

Інтернет + Бібліотека

97.81% Оригінальність	2.19% Схожість	298 Джерела
-----------------------	----------------	-------------

Джерела з Інтернет : 163 джерела знайдено

1. https://docplayer.net/71925952-Nacionalniy-tehnichniy-universitet-ukrayini-kiyivskiy-politehnichniy-...	0.45%
2. http://cad.kpi.ua/attachments/093_2017d_Voitenko.pdf	0.45%
3. http://uk.x-pdf.ru/6tehnichekie/86636-1-magisterska-disertaciya-zi-specialnosti-805020102-kompy...	0.45%
4. http://cad.kpi.ua/attachments/093_2015_%D0%B5%D0%B2%D1%82%D1%83%D1%85.pdf	0.45%
5. http://cad.kpi.ua/attachments/093_2015_%D0%9A%D0%BE%D1%80%D0%B5%D0%BD%D0%B...	0.45%
6. http://cad.kpi.ua/attachments/093_2016d_Azzuz.pdf	0.45%
7. http://donntu.edu.ua/2015/publ/ek2015.pdf	0.44%
8. http://fmf.kpi.ua/media/imperavi/%D0%BE%D1%84%D0%BE%D1%80%D0%BC%D0%BB%D0%...	0.43%
9. http://cad.kpi.ua/attachments/093_2017d_Chantsova.pdf	0.43%
10. https://StudFiles.net/preview/3740743/page:34	0.43%
11. http://management.fmm.kpi.ua/wp-content/uploads/2017/05/mag_metod_UI.pdf	0.43%
12. http://management.fmm.kpi.ua/wp-content/uploads/2017/05/mag_metod_UV.pdf	0.43%
13. http://management.fmm.kpi.ua/wp-content/uploads/2017/05/mag_metod_UL.pdf	0.43%
14. http://cad.kpi.ua/attachments/093_2016d_Akymov.pdf	0.43%
15. http://lectmania.ru/1x7519.html	0.43%
16. http://cad.kpi.ua/attachments/093_2017d_Belenok.pdf	0.43%
17. http://referatu.in.ua/ministerstvo-osviti-i-nauki-ukrayini-v7.html?page=11	0.43%
18. http://aesitf.kpi.ua/wp-content/uploads/2015/06/oformlennya_tekstovih_dokumentiv_v_proektah_...	0.43%
19. http://ela.kpi.ua/bitstream/123456789/23470/1/Turanska_magistr.pdf	0.43%
20. http://referatu.in.ua/polojennya-v3.html?page=56	0.43%
21. http://uchika.in.ua/atestacijnoyi-roboti.html	0.43%
22. http://cad.kpi.ua/attachments/093_2016d_Kravchuk.pdf	0.43%
23. http://referatu.in.ua/navchalenij-posibnik-dlya-studentiv-yaki-navchayutesya-za-spec-v3.html?pag...	0.43%
24. http://referatu.in.ua/magisterseka-disertaciya.html	0.43%
25. https://docplayer.net/amp/85780589-Nnk-institut-prikladnogo-sistemnogo-analizu-povna-nazva-ins..	0.43%
26. https://docplayer.net/85780589-Nnk-institut-prikladnogo-sistemnogo-analizu-povna-nazva-institutu..	0.43%
27. http://scs.kpi.ua/sites/default/files/files/%D0%A2%D0%B8%D1%82%D1%83%D0%BB%20%D0...	0.38%
28. http://ela.kpi.ua/bitstream/123456789/23831/4/Hrygorovych_magistr.pdf	0.38%
29. http://its.kpi.ua/tk/article/Turanska_magistr.pdf	0.38%
30. http://ela.kpi.ua/bitstream/123456789/23010/1/Nosik_magistr.pdf	0.32%
31. http://scs.kpi.ua/sites/default/files/files/GLUSCO.doc	0.27%
32. http://its.kpi.ua/tk/article/Kisilov_magistr.pdf	0.23%
33. https://www.kpba.edu.ua/publikatsii/all-news/55-anonsy/2094-pryklad-oformlennia-tytulnoho-arkus...	0.22%
34. http://cad.kpi.ua/attachments/093_2016d_Voronin.pdf	0.22%



Схожість



Цитата



Схожість з обраним джерелом



Посилання




Заміна літер абетки

35. http://cad.kpi.ua/attachments/093_2016d_Kalmanovich.pdf	0.21%
36. http://compi.com.ua/sposobi-optimizaciyi-strukturi-kompyuternih-merej-tehnologiyi.html	0.2%
37. http://www.shag.com.ua/nacionalenij-tehnichnij-universitet-ukrayini-kiyivsekij-polite-v6.html	0.2%
38. http://tc.kpi.ua/content/diplom/studwork/MW_Slyusar.pdf	0.2%
39. http://cad.kpi.ua/attachments/093_2016d_Avramenko.pdf	0.2%
40. http://cad.kpi.ua/attachments/093_2015_%D0%A1%D0%B0%D0%B2%D1%87%D1%83%D0%B...	0.2%
41. http://bmc.fbmi.kpi.ua/uploads/diplom/levchuk--andr%D1%96y-svyatoslavovich.pdf	0.2%
42. http://cad.kpi.ua/attachments/093_2017d_Yevtukhov.pdf	0.19%
43. http://tak.donntu.edu.ua/wp-content/uploads/2018/01/%D0%A2%D0%90%D0%9A_2017.pdf	0.17%
44. http://zfftt.kpi.ua/images/Anotations/Diploma/dipChubinska.pdf	0.16%
45. http://gamehub-cbhe.eu/ua/news/page/4	0.15%
46. http://donntu.edu.ua/council/rishennya-vchenoi-radi	0.15%
47. http://kivra.kpi.ua/wp-content/uploads/file/work/2015/Zinger/Zinger_PZ.pdf	0.15%
48. http://cad.kpi.ua/attachments/093_%D0%B4%D0%B8%D0%BF%D0%BB%D0%BE%D0%BC_%...	0.15%
49. http://www.sociology.kpi.ua/wp-content/uploads/2015/06/Bilan.pdf	0.15%
50. https://freedocs.xyz/docx-92596115	0.15%
51. http://gamehub-cbhe.eu/%D0%BA%D1%80%D1%83%D0%B3%D0%BB%D0%B8%D0%B9-%D1...	0.15%
52. http://kivra.kpi.ua/wp-content/uploads/file/work/2015/Korniichuk/Korniichuk_PZ.pdf	0.15%
53. http://gamehub-cbhe.eu/ua/news/page/7	0.15%
54. http://repository.kpi.kharkov.ua/bitstream/KhPI-Press/37400/1/prohramy_2018_Mulyahentni_sys...	0.15%
55. http://science.donntu.edu.ua/wp-content/uploads/2018/06/glinska.pdf	0.15%
56. http://erasmusplus.org.ua/en/news/703-10-projects-with-ukrainian-partners-selected-for-funding-wi..	0.15%
57. http://donntu.edu.ua/ms/gamehub-university-enterprises-cooperation-game-industry-ukraine.html	0.15%
58. http://donntu.edu.ua/en/ic/gamehub-university-enterprises-cooperation-game-industry-ukraine-2.ht..	0.15%
59. http://web.kpi.kharkov.ua/si/grant-yevrosoyuzu-gamehub-university-enterprises-cooperation-in-ga...	0.15%
60. http://gamehub-cbhe.eu/gamehub-in-ukraine/gamehub-kntu	0.15%
61. http://gamehub-cbhe.eu/gamehub-in-ukraine/gamehub-onpu	0.15%
62. http://science.donntu.edu.ua/wp-content/uploads/2015/11/aref_vovna.pdf	0.14%
63. http://matan.kpi.ua/public/files/2017/dis/Shaforostov.pdf	0.14%
64. http://bmc.fbmi.kpi.ua/uploads/diplom/shkurat-oksana-serg%D1%96ivna.pdf	0.14%
65. http://science.donntu.edu.ua/wp-content/uploads/2015/11/aref_laktionov1.pdf	0.14%
66. http://tc.kpi.ua/content/diplom/studwork/MW_AkhalA.pdf	0.14%
67. http://cad.kpi.ua/attachments/093_2016d_Svietlova.pdf	0.14%
68. http://ahv.kpi.ua/graduation_work/master/2018/Aliekseienko_magistr.pdf	0.14%
69. http://ela.kpi.ua/bitstream/123456789/23154/1/Aliekseienko_magistr.pdf	0.14%
70. https://docplayer.net/84383526-Magisterska-disertaciya.html	0.13%
71. http://ela.kpi.ua/bitstream/123456789/23216/1/Morozko_magistr.pdf	0.13%
72. http://referatu.in.ua/kafedra-informacijno--vimiryuvalenoyi-tehniki.html	0.13%
73. http://imt.kpi.ua/wp-content/uploads/2018/09/MD-Prosikin-2018.docx	0.13%
74. http://bmc.fbmi.kpi.ua/uploads/diplom/pugach-pavlo-romanovich.pdf	0.13%
75. http://bmc.fbmi.kpi.ua/uploads/diplom/boyko-vladislav-volodimirovich.pdf	0.13%
76. http://bmc.fbmi.kpi.ua/uploads/diplom/teplyakov-kirilo-%D0%86gorovich.pdf	0.13%
77. http://www.er.nau.edu.ua/bitstream/NAU/18537/3/Diss_Zhmurko.pdf	0.12%
78. http://www.er.nau.edu.ua/bitstream/NAU/14308/1/%D0%94%D0%B8%D1%81%D1%81%D0%B5...	0.12%
79. http://MirZnani.com/a/253830/analz-trudovikh-pokaznikov	0.11%
80. http://mses.kpi.ua/diplom/bar.pdf	0.11%

 Схожість

 Схожість з обраним джерелом

 Заміна літер абетки

 Цитата

 Посилання

81. http://phone.kpi.ua/wp-content/uploads/2015/01/%D0%94%D0%B8%D0%BF%D0%BB%D0%BE...	0.11%
82. http://bmc.fbmi.kpi.ua/uploads/diplom/matushevich-natal%D1%96ya-anatol%D1%96ivna.pdf	0.11%
83. http://cad.kpi.ua/attachments/475_Petrishenko_magistr.pdf	0.11%
84. http://refleader.ru/jgernaotrmerpol.html	0.09%
85. https://github.com/junior-devleague/unity-3D-racing/wiki/Car-Controller-Script	0.09%
86. https://stackoverflow.com/questions/46207445/unity-multiplayer-car-networking-particles-and-sou...	0.09%
87. https://pastebin.com/pGusjeZr	0.09%
88. http://gamehub-cbhe.eu/wp-content/uploads/2016/04/AboutProject.pdf	0.09%
89. https://StudFiles.net/preview/5692877	0.09%
90. http://compi.com.ua/navchaleno-metodichnij-posibnik-do-vikonannya-laboratornih-rob.html?page=3	0.09%
91. http://ua.coolreferat.com/%D0%90%D0%BD%D0%B0%D0%BB%D1%96%D0%B7_%D1%82%D...	0.09%
92. https://www.KazEdu.kz/referat/176267	0.09%
93. http://ref.rushkolnik.ru/v105865	0.09%
94. http://en.coolreferat.com/%D0%90%D0%BD%D0%B0%D0%BB%D1%96%D0%B7_%D1%82%D...	0.09%
95. https://xreferat.com/113/10900-1-anal-z-trudovih-pokaznik-v.html	0.09%
96. http://diplomba.ru/work/67797	0.09%
97. https://www.KazEdu.kz/referat/184601	0.09%
98. http://refeteka.ru/r-185323.html	0.09%
99. https://docplayer.ru/57000687-Tak-telekomunikaciyi-avtomatika-komp-yuterno-integrovani-tehnolo...	0.09%
100. http://ahv.kpi.ua/wp-content/uploads/2018/03/Cherevko.pdf	0.09%
101. http://ela.kpi.ua/bitstream/123456789/23209/1/Sergiienko_magistr.pdf	0.09%
102. http://ela.kpi.ua/bitstream/123456789/23193/1/Petrychuk_magistr.pdf	0.09%
103. http://vuzlib.com.ua/articles/book/14813-Tekhnolog%D1%96ja_sporudzhennja_g%EF%BF%BD...	0.09%
104. http://topreferat.znate.ru/docs/index-1469.html?page=54	0.09%
105. https://docplayer.ru/58460073-Tak-telekomunikaciyi-avtomatika-komp-yuterno-integrovani-tehno...	0.09%
106. http://kn.lib-i.ru/27raznoe/24504-1-ministerstvo-osviti-nauki-ukraini-derzhavny-vischij-navchalni...	0.09%
107. https://globuss24.ru/doc/analiz-trudovih-pokaznikiv	0.09%
108. http://coolreferat.com/%D0%90%D0%BD%D0%B0%D0%BB%D1%96%D0%B7_%D1%82%D1...	0.09%
109. http://turboreferat.ru/organization-economy/kompleksnij-ekonomichnij-analz-rezultatv-gospodarsk...	0.09%
110. https://uk.wikipedia.org/wiki/%D0%9E%D1%86%D1%96%D0%BD%D0%BA%D0%B0_%D0%B...	0.09%
111. http://svgorbatko.ucoz.ru/MONITORING/Lab_1.pdf	0.09%
112. http://www.sociology.kpi.ua/wp-content/uploads/2016/01/%D0%91%D0%BE%D1%80%D1%82%...	0.08%
113. http://vuzlib.com.ua/articles/book/40931-Aerolog%D1%96ja_g%D1%96rnichikh_p%D1%96dpr/1...	0.08%
114. http://znaimo.com.ua/%D0%9D%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%...	0.08%
115. http://cad.kpi.ua/attachments/093_2017dm_myronenko.pdf	0.08%
116. http://cad.kpi.ua/attachments/093_%D0%94%D0%B8%D0%BF%D0%BB%D0%BE%D0%BC%...	0.08%
117. https://otherreferats.allbest.ru/ecology/00102270_0.html	0.08%
118. http://www.sociology.kpi.ua/wp-content/uploads/2016/01/%D0%9B%D1%8F%D1%89%D0%B5%...	0.08%
119. http://uk.x-pdf.ru/5mehanika/1580515-3-61051-harkiv-vul-klochkivska-333-tel-0572-3368979-faks..	0.08%
120. http://uk.x-pdf.ru/5mehanika/71700-7-tezi-dopovidey-zagalnouniversitetskoi-naukovo-tehnichnoi-...	0.08%
121. http://bukvar.su/menedzhment/116634-Kompleksnaya-ocenka-effektivnosti-innovacionnogo-proe...	0.08%
122. http://uk.x-pdf.ru/5tehnicheskoe/2256010-2-sistemi-dlya-viznachennya-geometrichnih-rozmiriv-to...	0.08%
123. http://cad.kpi.ua/attachments/093_Babenko_diplom.pdf	0.08%
124. https://docplayer.net/72881621-Kriteriyi-ocinki-rivnya-rozvitku-profesiyno-vazhlyviv-psihofiziologi...	0.07%
125. http://en.coolreferat.com/%D0%A4%D1%96%D0%B7%D0%B8%D0%BA%D0%BE-%D1%82%...	0.07%
126. https://revolution.allbest.ru/law/00474232_0.html	0.07%

127. https://docplayer.net/50679263-Ocinyuvannya-efektivnosti-sistemi-upravlinnya-socialno-ekonom...	0.07%
128. http://it-ua.info/news/2016/10/12/neyronn-merezh-dlya-pochatkvcv-chastina-1.html	0.07%
129. http://tc.kpi.ua/content/diplom/studwork/MW_Gorbatiuk.pdf	0.07%
130. http://aop.nmu.org.ua/ua/metodicki/specialist/opg/lp/ukr_perenos_metanom%203.pdf	0.07%
131. http://opu.ua/eng/inter_coop/internationalprograms/projects/project3/gamehub	0.07%
132. https://www.KazEdu.kz/referat/160989	0.07%
133. http://zavantag.com/docs/index-16235849-3.html?page=3	0.07%
134. http://www.shag.com.ua/ministerstvo-osviti-i-nauki-ukrayini-derjavnij-vishij-navchale-v17.html	0.07%
135. http://uk.x-pdf.ru/5ekonomika/2374105-3-aktovi-knigi-dzherelo-rekonstrukcii-zemlevolodin-kiivsk...	0.07%
136. https://docplayer.net/70982294-Tendenciyi-ta-perspektivi-rozvitku-farmaceutichnogo-rinku.html	0.07%
137. http://mip.opu.ua/eng/project/project3/gamehub	0.07%
138. https://docplayer.net/52200972-Logistichniy-menedzhment-za-vimogami-kreditno-modulnoyi-sist...	0.07%
139. http://zavantag.com/docs/2010/index-22427.html?page=9	0.07%
140. http://uk.x-pdf.ru/5jazykoznanie/2292405-63-leksiko-grammaticheskie-innovacii-sovremennih-sla...	0.07%
141. https://mydisser.com.ua/catalog/view/552/628/15583.html	0.07%
142. https://docplayer.net/77915107-Marketingovi-strategiyi-pidpriemstva-restorannogo-biznesu-yak-...	0.07%
143. https://docplayer.net/54207829-Informaciyna-bezpeka-yak-element-pidvishchennya-efektivnosti-...	0.07%
144. https://docplayer.net/33440985-Administrativniy-menedzhment.html	0.07%
145. https://docplayer.net/44858731-Vprovadzhennya-metodiki-vikoristannya-shem-modeley-u-leksic...	0.07%
146. https://docplayer.net/51775573-Ocinyuvannya-realizaciyi-innovaciynogo-potencialu-pidpriemstv...	0.07%
147. https://docplayer.net/76250534-Formuvannya-strategiyi-innovaciynogo-rozvitku-mashinobudivno...	0.07%
148. https://docplayer.net/78072379-O-yu-baydak-tov-makhaus-a-p-gavrich-r-a-hohlova-v-v-kalinichen...	0.07%
149. http://stud24.ru/dcb/valjutnij-kurs-ta-kursova-politika/518079-2362372-page1.html	0.07%
150. http://cad.kpi.ua/attachments/093_2016d_Zagorodniuk.pdf	0.06%
151. http://ua.z-pdf.ru/7tehnicheskije/942005-12-prisvyachu-tsya-95-richchyu-zasnuvannya-doneckog...	0.06%
152. http://nashaucheba.ru/v10085/%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B...	0.06%
153. http://ea.donntu.org:8080/bitstream/123456789/4447/1/%D0%9A%D1%83%D1%80%D1%81%D...	0.06%
154. http://phone.kpi.ua/wp-content/uploads/2015/01/%D0%94%D0%B8%D0%BF%D0%BB%D0%B...	0.06%
155. http://mediku.com.ua/diplomna-robota-na-zdobuttya-osvitneo-kvalifikacijnogo-rivnya-v5.html	0.06%
156. http://www.er.nau.edu.ua/bitstream/NAU/22464/2/diser_ua_2.0.pdf	0.06%
157. http://uk.x-pdf.ru/5mehanika/1243996-1-nacionalniy-universitet-bioresursiv-prirodokoristuvannya-..	0.06%
158. https://donnaba.edu.ua/docs/navchannya/pidsumkova_atestecia.pdf	0.06%
159. http://www.mediku.com.ua/diplomna-robota-na-zdobuttya-osvitneo-kvalifikacijnogo-rivnya-v5.html	0.06%
160. http://uk.x-pdf.ru/5tehnicheskije/2256219-8-pobudovi-kompyuterizovanih-optoelektronnih-vimiryuv..	0.06%
161. http://uchika.in.ua/diplomna-robota-na-zdobuttya-osvitneo-kvalifikacijnogo-rivnya-v7.html	0.06%
162. http://uk.x-pdf.ru/5himiya/252539-1-avtomatizaciya-tehnologichnih-procesiv-galuzi-lokalni-sistem...	0.06%
163. https://otherreferats.allbest.ru/radio/00329800_0.html	0.06%

Джерела з Бібліотека : 135 джерел знайдено

2018_M_КН17м_Федорко.pdf	0.77%
2018_M_ИПЗм16_Лефтеров.pdf	0.76%
2018_M_ИПЗм16_Дзюбинский.pdf	0.73%
2018_M_ИПЗ16м_Ермолов_2.pdf	0.73%
Мат роб Селіванов.pdf	0.7%
2018_M_КНм-17_Нгуен.pdf	0.7%

Kusch_plag.pdf	0.67%
Styan_plag.pdf	0.67%
Шинкаренко плагіат.pdf	0.67%
2018_М_ИПЗм16_Роговец_Е.pdf	0.66%
2018_М_ИПЗм16_Роговец_Е-2.pdf	0.66%
2018_М_ИПЗм16_Драченко.pdf	0.66%
2018_М_МПЗ16м_Ермолов.pdf	0.66%
2018_М_ИПЗ16м_Бищенко.docx	0.66%
2018_М_ИПЗ16м_Псел.pdf	0.64%
2018_М_КНм16_Клокова_МВ.pdf	0.63%
2018_М_КНм16_Переяславський_ВС.pdf	0.61%
2018_Б_ПІ_14_Ситников_А.Е.pdf	0.6%
2018_М_КН17м_Нгуен.pdf	0.6%
2018_Б_ПІ14-Беломеря.pdf	0.58%
2018_Б_ПІ13з_Литвиненко-2кор.pdf	0.58%
2018_Б_ПІ14_Драмачев.pdf	0.58%
2018_М_ИПЗм16_Прядка_ДА.pdf	0.57%
2018_Б_ПІ14_Андросян.pdf	0.57%
2018_Б_ПІ14_Коврыгин.pdf	0.57%
2018_Б_ПІ13з_Литвиненко.pdf	0.54%
2018_М_ИПЗ17м_Бобылева.pdf	0.54%
2018_М_МСМзм16_Кузін_ОА.pdf	0.52%
Минзоров_КІ14.docx	0.52%
2018_Б_ПІ14_Ольшевский.docx	0.51%
2018_М_ИПЗм17_Glinska_U.pdf	0.41%
Ященко_КІ14а.docx	0.35%
2018_Б_КН14_Юпатов_ЄА.pdf	0.35%
2018_М_МСМзм16_Харін_ОІ.pdf	0.35%
2018_Б_КН14_Бородкін_ДВ.pdf	0.35%
Косинський плагіат.pdf	0.35%
Плехова_КІ14.docx	0.35%
2018_М_ЕЛТ16_Михайлюта_ІО.pdf	0.33%
2018_М_КН17м_Глушко.pdf	0.3%
2018_М_ЕЛТ16_Карабаджак_ЄЄ.docx	0.23%
Diplom_Misko_v2_nolit.pdf	0.22%
2018_Б_ПІ13з_Компаниец.pdf	0.22%
2018_М_ЕЛТ16_Сердюк_ОО.pdf	0.22%
КривенкоО. М.txt	0.2%
Диплом_Пастерский_на_проверку.pdf	0.19%
Васильченко_КІМ-16_пров.docx	0.19%
2018_М_МСМм16_Демченко_ІА.pdf.pdf	0.19%
2018_М_МСМм16_Раєнко_ЄГ.pdf	0.19%
2018_Б_ХТзск15_Бутовський_МС.pdf	0.19%
2018_М_ХТм16_Пономарьов_ММ.pdf	0.19%
2018_Б_ІУСзск15_Лісний_ВС.pdf	0.18%
2018_М_ЕСЕМ16_Нечай_ГП.pdf	0.17%



