ФИТ НГУ, курс ООП, осенний семестр 2012

Задача 3а. Дилемма заключённого

25 баллов

История изменений

- Версия 3. 28.11.В пункт 4а добавлено уточнение о выходе по команде quit
- Версия 2. 07.11.2012. Добавлено уточнение о том, что очки за каждый раунд вычисляются по матрице игры. Добавлен пропущенный параметр командной строки программы - режим соревнования.
- Версия 1. 06.11.2012. Документ создан.

Общие сведения

Требуется реализовать консольное приложение, позволяющее симулировать повторяющуюся “Дилемму заключённого” (детали в Википедии: http://en.wikipedia.org/wiki/Prisoner's_dilemma) для троих заключённых.

Устройство игры

1. Программа позволяет “соревноваться” трём “заключённым” (далее - стратегиям). которые пытаются максимизировать набранные очки. Выигрывает стратегия, по результатам игры (некоторого фиксированного количества ходов) набравшая наибольшее количество очков.

2. Игра протекает следующим образом: пошагово, в течение некоторого количества шагов, симулятор выясняет выбор каждой стратегии: сотрудничать (C) или предать (D). После этого, симулятор сообщает каждой стратегии выбор её оппонентов и вычисляет очки за текущий раунд. Очки вычисляются по матрице игры (пример матрицы в приложении).

3. Стратегии делают свой выбор основываясь на накопленном опыте: истории ходов оппонентов и собственной. Алгоритмы стратегий могут быть самыми различными, начиная от тривиальных “всегда сотрудничать” или “выдавать случайный результат”. Изначально, история ходов каждой стратегии пуста.

4. Симуляция может проходить в трёх режимах:

a. Соревнование с детализацией. На каждом шаге программа ожидает команды от пользователя `tick` или `tick <n>`, после которой делает 1 или n шагов. Подробное состояние симуляции выводится после каждого шага (выбор каждой стратегии, очки за текущий ход, очки за текущую игру). Игра прерывается по команде quit.
b. Соревнование без детализации. Программа вычисляет заданное количество ходов и выводит результат.
c. Турнир. Программа перебирает все возможные тройки (без повторений, т.е. s1,s2,s3 и s3,s2,s1 - одна и та же тройка) указанных стратегий и по общему результату выявляет победителя. Результаты каждого соревнования и итоговый протокол выводятся на экран

Технические детали

1. На старте программе подаются следующие аргументы командной строки:
    a. Три (или более - для турнирного режима) имени соревнующихся стратегий.
    b. Название режима `--mode=[detailed|fast|tournament]` (опциональный, по умолчанию - `detailed` для трех стратегий, `tournament` для >3 стратегий)
    c. Число шагов симуляции `--steps=<n>` (опциональный)
    d. Директория с конфигурационными файлами стратегий `--configs=<dirname>` (опциональный)
    e. Файл с матрицей игры `--matrix=<filename>`(опциональный)

2. Технически, каждая стратегия представляется в виде класса. У стратегий выделяется общий абстрактный интерфейс. Нетривиальные стратегии хранят свои параметры в собственном конфигурационном файле произвольного формата.

3. При сдаче необходимо продемонстрировать 3-5 тривиальных и 3-5 нетривиальных стратегии, включая “метастратегии” - использующие несколько других стратегий для принятия решения. Сложность/интересность стратегий не ограничивается.

4. Для конструирования классов-стратегий следует использовать шаблон проектирования “Абстрактная фабрика”.

5. Матрица игры описывает количество очков, получаемых стратегиями за каждый ход. Матрицу по умолчанию можно вшить в код программы. Пример правильной матрицы для игры с тремя заключёнными содержится в приложении. Формат файла с матрицей произволен.

6. При реализации программы подразумевается максимальное использование STL. Для работы с файлами необходимо использовать файловые потоки, переключенные в режим сообщения об ошибках исключениями (см. метод `std::ios::exceptions`)

7. При сдаче, преподаватель может потребовать интегрировать в симулятор студента стратегии другого учащегося и устроить турнир.

8. Поощряется (но не является обязательной) реализация возможности динамической загрузки стратегий из динамических библиотек.

Приложение. Матрица игры

```
з1 з2 з3     з1 з2 з3
 C  C  C  =>  4  4  4  //заключённые 1, 2 и 3 получают по 4 очка
 C  C  D  =>  2  2  5
 C  C  D  =>  2  2  5
 C  D  C  =>  2  5  2
 D  C  C  =>  5  2  2
 C  D  D  =>  0  3  3
 D  C  D  =>  3  0  3
 D  D  C  =>  3  3  0
 D  D  D  =>  1  1  1
```