Миссия 1

Общая цель: обучить игрока основным понятиям в программировании (переменные, условные операторы и циклы)

Цели миссии: починить робота механиком, после чего отправить его в командный центр -> протестировать робота, задав ему разовую задачу, а именно походить по окрестностям в поисках ресурсов -> добыть ресурсы -> доставить ресурсы на базу -> построить здание для переработки металла -> переработать весь имеющийся металл в части для роботов -> построить здание для создания роботов -> создать 5 роботов механиков.

Предыстория: (надо проработать, не так срочно)

Левел дизайн: на карте находится командный центр, разрушенный робот, главный персонаж (механик, управляемый игроком) и блок ресурсов (камня или чего-то подобного), скрытый за туманом войны

Геймплей:

Выводится окно с инструкцией по починке сломанного робота, после чего, механик, управляемый игроком напрямую, чинит разобранного робота. Затем, игрок отправляет механика в командный центр. После этого, появляется задача-протестировать робота.

Итак, мы починили робота, и такие  
"Надо потестить как он ходит"  
Говорим игроку написать разовую команду для робота:  
  
int moveForward=1;  
int moveBackwards=2;  
int turnRight=3;  
int turnLeft=4;  
  
unit.move(moveForward);  
unit.move(turnLeft);  
unit.move(moveForward);  
  
Юнит продвигается вперёд на условные 10 метров за каждую команду, но он не может остановиться, пока выполняет команду, потому, в первый раз он проходит ресурсы мимо  
  
После этого команды игрока очищаются, и ему нужно прописать возвращение робота к ресурсам командой unit.move(moveBackwards) и сделать так, чтобы он повернулся лицом к ресурсам (они будут слева от него). Команда moveBackwards будет возвращать робота на 1 условный метр назад. Переменные не изменяются (их вообще можно прописывать в отдельном окошке на самом деле, чтоб игроку было по игре понятно, где что). Итак, вторая разовая команда для робота будет:  
  
unit.move(moveBackwards);  
unit.move(moveBackwards);  
unit.move(turnLeft);  
  
Таким образом, робот дошел до ресурсов и повернулся к ним лицом  
  
Теперь ему надо понять, полезен ресурс или нет (это можно сделать фишкой такой, что типа если нет какого-то условия для нормальной добычи и хранения ресурсов, то собирать его нет смысла)  
  
Игрок пишет следующую программу:  
  
bool message=0;  
string resource;  
  
unit.skanPlace();  
cin»resource;  
unit.sendMessage(resource);  
if(message==true){  
unit.collect();  
}  
  
Тут булевая переменная отвечает за то, полезен ресурс или нет (это будет базовый ресурс для постройки зданий, который можно добывать при любых условиях). Стринг нужен для того, чтобы игрок вбил название ресурса.  
Первым делом, чтобы узнать название ресурса, игрок его сканирует, после чего ему возвращается в консоли название ресурса.  
Игрок вбивает название ресурса, после чего робот отправляет сообщение на базу. Это занимает какое-то время, пока не пришла обратная смс-ка с базы, что надо добывать.  
Как только робот получил смс-ку, он добывает ресурс.

После этого роботу надо донести ресурсы до командного центра.  
Я предлагаю сделать карту через клеточки, у каждой из которых будет x и у координаты. Так будет легче простраивать маршрут через программирование. Командный центр всегда является 0:0 в таком случае, ТК это единственная определенная точка на карте, присутствующая в любой миссии.  
Игрок снова прописывает разовую команду:  
  
string place;  
  
unit.move(turnLeft);  
unit.move(moveForward);  
unit.move(moveBackwards);  
unit.move(moveBackwards);  
unit.move(turnLeft);  
unit.move(moveForward);  
cin»place;  
unit.putIn(resource, place);  
  
Стринговая переменная place нужна для того, чтобы игрок вбил название места, куда он кладет ресурсы (в данном случае, это-командный центр)  
Функция putIn используется для того, чтобы положить что-то куда-то (в данном случае ресурсы в командный центр)

Изначально у игрока будет какое-то количество ресурсов, а того, что он донесет до базы-будет достаточно для того, чтобы построить здание по переработке ресурсов.  
Игрок снова прописывает разовую команду:  
  
int x;  
int y;  
  
cin»x»y;  
unit.moveToCoordinates(x, y);  
cin»place;  
if(unit.skanPlace==0){  
[unit.build](https://vk.com/away.php?utf=1&to=http%3A%2F%2Funit.build)(place);  
} else {  
unit.move(moveBackwards);  
[unit.build](https://vk.com/away.php?utf=1&to=http%3A%2F%2Funit.build)(place);  
}  
  
Таким образом, игрок построит здание там, где ему захочется, после чего идет проверка на идиота, которая состоит в том, что игрок не сможет построить здание на клетке, которая занята каким-либо объектом. Функция moveToCoordinates работает так, что если игрок ввел координаты клетки, которая занята каким-либо объектом, то юнит останавливается рядом с ней.  
  
Как только игрок построил здание, ему нужно переработать имеющийся у него другой ресурс (железо/сталь/что-нибудь подобное) в части роботов. Для этого он пишет разовую команду для здания по переработке материалов.  
  
int details=1;  
  
while (resource.metal()!=0){  
constructionResources.makeResources(details);  
}  
  
Переменная details нужна для функции makeResources для того, чтобы металл перерабатывался именно в детали для роботов. В следствие этого цикла, весь металл который был у игрока, переработается в детали для роботов.  
  
Далее игроку предлагается самостоятельно (по предыдущему алгоритму) построить здание для разработки новых роботов.  
  
После этого, для завершении миссии, игроку требуется создать 5 новых роботов, для чего он пишет еще один разовый код:  
  
int workers=1;  
  
for(int i=0; i++; i<5) {  
constructionRobots.makeRobots(workers);  
}  
  
Переменная workers нужна для функции makeRobots для того, чтобы собрать именно роботов рабочих. В следствие цикла, игроком будет создано 5 роботов рабочих, после чего миссия завершится.