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

м_ТКР-17_ПрокушевА.М.-_пдф.pdf	0.17%
2018_м_ЕЛТ17_Мітіна_ТІ.pdf	0.17%
2018_Б_ЕМ13з_Земляніцина_ОА.pdf	0.13%
18-лях-проверка.pdf	0.13%
2018_Б_ЕМ14_Першин_ОО.pdf	0.13%
2018_Б_ЕМ13з_Канівець_ВВ.pdf	0.13%
2018_Б_ЕМ13з_Матолинець_АВ.pdf	0.13%
2018_Б_ЕМ14_Тимошин_МД.pdf	0.13%
2018_Б_ЕМ13з_Зайдун_ОО.pdf	0.13%
2018_Б_ЕМ13з_Баранов_СО.pdf	0.13%
Дипломная работа Ищенко (1).pdf	0.13%
18-кучер-проверка.pdf	0.13%
ПЗ_Кузьменко_КИ14б.docx	0.13%
2018_Б_ЕСЕ14_Васін_ВЕ.pdf	0.13%
2018_Б_ЕСЕ14_Севостьянов_КЮ.pdf	0.13%
18-Гладков-проверка.pdf	0.13%
2018_Б_ЕСЕ14_Твердохлеб_РВ.pdf	0.13%
2018_Б_ЕСЕ14_Доценко_ОС.pdf	0.13%
2018_Б_ЕСЕ14_Бріжніченко_ВО.pdf	0.13%
Любимов_ПЗ.docx	0.13%
2018_М_АКТм16_Аношка_ВО.pdf	0.13%
2018_М_ГСм16А_Богданова_ОМ.pdf	0.13%
СУС_плагіат.pdf	0.13%
2018_Б_ЕСЕ14_Громов_КЮ.pdf	0.13%
2018_М_РККм-16_Береговий_ПК.docx	0.11%
2018_М_Мм16_Карабахцян_РТ.pdf	0.11%
ауп14_Рак.docx	0.11%
2018_М_ЕЛТ17_Рубан_АО.pdf	0.1%
2018_М_ЕЛТ17_Запорожченко_ВМ.pdf	0.1%
2018_М_ЕЛТ17_Дульченко_СЮ.pdf	0.1%
2018_М_ГМБм16_Соболь_ЄА.pdf	0.1%
2018_М_ЕЛТ17_Понікаров_БК.pdf	0.1%
2018_М_ЕЛТ17_Васильченко_ОВ.pdf	0.1%
18-09-21-диссертация Адарюкова Л.Б.docx	0.1%
2018_М_М16_Масик_ОО.pdf	0.1%
2018_М_ГМБм16_Шульженко_ВВ.pdf	0.1%
2018_М_ЕСЕм16_Алексєєва_МВ.pdf	0.1%
2018_М_ЕЛТ16_Саланжій_ВМ.pdf	0.1%
2018_М_ЕСАм16_Хоменко_ЛБ.pdf	0.1%
2018_М_ЕЛТ16_Бричка_ЄО.pdf	0.1%
2018_М_ЕЛТ16_Ігнатєєва_ВЮ.pdf	0.1%
2018_М_ЕЛТ16_Оленич_ВВ.pdf	0.1%
2018_М_ЕСАм16_Юр'єв_СА.pdf	0.1%
2018_М_ЕСАм16_Соловійов_ІД.pdf	0.1%
2018_М_ТКРм16_НесвітаєвАА.doc	0.1%
dis5 (1).pdf	0.1%



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

full text ua.pdf	0.1%
Метода для курсового по ТСГВ 2016 без обложки.pdf	0.09%
2018_М_КНм16_Малиновський_ДС.pdf	0.09%
2018_М_МСМм16_Безелюк_ОА.pdf	0.09%
2018_М_МСМзм16_Баша_ОО.pdf	0.08%
МР_МІКм16_Гайдук_antiplag.docx	0.08%
2018_М_КНм16_Смірнов_АВ.pdf	0.08%
2018_М_МСМм16-Таран_ЯІ.pdf	0.08%
2018_М_МСМм16_Короткий_ОВ.pdf	0.08%
2018_М_КНм16_Толстих_ЯВ.pdf	0.08%
2018_М_КНм16_Яковлев_МВ.pdf	0.08%
2018_М_МСМзм16_Переверзева_ОС.pdf	0.08%
2018_Б_ЕК-14_Маргієв_А.Е..pdf	0.07%
ДР_бакалавр_Столяр_ЕК-14.pdf	0.07%
2018_Б_ХТ14_Баласанян_ЛО.pdf	0.07%
2018_М_ЕКм-16_Мась_Д.В..pdf	0.07%
2018_Б_ЕК14_Чернов_ЛІ.pdf	0.07%
Очкурова.docx	0.07%
антиплагиат_Авраменко_Н_И.pdf	0.06%
Дисертація Сахно С.В.pdf	0.06%
2018_М_Мм16_Бутенко_ДВ.docx	0.06%
2018_М_КНм16_Бесхмельницин_ЄД.pdf	0.06%
2018_М_ЕКОм16_Черкашина_АС.pdf	0.06%
2018_М_ЕЛКм16_Чухно_РО.pdf	0.06%
2018_М_КНм16_Кучеренко_ОА.pdf	0.06%
2018_М_РЕЛм16_Костов_ОВ.doc	0.06%
03_Кондратенко.docx	0.06%
2018_Б_КН14_Хоменко_МО.pdf	0.06%
Карагодов_ПЗ_1.docx	0.06%
2018_М_Щербаков.pdf	0.06%
2018_М_ХТм16_Циганкова_КС.pdf	0.06%
2018_М_КНм16_Чекулаєва_КЄ.pdf	0.06%
НАУКОВІ ПРАЦІ Серія Економічна.doc	0.06%
04_Костюченко.docx	0.06%
1.3 Сахно, Сахно.docx	0.06%
1.5 Иорданов и др..docx	0.06%
2018_м_ЕЛТ17_Хабаров_КС.pdf	0.06%



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»

Факультет комп'ютерних наук і технологій
Кафедра прикладної математики і інформатики

«До захисту допущений»
Завідувач каф. ПМІ

О.А. Дмитрієва
(підпис, дата) (ініціали, прізвище)
« » 2018 р.

Випускна кваліфікаційна робота
магістра
(освітній ступінь)

на тему
«Побудова ігрового процесу на базі
методів машинного навчання і інтелекту»

Виконав: студент 2 курсу, групи ІПЗм–17 напрямку підготовки
(спеціальності) спеціальності 121 «Інженерія програмного забезпечення»
Кривенко О.М.
(прізвище та ініціали) (підпис)

Керівник зав. каф. ПМІ, д. т. н., проф. Дмитрієва О. А.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Нормоконтролер доц. каф. ПМІ, к. т. н. Назарова І. А.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент
(підпис)

Покровськ – 2018

ВСТУП

Протягом останніх двох десятиліть машинне навчання стало одним з основних елементів інформаційних технологій [1], а з цим – досить важливою, хоча й, зазвичай, прихованою частиною нашого життя [2]. На сьогоднішній день поступове поліпшення підходів машинного навчання і інтелекту дозволило досягнути нового якісного рівня у підходах штучного інтелекту [2].

Враховуючи постійно зростаючу кількість даних, стають доступними нові можливості з їх аналізу [3]. Відповідно до цього існують підстави вважати, що розумний аналіз даних стане ще більш поширеним, як необхідний компонент технологічного прогресу [4]. Показовим результатом такої еволюції слугує те, що штучний інтелект почав перевершувати можливості людей в деяких вузько спеціалізованих задачах.

Індустрія відеоігор також не стояла на місці, ставши найбільшим сегментом індустрії розваг. Індустрію відеоігор неможливо розглядати окремо від розробки відеоігор. Коли на ринок виходить новий ігровий продукт, він відразу ж опиняється в опозиції по відношенню до всіх наявних продуктів індустрії, які актуальні на даний момент [5]. Завдяки цьому дана індустрія це висококонкурентне середовище з глобальною конкуренцією, де велика кількість компаній і незалежних команд займаються створенням відеоігор.

Для того, щоб ігровий продукт був конкурентоспроможним і успішним на ринку відеоігор, випускаючи новий продукт необхідно продемонструвати не тільки гарну графічну складову, а й запропонувати щось нове та інноваційне, що зробить продукт кращим за інші [6]. Можна сказати, що ігрова індустрія знаходиться у пошуку нових підходів для покращення ігрового процесу та ігрового досвіду. Одним з таких підходів може бути використання методів штучного інтелекту для поліпшення ігрового досвіду.



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

В розрізі поліпшення ігрового досвіду гравців методи машинного навчання та інтелекту теоретично можуть вирішувати великий обсяг завдань забезпечення ігрового процесу. Теоретично сумісне використання методів машинного інтелекту і навчання дозволить отримати ігровий процес, який зможе підлаштовуватися під гравця і адаптуватися під різні ігрові ситуації [7]. Це, в свою чергу, зробить процес гри більш цікавим, а також буде спонукати гравця знаходити нові стратегії і методи гри.

Актуальність роботи пояснюється недостатнім ступенем дослідження і дефіцитом емпіричного матеріалу, що перешкоджає подальшим розробкам, пов'язаним з питаннями використання методів штучного інтелекту для реалізації ігрових механік, а саме, використання штучних нейронних мереж і генетичних алгоритмів для побудови штучного інтелекту ігрових агентів.

Зв'язок дослідження з науковими програмами, планами, темами. Спрямованість кваліфікаційної роботи обумовлена проведенням досліджень в рамках міжнародного проекту Erasmus+ з навчання персоналу для ігрового сектора на IT-ринку України 561728-EPP-1-2015-1-ES-EPPKA2-CBHE-JP «GameHub: University-Enterprises Cooperation in Game Industry in Ukraine».

Мета і задачі дослідження. Мета кваліфікаційної роботи полягає у розробці і обґрунтуванні формальної моделі ігрового агента, заснованої на методах машинного навчання та інтелекту, програмної реалізації ігрового процесу з застосуванням нейронних мереж та еволюційних методів оптимізації при множинних генераціях популяцій ігрових агентів.

Досягнення поставленої мети вимагає розв'язання наступних основних задач дослідження:

- проведення порівняльного аналізу сучасних методів та алгоритмів машинного навчання і інтелекту, які застосовуються при побудові ігрового процесу, визначення підходів до організації навчання нейронних мереж з базовими топологіями;
- обґрунтування та розробка формальної моделі ігрового агента, заснованої на методах машинного навчання та інтелекту;

- визначення підходів до навчання розробленого агента з отриманням даних про навколишнє середовище;
- розробка програмної системи для симуляції гоночного процесу з реалізацією логіки керування ігровими агентами при множинній генерації популяцій;
- тестування розробленої програмної системи з візуалізацією процесу навчання та оцінювання достовірності отриманих результатів.

Об'єктом дослідження є процеси створення і навчання ігрових агентів на основі еволюційних підходів штучного інтелекту. **Предметом дослідження** є формальні і програмні моделі ігрових агентів.

Методи дослідження. При виконання кваліфікаційної роботи було застосовано методи машинного навчання, штучного інтелекту, математичного моделювання, теорії обчислювальних процесів, теорії ігор. А при створенні системи моделювання було використано засоби конструювання, моделювання та тестування програмного забезпечення, комп'ютерної графіки та генетичних алгоритмів.

Наукова новизна одержаних результатів полягає у наступному:

- запропоновано удосконалити формальну модель агента ігрового автомобіля, засновану на методах машинного навчання і інтелекту;
- вперше запропоновано використання модифікованої функції безпечної мутації для навчання агента гоночного автомобіля.

Практичне значення. Отримані в дипломній роботі результати можуть бути використані при розробці відеоігор, побудованих на базі штучного інтелекту ігрових агентів, що гуртуються на методах машинного навчання (штучних нейронних мереж і генетичного алгоритму), а також в рамках інших наукових досліджень.

Практична цінність роботи полягає в наступному:

- проведена порівняльна оцінка доцільності використання класичного підходу та підходу на базі машинного навчання для побудови інтелекту ігрових агентів;

- спроектована та створена програмна платформа для тестування алгоритмів, які реалізують різні підходи до побудови ігрового штучного інтелекту на базі обраних засобів реалізації;
- отримані оцінки складності розробки і налагодження агентів, а також оцінки достовірності результатів, отриманих в рамках дослідження;
- результати дипломної роботи можуть бути використані розробниками відеоігор, а також у навчальному процесі кафедр, які здійснюють підготовку фахівців у даному напрямку.

Особистий внесок. Основні положення і результати дипломної роботи, що виносяться до захисту, отримані автором самостійно. У роботах, написаних у співавторстві, автору належать усі розроблені програмні моделі та програмні реалізації. З робіт, опублікованих у співавторстві, для вирішення проблеми та задач, поставлених у дослідженні, використовуються результати, отримані автором особисто.

Апробація результатів. Основні положення даної роботи доповідались і обговорювались на науково-технічних конференціях, семінарах, серед яких:

1. III Всеукраїнську науково-технічну конференцію молодих учених, аспірантів та студентів «Автоматизація, контроль та управління: пошук ідей та рішень» ДВНЗ «ДонНТУ» 22-23 травня 2018 р. (м. Покровськ).
2. Університетська наука – 2018: XI Міжнародна науково-технічна конференція, ДВНЗ «Приазовський державний технічний університет» 23-24 травня 2018 р. (Маріуполь).
3. XII Регіональна студентська науково-технічна конференція «Наука – перші кроки» кафедри інформатики ДВНЗ «ПДТУ» 20-24 травні 2018 р. (м. Маріуполь).
4. III Всеукраїнська науково-практична конференція молодих вчених «ТАК»: телекомунікації, автоматика, комп'ютерно-інтегровані

технології ДВНЗ «ДонНТУ» 29-30 листопада 2017 р. (м. Покровськ).

5. XXXVI Всеукраїнська науково-практична Інтернет-конференція «Перспективи розвитку наукових досліджень в XXI столітті» у 2017 р. (м. Дніпропетровськ).
6. XI Регіональна студентська науково-технічна конференція «Наука – перші кроки» кафедри інформатики ДВНЗ «ПДТУ» 20-24 травні 2017 р. (м. Маріуполь).

Публікації. Матеріали дослідження опубліковано в вигляді тез доповідей у конференціях [8]–[13].

Структура роботи та її обсяг. Робота складається із анотації, вступу, змісту, шести розділів, загальних висновків, списку посилань і додатків. Підсумковий обсяг основної частини роботи складає - сторінок. Список посилань на даному етапі виконання роботи містить - пунктів.

РОЗДІЛ 1

ОГЛЯД СУЧАСНИХ МЕТОДІВ МАШИННОГО НАВЧАННЯ І ІНТЕЛЕКТУ У ПОБУДОВІ ІГРОВОГО ПРОЦЕСУ

1.1 Огляд сучасних технологій машинного навчання і інтелекту

В широкому сенсі до категорії машинного навчання інтелекту підпадає достатньо велика кількість концепцій, підходів, методів, алгоритмів, що застосовуються для автоматичного отримання будь-яких нетривіальних висновків на основі будь-яких емпіричних даних [14]. Дана категорія тісно пов'язана з багатьма іншими категоріями і має спільні методи, наприклад з категорією добутку даних, оптимізацією, статистикою, тощо.

Класично у дану категорію відносять технології, характерною рисою яких є непряме рішення задачі [3], тобто система програмується не на вирішення конкретного завдання, а на навчання вирішенню безлічі подібних завдань [14]. Відповідно до цього можна зробити загальну постановку задачі машинного навчання і інтелекту, де існує безліч об'єктів (ситуацій) і безліч можливих відповідей (відгуків, реакцій), а також існує деяка невідома неявна залежність між відповідями і об'єктами і її потрібно знайти [4].

Під знаходженням даної неявної залежності можна розуміти побудову алгоритму, який здатний для будь-якого можливого вхідного об'єкта видати досить точну відповідь. Ця залежність не обов'язково виражається аналітично, і тут нейроні мережі реалізують принцип емпірично формованого рішення [14]. Важливою особливістю при цьому є здатність навченої системи до узагальнення, тобто до адекватного відгуку на дані, що виходять за межі наявної навчальної вибірки [3]. Для вимірювання точності відповідей вводиться оціночний функціонал якості.

Одним з основних критеріїв класифікації задач машинного навчання і інтелекту є спосіб машинного навчання. На даний момент не існує сталої



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

класифікації методів машинного навчання, але можна виділити за характерними рисами: навчання з вчителем, навчання без вчителя та змішані підходи навчання. Також часто вдаються до класифікації за допомогою широких категорій завдань машинного навчання, які описані далі [14].

Навчання з вчителем (англ. supervised learning) представляє собою процес навчання, коли навчальна система примусово навчається за допомогою навчальної вибірки, яка представляється у вигляді прикладів до яких входить ситуація і відповідна відповідь.

Змішане навчання (англ. semi-supervised learning) – процес навчання аналогічний навчанню з вчителем, але з неповним набором навчальної вибірки, тобто в вибірці відсутні деякі набори ситуацій і відповідей. Для невизначеного набору відповідь припускається автоматично.

Активне навчання (англ. active learning) – аналогічно змішаному навчанню, але існує можливість до визначення бажаного результату в необхідних ситуаціях. При такому навчанні існує ризик того, що алгоритм перевчиться на інформативними прикладами.

Навчання без вчителя (англ. unsupervised learning) – представляє собою протиставлення процесу навчання з вчителем, в якому кожній ситуації випадково задається правильна відповідь. Після чого відбувається пошук залежності. Процес підходить для задач з відомим описом множин навчальної вибірки, де необхідно виявити існуючі взаємозв'язки.

Навчання з підкріпленням (англ. reinforcement learning) – на відміну від навчання з вчителем, цей процес ґрунтується на взаємодії з середовищем. При такому навчанні сигнали підкріплення (нагороди чи покарання), які формують функцію фітнесу (пристосованості), є відгуком середовища на прийняті рішення, а не системи управління підкріпленням, як це відбувається в навчанні з учителем. В якомусь сенсі таке навчання можна назвати окремим випадком навчання з учителем, але в ролі учителя виступає певне середовище з встановленими правилами або його модель.

З іншої сторони, системи машинного навчання і інтелекту можна

охарактеризувати з точки зору вирішуваних ними задач і бажаного виходу навченої системи [15]. З такої точки зору можна виділити декілька наступних категорій, описаних далі.

Задача класифікації полягає в наданні класифікаційного розподілу у відповідність до вхідних сигналів. В даному випадку модель виконує пошук неявної залежності між вхідними даними і результуючим розподілом.

Задача регресії полягає в пошуку бажаного результату в невідомій області, тобто знаходження залежності між вхідними сигналами і відповідями, де відомі деякі пари вхідних і вихідних сигналів, а нам потрібно зробити припущення про значення при вхідних сигналах з невідомою відповіддю.

Задача кластеризації полягає у розбитті на підмножини чи кластери у відповідності до вхідних сигналів. На виході знаходиться відповідь у вигляді класифікації за кластерами, які включають в себе схожі об'єкти, а об'єкти інших кластерів істотно відрізняються. На відміну від класифікації, групи не відомі заздалегідь, в результаті чого вирішення зазвичай є безконтрольним.

Задача оцінки щільності представляє оцінку неспостережуваної підлеглої функції щільності ймовірності на основі даних спостереження. Ця неспостережувана функція щільності розглядається як густина, відповідно до якої розподілено велику сукупність, а дані зазвичай розглядаються як випадкова вибірка з тієї сукупності.

Задача зниження розмірності полягає у процесі скорочення кількості випадкових змінних шляхом отримання множини головних змінних. Цей процес можна поділити на обирання ознак та виділення ознак.

Окрім цих класичних задач машинного навчання існує велика кількість інших задач, які пов'язані з машинним навчанням [15]. Наприклад, серед таких задач, ще можна виділити задачі, що пов'язані з навчанням навчання, задачі керування, задачі побудови рангової залежності, тощо.

Є логічним, що вирішення певних задач машинного навчання і інтелекту вимагає використання певних концепцій, підходів, методів чи



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

алгоритмів. Але обраний підхід може бути не єдиним можливим.

1.2 Задачі машинного навчання і інтелекту у відеоіграх

Як вже було зазначено, методи машинного навчання і інтелекту мають достатньо широкий спектр застосування і відеоігри не виключення.

Відповідні методи можуть застосовуватися, як для побудови самого ігрового процесу, так і для вирішення завдань, пов'язаних з ігровим процесом [16]. З точки зору ігрового процесу відповідні алгоритми можуть бути застосовані у реалізації штучного інтелекту ігрових опонентів і механік, а з точки зору завдань, пов'язаних з ігровим процесом – анімації руху, постобробки, тощо.

Першочергово ігрова система повинна забезпечувати широкий спектр потенційних цілей і функцій, пов'язаних як з самою грою, так і з її жанром. Основним фактором, який впливає на якість ігрового досвіду прийнято вважати ігровий штучний інтелект, що відповідає за забезпечення гравця правдоподібним ігровим процесом. Такий штучний інтелект здебільшого полягає в моделюванні та імітації поведінки гравців чи об'єктів [17], [18].

Тобто, концепція ігрового штучного інтелекту полягає в імітації сильного штучного інтелекту ігрових опонентів на базі використання слабого штучного інтелекту та ряду спеціалізованих методів. Загальноприйнятою є класифікація персонажів за рівнем інтелекту – персонажі без інтелекту (англ. NPC), зі слабким інтелектом (англ. mob) і сильним інтелектом (англ. bot).

Створити ігрового суперника, який буде завжди перемагати гравця досить легко, оскільки людині необхідно покладатися на зір і слух, в той час як штучний інтелект апелює абстракціями гри [17], [18]. Також не варто забувати, що кінцевою метою штучного інтелекту ігрового персонажу є створення правдоподібного ігрового опонента, якого зможе подолати гравець.

Створення реалістичного інтелекту не є тривіальною задачею, оскільки створення такого інтелекту потребує врахування багатьох факторів [19]. У погоні за поліпшення якості і реалістичності в комп'ютерних іграх

прибігають до використання методів машинного навчання і інтелекту.

Використовуються як спеціалізовані, так і загальні методи. Яскравими прикладами таких методів і підходів можна назвати концепцію скінченних автоматів, яка вже стала класичною для опису стану ігрових агентів чи алгоритм A^* для планування шляху, який використовуються майже у всіх іграх, де використовується ігрова механіка відповідного роду [20].

Можна сказати, що найбільш широкою областю для застосування методів машинного навчання і інтелекту виступає саме створення реалістичних ігрових супротивників. В цьому випадку відповідні методи можуть, як в прямому вигляді контролювати поведінку ігрових агентів, так і в більш складному випадку відповідати за вибір стратегії поведінки відповідно до ігрових обставин [21].

