Новый разум. Прорыв

Требования к ПО. Версия 0.2

Платформа и язык программирования

- Язык программирования C++
- Кросс-платформенное приложение, консольное, нет зависимостей от внешних библиотек.

Концепция

Игра представляет собой парсерный шутер. Парсерный означает, что игра должна обрабатывать введённую текстовую команду игрока. Дополнительно, существует менюшный подрежим синтеза в котором можно выбрать по пунктам что синтезировать.

Навигация

Существует два вида карты. Карта местности (2D), перемещение по которой выполняется командами "север, юг, запад, восток, вверх, вниз" и карта боя (1D), перемещение по

которой происходит командами вперёд и назад. При появлении ГГ на одной локации с монстром карта боя включается автоматически. При этом, есть возможность уйти из текущей локации во время боя (сбежать). Также и монстр может покинуть локацию и, если герой останется один, то карата боя выключится. Основная цель - достигнуть босса уровня и победить его. Топология карты представлена в **Приложении 1**. Существует единственный центральный коридор (локации 0-9, в зеленом контуре), продвигаясь по которому игрок достигает босса (который на клетке 9). Секции коридора с 1-9 закрыты автоматическими дверями, которые открываются, когда игрок находит ключ в боковом ответвлении. Например, чтобы отпереть проход 1->2, ему необходимо из 1-й локации отправиться на северо-запад(nw) в локацию 10. При входе его встречает команда стражей (g1). После победы над ними, игрок двигается наверх (u) и попадает в локацию 13. Затем, оттуда проходит наверх (u) в локацию 15, где находиться ключ, естественно под охраной.

Оружие

Крио-нож

Штырь, кромка которого охлаждается до температуры, близкой к абсолютному нулю.

Эффекты (списываются автоматически, если были синтезированы):

- Прыжок. Если вы выберете нож, и будете двигаться вперёд к противнику в течение двух ходов подряд, то вы подскочите к ближайшему противнику и поразите его ударом ножа.
- Микровозврат. При ударе ножом наноситься второй удар автоматически.
- Вихрь. При попаданию ножом по противнику, нож описывает дугу в радиусе 5 клеток и поражает всех стоящих вокруг на 30% от силы основного удара.

Роботизированный револьвер.

Старая-добрая пушка, но с более удобной управлением обоймами.

Патроны:

- Обычные
- Разрывные. Двойной урон.
- Толкатели. Отбрасывают монстра назад на 5 клеток после попадания.
- Зажигательные. Продолжают наносить урон 100% в течение последующих 4-х ходов.

Система синтеза

Подсистема синтеза нужна чтобы генерировать из кристаллов энергии (КЭ) полезные девайсы - патроны, эффекты, лечилку.

> Вас приветствует интегрированная подсистема синтеза! Для вашего удобства работа с ней идёт в режиме диалога. Сначала выдаётся статистика по имеющимся у вас ресурсам. Выберите подходящий элемент и введите количество, синтезатор выдаст вам готовый продукт. Вы всегда сможете узнать подробнее про каждый элемент перед покупкой, просто выберите номер и перейдите в справку. Удачного пользования!
>
> Имеется: кристаллов энергии(Э)-40. Выберите номер позиции для синтеза:
>
> 0 - выход
>
> 1- патроны для револьвера (Э - 10)
>
> 2- шприц (Э - 20)
>
> 3 - модуль прыжка (крионож) (Э - 30)
>
> 4 - модуль микровозврата (крионож) (Э - 30)
>
> 5 - модуль "вихрь" (крионож) (Э - 30)
>
> 6 - разрывные патроны (Э - 30)
>
> 7 - патроны-толкатели (Э - 30)
>
> 8 - зажигательные патроны (Э - 30)

После выбора необходимо ввести количество, проверить хватает ли средств и пополнить их количество у героя. Количество энергии приведено для примера.

Правила боя

1. Бой включается автоматически, если на текущей локации есть вражеские монстры. Обоснование: у нас аналог шутера, частые бои, бесшовный переход от режима боя к режиму не боя
2. Бой выключается автоматически, если на текущей локации нет вражеских монстров. Обоснование: у нас аналог шутера, частые бои, бесшовный переход от режима боя к режиму не боя

