ЛАБОРАТОРНАЯ РАБОТА № 5«ИССЛЕДОВАНИЕ ДИНАМИЧЕСКИХ СЕТЕЙ БАЙЕСА»

Цель работы

Исследование методов точного и приближенного вероятностного вывода с использованием динамических сетей Байеса, приобретение навыков программирования интеллектуальных агентов, знания которых представляются условными высказываниями с определенной степенью уверенности.

Ход работы

Для начала для класса InferenceModule был реализован метод getObservationProb (код представлен в приложении А), после был вызван автооцениватель (рисунок 1).

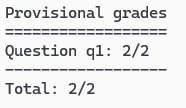


Рисунок 1 – Результат автооценивания первого задания

Далее для класса ExactInference был реализован метод observeUpdate (код представлен в приложении А), после был вызван автооцениватель (рисунок 2), а также тесты ExactUpdate (рисунок 3).

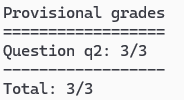


Рисунок 2 – Результат автооценивания второго задания

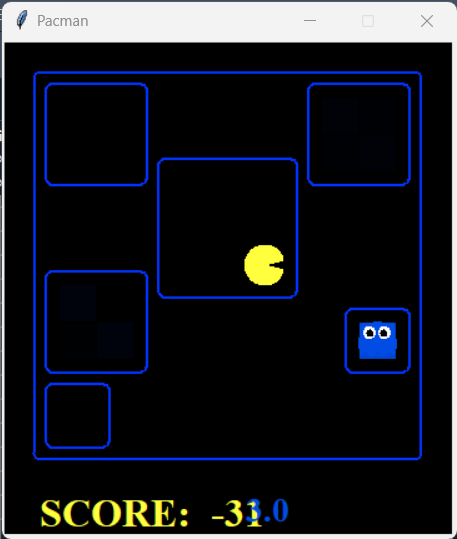


Рисунок 3 – Тест ExactUpdate

Далее для класса ExactInference был реализован метод elapseTime (код представлен в приложении А), после был вызван автооцениватель (рисунок 4), а также тесты ExactPredict (рисунок 5).

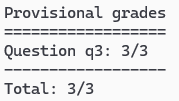


Рисунок 4 – Результат автооценивателя для третьего задания

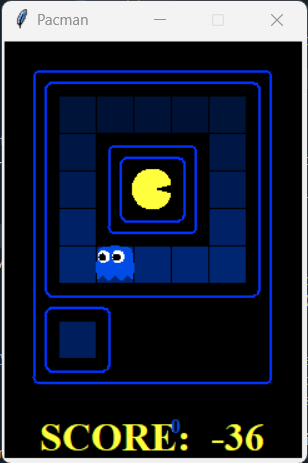


Рисунок 5 – Тест ExactPredict

Далее для класса GreedyBustersAgent был реализован метод ChooseAction (код представлен в приложении Б), после был вызван автооцениватель (рисунок 6), а также тесты ExactFull (рисунок 7).

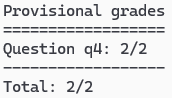


Рисунок 6 – Результат автооценивания четвертого задания

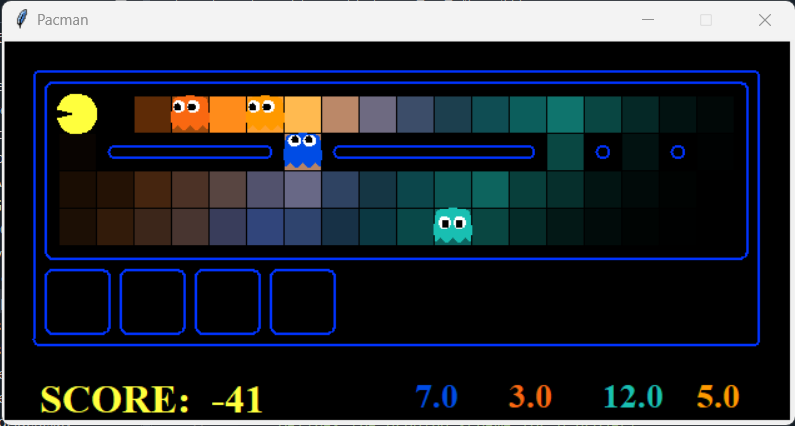


Рисунок 7 – Тест ExactFull

Далее для класса ParticleFilter был реализованы методы initializeUniformly и getBeliefDistribution (код представлен в приложении А), после был вызван автооцениватель (рисунок 8), а также тесты ParticleInit (рисунок 9).