Можливість використання певних методів машинного навчання і інтелекту прямо залежить від ігрової концепції, які реалізує відеогра [16]. Здебільшого основними характерними рисами, які обумовлюють можливість використання певних методів, це наявність шуму у даних і виникнення апіорних ситуацій [22]. Одним з найбільш універсальних засобів, який може використовуватися в даних ситуаціях, є нейронні мережі.

1.3 Обґрунтування використання нейронних мереж

1.3.1 Сучасні підходи до організації нейронних мереж

Під поняттям нейронної мережі в контексті даної роботи, як і в більшості публікацій в інтернеті, розуміється поняття штучної нейронної мережі (англ. artificial neural networks). Штучна нейронна мережа в свою чергу представляє собою обчислювальну систему, натхнену біологічними нейронними мережами (англ. biological neural networks), що складають мозок тварин [23].

Нейронна мережа не являє собою якогось певного алгоритму, а скоріше



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

є основою для багатьох різних алгоритмів машинного навчання для спільної роботи і обробки даних. Як правило, відповідні системи навчаються виконувати завдання без програмування правил під конкретні завдання [23]. Що забезпечує велику гнучкість в плані застосування, завдяки чому відповідний метод набув широкої популярності.

Штучні мережі представляють собою сукупності з'єднаних вузлів, що називають штучними нейронами (у відповідність до біологічних нейронів). З'єднання між нейронами називають мостами. Мости передають сигнал між нейронами, змінюючи його відповідно до ваги. Штучний нейрон отримує сигнал, обробляє його за допомогою функції активації, після чого сигналізує наступним нейронам, приєднаним до нього мостами [24].

Штучні нейрони зазвичай розділяють за шарами, а шари в свою чергу поділяють на вхідний, приховані, вихідні [25]. В поширених реалізаціях нейронних мереж сигнали і ваги мостів представляють собою дійсні числа, а значення входів штучних нейронів обчислюється зведеною функцією від вхідних сигналів [24]. Загально прийнятим є, що вага мостів підлаштовується в перебігу навчання. Окрім цього в деяких випадках шари нейронної мережі виконують різні види перетворень своїх входів [24].

1.3.2 Дослідження базових топологій нейронних мереж

Нейроні мережі представляють собою дуже поширений засіб з точки зору використання в програмних системах, в результаті чого формуються все нові і нові типи нейронних мереж [26]. Дивлячись на кількість нових розробок, які торкаються нейронних мереж можна сказати, що кількість нових типів і модифікацій нейронних мереж зростає маже в експоненціальній швидкості.

Сьогодні нереально знайти класифікацію, яка охоплює всі існуючі типи і модифікації нейронних мереж. Однією з більш менш повних класифікацій можна вважати класифікацію Fjodor van Veen з Asimov institute



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

[27]. Відповідна класифікація торкається великої кількості базових типів нейронних мереж і наведена на рис. 1.1.

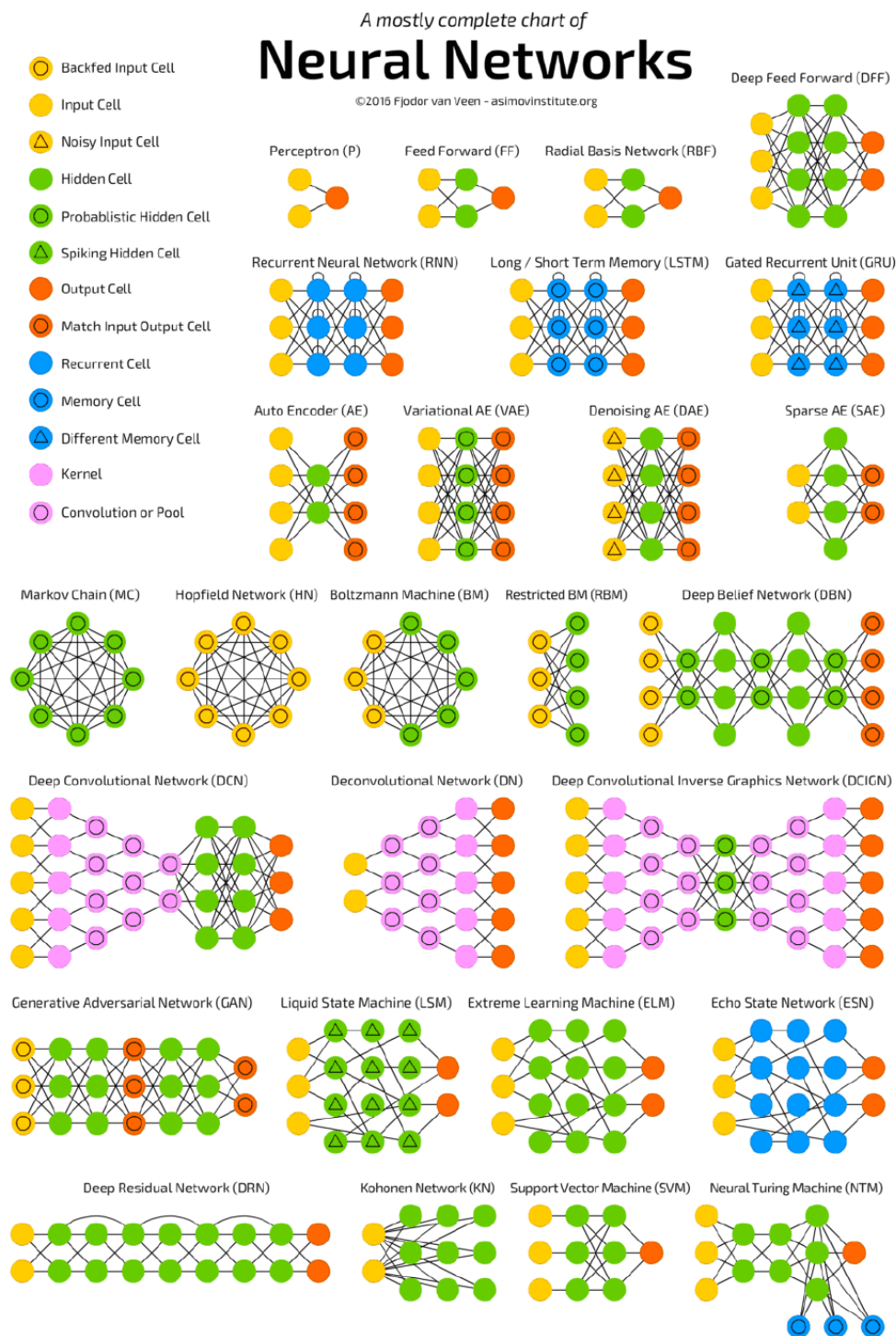


Рисунок 1.1 – Класифікація нейронних мереж

Враховуючи факт того, що прийняття рішення нейронною мережею повинно виконуватися в реальному часі, під час ігрового процесу найбільш відповідним типом нейронної мережі можна вважати глибокі нейронні мережі з обмеженою кількістю внутрішніх слоїв [25].

Використання більш складних типів нейронних мереж буде приводити до зменшення швидкості обчислювання, що, в свою чергу, є дуже важливим, оскільки системи реального часу, які представляють собою відеогри, мають часові обмеження [28], [29]. Окрім цього, використання більш складних нейронних мереж може суттєво вплинути на швидкість навчання мережі [29].

Враховуючи вищенаведений опис використання більш складних типів нейронних мереж в ракурсі обмеженої обчислювальної потужності використання в системах реального часу таких, як ігри, можна назвати недоцільним ускладненням.

1.3.3 Практичне значення нейрону зміщення

Окрім нейронів у наведеній вище класифікації існує і додатковий тип нейронів – упереджений нейрон, чи як його ще називають, нейрон зсуву. Він відрізняється від основного виду нейронів тим, що його вхід і вихід в будь-якому випадку дорівнює одиниці. При цьому вхідних синапсів такі нейрони не мають. Розташування таких нейронів відбувається по одному на шар і не більше, також вони не можуть з'єднуватися синапсами один з одним [30]. Розміщувати такі нейрони на вихідному шарі не доцільно.

Відповідні нейрони представляють собою засіб розширення області визначення нейронної мережі. Бувають ситуації, в яких нейрона мережа просто не зможе знайти вірне рішення через те, що потрібна точка буде знаходитися за межами досяжності [31], [32]. Саме для цього і потрібні такі нейрони, щоб мати можливість змістити область визначення.

Тобто вага мосту змінює вигин графіка функції, тоді як нейрон зміщення дозволяє здійснити зрушення по осі координат X, таким чином,



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

щоб нейрона мережа змогла захопити область недоступну їй без зсуву [31]. При цьому зрушення можуть бути здійснені як вправо, так і вліво. Схематично нейрони зсуву можуть позначатися в вигляді окремого нейрону чи не позначаються зовсім, але їх вага враховується за замовчуванням при розрахунку.

Також нейрони зсуву дозволяють отримати результат в тому випадку, коли всі інші нейрони видають 0 в якості вихідного параметра [31]. У цьому випадку незалежно від ваги синапсу на кожен наступний шар буде передаватися саме це значення. Тобто, наявність нейрона зміщення дозволить виправити ситуацію і отримати інший результат. Доцільність використання нейронів зміщення зазвичай визначається шляхом тестування мережі з ними і без них і порівняння результатів.

1.3.4 Функції активації нейронів

Функція активації нейронів приймає значення від сумарної функції входів чи як її ще називають зважена функції від входів [33]. Сумарна функція може бути представлена графічно на моделі нейрону (рис. 1.2) чи формулою (1.1), де \bar{x} – вектор вхідних активацій, \bar{w} – вектор вагових коефіцієнтів мостів, b – зсув функції у відповідності до нейрону зміщення.

$$S(\bar{x}, \bar{w}, b) = \sum_{i=1}^n x_i \cdot w_i + b = \bar{x} \cdot \bar{w} + b \quad (1.1)$$

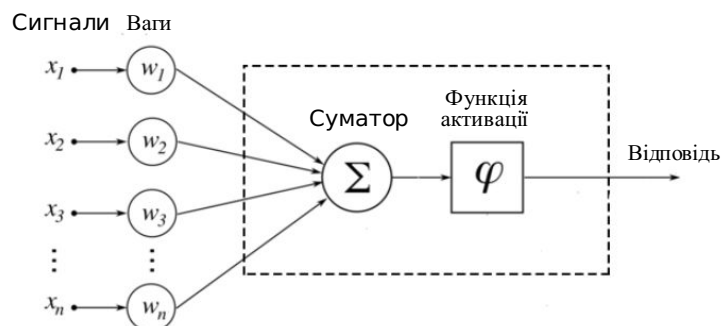


Рисунок 1.2 – Модель обчислення сумарної функції

В загальному плані функція активації являє собою спосіб нормалізації вхідних даних. Тобто, якщо на вході у вас буде велике число, то пропустивши його через функцію активації буде отримано вихід в потрібному діапазоні [32]. Існує досить велика кількість функцій активації, але умовно їх можна поділити на категорії за критерієм нормалізації вхідних даних: функції без нормалізації, функції жорсткої нормалізації, функції плавної нормалізації.

Перша категорія представлена лінійною функцією активації. Друга категорія представлена функціями одиночного стрибка і функцією лінійного порогу, поліпшеної лінійної одиниці. Третя категорія включає в себе плавні монотонно зростаючі всюди диференційовані нелінійні функції з насиченням, як приклад функцій можна розглядати функцію сигмоїда, гіперболічний тангенсу чи гладкого наближення функції лінійної одиниці. Відповідна природа функцій третьої категорії дозволяє істотно скоротити обчислювальну складність методу зворотного поширення помилки.

В процесі реалізації часто використовується лінійна функція активації (рис. 1.3), яка передає значення збільшеної функції входів в якості значення активації нейрону. З цією функцією значення передаються без перетворень. Відповідну функцію активації нейрона можна представити за допомогою формули (1.2).

$$F(\bar{x}, \bar{w}, b) = S(\bar{x}, \bar{w}, b) = \bar{x} \cdot \bar{w} + b \quad (1.2)$$

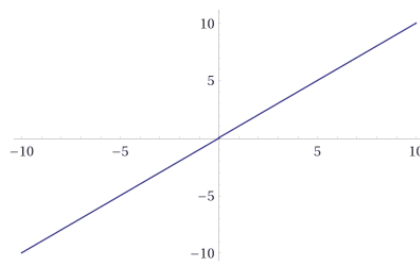


Рисунок 1.3 – Лінійна функція активації

Можна сказати, що лінійний нейрон являє собою модель лінійної регресії, яку переформулювали на мові нейронних мереж. Відповідну функцію дуже часто використовується для тестування реалізації нейронної мережі, оскільки вона передає свої значення без змін. Окрім цього її можна дуже часто зустріти на вихідному шарі нейронної мережі. Це пояснюється необхідністю отримання ненормалізованих даних на виході.

Функція одиничного стрибка (рис. 1.4) чи як її ще називають порогова функція вона представляє собою просту кусочно-лінійну функцію, де якщо вхідне значення менше порогового, то значення функції активації дорівнює мінімальному допустимому, інакше - максимально допустимому. Відповідно функцію активації нейрона можна записати формулою (1.3).

$$F(\bar{x}, \bar{w}, b) = \begin{cases} 1 & S(\bar{x}, \bar{w}, b) > a \\ 0 & S(\bar{x}, \bar{w}, b) \leq a \end{cases} \quad (1.3)$$

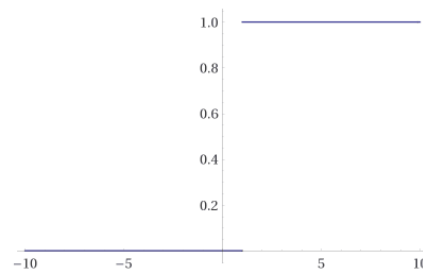


Рисунок 1.4 – Функція активації одиничного стрибка

Функція лінійного порогу або гістерезису (рис. 1.5) представляє собою функцію, яка має дві лінійні ділянки, де функція активації тотожно дорівнює мінімально допустимому і максимально допустимому значенням і є ділянка, на якій функція строго монотонно зростає. Відповідна функція активації нейрону записана за допомогою формули (1.4), де k – деякий коефіцієнт перетворення.

$$F(\bar{x}, \bar{w}, b) = \begin{cases} 1 & k \cdot S(\bar{x}, \bar{w}, b) > 0 \\ k \cdot S(\bar{x}, \bar{w}, b) & 0 \leq k \cdot S(\bar{x}, \bar{w}, b) \leq 1 \\ 0 & k \cdot S(\bar{x}, \bar{w}, b) < 0 \end{cases} \quad (1.4)$$

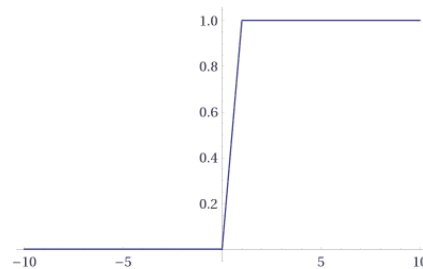


Рисунок 1.5 – Функція активації лінійного порогу

Функція сигмоїда (рис. 1.6) або як її ще називають логістична функція представляє собою S-подібну криву, діапазон значень якої відповідає відрізьку $[0; 1]$. Відповідна функція на відміну від попередньо розглянутих вирішує дилему шумового насичення, оскільки вводить поняття впевненості, завдяки якому відбувається посилювання слабких сигналів і не відбувається насичення від сильних сигналів. Відповідна функція активації нейрону записана за допомогою формули (1.5).

$$F(\bar{x}, \bar{w}, b) = \sigma(S(\bar{x}, \bar{w}, b)) = \frac{1}{1 + e^{-S(\bar{x}, \bar{w}, b)}} \quad (1.5)$$

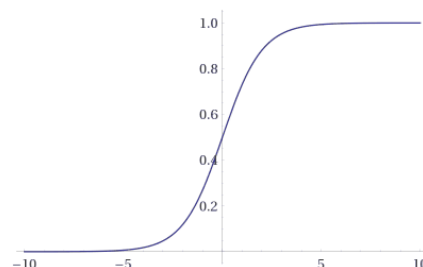


Рисунок 1.6 – Функція активації сигмоїда

Гіперболічний тангенс (рис. 1.7) також представляє собою функцію, схожу на функцію сигмоїда, але гіперболічний тангенс зростає більш швидко і центрований на 0, а його діапазон значень відповідає відрізьку $[-1; 1]$. Окрім цього, на відміну від сигмоїда, вона дозволяє оброблювати негативні і позитивні сигнали, що робить можливість застосування даної функції більш широкою, але потрібно враховувати, що використання цієї функції тільки з



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

позитивними значеннями може значно погіршити результати навчання нейронної мережі. Відповідна функція активації нейрону записана за допомогою (1.6).

$$F(\bar{x}, \bar{w}, b) = \tanh(S(\bar{x}, \bar{w}, b)) = \frac{e^{S(\bar{x}, \bar{w}, b)} - e^{-S(\bar{x}, \bar{w}, b)}}{e^{S(\bar{x}, \bar{w}, b)} + e^{-S(\bar{x}, \bar{w}, b)}} \quad (1.6)$$

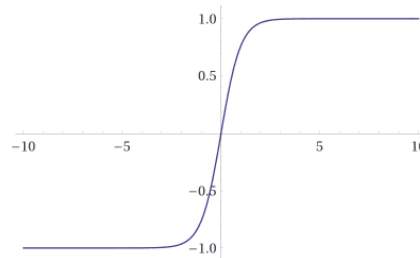


Рисунок 1.7 – Функція активації гіперболічний тангенс

Поліпшена лінійна одиниця (рис. 1.8) або ReLU (від англ. rectified linear unit), на відміну від функції сигмоїда і гіперболічного тангенсу ця функція виправляє значення лише на половину. Функція та її похідна монотонні, але вона не всюди диференційована. Основна проблема функції полягає в тому, що всі негативні значення стають нульовими негайно, що знижує здатність моделі правильно вписуватися або тренуватися з даних. Це означає, що будь-який негативний ввід, наданий для функції активації негайно повертає значення до нуля, не наближаючи відповідні негативні значення. Відповідна функція активації нейрону записана за допомогою формули (1.7).

$$F(\bar{x}, \bar{w}, b) = \max(0; S(\bar{x}, \bar{w}, b)) \quad (1.7)$$

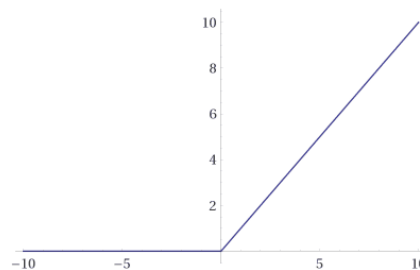


Рисунок 1.8 – Функція активації – поліпшена лінійна одиниця

Гладке наближення функції лінійної одиниці (рис. 1.9) чи як її ще називають SoftPlus невідмінно від поліпшеної лінійної одиниці функція є всюди диференційована. Цікавим фактом є те, що похідною даної функції є функція сигмоїда.

$$F(\bar{x}, \bar{w}, b) = \log(1 + \exp(\mathcal{S}(\bar{x}, \bar{w}, b))) \quad (1.8)$$

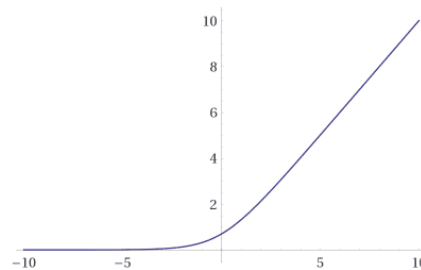


Рисунок 1.9 – Функція активації – гладке наближення лінійної одиниці

Також існують і інші модифікації функції поліпшеної лінійної одиниці. Наприклад, поліпшена лінійна одиниця з витокм (англ. Leaky ReLU) покликана вирішити проблему з негативним значеннями сигналів. Окрім цього є функції параметрична лінійна одиниця (англ. Parametric ReLU) і випадкова єдина одиниця (англ. Randomized ReLU), в яких ключовою особливістю є те, що кутовий коефіцієнт на негативному інтервалі не задано попередньо, а визначається на основі даних під час навчання чи генерується випадково з заданого інтервалу відповідно. Окрім цього існують і варіанти гладкого наближення відповідних функцій.

В якості критерію вибору функції активації можуть слугувати вхідні сигнали та бажані відповіді. В якості можливості використання декількох функцій активації зазвичай обираються емпіричним шляхом на базі аналізу процесу навчання.

1.4 Парадигми та підходи навчання нейронних мереж

Відповідно до згаданих попередю підходів машинного навчання можна



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

зрозуміти, що існує достатньо широкий перелік можливих підходів до навчання нейронних в цілому, але для навчання глибинних нейронних мереж сьогодні найчастіше використовується декілька базових підходів та їх модифікацій [26]. Відповідні базові підходи описані далі.

Метод зворотного поширення (англ. backpropagation) заснований на підході градієнтного спуску для знаходження локального максимуму і мінімуму. У цьому методі сигнали послідовно передаються від ввідних нейронів до вихідних. Після чого обчислюється похибка між бажаним і отриманим виходами та засновуючись на даній похибці робиться зворотна передача, яка засновується на тому, щоб послідовно змінювати ваги нейронів до отримання бажаного результату.

Метод пружного поширення (англ. resilient propagation чи Rprop) на відміну від методу зворотного алгоритм використовує тільки знаки приватних похідних для налаштування вагових коефіцієнтів і використовує так зване «навчання по епохах», коли корекція ваг відбувається після надання мережі всіх прикладів з навчальної вибірки.

Метод генетичного навчання (англ. genetic algorithm) представляє собою спрощену інтерпретацію природного алгоритму. Даний алгоритм, як і алгоритм пружного навчання здійснює навчання за епохами з декількох особин після чого з них обираються найкращі відбуваються модифікації найкращих. Метод на відміну від попередніх дозволяє здійснювати навчання без навчальної вибірки.

1.5 Висновки за розділом

Обґрунтовано, що серед методів машинного навчання і інтелекту прийнятними для використання при побудові логіки ігрових агентів є нейронні мережі. Але треба враховувати, що методи машинного інтелекту з навчанням привносять свої специфіку до створення і налагодження ігрової системи.



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

Однією з головних проблем, яка виникає, є необхідність передбачення усього набору входних ситуацій і можливих відповідей на моменті проектування і неможливість розширення їх списку без перенавчання. Що значно звужує можливість їх використання у реальних ігрових системах.

Відповідно, підходи, що розглядаються, можуть використовуватися в ігрових механіках з мінімальною кількістю зовнішніх взаємодій, які впливають на поведінку між ігровими агентами. До такого роду можна віднести обмежений перелік ігрових механік, в яких може бути застосовано даний підхід. Серед таких механік – автомобілі з гоночних симуляторів, які не мають зовнішніх взаємодій. Саме їх було обрано для подальшого розгляду.

У відповідності до обмежених обчислювальних ресурсів і специфіки систем реального часу, які представляють собою відеоігри, для розгляду було обрано глибинні нейронні мережі з обмеженою кількістю внутрішніх слоїв.

