

Вариант Р Операция [ДАННЫЕ УДАЛЕНЫ] (***)

Команда оперативников “Организации” пытается уничтожить группу аномальных неопознанных существ, расположившихся в комплексе помещений. Помещения состоят из стен, перегородок и стёкол. Каждый оперативник имеет характеристики и набор снаряжения. Цель оперативников - уничтожить противников с минимальными потерями.

Команды оперативников и их противников ходят по очереди, при этом любой может ходить и выполнять действия в ход своей команды. Каждое действие или шаг отнимает определённое кол-во *очков времени* у сходявшего, которые пополняются каждый раз в начале хода команды. Когда у всех в команде кончились *очки времени*, ход передаётся другой команде.

В качестве предметов оперативники могут носить в инвентаре оружие, контейнеры с патронами и аптечки. Оперативники могут атаковать только используя оружие, извлечённое из инвентаря и находящееся в руках, и которое может быть заряжено определённым количеством патронов. В случае разрядки оружия оперативник должен его перезарядить, используя патроны из контейнера соответствующего типа. Каждый предмет имеет свой вес, невозможно носить предметов больше чем позволяет сила носящего.

Существа делятся на диких, разумных и одомашненных фуражиров. Дикое существо бросается на оперативников, могут атаковать только в ближнем бою и не имеют инвентаря. Разумные также не имеют инвентаря, но могут нести одно оружие, и атаковать с помощью него. Перезарядить его, или использовать какой-либо другой предмет они не могут, однако могут выбросить имеющееся оружие и поднять другое. Фуражиры имеют инвентарь, но не умеют атаковать, вместо этого они ищут предметы по всему полю, собирают их, и сбрасывают в точках складирования. Фуражиры и разумные знают о местоположении складских точек на карте, и могут использовать это знание для более продуманного взаимодействия. Однако никто из них не знает о местоположении разбросанных предметов, поэтому вынуждены их искать. В случае смерти оперативника, фуражира или разумного существа, предметы из его инвентаря падают на пол.

Каждый оперативник и существо имеет определённый радиус обзора, при попадании в который становятся видны противники, и по нему становится можно стрелять. Если противник вышел за пределы радиуса обзора, он становится снова невидимым для команды. Линия обзора преграждается стенами или перегородками, но не стеклом. Однако стрелять по противнику за стеклом нельзя. Если выстрелить по стеклу или перегородке, они разрушатся, переставая мешать стрельбе.

Разработать приложение, позволяющее описывать сражение и процесс игры. Обеспечить загрузку информации о предметах и солдатах 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 предметах, персонаже, врагах и нежити используются соответствующие описатели.

Описатель оружия содержит название, параметры урона, затраты очков времени на стрельбу и перезарядку, тип патронов, текущее и максимальное количество патронов в оружии, вес.

Описатель контейнер патронов содержит название, тип патронов, максимальное и текущее кол-во патронов в контейнере, вес.

Описатель аптечки содержит название, кол-во восстанавливаемого здоровья и кол-во очков времени на использование, вес.

Информация обо всех носимых предметах оперативником сведена в таблицу инвентаря.

Каждый элемент таблицы содержит описатель предмета и его номер в инвентаре. Для удобства доступа к таблицам определен класс-итератор.

Описатель оперативника содержит следующую информацию: имя, характеристики(максимальное и текущее здоровье, максимальное и текущее кол-во *очков времени*, кол-во очков времени затрачиваемое на шаг и перезарядку, сила(носимый вес), точность стрельбы, радиус обзора), таблица носимых предметов, текущее активное оружие.

Описатель дикого существа содержит имя и характеристики(максимальное и текущее здоровье, максимальное и текущее кол-во *очков времени*, кол-во очков времени затрачиваемое на шаг, точность и урон от атаки, радиус обзора).

Описатель разумного существа содержит имя, носимое оружие и характеристики(максимальное и текущее здоровье, максимальное и текущее кол-во *очков времени*, кол-во очков времени затрачиваемое на шаг, точность стрельбы, радиус обзора).

Описатель фуражира содержит имя, таблицу носимых предметов и характеристики(максимальное и текущее здоровье, максимальное и текущее кол-во *очков времени*, кол-во очков времени затрачиваемое на шаг, радиус обзора).

Уровень представляет собой списки с командами и набор прямоугольных полей, разделённых на клетки: пол, стена, стекло, перегородка, складская точка. Каждая клетка содержит список предметов, лежащих на ней.

Обеспечить выполнение как минимум следующих операций, при помощи которых можно реализовать процесс игры:

- Для уровня:
 - получение/изменение размеров игрового поля;
 - получение/изменение типа клетки с заданной координатой;
 - добавление/удаление предметов на разрешённые клетки поля;
- Для таблицы:
 - получить кол-во предметов в инвентаре;
 - получить/изменить предмет инвентаря;
 - добавить предмет в инвентарь;
- Для оружия:
 - получить наносимый урон и сделать выстрел;
 - получить результат попадания/непопадания оружием(зависит от характеристик бойца, оружия и дальности стрельбы);
 - перезарядить оружие;
- Для контейнера патронов:
 - получение/изменение параметров контейнера;
 - извлечь определённое кол-во патронов;
- Для оперативника:
 - получение/изменение параметров персонажа;
 - получить текущий носимый вес;
 - выбрать оружие для использования;
 - взять предмет, если возможно;
 - бросить предмет(кроме активного оружия);
 - получить наносимый урон;
 - сделать шаг;

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(оперативник, враги, предметы). Отладку методов можно реализовать с помощью Unit тестирования.
3. Разработать и реализовать контейнерный класс вместе с необходимыми итераторами (таблицы). Выбор шаблона классов согласовать с преподавателем.
4. Разработать и отладить класс, реализующий работу приложения (уровень). 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.

Вариант Р

Операция [ДАННЫЕ УДАЛЕНЫ](***) 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 работает в два этапа:

- ❖ Загрузка из конфигурационных файлов описания предметов, врагов и оперативников, а также карты с из файлов карты.
- ❖ Режим игры, в котором каждый боец может передвигаться по уровню, брать предметы и стрелять.
- ❖ Режим инвентаря персонажа, в котором можно бросить предметы, а также выбрать активное оружие и использовать аптечки.

Пример доступных действий для игрока:

- 1, 2, 3 - переключиться между оперативниками;
- стрелки/h,j,k,l - передвижение;
- g - поднять предмет, если можно;
- i - перейти в режим инвентаря;
- f - указать цель и выстрелить;
- r - перезарядить активное оружие;
- e - завершить ход команды.

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. = - перегородка;
4. - - стекло;
5. a, b, c - оперативники, подконтрольные пользователю;
6. A, B, C - противники;
7.) - предметы на земле;

Игра происходит в пошаговом режиме, каждое действие стоит соответствующее число очков времени. Противники ходят только в ход своей команды.

Вариант графического приложения

Практически аналогичен консольному варианту, но вместо псевдографики и консольного диалога используется какой-либо графический фреймворк (oxugine, 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.