3. Во время боя доступны дополнительные глаголы боя.
4. Каждая локация имеет свой размер линейного поля боя и начальное положение для игрока и монстров. Обоснование: на разных локациях можно делать разную тактическую обстановку - враги по одну сторону, окружили, бегут за героем и т.п.
5. Чтобы переместиться к концу карты, надо использовать глагол “вперёд” (синоним “дальше”). Обоснование: глагол удобен для пространственного представления перемещения и не конфликтует со сторонами света. Конец карты положительное направление и персонаж как бы уходит вдаль.
6. Чтобы переместиться к началу карты, надо использовать глагол “назад” (синоним “ближе”). Обоснование: глагол удобен для пространственного представления перемещения и не конфликтует со сторонами света. Конец карты положительное направление и персонаж как бы подходит к началу - во весь экран.
7. Если враг приближается к герою, то выдача будет “противник подошел ближе”. Обоснование: Хоть враг идёт в противоположном направлении, для более удобного анализа ситуации говорим, что он подошел ближе. Причем, ближе имеется ввиду к главному герою.
8. Если враг отдаляется от героя, то выдача будет “противник отошел дальше”. Обоснование: Хоть враг идёт в противоположном направлении, для более удобного анализа ситуации говорим, что он отошел.
9. Разрешается проходить мимо врагов. Обоснование: если будем стопориться об врагов, то они будут тоже стопориться о нас, это сильно ограничивает разные тактики и приводит к неопредёленным ситуациям (союзник вдруг появился за спиной врага).
10. Разрешается стрелять в любого врага на карте. Обоснование: мощность оружия зависит от дальности, поэтому эффект понижения точности стрельбы учитывается. Дополнительно могут быть введены аникомбинации-штрафы, если стреляем в монстра, перед которым уже кто-то стоит.
11. Враги могут проходить мимо героя. Обоснование: аналогично п. 9
12. Если игрок попытается уйти с карты, он получает штраф в размере удара от всех монстров в ближнем бою. Не уверен, что работает... Обоснование: надо дать игроку возможность сбежать с поля боя в любом направлении, но надо делать это с наказанием, чтобы он не прыгал между локациями. Будем считать, что когда игрок сбегает, но пробегает мимо монстров и все его задевают.

13. После победы над монстром прибавляются кристаллы энергии для оперативного синтеза

Построение карты и отладка

В идеале должен быть текстовый файл (CSV) с таблицей вида для описания карты locations.csv:

```
id локации; описание; id-двери; id-ключа; количество стрелков; количество штурмовиков;
количество огневиков; количество лутбоксов;
0; Коридор бункера. Стартовая точка. Проход идёт на север; 0; 0; 0; 0; 0; 0
1; Коридор бункера уровень 1. Проход идёт на запад к следующему уровню; 1; 0; 0; 0; 0;
0;
...
10; Начало зеленого коридора. Можно выйти на юго-восток или двинуться дальше на верх,
на запад и вниз; 0; 0; 1; 0; 0; 1
...
15; Тупик под навесом. Можно выйти вниз; 0; 1; 0; 2; 0; 0
```

Получается, мы можем указать в каких локациях содержаться двери, которые открывают проход на следующий уровень (всегда id-больше предыдущего).

Отдельный файл отображает связь между идентификаторами локаций. connections.csv:

```
id исходной;id результата;направление
0;1;n
1;0;s
1;10;nw
10;1;se
...
```

Еще файл предназначен для установки параметров оружия, HP героя, монстров. Нужен для дальнейшего баланса.

Расчёт очков

Очки за игру рассчитываются как суммарный нанесенный урон.

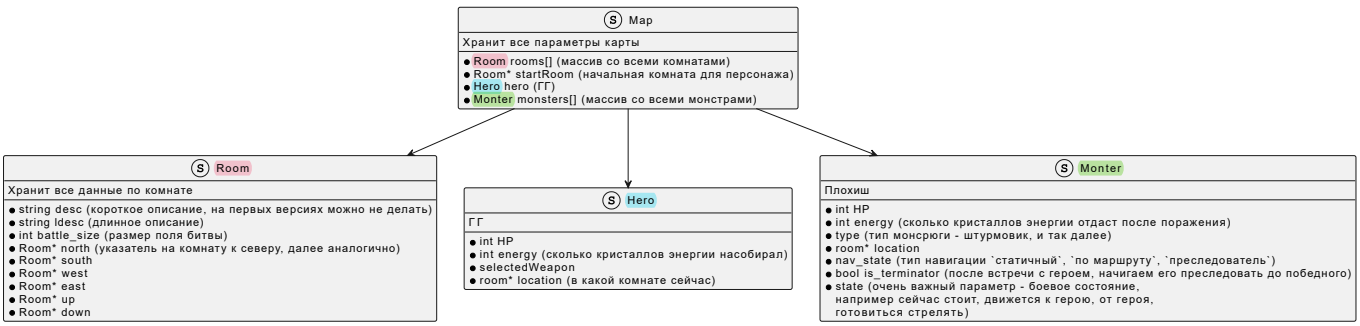
Порядок этапов разработки

1. Бетта 0.1 Только загрузка карты (можно урезанной) и навигация по ней.