Враховуючи факт фактичної відсутності вибірки навчання і можливої проблематичності генерації чистої вибірки, щоб запобігти можливій проблемі перенавчання нейронної мережі, то найбільш відповідним підходом до навчання виступає підхід навчання з підкріпленням на базі генетичного алгоритму. Обирати активаційні функції нейронів на даному етапі також не є доцільним, оскільки при виборі функції потрібно враховувати бажані дані входів і виходів, а також емпіричні данні під час реалізації ігрової системи.

Відповідно до тематики роботи і визначеної області застосування було проведено аналіз існуючих досліджень і реалізацій систем моделювання. Ознайомитися з відповідним аналізом можна у Додатку Б. Відповідно проведеного аналізу було підтверджено основні тези, отримані в результаті огляду сучасних підходів.

РОЗДІЛ 2

РОЗРОБКИ ПРОГРАМНИХ МОДЕЛЕЙ ІГРОВИХ АГЕНТІВ У МОДЕЛЮЮЧИХ СЕРЕДОВИЩАХ

Ідея використання методів машинного навчання і інтелекту для реалізації штучного інтелекту в іграх не є новою. В деякому сенсі можна сказати, що ігри стали однією з відправних точок розвитку відповідних методів. На початковому етапі розвитку галузі дослідження торкалися класичних ігор з ідеальною інформацією, а потім, з розвитком методів і технологій дослідження, розповсюдились і на відеоігри, які стали новим полем діяльності для дослідників.

Задача реалізації транспортних агентів на базі методів штучного інтелекту також не є новою. Під час виконання роботи вдалось знайти достатньо велику кількість матеріалів в інтернеті на тематику самокерування автомобілем. Знайдені реалізації умовно можна поділити на 2 класи. Перший клас представляє собою реалізації самокерованих автомобілів для ігрових механік, другий клас – спроби реалізації системи автопілотів автомобілів в реальному житті.

Відповідні класи мають суттєві відмінності з точки зору підходів до програмної реалізації, використаних методів і інших особливостей. Другий клас, хоча і близький за тематикою, але вирішує інші задачі. Це пояснюється тим, що в реальному світі машина потребує вирішення набагато більше питань, ніж у віртуальному. Окрім цього, системи другого класу мають значно вузькі можливості по отриманню даних з оточуючого середовища на відміну від першого типу систем, а також вони повинні враховувати додаткові проблеми, такі, як безпечність руху, щільність трафіку, погані погодні умови, дорожні знаки, тощо.

В результаті такої істотної різниці системи другого класу представляють набагато більш складні системи на відміну від ігрових. Тому,



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

їх хоча і можна використати для створення ігрових агентів, але фактична доцільність цього у ігрових системах відсутня, оскільки відповідні системи обумовлюють під собою інші принципи реалізації і призначені для інших задач.

Відповідно до цього огляд систем другого роду в рамках даної роботи не є доцільним. В той час майже всі знайдені реалізації систем першого класу представляють реалізацію механіки машинок, які рухаються у відповідності до показників променевих сенсорів, які знаходять перепони в певному радіусі від автомобілю. В якомусь роді це представляє механіку керування автомобілем з запобіганням зіткнень.

Знайдені матеріали по системах першого роду в свою чергу можна поділити на матеріали, які представляють собою керівництва дії і опубліковані в вигляді проектів і роботи, які носять науковий характер.

2.1 Формування моделей ігрових агентів в середовищах зі штучним інтелектом

За тематикою дипломної роботи були знайдені та досліджені різного роду матеріали як з напряму, пов'язаного з тематикою дослідження, так із суміжними областями. З приводу вивчення питання історичного ракурсу і базових концепцій застосування можна виділити ресурси [20], [34]–[37]. Більш детально концепції використання розглядаються у роботах [16]–[19], [21]. Хоча в даних матеріалах і проводиться огляд деяких концепцій, принципів, підходи до створення штучного інтелекту на базі машинного навчання і інтелекту, але не проводиться фактична оцінка складності розробки і налагодження агентів.

Оскільки дослідження проводиться на базі створення гоночного симулятора, де ігрові машини будуть керуватися методами машинного навчання і інтелекту, то окрім загального ракурсу вивчення робіт даної тематики було знайдено дослідження у відповідність до цієї тематики.



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

Однією з фундаментальних робіт, де детально розглядалося питання реалізації ігрового автомобілю, можна назвати роботу дослідників Kenneth S. і Risto M. Вони представили метод нейрона посилюючих топології (англ. Neuro Evolution of Augmenting Topologies чи скорочено NEAT), детальніше про це можна прочитати у джерелі [38]. Дослідження відповідної топології також можна зустріти і в роботі Воеводин О. М., яка знаходиться за посиланням [39].

Окрім цього в рамках тематики роботи вдалося знайти вивчення особливостей моделювання та симуляції автомобілів у двовимірному просторі. Відповідна робота належить автору Aramayis Amiraghyan і наведена за посиланням [40]. Автор даної роботи використовує глибину нейронну мережу і виконує навчання з підкріпленням на базі генетичного алгоритму. В якості функції активації використовується сигмоїда. Скріншот інтерфейсу реалізованої їм програми наведено на рис. 2.1.

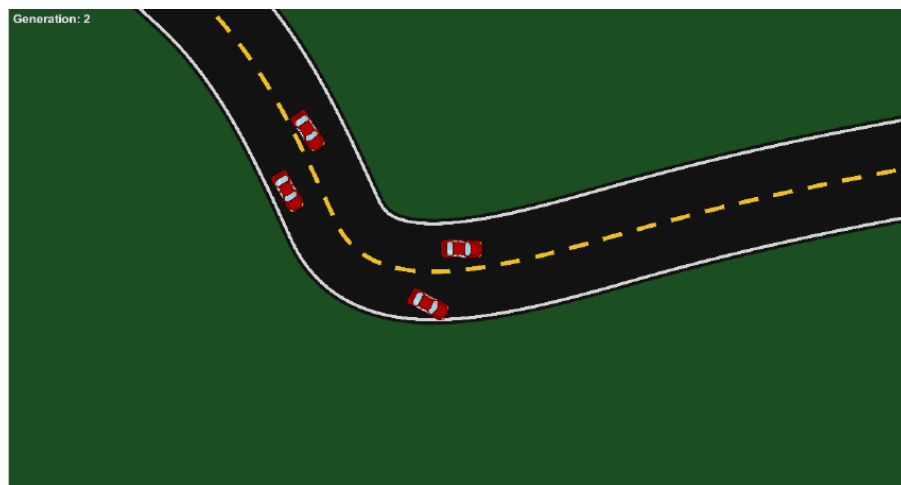


Рисунок 2.1 – Інтерфейс програми, реалізованої автором Aramayis Amiraghyan

Також було оглянуто тези Blair Peter Trusler і Dr Christopher Child наведені за посиланням [41]. В них робиться огляд підходу Q-навчання (англ.



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

Q-learning) до реалізації штучного гоночного ігрового автомобіля. Також огляд питання реалізації штучного інтелекту пересування на відповідних методах розглядається в роботі Yung-Ping Fang та I-Hsien Ting [42], реалізацію цієї ж ідеї можна знайти і в тезах і роботі Чернявского І. І. [43], [44]. В даних роботах оглядається один і той самий підхід, але з певними корективами. Інтелект переслідування заснований на навчанні з підкріпленням, а саме підході Q-навчання.

Ще одна цікава робота належить авторам Artur Bednarczyk і Dawid Grajewski [45]. В ній оглядається створення агента на базі глибинної нейронної мережі, в якості підходу до навчання якої використовується метод зворотного поширення. Вихідні данні до методу зворотного поширення отримуються в результаті процесу керування машини гравцем. Інтерфейс розробленої програми в рамках цієї роботи наведено на рис. 2.2.

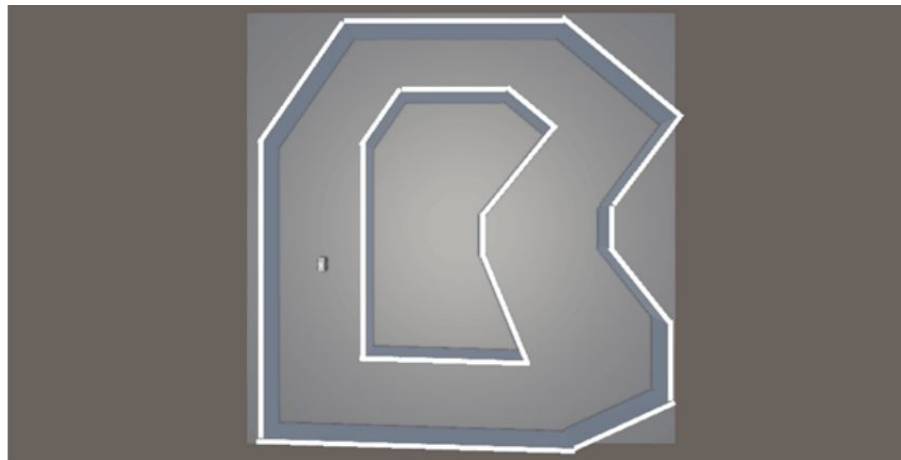


Рисунок 2.2 – Інтерфейс програми, реалізованої авторами Artur Bednarczyk і Dawid Grajewski

2.2 Формування моделей ігрових агентів в інших реалізованих програмних системах

На ресурсі Code Project представлено статті «ReInventing Neural

Networks» і вихідний код автора Byte-Master-101 в трьох частинах. Посилання на статті можна знайти в пунктах [46]–[48] в переліку літератури, а зображення основної форми проекту представлено на рис. 2.3. Автор описує реалізацію штучного інтелекту автомобілю за допомогою нейронних мереж.

В даній реалізації використовується глибинна нейрона мережа з 2 прихованими шарами. На вхідний шар нейронної мережі подаються показники 6 сенсорів автомобілю, приховані шари містять 3 та 2 нейрони, а вихід з мережі представлено 2 нейронами, які відповідають за вертикальну і горизонтальну осі керування. Навчання здійснюється за методом генетичного алгоритму.

У першій статті автора описується принцип функціонування нейронних мереж і деякі риси генетичного навчання, а також реалізацію класу логіки нейронної мережі. В якості функції активації використовується поліпшена лінійна одиниця з витоком. У другій статті реалізується система автомобілі на базі описаного у попередній статті класу.

Навчання здійснюється за допомогою оператора мутації з параметром вірогідності 0.3 і максимальною зміною матованої ваги 2. У третій частині за пропозицією Michael Bartlett було реалізовано навчання за допомогою безпечної мутації, що дозволило покращити результати навчання.

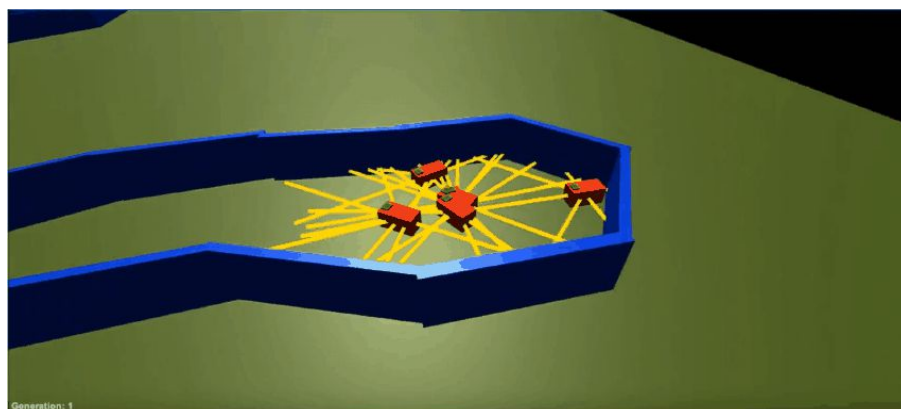


Рисунок 2.3 – Інтерфейс програми, реалізованої автором Byte-Master-101

Також цікаве рішення вдалось знайти на просторах GitHub, хоча сам проект не представлений в вигляді статті, як попередній. Це проект автора Samuel Arzt під назвою «Applying_EANNs» (скорочено від Applying Evolutionary Artificial Neural Networks). Посилання на проект наведено в переліку джерел і знаходиться за пунктом [49], а інтерфейс його головної форми наведено на рис. 2.4.

В відповідній реалізації автомобілі мають 5 передніх сенсорів, значення яких, як і о попередньому розглянутому аналізі, представляють собою вхідні сигнали нейронної мережі. У проекті також використовується глибинна нейрона мережа з 5 нейронами на вхідному шарі, з 4 та 3 нейронами на прихованих шарах і 2 нейронами на вихідному.

В якості функції активації нейронів використовується сигмоїда, а навчання здійснюється генетичним алгоритмом. Популяції представляють собою комбінацію двох найкращих агентів на минулому етапі, а всі інші агенти формуються в результаті перетину генів цих двох агентів за допомогою функції кросоверу, після чого додатково використовується оператор мутації.

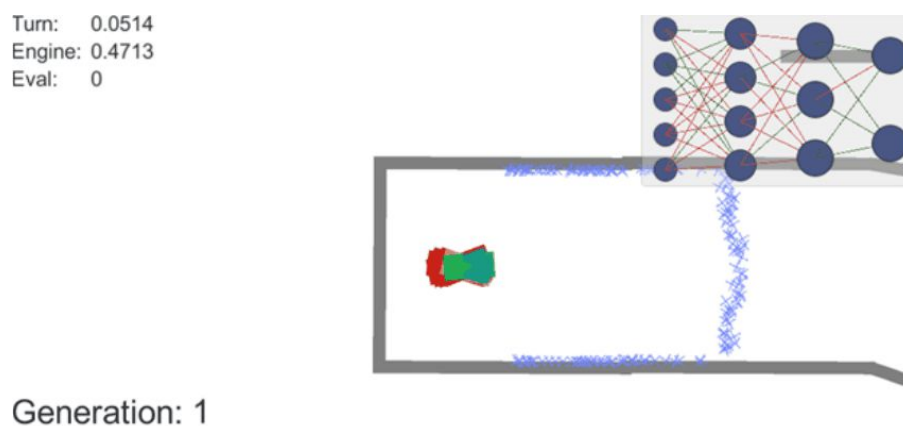


Рисунок 2.4 – Інтерфейс програми, реалізованої автором Samuel Arzt

На просторах GitHub окрім попереднього проекту вдалось знайти ще

один гарно документований проект автору Plunti під назвою «evolution-of-cars» (скорочено від Evolution of Cars Using Neural Networks and Genetic Algorithm). Посилання на відповідний проект наведено в переліку джерел за пунктом [50] і інтерфейс відповідного проекту наведено на рис. 2.5. Автор Plunti запозичив програмну логіку роботи генетичних автівок у Byte-Master-101, але реалізував власну модифікацію генетичного алгоритму і більш функціональне тестове середовище. Популяція формується з елітних агентів і їх нащадків в результаті виконання операторів елітизму і кросоверу.

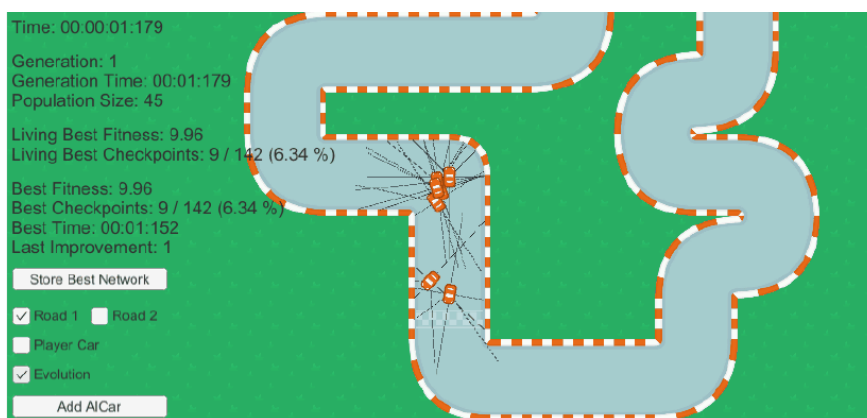


Рисунок 2.5 – Інтерфейс програми, реалізованої автором Plunti

2.3 Висновки за розділом

Як можна побачити з проведеного аналізу питання використання методів машинного навчання і інтелекту, а саме, питання реалізації штучного інтелекту ігрового автомобілю, розглядається не тільки науковою спільнотою, але й ще спільнотою програмістів в цілому.

Проаналізувавши роботи за тематикою можна виявити, що найбільш часто вживаною технологією реалізації штучного інтелекту автомобілю є нейронні мережі, а найчастішим типом виступають глибинні нейронні мережі.

Навчання нейронної мережі здебільшого здійснюється за допомогою навчання з підкріпленням. Основною причиною поширення методу навчання

з підкріпленням пояснюється відсутністю необхідності створення навчальних вибірок, що приводить до пришвидшення та зниження вартості розробки.

Серед методів навчання, які часто використовуються, найбільше представлено навчання за допомогою генетичного алгоритму. На другому місці за поширеністю Q-навчання.

В жодній з розглянутих робіт не проводилась повноцінна формалізація моделі агента ігрового автомобілю, заснованого на базі нейронних мереж, також не проводився аналіз автомобілів з різним розміщенням сенсорів. Окрім цього не проводився порівняльний аналіз механіки руху за маршрутом і навчених агентів на базі методів машинного навчання і інтелекту.



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

РОЗДІЛ 3

ФОРМАЛЬНА ПОБУДОВА МОДЕЛІ АГЕНТА АВТОМОБІЛЯ, ЗАСНОВАНОЇ НА МЕТОДАХ МАШИННОГО НАВЧАННЯ І ІНТЕЛЕКТУ

3.1 Визначення задач агента ігрового автомобіля

В гоночних симуляторах можна виділити дві базові концепції треків для заїзду (рис. 3.1). Перша з них є кінцевим треком, який має точку старту, від якої відбувається рух, і точку фінішу, до якої відбувається рух. Другий є круговим треком, який має лише точку старту і не має кінцевої точки завершення аналогічної до першого типу треку. Відповідна різниця між типами треків вимагає двох окремих формулювань задач для кожного типу.

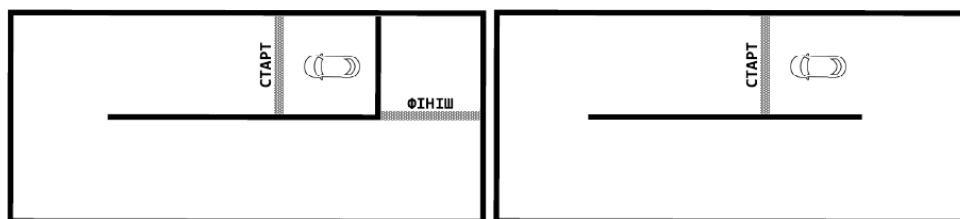


Рисунок 3.1 – Базове подання кінцевого і кільцевого треку

В випадку кінцевого треку задача агента ігрового автомобіля полягає в забезпеченні успішного подолання треку від старту до фінішу за мінімальний час. В випадку кільцевого треку – здійсненні успішного руху за напрямком треку з мінімальним часом проходження кола. Під успішністю в обох формуваннях задач розуміється рух автомобілів без здійснення зіткнень.

3.2 Загальний підхід до здійснення керування автомобілем

Ігровий автомобіль представляє собою спрощене представлення

реального автомобілю. Вона може прискорюватися, сповільнюватися, гальмувати і розвертатися. Фактично ці дані можна представити в вигляді коефіцієнту змінення швидкості і коефіцієнту повороту автівки. Відповідно до цього для керування автівкою їй потрібно передавати відповідні коефіцієнти.

Відповідна модель керування наведена на рис. 3.2. Керування автівкою здійснюється за допомогою двох осей керування, які відповідають за повороти і змінення швидкості. Часто відповідні осі керування називають горизонтальною і вертикальною.



Рисунок 3.2 – Модель керування автомобілем

Забігаючи наперед можна сказати, що відповідні осі керування в даній реалізації гоночного автомобіля мають діапазон значень від -1 до 1. Це важливо, як і для розуміння програмної логіки фізичного контролера, так і при реалізації програмної логіки контролерів автомобілю.

Якщо вертикальна вісь від'ємна, то автомобіль сповільнюється чи їде до заду, а якщо позитивна то до переду. У випадку максимального і мінімального значення автомобіль прискорюється чи сповільнюється з максимальною передбаченою швидкістю.

Якщо горизонтальна вісь від'ємна, то автомобіль повертає ліворуч, якщо позитивна, то праворуч. В крайніх точках діапазону значень осі автомобіль повертає на максимально можливий градус повороту, а в центральній точці діапазону автомобіль їде прямо.



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

3.3 Отримання даних про навколишнє середовище автомобіля

Однією з найбільш популярних механік отримання даних про навколишнє середовище автомобіля є механіка сенсорів зіткнень. Відповідна механіка полягає в тому, що від автомобіля пускається певна кількість промінів встановленої довжини під різними кутами. Сенсор зіткнення повертає значення, яке відповідає довжині проміння чи відстань, до перешкоди з якою відбулося зіткнення.

В вигляді графічного представлення відповідна механіка може бути представлена в вигляді наведеному на рис. 3.3. Хоча механіка є відносно простою, але вона дозволяє отримати достатню кількість даних для того, щоб здійснювати керування автомобілем.

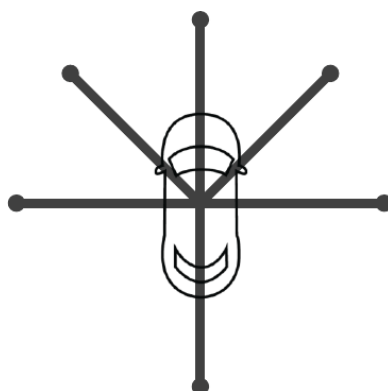


Рисунок 3.3 – Модель отримання даних

3.4 Контролер автомобіля на базі нейронних мереж

Контролер автомобілю на базі нейронних мереж фактично реалізує логіку нейронної мережі, яка на вхід в якості сигналів отримує дані від сенсорів зіткнення автомобілю, а в якості відповіді надає значення



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

горизонтальної та вертикальної осей координат. Як вже зазначалося раніше відповідні осі представляють коефіцієнт змінення швидкості і коефіцієнт повороту автівки. В вигляді схематичного представлення варіант реалізації відповідної механіки наведено на рис. 3.3.