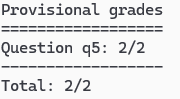


Рисунок 8 – Результат автооценивания пятого задания

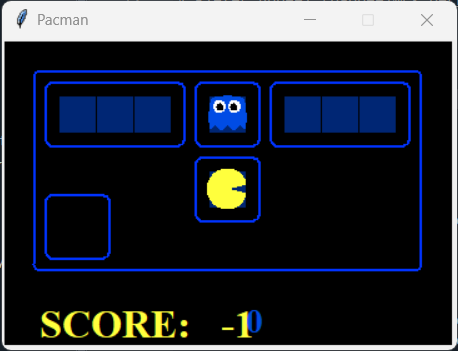


Рисунок 9 – Тест ParticleInit

Далее для класса ParticleFilter был реализован метод observUpdate (код представлен в приложении А), после был вызван автооцениватель (рисунок 10), а также тесты ParticleUpdate (рисунок 11).

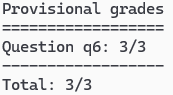


Рисунок 10 – Результат автооценивания шестого задания

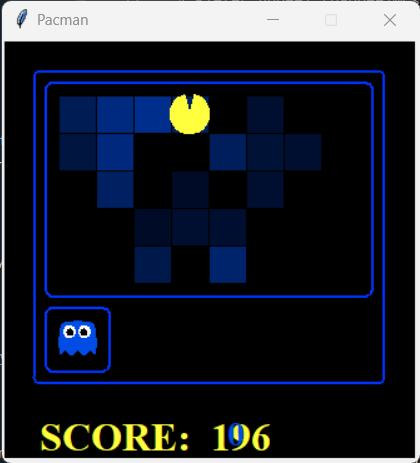


Рисунок 11 – Тест ParticleUpdate

Далее для класса ParticleFilter был реализован метод elapseTime (код представлен в приложении А), после был вызван автооцениватель (рисунок 12), а также тесты ParticlePredict (рисунок 13).

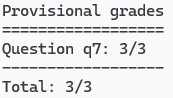


Рисунок 12 – Результат автооценивания седьмого задания

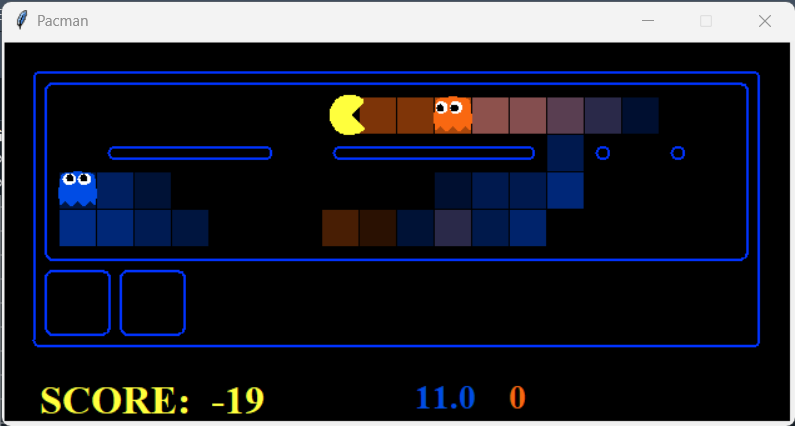


Рисунок 13 – Тест ParticlePredict

Далее для класса JointParticleFilter был реализован метод initializeUniformly (код представлен в приложении А), после был вызван автооцениватель (рисунок 14), а также тесты JointParticleInit (рисунок 15).