2. Бетта 0.2 Подключение битвы и сражение и фиксированным инвентарем с каждым из типов монстров (кроме босса).
3. Бетта 0.3 Подключение синтеза, расстановка по всей карте.
4. Бетта 0.4 Подключение всех команд, добавление босса.
5. Далее до версии 1.0 доработки.

Архитектура ПО

Самая простая архитектура - процедурно-ориентированная. Есть набор структур и массивов для хранения. Обработка по сути меняет карту.



Псевдокод главной функции:

```
int main()
{
    Map map;
    //1. Считывание комнат из файла locations.csv, заполнение map.rooms в части
описаний.
    //пока можно заменить коснтантами
    //id комнаты - просто позиция в массиве, напрямую в структуры не кладётся
    //параллельно надо заполнять массив с монстрами и давать указатель на текущую
комнату
    //2. Считывание направлений из connections.csv, заполнение map.rooms в части
указателей

    //3. Основной цикл игры
    while(true)
    {
       cin >> user_inpout;
       std::string res = game_loop(user_inpout, map);
       cout << res;
    }
}
```

Основной цикл обработки следующий:

```
std::string game_loop(std::string input, Map& map)
{
    //функция парсит строку и переводит текст в id команды и её строковый аргумент,
может отстусвовать
    CommandId, CommandArg = parse_input(input);
    //флаг, что необходимы дополнительные команды и действия для боевого режима
    bool is_battle_mode = getMonstersOnLocation(map.hero.location);
    std::string output;
    switch(CommandId)
    {
       case CmdNorth: //хотим идти на север
           if (map.hero.location->north) //можем ли пойти на север
           {
                map.hero.location = map.hero.location->north; //переходим
                output = map.hero.location->ldesc; //для возврата даём описание
           }
           else //не можем пройти
           {
               output = "Туда не пройти!"
           }
       break;
       //подключаем команды, которые разрешены в боевом режиме
      ...
    }
    //проверяем, нужна ли обработка боевого режима
    if (is_battle_mode)
    {
        processBattle(map.hero,getMonstersOnLocation(map.hero.location))
    }
    //после обработки ввода пользователя, надо выполнить обновление монстров на карте,
они могут перемещаться. Кто-то в режиме боя, кто-то ползает по карте
    //на выходе мы получаем строку от их действий (если они что-то выполнили, например
вошли в комнату к герою)
    output += updateMonsters()
    return output;
}
```

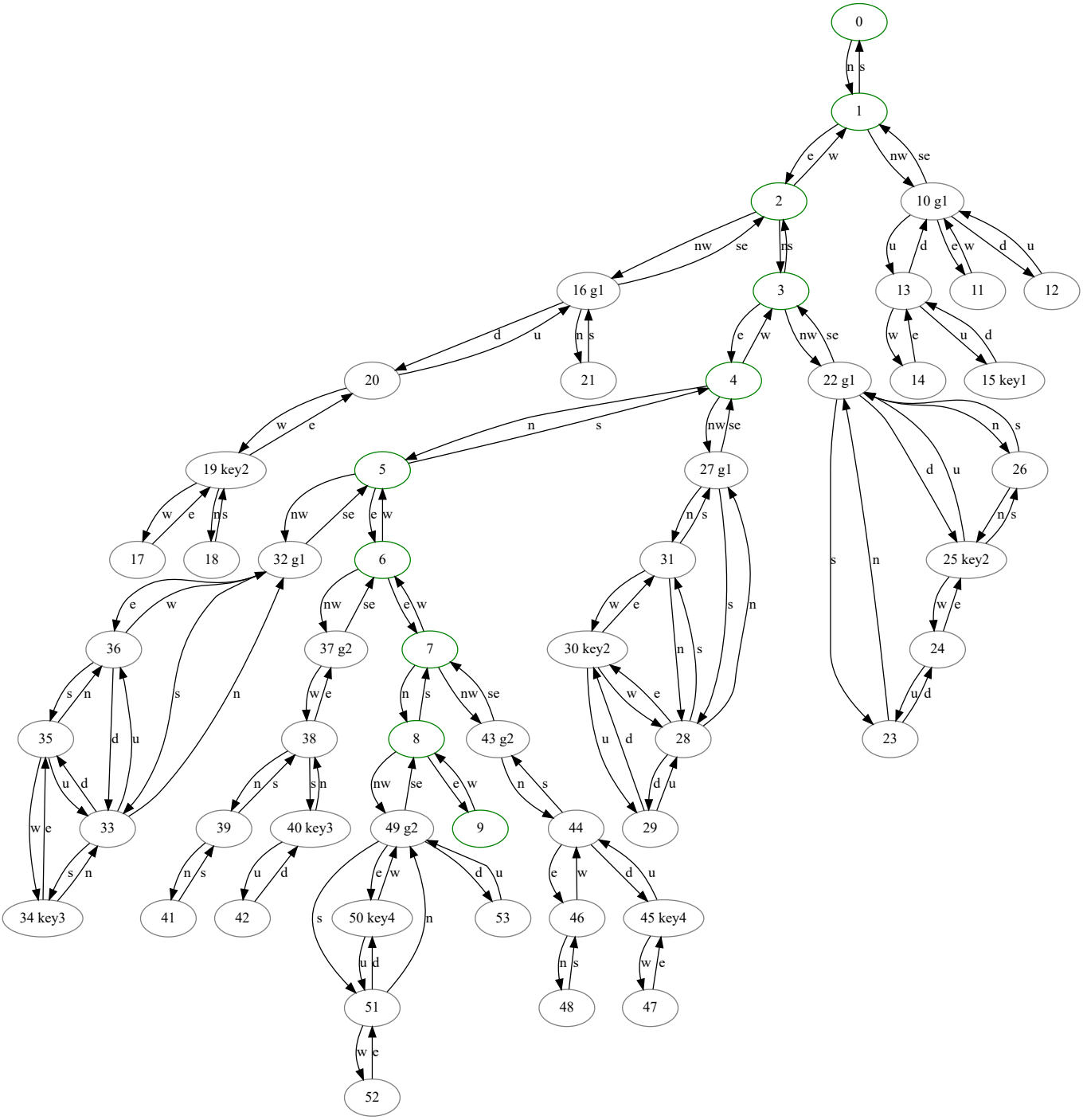
После смерти монстра, его локация становиться 0, его дальше не обслуживаем.

