Условия игрового тура

1.Введение 2.Правила игры 3.Игровые объекты 4.Игровая оболочка 5.Технический регламент

1.Введение

В рамках проведения 10го турнира ICL по программированию участникам предлагается принять участие в соревнованиях класса Code Game Challenge.

Code Game Challenge это соревнование искусственных интеллектов в заданной игре. Командам предлагаются правила игры и модель игрового мира. В рамках заданных условий и ограничений команды создают свои алгоритмы (боты), которые затем самостоятельно соревнуются, выявляя победителя.

В многолетней истории CGC были и космические баталии, и гонки, и логические игры. Классической игрой для CGC стала игра в «танчики». В рамках X ICL CGC участникам впервые предлагается сыграть в стратегическую игру – Dice Wars или «Костяные войны».

История игры ведется с 2001 года, когда появилась её первая реализация от GameDesign.jp, созданная на Flash. За прошедшие годы игра завоевывала армии поклонников по всему миру, появляются новые реализации и вариации. Сейчас в Dice Wars ежедневно играют он-лайн тысячи людей, и игра становится только популярнее.

Участникам турнира предлагается адаптированный вариант игры. Добавлены некоторые элементы, направленные на балансировку сил. Многие полагают, что слишком большое значение в этой игре имеет фактор везения, но на практике видно, что сильная стратегия побеждает.

2.Правила игры

Поле для игры представляет собой гексагональную сетку, произвольно поделенную на земли. Часть поля, не занятую землями, будем считать водой. Земля представляет собой связное множество шестиугольников в сетке. Связными являются клетки, граничащие сторонами. На каждой земле может находиться от 1 до 8 игральных костей - армия этой земли.

В начале игры случайным образом генерируется карта мира, делится на земли, земли распределяются между государствами, соответственно окрашиваются в разные цвета, и на каждую землю выкладывается некоторое количество кубиков. Генерация осуществляется таким образом, чтобы шансы игроков были максимально сбалансированы.

После генерации карты определяется очередность ходов соперников, и каждому предоставляется возможность выбрать государство на карте, за которое он будет играть. Тому, кто ходит последним, предоставляется право первым выбрать государство на карте. Предпоследний выбирает следующим, и так до первого, который играет за единственное невыбранное государство на карте.

Далее начинается война. Соперники по очереди делают ходы. За каждый ход можно сделать несколько шагов. На своем N-ном ходу игрок может сделать не более N шагов. Каждый шаг представляет собой нападение любой земли с армией более 1 кости на граничащую с ней землю соперника. При нападении все кости нападающей армии и кости обороняющейся выбрасываются и подсчитываются суммы выпавших чисел. Если сумма нападающей армии выше, то проигравшая армия уничтожается, земля объявляется захваченной, то есть переходит к нападавшей стороне, а армия атаковавшей земли переходит на захваченную землю, оставляя одну кость на старой земле. В противном случае уничтожается атакующая армия, кроме одной кости, которая должна остаться на земле. Игрок может закончить ход на любом шагу, или вообще не делая шагов.

Когда игрок заканчивает ход, его военный потенциал наращивается – на принадлежащие ему земли случайным образом разбрасываются кости в количестве, равном максимальному связному множеству земель этого игрока. В случае если на всех землях игрока размещено максимальное количество кубиков (по 8 на каждой земле), положенный ему военный потенциал кладется в запас. При окончании очередного хода запас прибавляется к числу выкладываемых костей. Величина запаса ограничена величиной 100 кубиков. При достижении лимита весь поступающий потенциал просто уничтожается.

Выигрывает игрок, который захватит весь мир. Остальные места распределяются в порядке выбывания соперников из игры.

3.Игровые объекты

Государства на карте обозначены флагами. Флаги изначально присвоены всем землям.

```
public enum Flag {

      RED, GREEN, BLUE, ORANGE, YELLOW, CYAN, MAGENTA, GRAY;

}
```