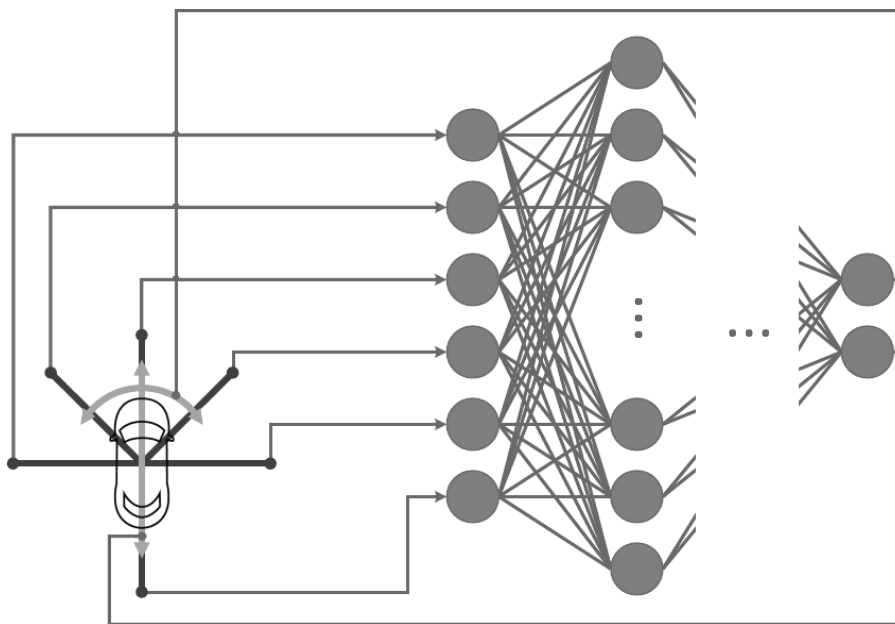


Рисунок 3.4 – Модель прийняття рішення контролером автомобілю

Як вже зазначалося раніше, діапазон вихідних значень знаходиться в проміжку від -1 до 1. Враховуючи цей факт найбільш відповідною функцією активації даної нейронної мережі є гіперболічний тангенс. Окрім цього необхідно розширити початковий діапазон значень мостів нейронної мережі від -1 до 1.

3.5 Особливості використання підходу навчання з підкріпленням

Для даної системи найбільш ефективним є виконання навчання з підкріпленням на базі генетичного алгоритму. Відповідну модель взаємодії

агента з середовищем в загальному випадку під час навчання з підкріпленням наведена на рис. 3.5.

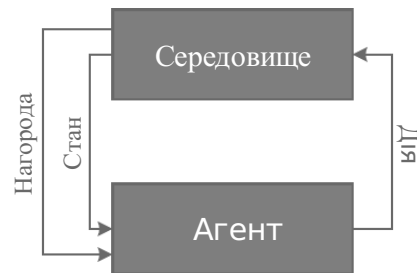


Рисунок 3.5 – Модель взаємодії агента з середовищем під час навчання з підкріпленням

В загальному випадку відповідна концепція полягає в тому, що агент взаємодії з середовищем в певні дискретні проміжки часу $t = 0, 1, 2, \dots$, де на кожному кроці t агент отримує інформацію про середовище – стан середовища: $s_1 \in S$, де S – множина станів середовища. На основі цього агент обирає дію $a_t \in A(s)$, де множина можливих дій доступних зі стану s .

На наступному кроці агент отримує новий стан s_{t+1} . Під час виконання певних дій агент може отримати від системи нагороду r_i , що представляє собою безпосередню оцінку дій, вчинених агентом. Важливо відзначити, що нагорода є оцінкою дій агента тільки в найближчій перспективі. Метою агента є максимізація суми нагород в довгостроковій перспективі. В огляді використання генетичного алгоритму сума нагород фактично представляє собою фітнес функцію, яку можна представити в вигляді формули (3.1), де n – кількість отриманих нагород.

$$F = \sum_{i=0}^n r_i \quad (3.1)$$

Відповідно до зазначеного у попередніх пунктах оглянемо відповідну концепцію для даного конкретного випадку. В даному випадку агент автомобілю взаємодіє з середою в певні дискретні проміжки часу, де отримує



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

інформацію про навколишнє середовище і приймає рішення про рух на основі стану навколишнього середовища. Після виконання прийнятого рішення агент автомобілю переходить в новий стан.

В якості одного з методів отримання нагород можна розглядати події чи деякі базові параметри ігрового середовища, а саме час активності, пройдений шлях. Але використання відповідних параметрів теоретично створює проблеми з навчанням.

В першому випадку реальною є ситуація, що машинка, яка просто стоїть на місці, буде найбільш результативною, чим машинка, яка поїхала та зіштовхнулася з перепорою. В другому випадку можлива ситуація, що машинка, яка їздить по колу, буде найбільш результативною. Сумісне використання відповідних параметрів до нового результату не приведе, тому реалізація навчання з підкріпленням потребує реалізації додаткової механіки.

Відповідною механікою представляється системою контрольних точок, розміщених за маршрутом з послідовним проходженням, а в якості дії для отримання нагороди виступає активація контрольної точки. Оскільки ми автомобіль навчаємо не просто їздити, а швидко їздити, то значення нагороди за активацію контрольної точки повинно бути прямо пропорційним номеру точки, а також зворотно пропорційним часу їзди до неї.

Відповідну функцію нагороди можна представити в вигляді формули (3.2), де i – номер активованої контрольної точки, а t_i – час їзди до контрольної точки. Але із-за недостатньої швидкості зростання чисельника з великою швидкістю зростання займенника існує можливість того, що машина, яка показала гірший результат взяття контрольних точок за більш короткий час буде мати кращий результат фітнес функції (3.1).

$$r_i = \frac{i}{t_i} \quad (3.2)$$

Для запобігання відповідної ситуації потрібно збільшити швидкість зростання чисельника чи зменшити швидкість зростання займенника. Оскільки займенник відповідає дискретно зростаючій величині, то найбільш



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

доцільно збільшувати швидкість зростання чисельника. Відповідно до цього можна отримати функцію нагороди, наведену у формулі (3.3).

$$r_i = \frac{i^2}{t_i} \quad (3.3)$$

Відповідно до цього фітнес функція (3.1) набуває вигляду (3.4).

$$F = \sum_{i=0}^n \frac{i^2}{t_i} \quad (3.4)$$

3.6 Особливості використання генетичного підходу навчання

Генетичні алгоритми є одним з найбільш потужних засобів багато параметричної оптимізації, здатних до ефективного пошуку глобального оптимуму цільової функції. Це легко пояснюється тим, що в умовах невизначеності генетичні алгоритми, мають найбільш високі шанси для досягнення необхідних результатів.

Одна з ключових особливостей даних алгоритмів полягає в паралельній обробці безлічі альтернативних рішень. При цьому пошук концентрується на найбільш перспективних з них, що каже про достатньо широкий спектр застосування. Модель здійснення навчання, характерна для генетичних алгоритмів, наведена на рис. 3.6.

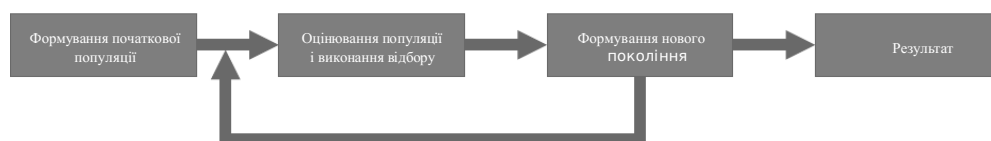


Рисунок 3.6 – Модель навчання за допомогою генетичного алгоритму

В даному конкретному випадку початкова популяція вибирається випадково, відповідно до наведеного вище, значення мостів лежать в проміжку від -1 до 1. Визначення кращих нащадків виконується за допомогою фітнес функції середовища, яка задана на формулою (3.4), чим більше



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

значення функції, тим більш успішний нащадок. Відповідний процес визначення кращих нащадків ще називають турнірною селекцією.

Навчання відбувається в результаті генерування нових популяції, це досягається тим, що до початкової популяції пропонується застосовувати варіанти простих генетичних принципів, які представлені турнірною селекцією, модифікованою функцією безпечної мутації, та рівномірний кросовер між двома особами.

Формування популяції відбувається в декілька етапів. Спочатку в результаті визначаються еліти, які отримуються в результаті проведення турнірної селекції. На наступному етапі створюються нащадки за допомогою модифікованої функції безпечної мутації. Після чого за допомогою функції рівномірного кросоверу створюються нові агенти нащадки. Відповідний процес формування популяції можна представити за допомогою лінійної моделі, наведеної на рис. 3.7.

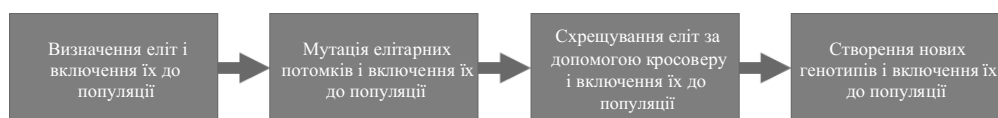


Рисунок 3.7 – Модель формування нової популяції

Безпечна мутація підмовляє собою достатньо поширену модифікацію функції мутації для нейронних мереж з функціями активації, визначеними на проміжку від -1 до 1. Відповідна функція мутації передбачає певну мутацію вхідних нейронів до певного мосту. Процеси функції безпечної мутації також представлені в вигляді алгоритмів на рис. 3.8 і рис. 3.9.

Модифікована функція безпечної мутації розуміє під собою дві вірогідні величини, які позначають вірогідність мутації вхідних мостів нейрону і вірогідність мутації певних мостів. Також функція безпечної мутації передбачає 4 різних способи мутації певних мостів, які трапляються з однаковою вірогідністю.

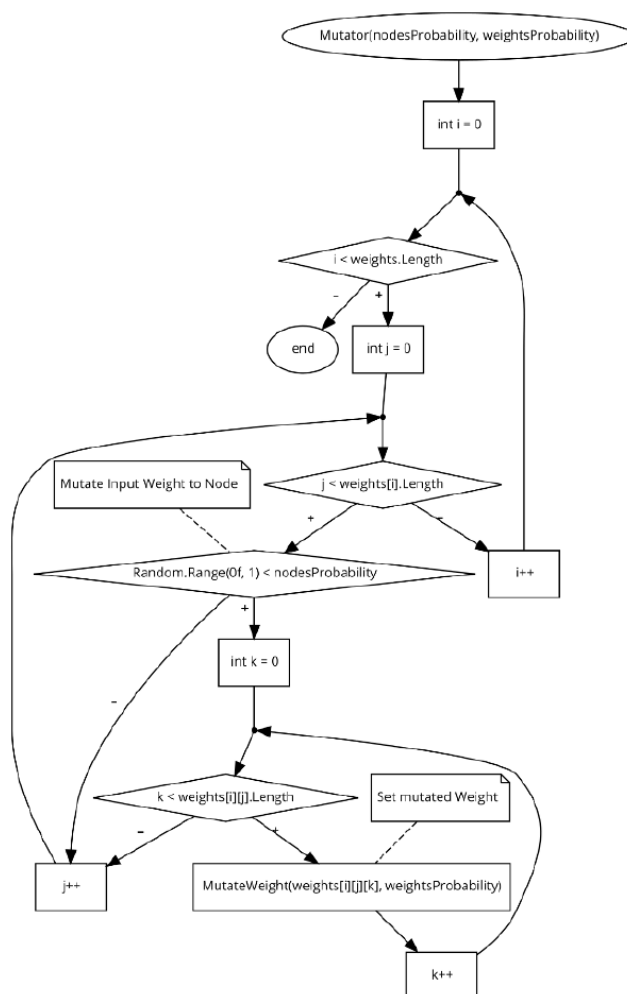


Рисунок 3.8 – Загальний алгоритм здійснення безпечної мутації

Перший варіант мутації мостів полягає в зміні знаку мосту, другий – генерації нового значення в проміжку від -0.5 до 0.5 , третій і четвертий – посиленні чи послабленні ваги за допомогою помноження на випадковий коефіцієнт у проміжках від 0 до 1 чи від 1 до 2 , відповідно.

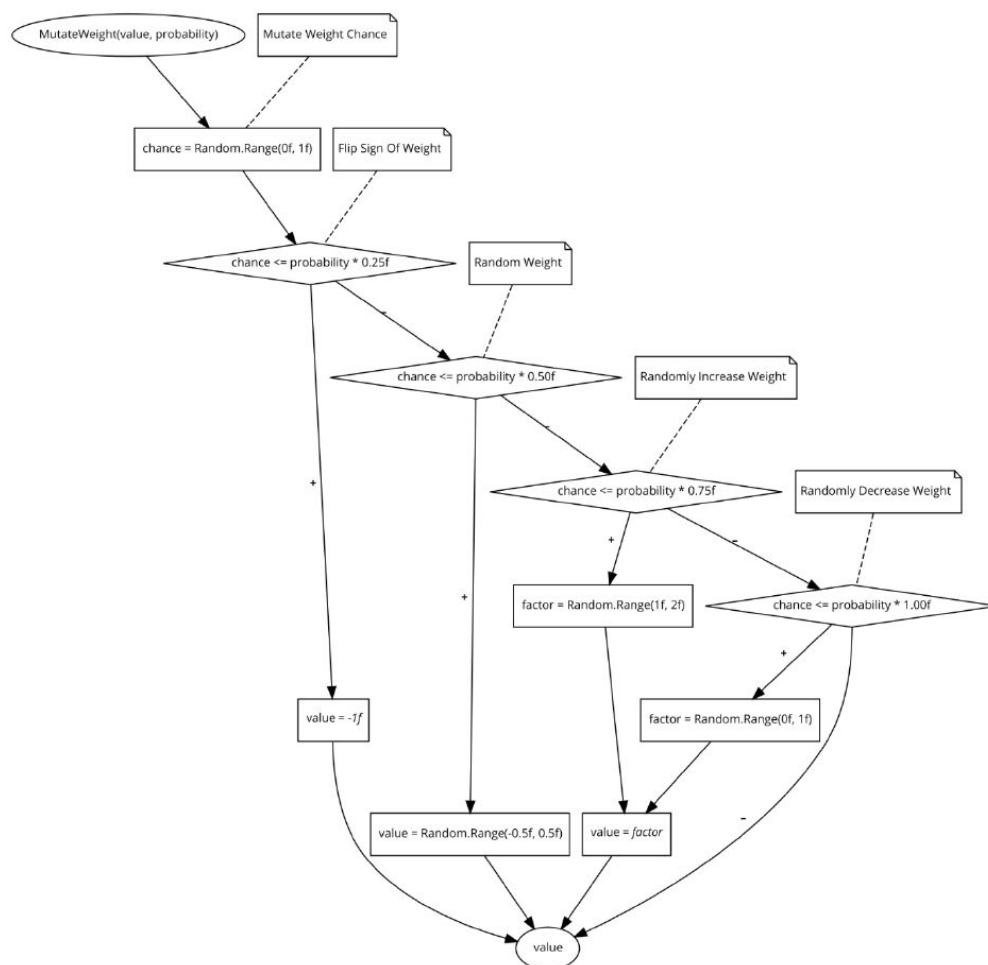


Рисунок 3.9 – Алгоритм здійснення безпечної мутації мосту

Завдяки оператору кросоверу, чи як його ще називають оператор схрещування, здійснюється генерація нового нащадку на базі поєднання генетичного матеріалу батьків у відповідності до якоїсь характеристичної риси. В випадку нейронних мереж це представляє собою створення нової нейронної мережі на базі поєднання ваги мостів батьківських мереж.

У даному випадку передбачається виконання рівномірного кросоверу між двома нейронними мережами з певною вірогідністю. Відповідний алгоритм наведено на рис. 3.6.

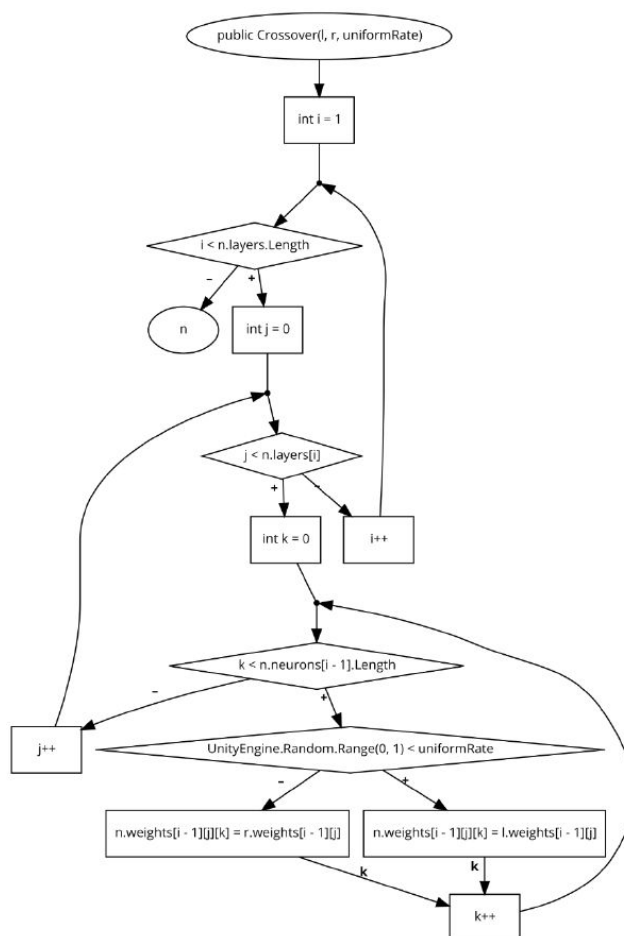


Рисунок 3.10 – Алгоритм рівномірного кросоверу нейронних мереж

3.7 Висновки за розділом

В розділі було визначено задачу ігрового агента автомобілю і розкрито методи і підходи для реалізації агента, який виконує описану задачу. Серед таких методів і підходів було обумовлено підходи до керування автомобілем, підходи до отримання даних і концепцію їх сумісного використання у контролері автомобілю. Також було розглянуто особливості використання підходів машинного навчання і інтелекту для вирішення цієї задачі.

РОЗДІЛ 4

ОБГРУНТУВАННЯ ВИМОГ ДО РОЗРОБЛЮВАННОГО СЕРЕДОВИЩА МОДЕЛЮВАННЯ І ОБРАНИХ ПРОГРАМНИХ ЗАСОБІВ

4.1 Формальний опис розроблюваної системи моделювання

У відповідності до зазначеного у попередніх пунктах можна зрозуміти, що середовище моделювання повинно реалізовувати гоночне середовище, мета якого здійснення тестування ігрового автомобілю, заснованого на методах машинного навчання та інтелекту, а також здійснювати тестування самого процесу навчання. У відповідності до цього можна сказати, що середовище повинно реалізовувати два режими поведінки.

Перший режим – стандартний режим чи як ще його можна назвати – режим тестового заїзду. Цей режим повинний дозволяти здійснювати заїзди, формуючи популяцію в ручну. Другий режим – навчальний, який обумовлює навчання ігрових агентів автомобілю, де можна задати певні параметри як ігрових агентів, так і параметри популяції і навчання.

В обох режимах середовище повинно передбачати перегляд інформації про певний автомобіль, який приймає участь в заїзді, а у випадку машини на базі нейронної мережі дозволяти її збереження. В режимі навчання повинна виводитись інформація, яка стосується процесу навчання, а також наглядно демонструватися процес навчання.

Також повинно бути передбачено налаштування параметрів тестового середовища, таких як трек заїзду і ігровий час. Окрім цього середовище повинно передбачати змінення режиму перегляду процесу навчання і змінення ракурсів перегляду. Окрім цього середовище повинно передбачати змінення режиму поведінки в будь-який час.

Окрім фізичних процесах ігрового рушія, що забезпечить додаткову складність до керування автомобілем в наслідок отримання шумового

насичення під час керування ігровою автівкою.

4.2 Визначення вимог до розроблюваної системи

У відповідності до формального опису для розроблюваного ігрового середовища можуть бути сформовані функціональні і не функціональні вимоги, які можна поділити на категорії згідно області реалізації, якої торкається бізнес вимога.

Відповідно до цього можна виділити наступні основні вимоги до програмних компонентів системи:

- скрипти середовища повинні бути розроблені як модульні за своїм характером і передбачити можливість розширення функціональності системи;
- у середовищі повинна бути реалізована контрольована фізична модель гоночного автомобілю, яка дозволить здійснювати керування автомобілем;
- фізична модель автомобілю повинна передбачати можливість встановлення ряду властивостей, які визначають потужність, максимальну швидкість та керованість автомобілем, тощо;
- звук двигуна і графічні ефекти (відображення обертів та передач) повинні ґрунтуватися на реальних показниках фізичної моделі автомобілю;
- у системі повинний бути реалізований контролер автомобілю, заснований на користувацьку ввіді, що дозволить здійснювати контроль автомобілем;
- у системі повинний бути реалізований контролер автомобілю, який виконує слідування у напрямку певної встановленої цілі у просторі;
- у системі повинна бути реалізована логіка точкового маршруту і переміщення точки у відповідності до маршруту з можливістю налаштування параметрів;



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

- у системі повинна бути реалізована модель кидання променів, яка повідомляє систему про дистанцію до об'єктів в радіусі максимальної дистанції променів;
- у системі повинний бути реалізований контролер автомобілю, що ґрунтується на нейронних мережах і здійснює рух відповідно до сигналів, отриманих з моделі кидання промінів.

Також система повинна відповідати наступним вимогам до інтересу користувача системи:

- середовище повинно бути зрозумілим у використанні і мати мінімально навантажений інтерфейс, який повинний відповідати загально прийнятим стандартам побудови інтересів;
- додаток повинний дозволяти здійснювати аналіз подій, що відбуваються на сцені і передивлятися інформацію про ігрові машини, що знаходяться на сцені;
- передбачити можливість контролю за ігровим часом у вигляді таймеру і можливість керування плином часу, такі як пауза, прискорення і уповільнення;
- інтерфейс програми повинний дозволити здійснювати вибір ігрового режиму і передбачати встановлення параметрів заданого режиму;
- під час навчання реалізувати візуальне інформування, щодо процесу змінення параметрів фітнес функції автівок у кожній інтеграції навчання;
- для автівок, що керуються нейронними мережами, під час перегляду інформації про автівку передбачити графічне відображення процесу обрахунку рішення нейронною мережею;
- передбачити можливість тимчасового збереження навчених автівок для їх подальшого застосування в ігровому режимі, а також передбачити можливість очищення відповідного списку.

4.3 Обґрунтування вибору середовища розробки

Фактично структура і засоби сучасних рушіїв заточені для того, що при створенні гри дозволяє абстрагуватися від опису команд низького рівня і інструкцій, а сконцентруватися на реалізації ігрового процесу. Відповідно до цього розробники відеоігор прямим чином взаємодіють тільки з частиною модулів ігрового рушія.

Під реалізацією ігрового процесу в широкому сенсі розуміється, як створення ігрового оточення і геймплею, так і реалізація ігрової логіки і штучного інтелекту. Ігрове оточення і геймплей зазвичай створюється за допомогою вбудованих до ігрового рушія редактору сцен і інтерфейсів, а реалізація ігрової логіки і штучного інтелекту – програмним чином.

Хоча ігрові рушії надають засоби і мають розширення для створення елементів ігрового інтерфейсу, але загальноприйнятою практикою є використання спеціалізованого програмного забезпечення, оскільки воно має більш зручний інтерес для задач редагування об'єктів і більший функціонал.

Фактично ігрова логіка формалізує в коді правила гри і ігрових взаємодій, які складають фактичний ігровий геймплей. Порівняно з штучним інтелектом ігрова логіка зазвичай є більш тривіальною. Тому для створення штучного інтелекту ігрові рушії мають вбудовану підсистему штучного інтелекту, яка включає ряд засобів, що можуть бути застосовані при реалізації штучного інтелекту.

