ekzekutimi i kodit dhe shfaqja e kornizës.

Në vazhdim kemi shfaqur kohën për të gjetur rrugën per numër te ndryshëm të gjeneratave. Si dhe tregon se sa i ështe afruar qëllimit.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MOVEMENTS | THREADS | POP | GENERATIONS | KOHA | DISTANCA |
| 500 | 100 | 100 | 3 | 14152ms | 345.03 |
| 500 | 100 | 100 | 5 | 28272ms | 152.97 |
| 500 | 100 | 100 | 10 | 58488ms | 143.17 |
| 500 | 100 | 100 | 20 | 85706ms | 93.13 |

**Përmbyllje**

Në këtë punim kemi treguar se si në një implementim të lojës me makinë mund të aplikojmë Algoritmin Gjenetik për të mësuar AI që të vozis makinën deri tek pika e dëshiruar. Nuk kemi treguar të gjitha tentimet e makinave por kemi shtuar mundësi për konfigurimin e shpejtësisë, popullsisë, gjatësisë, gjeneratave.

**Referencat**

[An Overview of Genetic Algorithms: Part 1, Fundamentals, David Beasley, David R. Bull, Ralph R. Martin, kapitujt 1-5, faqet 1-13].