Темы курсовых работ

1. Турнир роботов. Разработка ядра системы.
2. Турнир роботов. Разработка клиентского модуля.

Организация работы

Курсовая работа выполняется в рамках коллективного проекта и разбивается на три этапа.

На **первом этапе** из каждой студенческой группы выделяются (по желанию) не менее двух рабочих групп, состоящих из трех человек. В задачи рабочей группы входит:

1. Разработка архитектуры ядра системы (структура данных, способы взаимодействия с пользовательским интерфейсом и клиентскими модулями).
2. Разработка пользовательского интерфейса и средств протоколирования.
3. Разработка базового клиентского модуля, реализующего случайное поведение, и средств взаимодействия с ним.
4. Подбор нескольких наборов рабочих настроек, обеспечивающих приемлемое время функционирования участников турнира (не менее 500 шагов для 20% участников).

Взаимодействие рабочих групп осуществляется с помощью зарезервированных репозитариев:

https://github.com/asugubkin/AS.1204.SysProg.2
https://github.com/asugubkin/AS.1205.SysProg.1

В каждой студенческой группе проект, первым доведенный до рабочего состояния и соответствующий всем требованиям преподавателя, выбирается в качестве базового для следующих этапов, его разработчики получают окончательные оценки. Разработчики остальных проектов в зависимости от их готовности получают 0-30 баллов и принимают участие в следующих этапах на общих основаниях.

Срок выполнения первого этапа: **17.03.2015**.

Во **втором этапе** принимают участие все студенты, кроме победителей первого этапа. Задачей этого этапа является разработка клиентского модуля, взаимодействующего с ядром системы в соответствии со спецификациями, заданными на первом этапе. За основу может быть взят базовый клиентский модуль, разработанный на первом этапе.

Подведение итогов второго этапа осуществляется в форме турнира после окончательной проверки работоспособности модулей всех участников. Предоставление клиентского модуля для участия в турнире осуществляется путем добавления подкаталога с его кодом в общий репозитарий проекта и записи его атрибутов в общий конфигурационный файл, формат которого определяется разработчиками ядра.

В группе может быть выбран студент, отвечающий за корректность применения изменений в репозитарии и сборку проекта. В зависимости от успешности этой работы она оценивается в дополнительные 0-10 баллов.

Турнир проводится в 5 раундов, каждый из которых проходит с одним из ранее определеных рабочих наборов настроек. В каждом раунде принимают участие 5 стандартных роботов, все роботы, выжившие в предыдущем раунде, и по одному новому роботу от каждого участника. Баллы, получаемые по итогам каждого раунда:

Место	Баллы
1	15
2-6	11
7-13	9
14-22	7
23-33	5

Места для роботов, оставшихся работоспособными на момент завершения раунда, распределяются в соответствии с итоговым уровнем энергии, для остальных — в соответствии с числом шагов, в течение которых они сохраняли работоспособность. Каждый успешный вывод из строя робота противника добавляет к итоговому уровню энергии K бонусных единиц.

В зачет участника в каждом раунде идут баллы не более чем от трех его лучших роботов.

Срок выполнения второго этапа: **22.06.2015**.

Перед **третьим этапом** участникам дается возможность доработать свои модули с учетом результатов второго этапа. Правила остаются теми же, в первый раунд третьего этапа переходят роботы, выжившие в пятом раунде второго этапа. Студенты, набравшие желаемое число баллов на втором этапе, могут не принимать участия в третьем этапе.

Начиная с этого этапа вышедшие из строя роботы также получают K/2 бонусных единиц за вывод из строя других роботов.

Срок выполнения третьего этапа: **26.06.2015**.

Окончательная оценка может быть скорректирована по итогам защиты кода. В случае предоставления чужого кода либо неспособности его объяснения применяется стандартный штраф 20%.

Правила проведения раундов турнира

Турнирное пространство представляет собой тороидальное поле размером W на H. Каждый раунд турнира состоит из N шагов продолжительностью T миллисекунд. Каждый участвующий в турнире робот имеет следующие характеристики: уровень энергии E

(0-E_{max}), уровень технического состояния L (0-L_{max}), уровень атаки A, защиты P, скорости V (A+P+V = L).