Хоча ігрові движки мають досить багато спільного у загальних концепціях устрою і реалізації, але вони мають багато різного з точки зору використаних архітектурних рішень і конкретних засобів, оскільки кожен окремий ігровий рушій підумовлює різні архітектурні концепції та підходи. Це наділяє кожен ігровий рушій певними унікальними рисами. В результаті чого використання певного ігрового рушія приносить свої особливості і обмеження при проектуванні програмної архітектури гри.

В останні роки конкуренція на ринку ігрових рушіїв розгорнулась між



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

трьома основними суперниками, а саме Unity 3D, Unreal Engine, Cry Engine. Серед цих ігрових рушіїв неможливо сказати, який з них є абсолютно кращім оскільки кожен з рушіїв має свою галузь використання.

Ігровий рушій Unity 3D має широкий спектр можливостей, а також зручний і достатньо дружній інтерфейс. Підходить для розробки двомірних і тривимірних ігор невеликого розміру. Unreal Engine 4 представляє собою найпопулярніший ігровий рушій для створення фільмів і AAA-проектів. Має більш широкі графічні можливості порівняно з Unity. Ігровий рушій CryEngine на відміну від інших ігрових движків, націлений на створення ігор з максимально якісною графікою і позиціонуються, як рушій для розробки AAA-проектів.

Для виконання даної роботи є найбільш підходящим рушій Unity 3D, оскільки він заточений для розробки проектів невеликого розміру, а також на відмінно від інших розглянутих рушіїв має вбудований інструментарій, який призначений для підтримки машинного навчання.

4.4 Визначення апаратних та програмних вимог

Розроблювана програмна система не пред'являє специфічних вимог для систем, але є бажаним використовувати продуктивне обладнання для роботи з програмною системою. Оскільки здійснення навчання нейронних мереж вимагає ресурсів для обробки обрахунків усіх нейронних мереж у визначеній популяції. Окрім цього при великій кількості автомобілів під час здійснення тестування вимагаються достатньо великі обчислювальних ресурсів.

Відповідно до того, що тестове середовище розроблюється для роботи на персональному комп'ютері на базі Windows, то є актуальними наступні мінімальні вимоги ігрового рушія Unity: операційна система Windows SP1+ та новіше, графічна карта з підтримкою DirectX 10 (модель шейдерів 4.0), підтримка процесором програмних інструкцій SSE2.

Оскільки для реалізації програмного продукту використовується мова



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

програмування C# 7.0, а відповідна версія платформи .NET 4.7, а вона в свою чергу пред'являє наступні мінімальні системні вимоги: процесор 1 ГГц, оперативна пам'ять 512 МБ, дисковий простір 4.5 ГБ, операційна система Windows 10 Anniversary Update та новіше.

Серед мінімальні системних вимог операційної системи Windows 10 Anniversary Update можна виділити наступні: процесор IA-32 чи x86-64 з частотою 1 ГГц та вище з підтримкою PAE, NX та SSE2; оперативна пам'ять для IA-32 – 1 ГБ, для x86-64 – 2 ГБ; графічна карта з підтримкою DirectX 9, WDDM 1.0 або вище.

Враховуючи мінімальні системні вимоги суміжного програмного забезпечення, то можна висунути до системи наступні мінімальні системні вимоги: операційна система Windows 10 Anniversary Update і новіше; оперативна пам'ять 2 ГБ і більше, графічна карта з підтримкою DirectX 10 і новіше. Враховуючи зазначені мінімальні системні вимоги можна висунути наступні рекомендовані системні вимоги: операційна система Windows 10 Anniversary Update і **нзовіше**; частота процесор 1.5 ГГц і вище; оперативна пам'ять 6 ГБ і більше, графічна карта з підтримкою DirectX 10 і новіше з 512 МБ відео пам'яті і більше.

4.5 Висновки за розділом

В розділі було висунуто формальний опис системи моделювання і її особливості, визначено основні бізнес вимоги до системи, обрано ігровий відповідний рушій, а також було оцінено мінімальні і рекомендовані системні вимоги до системи на базі підходу, що відштовхується від системних вимог суміжного програмного забезпечення.



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

РОЗДІЛ 5

ПРОЕКТУВАННЯ І ПРОГРАМНА РЕАЛІЗАЦІЯ СЕРЕДОВИЩА МОДЕЛЮВАННЯ ГОНОЧНОГО ПРОЦЕСУ

5.1 Загальний порядок проектування і виконання розробки

На початку розробки загальне бачення системи не проектувалося, але було виділено перелік задач, які поступово реалізувалися. Під час розробки до відповідного переліку задач за необхідністю вносились зміни. Відсутність необхідності проектування загального бачення системи в такому випадку можна пояснити тим, що розроблювана система складається з незалежних компонентів, які реалізують певну ігрову логіку.

З початку розробки реалізовувались і тестувались ігрові механіки, пов'язані з ігровими автомобілями. Коли реалізовувалась механіка ігрового автомобілю, заснованого на нейронних мережах, було реалізовано програмну логіку нейронної мережі і її графічного відображення.

Після реалізації логіки ігрових автомобілів було реалізовано програмну логіку, яка відповідає за генерацію популяцій автомобілів. Окрім генераторів популяцій на цьому етапі було реалізовано зберігання параметрів автомобілів на базі нейронних мереж.

На наступному етапі було створено моделі ігрових треків і реалізовано користувацький інтерфейс. Під час реалізації користувацького інтерфейсу було створено компоненти вибору камери, треку, роботи з ігровим часом, відображення графіків.

5.2 Розробка агента, реалізуючого логіку нейронної мережі

Взагалі існує велика кількість реалізацій систем програмної логіки нейронних мереж. У випадку ігрового рушія Unity вони, як правило,



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

зовнішні. Зовнішня логіка отримує від ігрового рушія певні сигнали, а усі обрахунки нейронної мережі в середині ігрового рушія є чорною скринькою з тієї точки зору, що отримання результатів проміжних обрахунків не є можливим. Оскільки необхідна наглядна демонстрація процесу навчання ігрової системи, то відповідна така логіка повинна бути реалізована в середині ігрового рушія.

Логіка обчислення нейронної мережі, питання генерації ваги мостів, питання відображення нейронної мережі реалізуються трьома окремими скриптами NNAgent, NNMath і NNView, відповідно. Скрипти NNAgent і NNMath повністю забезпечують логіку функціонування нейронної мережі. Під час виконання обчислень скрипт NNAgent зберігає результуючі значення нейронів, для того щоб запобігти виконанню зайвих обчислень під час відображення за допомогою скрипту NNView.

Скрипт NNView виконаний в тенденціях і підходах характерних до організації елементів користувацького інтерфейсу Unity. Базова ієрархія компоненту інтерфейсу наведена на рис. 5.1, а приклад відображення нейронної мережі наведено на рис. 5.2. Відображення складається з шаблонних компонентів, які лежать в відповідному місці в ієрархії елемента.

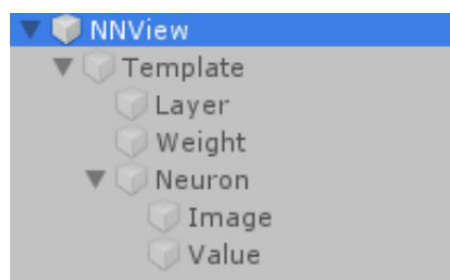


Рисунок 5.1 – Ієрархія компоненту NNView

Для відображення штучної нейронної мережі використовуються шаблони компоненти слою, нейрону і мосту. Основний компонент представляє собою горизонтальну групу з автоматичним розміщенням

елементів, компонент слою – вертикальну групу також з автоматичним розміщенням компонентів, нейрон представляє собою компонент слою, а мости розміщуються до нейронів, з яких вони виходять.

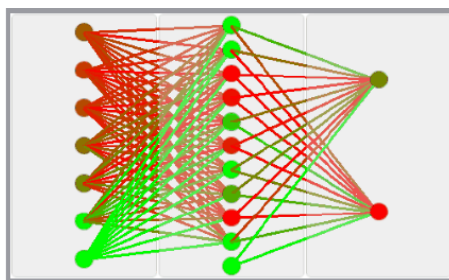


Рисунок 5.2 – Відображення штучної мережі на базі скрипту NNView

5.3 Формування загальної файлової структури проекту

Головним каталогом проекту Unity є каталог Assets і він повинний включати всі активи, які використовуються проектом. Вирішальний аспект формування файлової структури (рис. 5.3) полягає в тому, що матеріали проекту було умовно поділено на дві глобальні категорії, які було вирішено розмістити в окремі підкаталоги у каталозі Assets.

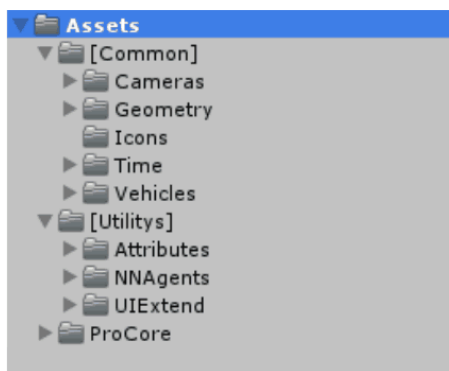


Рисунок 5.3 – Файлова структура проекту

Перша категорія представляє собою код реалізації ігрової логіки

самого проекту, відповідний код розміщується у підкаталог [Common], а друга категорія представляє зовнішні для логіки проекту скрипти і вони розміщується у підкаталог [Utilities]. Також у каталозі Asset присутній підкаталог ProCore, у якому знаходяться матеріали деяких пакетів системи і використовуваних активів, що розширюють можливості редактору Unity.

Структура розглянутих підкаталогів формувалась в подальшому процесі розробки за необхідністю розмежування певних компонентів системи за їх призначенням чи іншими ознаками.

5.4 Реалізація логіки керування ігрових автомобілів

Відповідно до вимог що пред'являються до реалізації ігрових агентів було визначено, що ігрові агенти зі стандартних ігрових активів Unity відповідають зазначеним вимогам з певною доробкою і розширеннями. Відповідно до цього їх було взято за основу для реалізації власних ігрових агентів. Відповідні матеріали для реалізації було розміщено до підкаталогу Vehicles каталогу [Common] (рис. 5.4).

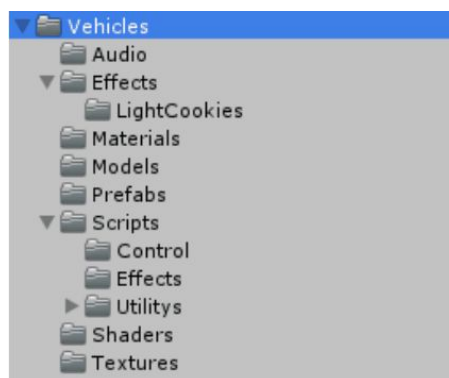


Рисунок 5.4 – Структура підкаталогу Vehicles каталогу [Common]

Стандартна структура ігрового об'єкту автівки (рис. 5.5) включає в себе наступні елементи: колайдери, колеса, джерела світла, систему часток,

допоміжні об'єкти і об'єкти моделі автомобілю.

Програмна логіка в свою чергу поділяється на логіку керування автомобілем, логіку реалізації графічних ефектів і додаткові утиліти, що розширюють функціональні можливості.

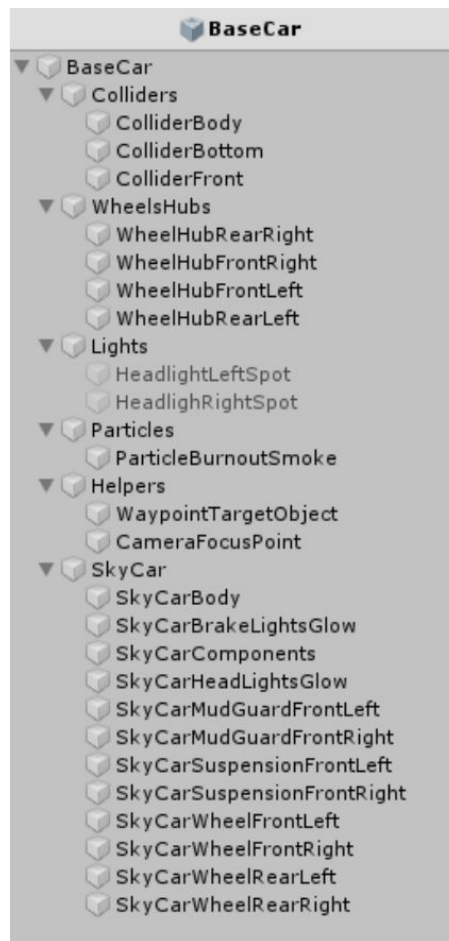


Рисунок 5.5 – Структура ігрового об'єкту автівки

Програмна логіка, яка відповідає за фізичний рух автомобілю, додана в вигляді компоненту до головного об'єкту автівки. Вона реалізується за допомогою скрипту Car Control System (рис. 5.6). Також цей скрипт відповідає за підрахунок часу життя авто і розрахунок фітнес функції.

Відповідний скрипт має ряд властивостей, серед яких показники на колеса автомобілю, потужність автомобілю, максимальну швидкість, керованість автомобілем, тощо. Відповідні властивості можуть бути знайдені у відповідному вікні редагування компоненту.

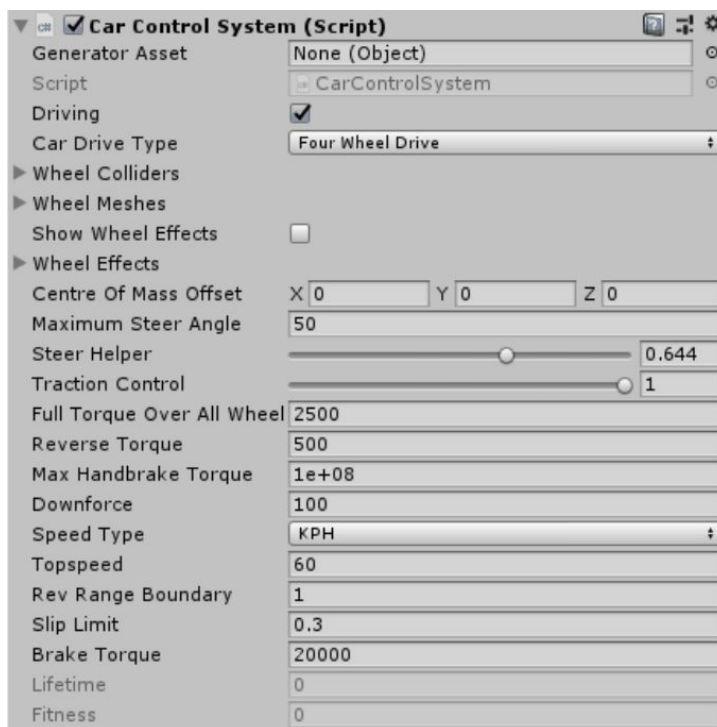


Рисунок 5.6 – Інтерфейс скрипту Car Control System в редакторі

Хочеться зазначити, що скрипт призначений для того, щоб розважати і надавати почуття простоти керування, а не намагатися бути реалістичним моделюванням керування автомобілем, хоча скрипт реалізує рух ігрової автівки на базі фізичних процесів ігрового рушія, що не робить керування машини занадто простим.

За керування автівкою відповідає функція Move скрипту Car Control System. Вона має наступний інтерфейс: `public void Move (float steering, float accel, float footbrake, float handbrake)`. Де steering – коефіцієнт повороту автівки в відповідності до встановленого максимального куту обертання,

accel – коефіцієнт прискорення автівки, footbrake – коефіцієнт ногоного гальмування автівки, handbrake – коефіцієнт ручного гальмування автівки.

Параметр steering може приймати діапазон значень від -1 до 1, accel – від 0 до 1, footbrake – від -1 до 0, handbrake – від -1 до 0. Фактично коефіцієнти accel і footbrake можуть бути представлені одним числом в діапазоні від -1 до 1, що дозволяє отримувати параметри коефіцієнту з однієї осі керування і в той час залишає інтерфейс функції зручним для автоматизованих контролерів, які ґрунтуються на скриптовій лозинці.

Інша група скриптів приносить чисто істеричні ефекти. До цих скриптів відносяться скрипти Car Audio, Car Brake Light, Car Mudguard, Car Suspension, Car Wheel Effects, Car Skid Trail. Кожен з скриптів реалізує певну істеричну функцію автомобілю.

Скрипт Car Audio (рис. 5.7) читає поточні властивості автомобіля і відповідно грає звуки. Відповідний скрипт було дезактивовано, оскільки він буде створювати непотрібний сторонній шум у даній системі. Скрипт розміщуються на головному об'єкті автомобіля.

Звук двигуна може бути простим єдиним затиском, який є петлею і розтягнутим, або це може бути перехресна суміш з чотирьох затискачів, які представляють тембр двигуна при різних ступенях обертання та дрoселя.



Рисунок 5.7 – Інтерфейс скрипту Car Audio в редакторі

Скрипт Car Brake Light (рис. 5.8) зчитує параметри руху автомобілю і вмикає задні вогні автомобілю при здійсненні гальмування. Скрипт розміщується на задніх вогнях автомобілю.

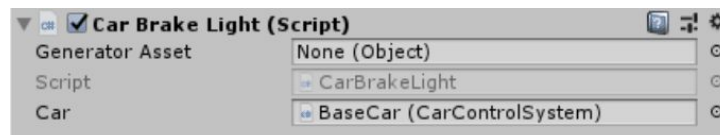


Рисунок 5.8 – Інтерфейс скрипту Car Brake Light в редакторі

Скрипт Car Mudguard (рис. 5.9) призначений для обертання засланок збирачів грязюки на передніх колесах автівки відповідно до параметрів руху автомобілю. Скрипт розміщується на грязьових засланках.

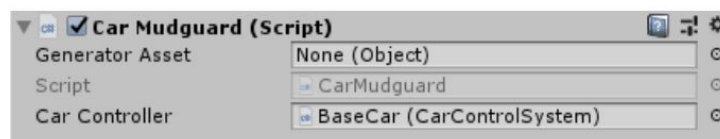


Рисунок 5.9 – Інтерфейс скрипту Car Mudguard в редакторі

Скрипт Car Suspension (рис. 5.10) керує концентратором підвіски, щоб змусити його рухатися відповідно до руху автомобілю й оброблювати ударні навантаження, які отримує автомобіль під час руху. Скрипт розміщується на колесах автомобілю.

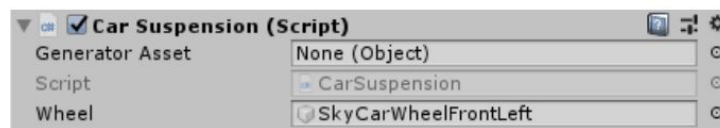


Рисунок 5.10 – Інтерфейс скрипту Car Suspension в редакторі

Скрипт Car Skid Trail (рис. 5.11) відповідає за програмну логіку слідів, які залишають колеса автомобіля, а саме зникненням з плином часу. Скрипт



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

розміщується на екземпляр слідів автомобіля.

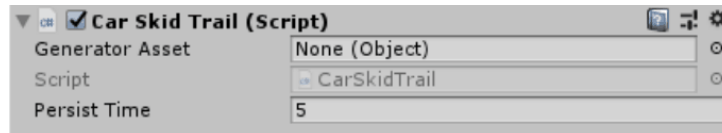


Рисунок 5.11 – Інтерфейс скрипту Car Skid Trail в редакторі

Скрипт Car Wheel Effects (рис. 5.12) відповідає за програмну логіку залишення слідів автомобіля. В даному випадку відображення слідів автомобіля вимкнено, оскільки таке відображення вимагає велику кількість ресурсів. Скрипт розміщується на колайдер колеса автомобіля.

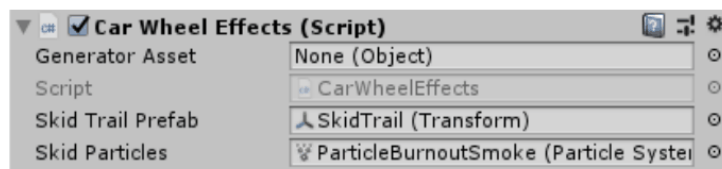


Рисунок 5.12 – Інтерфейс скрипту Car Wheel Effects в редакторі

5.5 Реалізація контролерів ігрових автомобілів

Для забезпечення поліморфізму при роботі з контролерами автомобілю було створено абстрактний скрипт контролеру Car Abstract Control, від якого успадковуються інші контролери. Діаграма, яка демонструє залежність між контролерами, наведена на рис. 5.13.

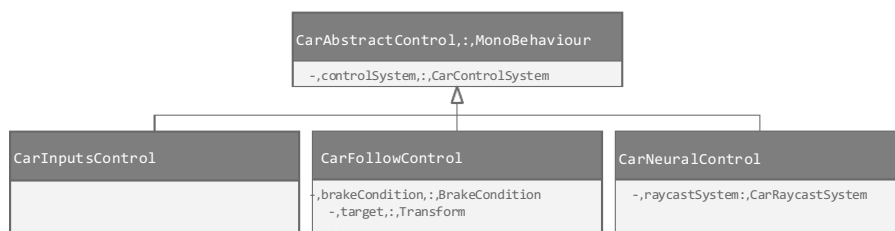


Рисунок 5.13 – Діаграма залежності ігрових контролерів

Першим і найбільш простим контролером є Car Inputs Control (рис. 5.14), який заснований на користувацькому ввіді. Відповідний контролер оброблює користувацький ввід за допомогою стандартного інтересу вводу Input, з якого отримуються ввід користувача за вертикальною та горизонтальною осями і передається до функції Move скрипту Car Control System.

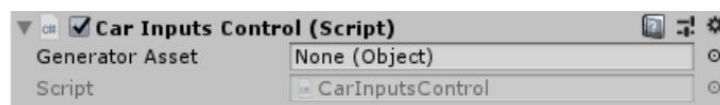


Рисунок 5.14 – Інтерфейс скрипту Car Inputs Control в редакторі

Другим і найбільш складним з усіх контролерів є Car Follow Control (рис. 5.15). Відповідний контролер реалізує переміщення автівки до певної встановленої цілі. Має змінні параметри руху, завдяки редагуванню яких можна добиватися певного стилю руху автівки. Обчислені на базі параметрів руху сигнали керування передаються до контролеру автівки.

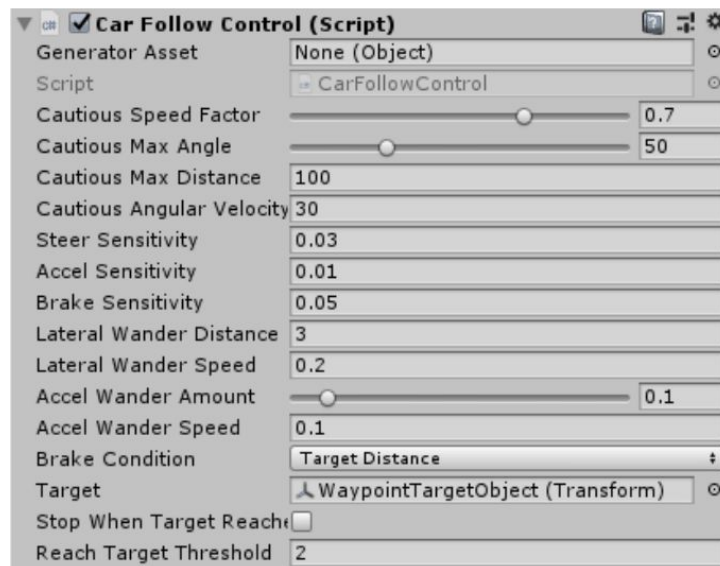


Рисунок 5.15 – Інтерфейс скрипту Car Follow Control в редакторі



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

Третім реалізованим контролером є Car Neural Control (рис. 5.16). Він заснований на базі механіки нейронних мереж. В якості входів до нейронної мережі подаються дані обчислюванні скриптом Car Raycast System, який є обов'язковим компонентом автівки даного типу. На виході з нейронної мережі отримуються відповіді, які можна порівняти з сигналами осей, отриманих від користувацького вводу.

