

Game Server

Краткое описание

Реализовать клиентское и серверное приложения: «игровой сервер». Предлагается создать программу, которая предоставит возможность двум (или более) пользователям играть друг с другом по сети.

Детальное описание

Сервер содержит список игроков – тех, кто сейчас подключен к серверу. Игроки делятся на две группы: «играющие» и «свободные». Группа «Играющие» состоит из игроков, которые участвуют в какой-либо партии. Соответственно в группе «свободные» - игроки, которые сейчас не участвуют ни в одной партии. При подключении к серверу игрок попадает в группу «свободные». Игрок переходит в группу «играющие» когда он присоединяется к уже существующей партии или начинает новую. Соответственно, когда «партия» заканчивается, игрок возвращается в группу «свободные».

Под термином «партия» подразумевается один сеанс связи между игроками (через сервер), в процессе которого они обмениваются информацией о сделанном ходе и текущем состоянии. Есть несколько вариантов контролирования этого процесса (обмена данными):

- сервер самостоятельно контролирует соблюдение правил, вычисляет новые состояния для всех игроков и рассылает эту информацию всем участникам; контроль завершения партии осуществляется сервером;
- клиенты самостоятельно контролируют правильность хода; при возникновении конца партии (например, мата в шахматах) один или несколько клиентов уведомляют об этом сервер;
- клиенты только реализуют отображение и пересылку информации; при возникновении конца партии один из игроков сообщает об этом серверу, после чего сервер спрашивает согласия остальных игроков на завершение партии (при этом запрашиваются информация о том, кто проиграл, а кто выиграл). Предполагается, что игроки честные и реализовывать проверки в программе не надо.

Сервер хранит некоторые данные об игроках: имя, пароль, ранг и др. Для вычисления ранга можно использовать, например, следующий алгоритм: каждому новому игроку дается определенное количество очков. Пусть k – коэффициент «значимости» партии ($k \ll 1$). При окончании партии сервер выполняет следующие изменения количества очков: если ничья, то каждому игроку прибавляется количество очков оппонента, умноженное на $k/2$. Иначе выигравшему начисляется количество очков проигравшего, умноженное на k .

Требования

Звездочками * отмечены обязательные требования. Набор обязательных требований может быть изменен или дополнен куратором группы.

1. * При подключению к серверу клиент получает список игроков из группы «свободные», после чего он может выбрать, с кем он бы хотел сыграть. Выбранному сопернику посылается запрос, согласен ли он сыграть с таким-то игроком и в случае получения утвердительного ответа начинается партия.
2. Завершение партии должен контролироваться программой (либо клиентом, либо сервером).
3. При долгом бездействии игрока в партии, ему выводится предупреждение «idling». Если он не проявит активности, то соединение с ним закрывается.
4. Использование XML для обмена данными между сервером и клиентом.
5. Для каждого игрока сервер должен вести статистику: количество выигранных и проигранных партий, сколько всего времени провел на сервере, ранг (очки).
6. При подключении к серверу клиент получает список всех игроков. При выборе в качестве соперника игрока из группы «играющие» сервер должен выполнить действия, описанные в пункте 1 после того, как выбранный игрок закончит текущую партию. То есть сервер поддерживает «ожидание игрока»: клиент выбирает игрока, который уже

- с кем-то играет, чтобы сыграть с ним, когда он освободится (сервер автоматически сообщает, что выбранный игрок закончил свою партию).
7. * Возможность банить игроков/ip адреса.
 8. Возможность восстановления игровой позиции в случае разрыва связи с одним клиентом. То есть при обрыве соединения клиенту дается возможность подключиться к серверу в течении некоторого времени (например, 2 минут) и если он подключится – партия должна продолжиться.
 9. * Сохранение пользовательских настроек на сервере.
 10. Ограничение времени, данного игроку на раздумывание (на партию 30 минут каждому игроку) – то есть в сумме игрок не может размышлять над своим ходом более какого либо времени.
 11. Поддержка «чемпионатов».
 12. Возможность сохранения партии в файл.
 13. Изменение Look-and-Feel на клиенте.
 14. * Организовать ведение логов на сервере.
 15. * Программный код должен удовлетворять Java Code Conventions и быть снабженным JavaDoc.
 16. * Логирование в коде организовать с помощью библиотеки Log4j.
 17. * Для сборки приложения использовать Maven.
 18. * Дизайн приложения. Дизайн приложения должен создаваться **до написания кода** и представляться группой куратору для обсуждения и утверждения. Дизайн должен состоять из:
 - Диаграмма классов (Class diagram) приложения
 - Диаграмма прецедентов (Use Case диаграмма)

Модификации задания

В качестве модификаций задания предлагается реализовать различные игры.

Игры:

1. Шашки.
 - Поле: 8x8.
 - Правила игры, контролируемые программой: окончание партии; если есть возможность, игрок должен «брать» максимальное количество фишек противника.
 - Реализация: Java2D; JTable+JLabels; Console
 - Ресурсы: <http://www.shashki.com/>
2. Шахматы.
 - Поле: 8x8.
 - Правила игры, контролируемые программой: окончание партии; ситуация «шах»; ситуация «пат».
 - Реализация: Java2D; JTable+JLabels; Console
 - Ресурсы: <http://www.sportzone.ru/sport/rules.html?sport=chess&chapter=02>
3. «Точки» («ГО»).
- Поле: размер поля на усмотрение студента.
- Правила игры, контролируемые программой: окончание партии; «окружение» точек противника
- Реализация: Java2D
- Ресурсы: <http://www.tochki-club.narod.ru/>; <http://forum.kido.com.ru/>
4. Морской Бой.
 - Поле: 10x10
 - Правила игры, контролируемые программой: окончание партии;
 - Реализация: Java2D или JTable+JLabels
5. Нарды.
 - Правила игры, контролируемые программой: окончание партии; корректность хода;
 - Реализация: Java2D

- Ресурсы: <http://amis.h11.ru/nardprav.htm>
- 6. Домино.
 - Правила игры, контролируемые программой: окончание партии; правильность хода
 - Реализация: Java2D
 - Ресурсы: <http://www.orgdosug.ru/pubcatalog.php?cid=83>
- 7. «Быки И Коровы»
 - Правила игры, контролируемые программой: окончание партии;
 - Реализация: Swing; Console
 - Ресурсы: <http://www.wowwi.orc.ru/games/bulls/help-ru.htm>

Дополнительная информация

Изучаемые темы

Swing, Threads, Sockets; Java2D

Справочные материалы

1. <http://logging.apache.org/log4j/2.x/>
2. <http://maven.apache.org/guides/getting-started/index.html>
3. http://www.info-system.ru/designing/methodology/uml/theory/class_diagram_theory.html
4. http://www.info-system.ru/designing/methodology/uml/theory/use_case_diagram_theory.html
- 5.