На каждом шаге роботу доступны следующие действия, которые могут комбинироваться:

1. перемещение в любом направлении на расстояние 0 - V_{max}*V/L_{max}*E/E_{max}.
2. атака соперника, находящегося на расстоянии 0 - R_{max}*V/L_{max}*E/E_{max}, с силой A*E/E_{max}.
3. перераспределение 0 - ΔL единиц технического состояния.

На каждом шаге робот теряет ΔE_S единиц энергии в случае простоя, ΔE_V в случае перемещения, ΔE_A в случае атаки, ΔE_P в случае ситуации, требующей защиты.

В процессе атаки атакующий и атакуемый взаимодействуют с реальными уровнями атаки A_r=RND*A и защиты P_r=(1-RND)*P, где RND — случайное число от RND_{min} до RND_{max}.

Если сила атаки атакующего A_r*E_A/E_{max} превышает текущую защиту атакуемого P_r*E_P/E_{max}, атака считается успешной. В этом случае уровень защиты атакуемого уменьшается на уровень $\Delta P = int(A_r*E_A/E_{max}-*P_r*E_P/E_{max})$. В противном случае уровень атаки атакующего уменьшается на $\Delta A = int(P_r*E_P/E_{max} -A_r*E_A/E_{max})$. Если соответствующие уровни защиты или атаки достигли нуля, происходит снижение уровня энергии пострадавшей стороны на величину $\Delta E = |int(P_r*E_P/E_{max} -A_r*E_A/E_{max})|*E_{max}/L_{max}$. В случае взаимной атаки используются значения A,P,E_A,E_P из предыдущего шага, до их снижения в результате атаки либо действий по перераспределению единиц технического состояния.

На поле размещаются (случайным образом либо в редакторе) N_E пунктов подзарядки и N_L пунктов технического обслуживания. Остановка в координате соответствующего пункта приводит к повышению уровня энергии на ΔE (но не более E_{max}) либо повышению (и перераспределению между A, P, V) уровня технического состояния на ΔL (но не более L_{max}).

Робот, чей уровень энергии E достиг нуля, выходит из строя. Другой робот, попавший в точку с его координатами, может повысить свой уровень технического состояния на величину ΔL (но не более L_{max}), при этом на ту же величину снижается и остаточный уровень технического состояния вышедшего из строя робота.

Первоначальная расстановка роботов в каждом раунде осуществляется случайным образом. Расстояние между любыми двумя роботами в первоначальной расстановке должно быть не менее 2*V_{max}.

Требования к ядру системы

Ядро системы обеспечивает хранение информации, описывающей текущее состояние поля; подключение клиентских модулей, описанных в конфигурационных файлах (при описании указывается имя загружаемой библиотеки, идентификатор, фамилия автора); перерасчет текущего состояния на основе действий пользователей.

Требования к пользовательскому интерфейсу

Интерфейс обеспечивает изменение и сохранение наборов настроек и стартовых условий, запуск/приостановку этапов турнира, визуализацию действий и текущих параметров участников, протоколирование и воспроизведение действий.

Требования к интерфейсу взаимодействия с клиентскими модулями

На каждом шаге для каждого клиента ядро создает новый поток, вызывающий функцию **DoStep** клиентского модуля и передающий ей указатель на структуру **StepInfo**. Структура содержит следующие поля: номер шага; сведения о текущем положении и характеристиках робота; информация о текущем состоянии поля; информация о действиях всех роботов на предыдущем шаге; значения текущих настроек; информация о выбранных действиях. В течение заданного времени T функция DoStep должна записать в переданную структуру информацию о выбранном действии и завершиться. В противном случае поток должен быть принудительно завершен ядром, а его действие проигнорировано. Запись и чтение в эти поля защищаются мьютексом.

Возможны другие реализации взаимодействия, основанные, например, на механизме передачи сообщений. Ключевые условия: гарантированное завершение клиентского действия в течение времени T, невозможность подделки действий и создания помех другим клиентам.

Клиентский модуль не получает информацию о предыстории действий, но может хранить ее самостоятельно. Модули могут использовать полученную информацию как для организации совместных действий с другими модулями участника, так и для формирования альянсов с другими участниками.

Предварительные ориентировочные значения настроек

W	200
H	200
N	1000
T	100 ms
E_{max}	1000
L_{max}	100
V_{max}	10
R_{max}	5
ΔL	10
ΔE_S	1
ΔE_V	2
ΔE_A	10

ΔE_P	5
ΔE	100
N_E	10
N_L	10
RND_{min}	0.4
RND_{max}	0.8
K	100