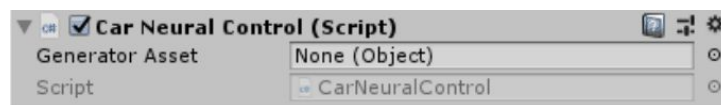


Рисунок 5.16 – Інтерфейс скрипту Car Neural Control в редакторі

5.6 Реалізація додаткових механік ігрового автомобілю

Скрипт Car Raycast System (рис. 5.17) відповідний за отримання сигналів від сенсорів про перешкоди в рамках зазначеного радіусу. Кількість сенсорів регулюється за допомогою параметрів скрипту. Окремо регулюється кількість передніх сенсорів, які охоплюють передню частину включно з боками, а також задніх сенсорів, які охоплюють задню частину без боків. Приклад роботи відповідного скрипту наведено на рис. 5.18.

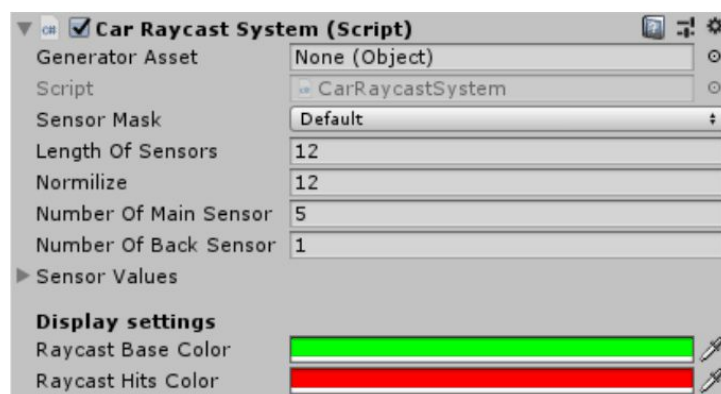


Рисунок 5.17 – Інтерфейс скрипту Car Raycast System в редакторі

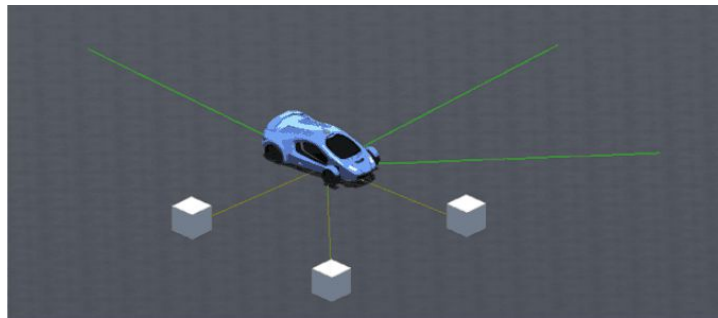


Рисунок 5.18 – Приклад роботи скрипту Car Raycast System

Скрипти Waypoint Circuit (рис. 5.19) і Waypoint Navigator (рис. 5.20) реалізують механіку пересування за маршрутом. Перший скрипт відповідає за формування маршруту на базі покладених в нього об'єктів, представляючи їх координати за точки маршруту. Скрипт Waypoint Navigator відповідає за пересування цільової точки за маршрутом.

Використовуючи Waypoint Navigator з автівкою, на базі Car Follow Control реалізується пересування автівки за певним маршрутом, заданим за допомогою Waypoint Circuit. Фактично це представляє собою реалізацію компонентного підходу до організації програмної логіки. Приклад відповідної моделі наведено на рис. 5.21.

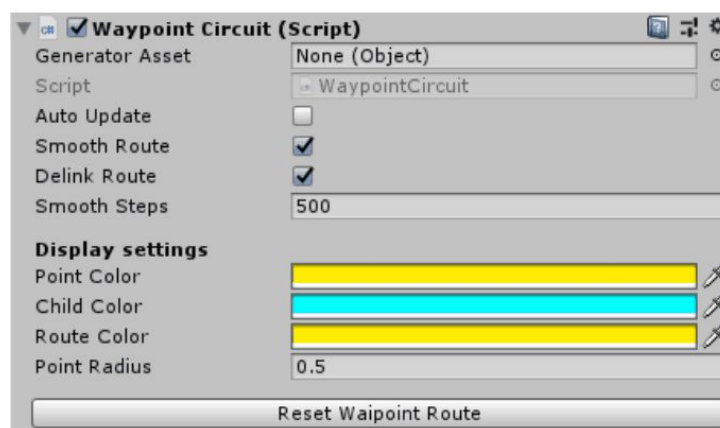


Рисунок 5.19 – Інтерфейс скрипту Waypoint Circuit в редакторі

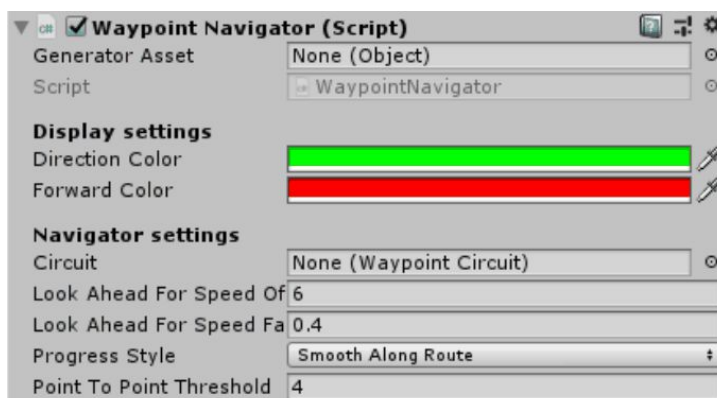


Рисунок 5.20 – Інтерфейс скрипту Waypoint Navigator в редакторі



Рисунок 5.21 – Застосування механіки пересування за маршрутом

Ігрова механіка контрольних точок реалізується за допомогою скриптів Checkpoint Circuit (рис. 5.22) і Checkpoint Agent (рис. 5.23). Відповідна ігрова механіка полягає в тому, що коли коллайдер автомобілю входить до тригер коллайдеру відповідної системи контрольних точок відбувається певний набір дії, який встановлюється в інспекторі. Також механіка контрольних точок дозволяє відслідковувати прогрес їх проходження і може виконувати дію, задану в інспекторі у випадку, якщо немає прогресу проходження контрольних точок протягом певного часу.

Відповідні скрипти в ігровій системі використовуються для проведення оцінки успішності проходження маршруту. Коли ігровий агент активує наступну контрольну точку, надсилається сигнал фізичному

контролеру ігрової автівки для збільшення фітнес функції, яка виконує оцінку успішності проходження маршруту. А також відповідний скрипт деактивує програмну логіку автівки, якщо вона не має поліпшень.

;

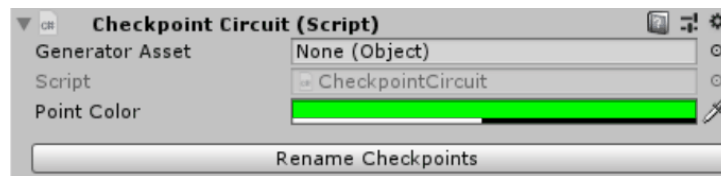


Рисунок 5.22 – Інтерфейс скрипту Checkpoint Circuit в редакторі

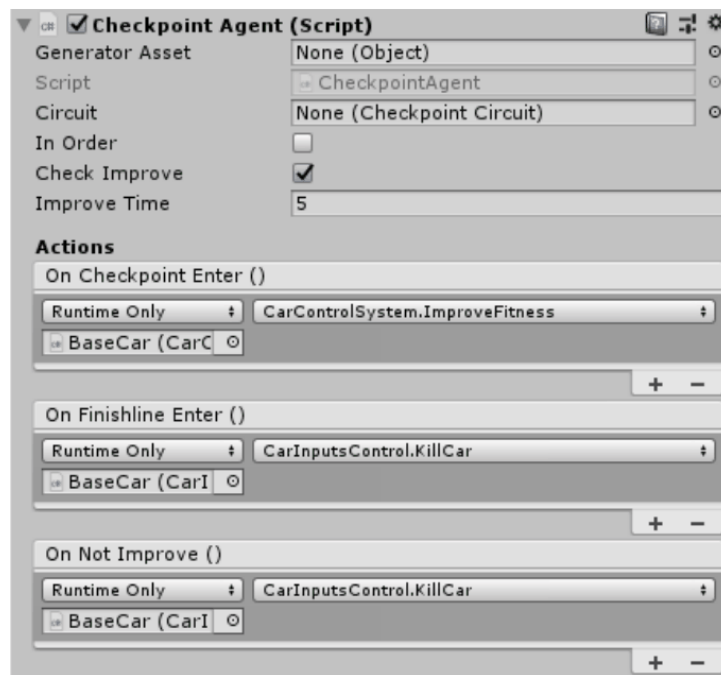


Рисунок 5.23 – Інтерфейс скрипту Checkpoint Agent в редакторі

5.7 Реалізація підсистеми генерації популяцій ігрових агентів

Для запобігання повторюваності програмного коду було створено абстрактний клас генератору популяцій, у якому реалізовано спільну програмну логіку. Діаграма, що показує залежності між класами, які

реалізують генерацію популяцій, наведена на рис. 5.24.

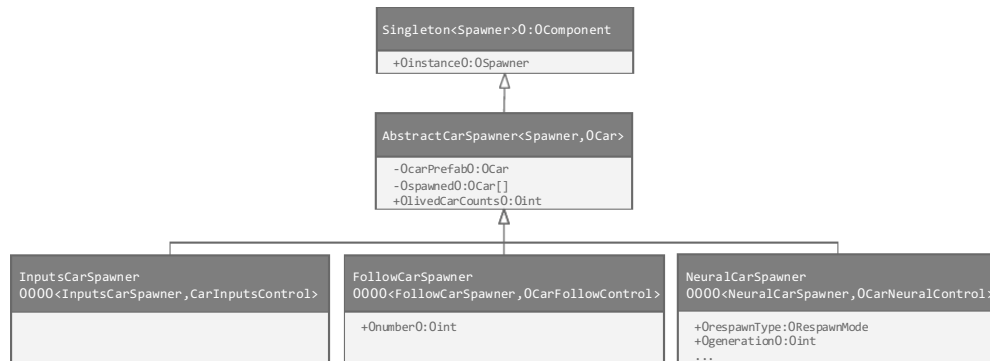


Рисунок 5.24 – Діаграма залежності генераторів популяцій

Скрипти Input Car Spawner (рис. 5.25), Follow Car Spawner (рис. 5.26), Neural Car Spawner (рис. 5.27), відповідно, реалізують логіку популяції автівок з контролерами Car Inputs Controller, Car Follow Controller і Car Neural Controller. Відповідна логіка полягає в доданні нової автівки до сцени і налаштуванні параметрів доданої до сцени автівки.

Найбільш простими є Input Car Spawner і Follow Car Spawner, перший з них дозволяє згенерувати лише одну автівку заданого типу, а інший генерує вказану кількість автівок. Найбільш складним є Neural Car Spawner, він підтримує два варіанти генерації автівок – стандартний і навчальний.

Стандартний завантажує збережені автівки в статичному класі Neural Car Store. Другий навчальний, який послідовно генерує популяції автівок з позначеними параметрами відповідно до генетичного алгоритму, особливості якого були описані раніше.

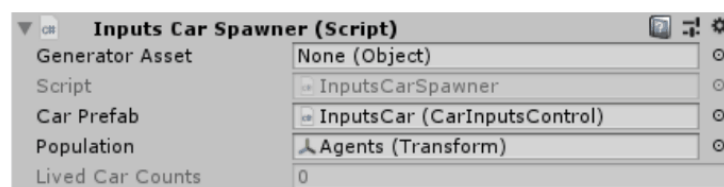


Рисунок 5.25 – Інтерфейс скрипту Input Car Spawner в редакторі

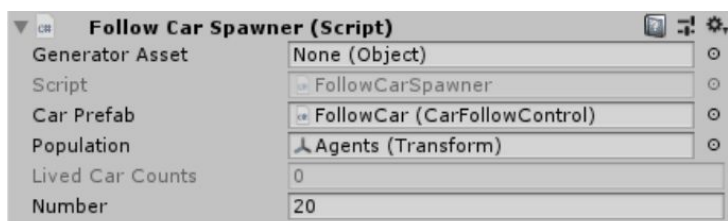


Рисунок 5.26 – Інтерфейс скрипту Follow Car Spawner в редакторі

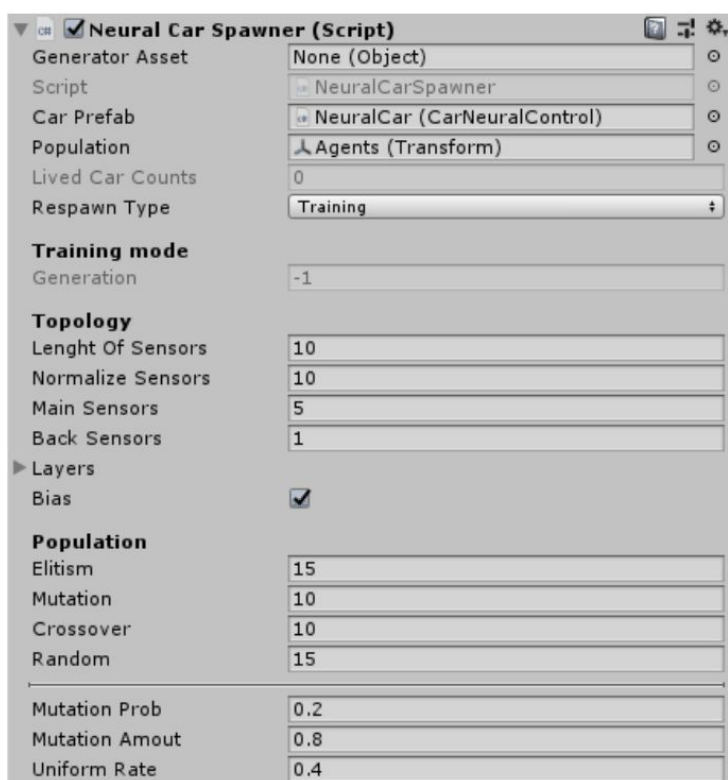


Рисунок 5.27 – Інтерфейс скрипту Neural Car Spawner в редакторі

5.8 Реалізація інформаційного елементу інтерфейсу автомобіля

Програмна логіка елементу відображення інформації реалізується за допомогою скрипту UICarManager. Ієрархія відповідного компоненту наведена на рис. 5.30, а інтерфейс компоненту наведено на рис. 5.31. Компонент співпрацює з менеджером популяції генетичних автомобілів, а

також дозволяє зберігати параметри автомобілів на базі нейронних мереж для того, щоб потім використати збережені машини у стандартному режимі.



Рисунок 5.28 – Ієрархія компоненту відображення інформації про автомобіль

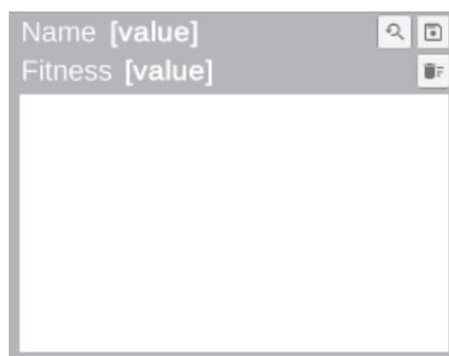


Рисунок 5.29 – Інтерфейс елемента відображення інформації про автомобіль

5.9 Реалізація інтерфейсу користувача програмної системи

5.10 Визначення базової концепції побудови інтерфейсу

Користувацький інтерфейс реалізовувався з логіки науковості системи. Відповідно до цього він повинен бути максимально інформативним. Для забезпечення відповідної інформативності повинні постійно виводитись параметри функціонування системи. Для того, щоб не перекривати область відображення було вирішено розмістити елементи інтерфейсу з лівої частини.



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

Оскільки інтерфейс займає ліву частину екрану і для запобігання зсуву камери від нульової осі було змінено параметр Viewport Rect камери, який змінює позицію відображення картинки камери. Відповідно до цього у батьківського елемента інтерфейсу було змінено якорі для того, щоб інтерфейс виводився з лівої частини області відображення.

Для запобігання проблем з відображенням при різному розширенні екрану були вирішено розроблювати інтерфейс тільки для розширень екрану з відношенням сторін 16:9, для чого було виставлено відповідне обмеження в параметрах проекту, а для відображення елементів інтерфейсу в фіксованому розмірі було встановлено фіксоване розширення завдяки зміні параметру UI Scale Mode на Scale With Screen Size у компоненті Canvas Scaler.

Головне меню (рис. 5.30), яке дозволяє обрати параметри генерації популяції і переглядати поточну інформацію ігрової системи, було реалізовано в вигляді системи вкладок з подіями, призначеними через інспектор Unity.

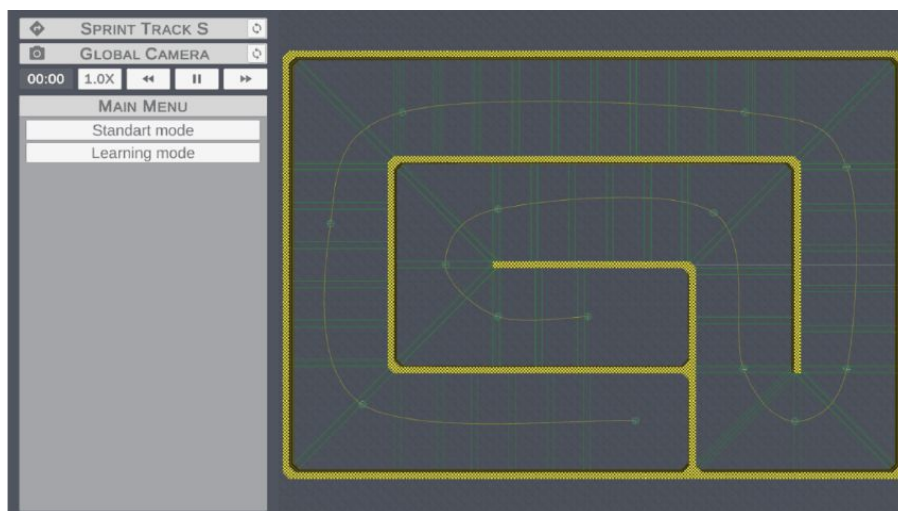


Рисунок 5.30 – Головний екран інтерфейсу користувача

5.11 Реалізація деяких компонентів інтерфейсу меню

В вигляді окремих компонентів було реалізовано меню вибору треку (рис. 5.31), меню вибору камери (рис. 5.32), меню роботи з ігровим часом (рис. 5.33), область відображення графіку (рис. 5.34). Логіка забезпечення робото здатності відповідних компонентів забезпечується скриптами Cameras Manager, Geometry Manager, Time Manager, UIChart.



Рисунок 5.31 – Меню вибору треку



Рисунок 5.32 – Меню вибору камери



Рисунок 5.33 – Меню керування часом

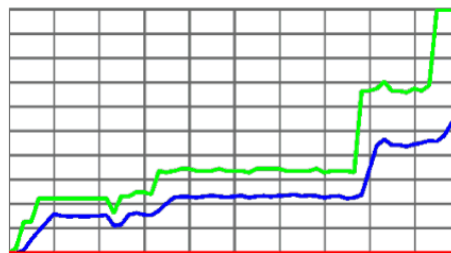


Рисунок 5.34 – Область відображення графіків

Елементи здійснення вибору треку і камери мають однакову ієрархічну структуру, а саме вони включають до себе елемент картинки, тексту та кнопку. Здійснення зміни активної камери чи треку відбувається за допомогою натискання кнопки з відповідною іконкою, після натиснення якої відбудеться дія зміни треку або камери.

Меню керування часу складається з одного текстового поля, в якому виводиться ігровий час, що минув від початку симуляції і чотирьох кнопок, які зліва на право виконують наступні дії, скинути швидкість до базової, задовільнити ігровий час, зупинити чи запустити ігровий час, пришвидшити ігровий час.

Область відображення графіків включає в себе 4 елементи – один UIGrid і чотири UILine. Значення точок передаються скрипту контролеру, після чого він визначає необхідний масштаб для відображення графіку, виконує перетворення точок і встановлює їх, як точки прямої.

5.12 Висновки за розділом

У розділі було описано загальний процес створення системи і наводиться інформація про деякі особливості реалізації системи. Як вже зазначалося, розроблена система носить за собою модульний характер і її компоненти можуть бути перенесені в інші ігрові програмні системи на базі Unity. Хоча наводиться опис базових складових і деяких аспектів реалізації, але опущені деякі деталі реалізації, які не є важливими для викладення і розуміння процесів функціонування системи.



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

РОЗДІЛ 6

ПРОГРАМНЕ МОДЕЛЮВАННЯ І АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

6.1 Виконання навчання агентів автомобілів на базі нейронних мереж

Під час проведення експериментів (рис. 6.1 – рис. 6.4) було встановлено залежність параметрів, з якими виконується навчання до швидкості навчання агента автомобілю. Під час навчання на графічному відображенні фітнес функцій можна побачити стрибки фітнес функції, це пояснюється природою реалізації ігрового агента автомобілю. А саме тим, що рух автомобілю забезпечується на базі фізичного рушія, що здійснює шумове насичення руху.

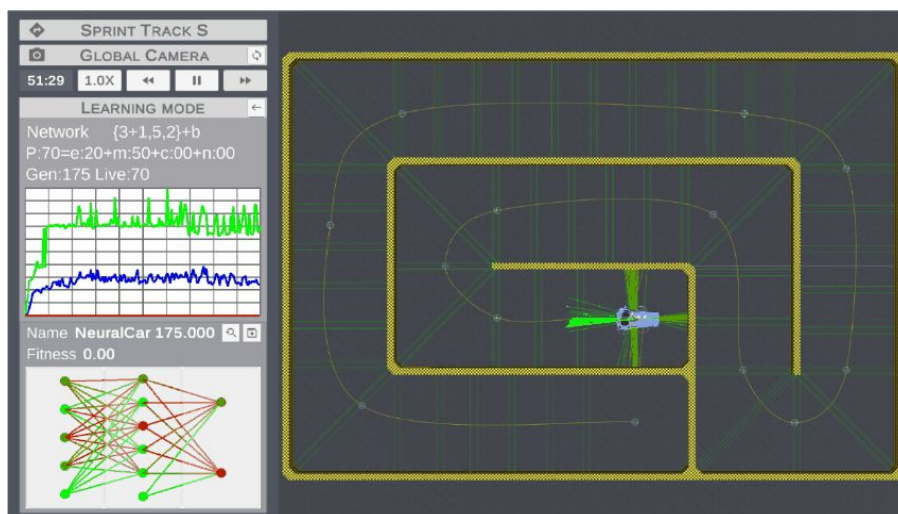


Рисунок 6.1 – Інтерфейс користувача під час симуляції навчальної популяції з недостатнім розміром нейронної мережі

Генерування нових особин в популяції не приносять неякої фактичної користі, тому що згенеровані особини поступаються особинам, які отримані в результаті функції кросоверу і мутації. Тому найбільш оптимальним виходом

є не використовувати їх для популяції.

