

На небольшой территории происходит сражение двух призывателей. Призыватели имеют на своей стороне несколько отрядов войск, каждый из которых состоит из определенного количества одинаковых существ. Цель каждого призывателя - используя свои отряды уничтожить вражеского призывателя.

Каждый призыватель имеет некоторое значение опыта, которое увеличивается при уничтожении отрядов противника, и значение энергии, которое можно потратить на призыв отряда существ рядом с собой. Умения призыва относятся к одной из нескольких школ (некромантия, анимализм, демонология, иные реальности, элементаризм... или робототехника, нанотехнологии, энергосущности, параллельные вселенные, биотехнологии...).

В зависимости от знания соответствующей школы зависит насколько сильных существ можно призывать, и в каком количестве. Каждая школа может быть доминирующей по отношению к другой, т. е. существа этой школы всегда наносят увеличенный урон по существам над которыми доминируют, а сами от них получают уменьшенный урон.

Каждый отряд и призыватель имеет значение инициативы, в зависимости от которого формируется очередность ходов. В свой ход призыватель может либо призвать за энергию отряд доступных существ, либо аккумулировать энергию, либо улучшить знание одной из школ за опыт. Призыватель в некоторой степени считается отрядом, так как имеет инициативу.

Отряды в свой ход могут либо перейти на клетку, дальность которой определяется скоростью отряда, либо совершить атаку. Отряды существ могут быть следующих видов: пехота - могут атаковать только отряд на соседней клетке, стрельки - могут атаковать по прямой любую клетку в радиусе стрельбы, если она не преграждена стеной, однако не могут стрелять в соседнюю от себя, поэтому бесполезны в ближнем бою; универсальные - могут как стрелять, так и атаковать вблизи; воскрешающиеся - должны являться одной из приведенных выше разновидностей, могут вместо атаки и передвижения попытаться воскресить умерших в своем отряде существ.

Разработать приложение, позволяющее описывать сражение и процесс игры. Обеспечить загрузку информации о школах и характеристиках существ из конфигурационных файлов, загрузку карты из файлов карты, а также возможность загрузки и сохранения текущего состояния приложения на диск. Для хранения в памяти информации о призывателях, отрядах, умений и местности используются соответствующие описатели.

Описатель пехоты содержит следующую информацию: название существа, школа, принадлежность, инициатива, скорость, параметры урона от удара, параметры защиты, количество, количество умерших, опыт за уничтожение.

Описатель стрелков содержит следующую информацию: название существа, школа, принадлежность, инициатива, скорость, параметры урона от стрельбы, дальность стрельбы, параметры защиты, количество, количество умерших, опыт за уничтожение.

Информация обо всех школах сведена в таблицу школ. Каждый элемент таблицы содержит название школы и список умений. Каждое умение содержит название, описатель существа, минимальное знание школы для этого существа, затрачиваемая энергия, коэффициент численности от знания школы. Для удобства доступа к таблице определен класс-итератор.

Описатель призывателя содержит следующую информацию: название призывателя, инициатива, принадлежность, максимальное и текущее здоровье, максимальная и текущая энергия, коэффициент аккумуляции, непотраченный опыт, список школ со значением знания этой школы.

Уровень представляет собой таблицу школ, список со всеми отрядами и призывателями каждой команды ранжированный по инициативе, прямоугольное поле разделенное на клетки: земля, стена, пропасть.

Приложение

Здесь приведён только пример приложения. Для создания карты для игровой сессии можно воспользоваться одним из существующих универсальных редакторов карт (например Tiled). Для реализации прикладной задачи можно добавить недостающие методы.

В данном примере программа имеет несколько режимов:

- ❖ Загрузка из конфигурационных файлов описания предметов и бойцов, а также карты с из файлов карты.
- ❖ Режим игры, в котором в зависимости от инициативы призыватели и отряды по очереди совершают ходы.
- ❖ Режим улучшения школы, в котором можно улучшить знание какой-либо школы.
- ❖ Режим призыва, в котором можно выбрать существ для последующего призыва.

Пример доступных действий для игрока:

- g - сделать передвижение в указанную точку;
- a - атаковать указанную точку;
- f - выстрелить указанную точку;
- h - восстановить свой отряд;
- t - аккумулировать энергию;
- c - меню призыва;
- o - меню улучшение школы;

Вариант консольного приложения

Режим игры, представляет собой псевдо графическую карту, на которой изображены объекты к примеру следующим образом (можно любым другим):

1. . - пустая клетка пола;
2. # - стена;
3. (ничего) - пропасть;
4. @ - призыватель пользователя;
5. * - вражеский призыватель;
6. a, b, c - отряды пользователя;
7. A, B, C - вражеские отряды;

Игра происходит в пошаговом режиме, очередность ходов зависит от инициативы. Режим улучшения школы и призыва представляет собой простой консольный диалог.

Вариант графического приложения

Практически аналогичен консольному варианту, но вместо псевдографики и консольного диалога используется какой-либо графический фреймворк (oxygine, Cocos2d, Qt/QML, Allegro). Красота спрайтов и анимации не важна (можно взять любые). Игра так же происходит пошагово. Вместо нажатия клавиш для некоторых действий может использоваться мышь.

Обеспечить выполнение следующих операций, при помощи которых можно реализовать процесс игры.

- Для уровня
 - получение/изменение размеров игрового поля;
 - получение/изменение типа клетки с заданной координатой;
 - установить отряд в заданную клетку;
- Для таблицы
 - добавить школу в таблицу;
 - добавить умение в школу;
 - изменить/получить существо по названию школы и умения;
 - получить количество школ;
 - получить количество существ в школе;
- Для любого отряда
 - получение/изменение параметров отряда;
 - нанести некоторый урон отряду;
 - получить урон, наносимый отрядом;
- Для воскрешаемого отряда
 - попытаться воскресить существа в отряде;
- Для призывателя
 - аккумулировать энергию;
 - получить опыт;
 - улучшить одну из школ;
 - нанести урон;

Порядок выполнения работы

1. На основе описания задачи определить состав классов, изобразить иерархию классов и схему их взаимодействия, а также состояния и необходимые методы с помощью UML.
2. Разработать, реализовать и отладить основные классы (призыватель, пехота, стрелки). Отладку методов можно реализовать с помощью JUnit тестирования.
3. Разработку и реализовать контейнерные классы вместе с необходимыми итераторами (таблицы). Выбор шаблона класса согласовать с преподавателем.
4. Разработать и отладить класс, реализующий работу приложения (ландшафт). Предусмотреть в классе возможность загрузки информации из конфигурационных файлов, а также сохранения и загрузки текущего состояния.
5. Реализовать консольное приложение.
6. (*) Реализовать приложение с графическим интерфейсом. В этом случае консольное приложение можно не реализовывать.