Земли выглядят так:

```
public interface Land extends Serializable{

      public int getLandId();

      public int getDiceCount();

      public Flag getFlag();

      public Set<Land> getNeighbouringLands();

}
```

getLandId возвращает идентификатор земли

getDiceCount возвращает количество костей на земле

getFlag возвращает флаг государства, которому на данный момент принадлежит земля

getNeighbouringLands возвращает множество земель, граничащих с данной землей

Игровой мир имеет следующий интерфейс:

```
public interface World extends Serializable{

	public Set<Land> getLands();

	public List<Flag> getFlags();

	public Flag getMyFlag();

	public int getDiceCountInReserve(Flag flag);

	public int getAvailableAttackCount();

}
```

getLands возвращает все множество земель на карте

getFlags возвращает список государств мира

getMyFlag возвращает флаг, принадлежащий игроку, вызвавшему метод

getDiceCountInReserve возвращает величину резерва данного государства

getAvailableAttackCount возвращает количество допустимых шагов до завершения хода для игрока, вызвавшего метод

Шаги, которые делают игроки, имеют следующий интерфейс:

```
public interface Attack {

	int getFromLandId();

	int getToLandId();

}
```

Первый метод возвращает идентификатор нападающей земли, а второй возвращает идентификатор земли, на которую осуществляется нападение.

Игроки должны реализовать класс со следующим интерфейсом (в скобках указано ограничение времени на выполнение метода):

```
public interface Player extends Serializable{

	public String getName();
```

Данный метод должен возвращать имя бота. (0.1 сек)

```
	public void init();
```

Метод запускается однажды в самом начале до вызова любых других методов. (2 сек)

```
	public Flag chooseFlag(World world, Set<Flag>
availableFlags);
```

Метод вызывается перед началом войны. Каждому игроку предоставляется возможность выбора любого флага из доступных. В случае, если данный метод вернет некорректный результат (выбор уже занятого флага, NULL или вызов ошибки), игрок пропускает очередь и по итогам распределения ему достанется невыбранный другими игроками флаг. Если несколько игроков вернули некорректный результат, оставшиеся флаги распределяются между ними случайным образом. Метод вызывается и в том случае, если остался лишь один доступный флаг – его же и необходимо вернуть.(1 сек)

```
	public void opponentAttack(Flag opponentFlag, Attack attack,
World beforeWorld, boolean wasAttackWon);
```

Этот метод вызывается после каждой атаки всех противников (но не после своих атак) и несет исключительно информационный смысл – игрок информируется о том, что игрок с флагом **opponentFlag** совершил действие **attack** при конфигурации мира **beforeWorld**, с успешностью **wasAttackWon**. (0.1 сек)

```
	public Attack attack(World world);

}
```

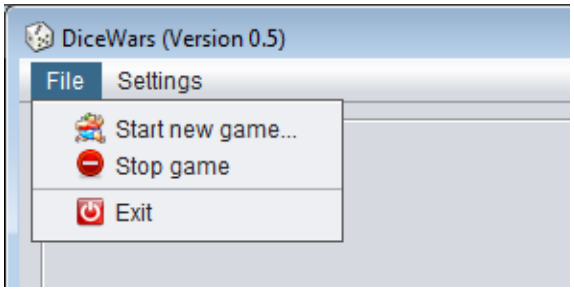
Этот метод вызывается на каждом шагу игрока. Метод должен вернуть корректный объект **Attack**. В случае некорректного нападения или ошибки исполнения будет зачтен конец хода. Для умышленного завершения хода следует возвращать **NULL**. (0.25 сек)

На все объекты игрока отводится не более 16 мегабайт оперативной памяти.

В случае если метод не укладывается в отведенное для его выполнения время или превышает ограничение памяти, выполнение метода останавливается, и объекты класса возвращаются в состояние, предшествующее вызову метода. Если из ограничений времени выбивается метод chooseFlag или attack, то засчитывается переход хода.

4.Игровая система

Для разработки игры всем участникам доступна игровая оболочка. Игра запускается с помощью **run.bat**. Для запуска потребуется установленный JDK, рекомендуется версия 6u18. В графическом интерфейсе нетрудно разобраться:



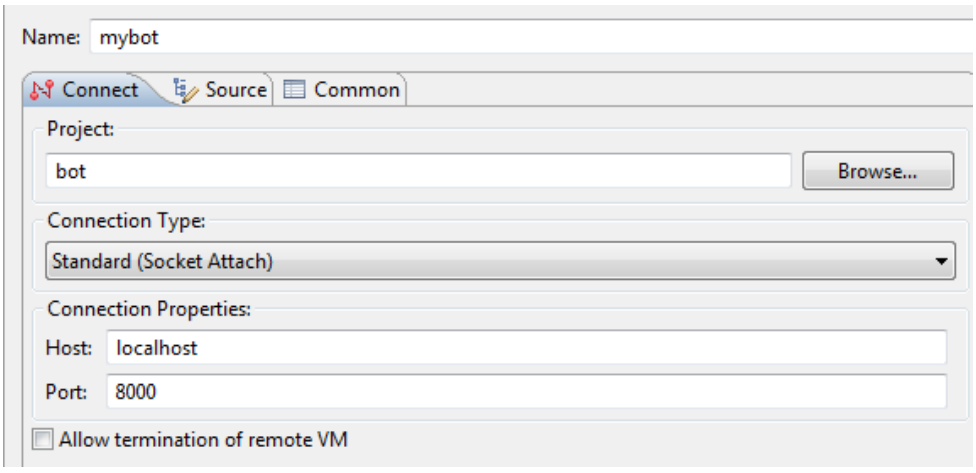
В разделе **Settings -> Players** можно выбрать необходимый набор игроков для матча.

Для подключения к игре своего бота необходимо разместить код класса формата .java в структуре пакетов в папку players. К примеру класс ru.*icl.dicewars.MyBot* должен располагаться в файле *…/players/ru/icl/dicewars/MyBot.java* . Далее класс нужно скомпилировать командой **compile** с путём от корня папки **players**, то есть для приведенного выше примера команда будет выглядеть так:

```
compile.bat ru/icl/dicewars/MyBot.java
```

Для корректной работы необходима правильная настройка переменной окружения **JAVA_HOME**. Новый класс сразу же появится в списке доступных игроков.

Для дебага ботов можно подключиться к игре из eclipse. Для этого в меню **Run → Debug Configurations…→ Remote Java Application** создаем новую конфигурацию следующего вида:



Разумеется, порт TCP:8000 необходимо разрешить на локальной машине в фаерволах. Настройки порта можно поменять непосредственно в файле **run.bat**.

Предоставляемый вариант игровой системы не отслеживает ограничения по памяти и времени.

5.Технический регламент

Команды, желающие принять участие в соревнованиях Code Game Challenge, должны подтвердить своё желание, отметив соответствующее поле в форме регистрации на сайте турнира http://icl.ru/turnir/order.php . Подтверждая своё участие, команды соглашаются с данным регламентом и гарантируют предоставление своего кода до конца срока приема решений. Если команда отказывается от участия в Code Game Challenge после того, как в регистрационной анкете было указано обратное, она должна предупредить об этом жюри не позднее, чем за неделю до последнего срока приема готовых ботов.

Окончательная схема проведения соревнований будет опубликована на сайте турнира после окончания срока приема заявок на участие. Способ проведения соревнований и турнирная сетка напрямую зависят от количества участников, поэтому команды, не заявившие о своем участии, не смогут быть допущены. Все сроки указаны на сайте турнира.

Команды обязаны выслать свои решения организаторам в указанные сроки. Решения принимаются в исходных кодах *.java. Созданный командами код должен играть честно – любая попытка вмешательства в работу игровой системы или других игроков может повлечь за собой дисквалификацию команды с соревнований, в том числе и с основного тура. Жюри соревнований оставляет за собой право визуальной проверки кода и может потребовать от команд пояснений по подозрительным фрагментам. Решения, регулярно вызывающие ошибки времени исполнения, переполняющие память или занимающие излишнее время могут быть сняты с турнира по усмотрению жюри.

Все матчи турнира будут проводиться на разных полях сражений. Поля сражений гарантированно связны и количество земель на каждом не превышает 100. В одном матче играют не более 8 команд. Игроки получают очки за каждый матч в зависимости от занятого места. Места распределяются в порядке вылета игроков. В случае если матч не заканчивается после 200 ходов, объявляется конец игры, и среди участников, оставшихся на поле, места распределяются по величине армии с учетом запаса. Если два и более игрока имеют одинаковую армию, то играет роль количество земель. Если и этот показатель равен, победителем становится тот игрок, чей первый ход был сделан позже.

По итогам матчей игрокам начисляются очки в зависимости от занятых мест. Чем дольше игрок продержался на поле боя, тем больше очков он получит.

Все подробности - на сайте турнира icl.ru/turnir

Разрешение спорных моментов реализуется через систему обмена сообщениями на сайте.

Игра Dice Wars не защищена авторскими правами. Концепция игры используется на некоммерческой основе.

Игровая система, все её компоненты и печатные материалы разработаны командой организаторов турнира и являются их интеллектуальной собственностью. Любое использование данных продуктов вне рамок турнира возможно только с разрешения разработчиков.