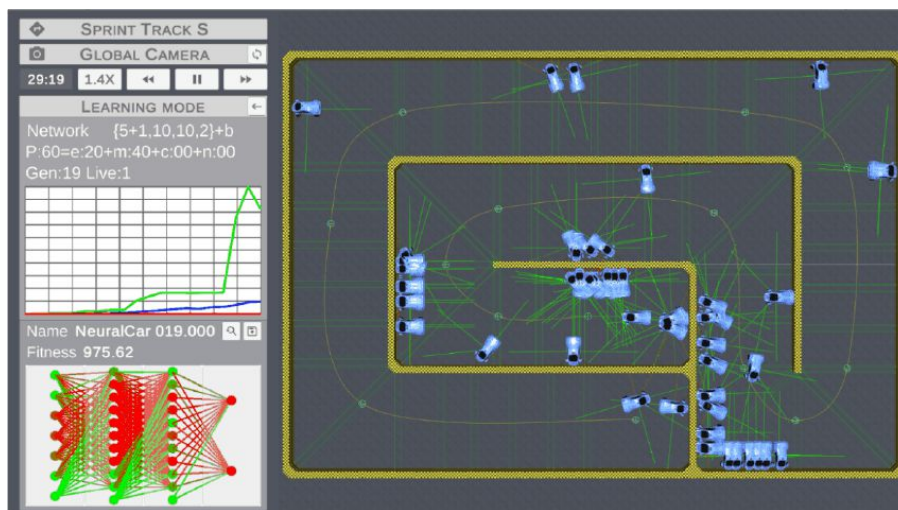


Рисунок 6.2 – Інтерфейс користувача під час симуляції навчальної популяції з одним з результативних наборів параметрів

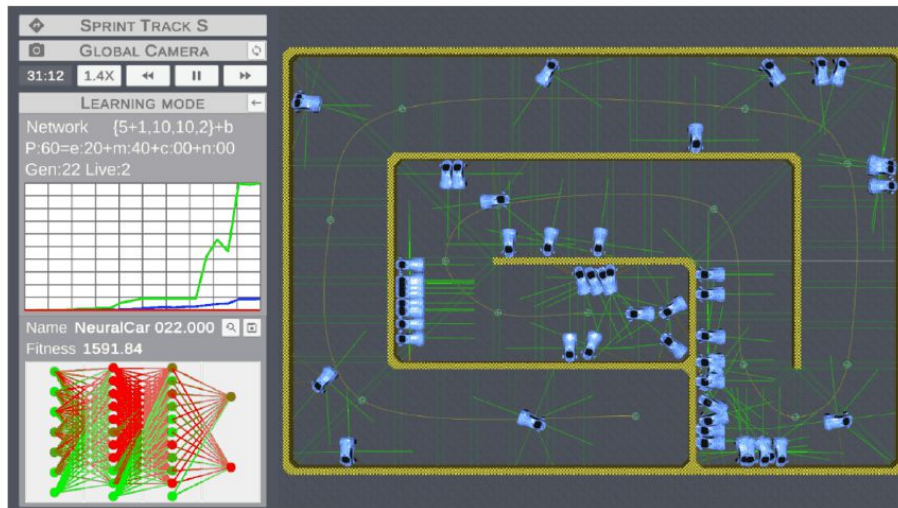


Рисунок 6.3 – Інтерфейс користувача під час симуляції навчальної популяції з результативним набором параметрів і позитивним проходженням треку

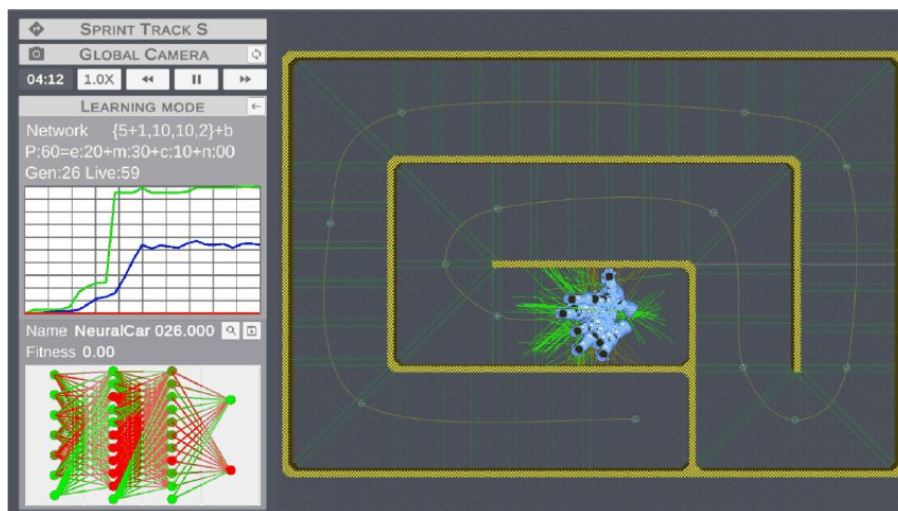


Рисунок 6.4 – Інтерфейс користувача під час симуляції навчальної популяції з не результативним набором параметрів

З інших операторів генерації нових нащадків в даній реалізації більш результативним показує себе оператор мутації. Однією з теоретичних причин цього може бути те, що схрещення за допомогою функції кросоверу відбувається між найбільш результативною особиною з близькими за результативністю особинами, але вони мають дуже сильну різницю у вагах мостів. В результаті чого відбувається не наближення до результату, а віддалення від нього.

Орім цього, на швидкість навчання впливають кількість сенсорів автівки, кількість шарів нейронної мережі, наявність упередженого нейрону. Практичним чином було підібрано параметри навчання, які демонструють себе достатньо результативними з точки зору швидкості навчання і адекватності поведінки автівки при будь-якій складності треку. Відповідні параметри наведені на рис. 6.5.

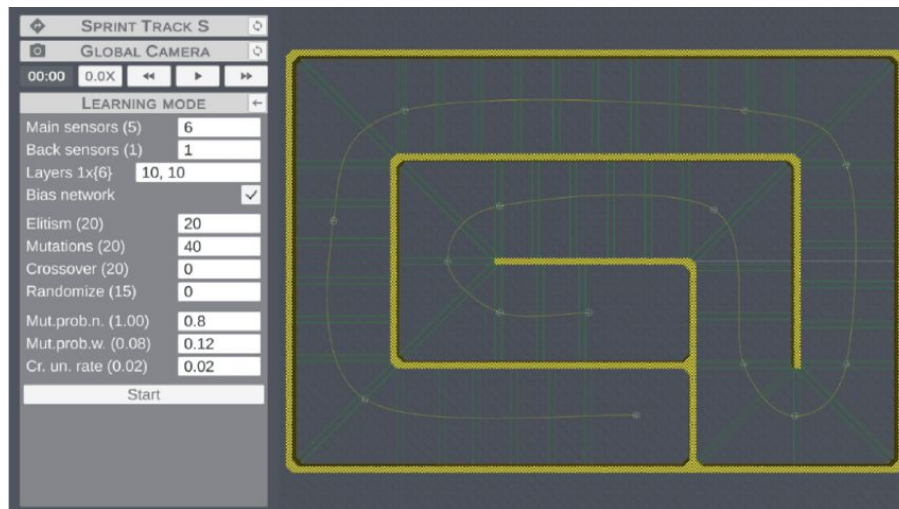


Рисунок 6.5 – Екрана форма з прикладом ініціалізації параметрів навчальної популяції

Змінення кількості шарів і кількості нейронів достатньо сильно впливають на процес навчання і поведінки автівки. Зменшення і збільшення кількості шарів відповідним чином відзначається і на часі навчання автівки і на якість отриманого результату. Так з точки зору навчання автівки з одним прихованим шаром нейронної мережі навчаються більш швидко, ніж автівки з більшою кількістю шарів.

В загальному випадку машинки, навчені на більш складних треках, показують себе однаково результативною на треках з відповідною і меншою складністю, а машинки, навчені на треках з меншою складністю, можуть не доїжджати до кінця більш складних треків. Відповідна ситуація легко пояснюється тим, що агент автомобілю зтикається з незнайомою ситуацією, для якого не здійснювалася оптимізація значень мостів.

6.2 Порівняльний аналіз навчених агентів з різною структурою

Як вже зазначалось під час виконання експериментів, щодо пошуку параметрів оптимальних для навчання автівки проводились експерименти з

різними структурами агентів ігрових автомобілів на різних треках. Що, в свою чергу, спонукало до здійснення порівняння навчених ігрових автівок різної структури. Для здійснення порівняння було проведено ряд експериментів, результати двох з них наведено на рис. 6.6 і рис. 6.7.

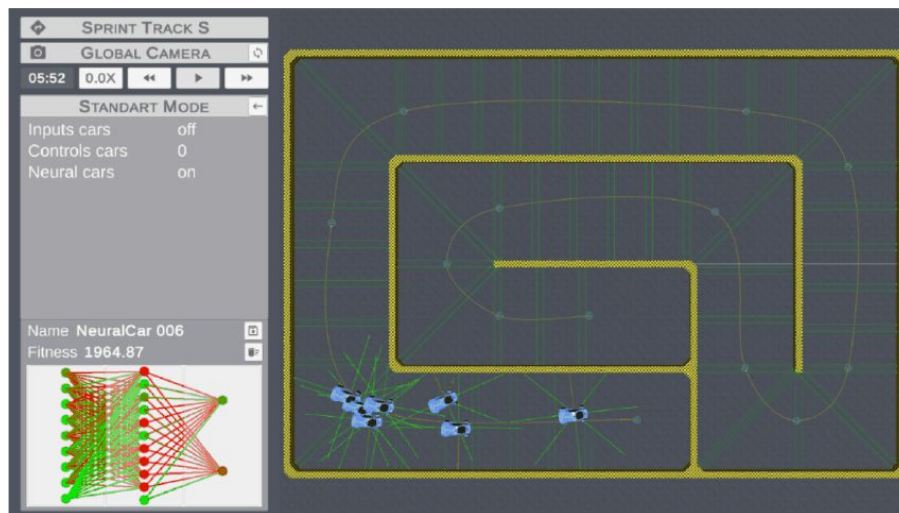


Рисунок 6.6 – Екрана форма експерименту з порівнянням автівок з різною структурою агентів на треку з мінімальною складністю

За результатами виконання навчання і проведення експериментів можна зазначити, що оптимальними для здійснення задачі керування автомобілем є агент з одним чи двома прихованими шарами. Тому що за якістю водіння вони відрізняються не суттєво, але навчаються у рази більш швидко. Також можна зазначити, що навчання агентів було помічено, що використання кросоверу максимально гарно показує себе при одному прихованому шарі.

Суттєвою особливістю нейронних мереж з одним шаром є те, що для успішного навчання на важкому треку вони повні мати більшу кількість нейронів на прихованому шарі у відповідності до двошарових мереж, які при тих самих обставинах можуть мати меншу кількість нейронів на своїх шарах.

Маленька кількість нейронів приводить до двох альтернативних

варіантів. Перший з них можна представити, як п'яне водіння, при якому автомобіль дуже сильно вілляє за треком. Другий варіант полягає в тому, що агент автомобілю взагалі не може навчитися проходити поворот.

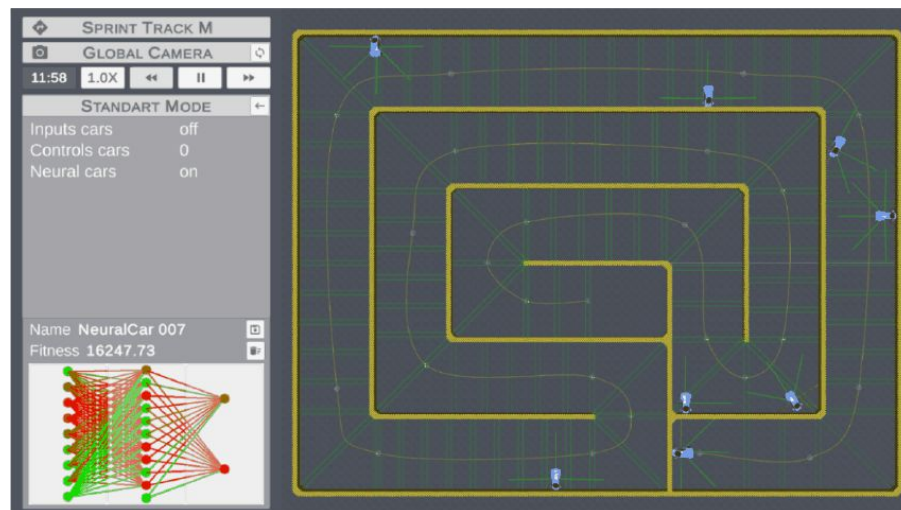


Рисунок 6.7 – Екрана форма експерименту з порівнянням автівок з різною структурою агентів на більш складному треку

В незалежності від кількості шарів та нейронів ситуація перевірки машинок на більш складних та легких треках показує ситуацію, аналогічну описаній у попередньому пункті. Але з цього приводу можна сказати, що машинки з одним шаром, які навчалися на більш легкому треку, здебільшого показують себе більш якісно.

Окрім цього тестувалися машинки з різною кількістю сенсорів. З точки зору навчання надмірне збільшення кількості сенсорів на автомобілі приводило до збільшення часу навчання і необхідної кількості нейронів у мережі, без отримання якого нового якісного рівня водіння.

Під час проведення відповідних тестів було практично визначено, що задні сенсори фактично не впливають на їзду автомобіля. А найкращій результат водіння досягається, коли кількість передніх сенсорів знаходиться в діапазоні від 3 до 7.

6.3 Порівняння механіки руху за маршрутом і навчених агентів

Як вже було зазначено, класичним варіантом реалізації гоночного автомобілю є автомобіль, що реалізовано на базі механіки слідування маршруту. В якомусь розрізі відповідний варіант реалізації відповідної автівки є діаметрально протилежним реалізації автівки на базі нейронних мереж.

Під час виконання відповідних експериментів (рис. 6.8) з порівняння автівок було виявлено те, що навіть з внесенням модифікації до параметрів контролеру ігрової автівки, що реалізує рух за маршрутом, вони поступаються ігровим автівкам, що базуються на нейронних мережах. Фактично автівки, що базуються на нейронних мережах більш швидкі і проворні.

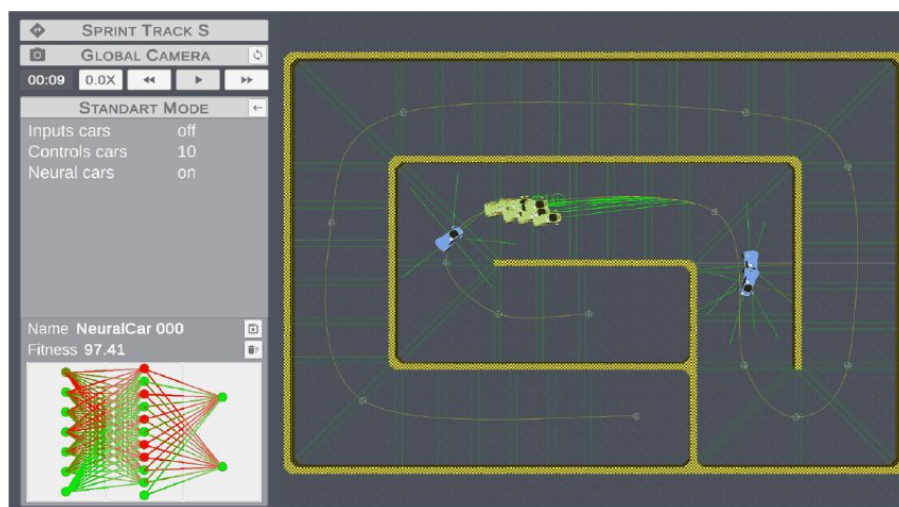


Рисунок 6.8 – Один з експериментів з порівняння автівок з різними механіками руху

Це легко пояснити тим, що автівка, заснована на мазниці руху за маршрутом обмежена певним раніш передбаченим алгоритмом з багатьма

параметрами для того, щоб можна було адаптувати відповідну автівку до певних особливостей треку. Наприклад, в даному випадку алгоритм руху за маршрутом не справляється з поворотами, якщо сильно розігнати автівку, а у випадку здійснення гальмування з великої швидкості на повороті втрачається більше часу ніж їхати з певною оптимальною швидкістю.

В той час автівки, засновані на нейронних мережах, часто взагалі входять до повороту не гальмуючи і використовують фізичні особливості рушія максимально в своїх цілях. Як приклад цього можна навести диктування автівки після повороту без скидання швидкості. Реалізація відповідних механік драфту класичними способами є більш важкою задачею.

Вище описані факти пояснюють кардинальну перевагу автівок на базі нейронних мереж. Вони представляють собою набагато більш сильного суперника, ніж класичні автівки. Відповідно конкурувати з даним типом автівок під час гри буде набагато складніше в зв'язку з тим відповідний тип автівок може з легкістю непереборною перешкодою для багатьох гравців.

6.4 Оцінка складності розробки і налагодження агентів

З точки зору роботи з побудови ігрової системи і здійснення її налагодження ігровий персонаж, заснований на базі агента нейронної мережі, представляє собою чорну скриньку. Але це не чорна скринька в сенсі розуміння внутрішніх процесів роботи, оскільки внутрішній алгоритм в даному випадку цілком відомий. Це чорна скринька в сенсі розуміння залежності, яка була визначена під час навчання.

Відповідно до цього внесення невеликих поправок до визначеної залежності не є можливим. Хоча і існує фізична можливість ручного внесення виправлення ваги певних мостів нейронних мереж. Але передбачити заздалегідь до яких наслідків приведене відповідне виправлення зазвичай представляється не можливим. В випадку не повного розуміння результатів, до яких приведе внесення такого виправлення, необхідно



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

проведення тестування виправленого агента у всіх можливих сценаріях.

Фактично ми приходимо до того, що у випадку необхідності внесення правок до агентів на базі нейронних мереж необхідно здійснювати повторно навчальний процес з деякими корективами, які повинні привести систему до бажаного результату. Фактично в багатьох випадках це робить процес корегування інтелекту є достатньо трудомісткий і займає багато часу.

Тут можна навести у порівняння ситуацію, яка відбувається при необхідності зміни поведінки при використанні класичних підходів. На відмінно від агентів, заснованих на нейронних мережах класичні підходи дозволяють дуже легко коректувати логіку взаємодії і власні параметри, а розширення логіки зазвичай не представляє нічого важкого і здійснюється доданням нових методів.

Фактично кожен з підходів до створення ігрових агентів має свої плюси і мінуси, але в цілому обидва підходи мають право на існування. Підхід, заснований на базі агентів нейронних мереж, має своє місце і навіть в деяких випадках може спростувати процес розробки і дозволяти виходити на новий якісний рівень ігрового інтелекту. Але, прибігаючи до використання, обов'язково необхідно враховувати специфіку.

Даний підхід дуже гарно може бути застосований у ігрових персонажах, взаємодії і параметри яких точно не будуть змінюватися як під час самої гри, так і бажано під час створення ігрового продукту. Тоді один раз правильно навчений ігровий агент буде реалізовувати логіку певного персонажу, що в багатьох випадках може скоротити час на написання великої кількості коду, який реалізує часто доволі просту логіку персонажу. І у багатьох випадках набагато більш якісніше, ніж в вигляді реалізованого класичним методом. Яскравим прикладом такої ситуації може слугувати те, що агенти ігрових автомобілів, реалізованих в рамках даної роботи, досить сильно перевершують контролери автомобілів, реалізованих на стандартних підходах.

6.5 Висновки за розділом



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання

В рамках проведених під час виконання даного розділу експериментів з моделюванням процесів навчання і конкуренції між автомобілями було зроблено висновки як стосовно процесу навчання, так і стосовно конкретного порівняння. Всі експерименти, що описані у розділі, проводилися по декілька разів, що дало змогу висунути певні припущення про певні закономірності. До деяких помічених закономірностей було висунуто найвірогідніше теоретичне обґрунтування відповідної закономірності.

У зв'язку з невеликою кількістю проведених експериментів отримані результати можуть мати похибки. Для отримання більш широкої картини необхідно проведення більшої кількості експериментів.

Проведені експерименти фактично підтверджують працездатність запропонованого підходу до навчання агентів гоночних автомобілів. В розділі було оцінено впливання параметрів автомобілю і алгоритму навчання на швидкість навчання. Також емпіричним шляхом було підібрано параметри алгоритму навчання, які достатньо швидко виконують навчання.

Окрім цього було зроблено порівняльний огляд роботи класичного контролеру і агенту ігрового автомобілю. В результаті цього протиставлення автомобілів, що базуються на нейронній мережі, перевершили класичні.

На базі отриманого під час розробки середовища моделювання досвіду і даних, отриманих в результаті моделювання, було зроблено оцінку складності розробки і налагодження агентів, а саме було визначено, що підхід має своє місце і в деяких випадках дозволяє спростувати розробку.

ВИСНОВКИ

В роботі було проведено порівняльний аналіз сучасних методів та алгоритмів машинного навчання і інтелекту, на основі проведеного аналізу відбулось обґрунтування і розробка формальної моделі ігрового агента на базі методів машинного навчання та інтелекту, визначено підхід до навчання розроблюваного агента, розроблено і протестоване програмне середовище, яке реалізує симуляцію гоночного процесу. За допомогою розробленого програмного середовища були реалізовані множинні симуляції, які дозволили отримати практичні результати навчання ігрових агентів. Разом з тим було отримано рекомендації відповідно роботоспроможності запропонованої концепції, отримано оцінки складності розробки і налагодження агентів.

Результатом виконання даної роботи стала формалізована модель агента ігрового автомобілю на базі методів машинного навчання та інтелекту і запропонований підхід до здійснення навчання даної моделі. Також результат роботи представлено середовищем моделювання гоночного процесу з фактичною реалізацією запропонованої моделі і підходу до його навчання. Відповідне середовище моделювання дозволяє проводити експерименти з навчання моделі агента, а також вільні експериментальні заїзди з різними типами ігрових автомобілів.

Вихідний код розробленого в рамках даної роботи середовища доступний в GitHub за посиланням <https://github.com/voossu/Driving-Range-Project> і розповсюджується за ліцензією MIT.



Схожість



Схожість з обраним джерелом



Заміна літер абетки



Цитата



Посилання