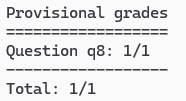


Рисунок 14 – Результат автооценивания восьмого задания

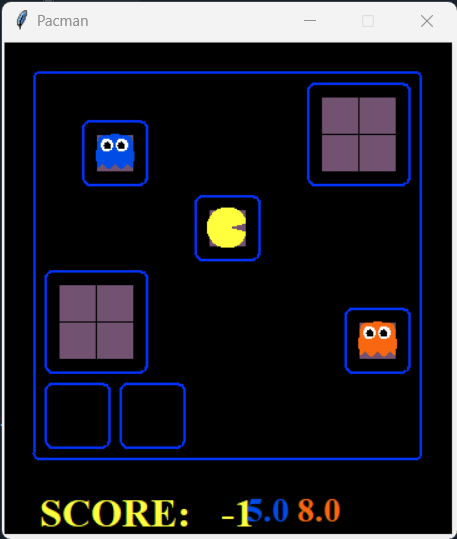


Рисунок 15 – Тест JointParticleInit

Далее для класса JointParticleFilter был реализован метод оbservUpdate (код представлен в приложении А), после был вызван автооцениватель (рисунок 15), а также тесты JointParticleUpdate (рисунок 16).

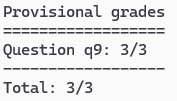


Рисунок 15 – Результат автооценивания девятого задания

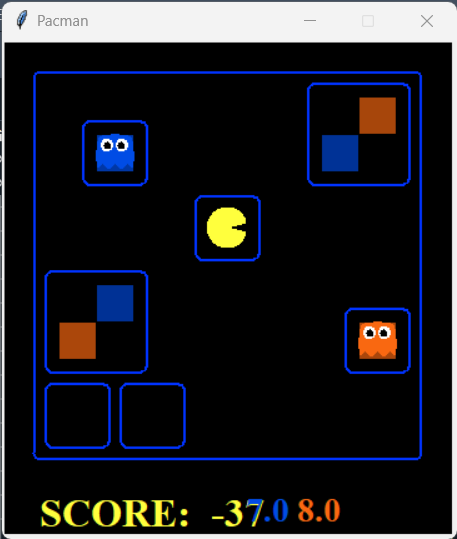


Рисунок 16 – Тест JointParticleUpdate

Затем для класса JointParticleFilter был реализован метод elapseTime (код представлен в приложении А), после был вызван автооцениватель (рисунок 17), а также тесты JointParticlePredict (рисунок 18).

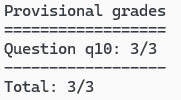


Рисунок 17 – Результат автооценивания десятого задания

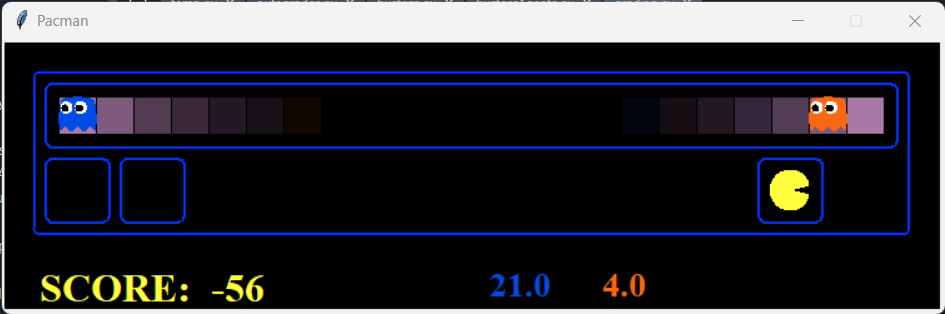


Рисунок 18 – Тест JointParticlePredict

Наконец был запущен автооцениватель для всех заданий (рисунок 19).

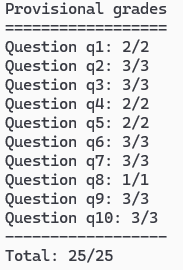


Рисунок 19 – Финальное автооценивание

**Выводы**

В ходе проделанной работы было проведено исследование методов точного и приближенного вероятностного вывода с использованием динамических сетей Байеса, были приобретены навыки программирования интеллектуальных агентов, знания, которые представляются условными высказываниями с определенной степенью уверенности.