Обслуживание боевого режима:

```
//получить список всех моснтров на локации
Monsters* getMonstersOnLocation(Room* room)
//обработка битвы
processBattle(hero, monsters_list)
{
    //для каждого монстра смотрим его тип и состояние
    for (monster : monsters_list)
    {
       if (monster->type == Shturmovic)
       {
           //проверяем в какой стороне герой и, если не в плотную смещаемся к нему
           //если рядом, то удар
       }
    }
}
```

Дополнительные функции, которые получат урон для данного типа оружия в диапазоне.

Приложение 1. Карта местности



Приложение 2. Команды игры

Команда (список однотипных команд)	Пример	Описание
север(с), юг(ю), запад(з), восток(в), северо-запад(сз), северо-восток(св), юго-запад(юз), юго-восток(юв), вверх(вв), вниз(вн)	>с	переводит ГГ в новую локацию. Если такого направления нет, необходимо сообщить - "вы не можете пройти в этом направлении"
вперед(вп), назад(нз)	>вп	Перемещает персонажа в рядом стоящую клетку во время боя.
ждать (ж)	>ж	Пропускает ход, при
осмотреть (осм, о)	>о себя >о синего >о	Осматривает себя (показывает характеристи хп, количество боеприпасов, ресурсов). Осматривает монстра - показывает его описание. Осматривает локацию - повторяет её описание.
уколоть(укол)	>уколоть себя	Прибавляет НР ГГ из синтезированной лечилки
нож, пистолет(револьвер)	>выбрать нож	если во фразе встречается оружие, то всегда его выбираем, при этом оповещаем игрока, что новое оружие выбрано.
удар	>ударить синего	если во фразе есть "удар", затем надо смотреть часть имени монстра. Работает, когда выбран нож.
стрелять	>застрелить синего	если есть "стрел", то затем надо смотреть часть имени монстра. Работает, когда выбран револьвер.
патроны(обойма)	>патроны разрывные	выбор типа патронов, если уже были синтезированы.
открыть	>открыть бокс	если есть "откры", то затем смотрим название лутбокса на локации. Для всего остального - ошибка

Команда (список однотипных команд)	Пример	Описание
синтез	>синтез	открывает меню синтеза
помощь (справка)	>помощь	выдаёт список команд
супержизнь, супероружие, открытьвсе, стопрандом	>супержизнь	чит-коды для отладки, много хп, оружия и всё открыто. Также можно выключить рандом.

Приложение 3. Характеристики персонажей (defaults)

ГГ. HP=100, КЭ=40.

"Вы выглядите как трехметровый гигант, вместо волос и одежды - голая плоть, без гендерных признаков. Кожа сверхпрочная, состоит из алмазных наностержней, которые, переплетаясь, выделяют темный переливающийся рельеф мышц. Толстый мускульный слой прикрывает искусственные органы, сосредоточенные в области груди. Ваши глаза - два черных алмаза, скрывающих мощную оптическую систему и искусственный разум. Оружие находится прямо в теле в районе таза, закрытое прочной мембраной. Часть спины отведена под синтезируемые предметы также закрыта мембраной и позволяет быстро их доставать. Декоративный рот и нос сделаны из эстетических соображений, для редких контактов с людьми."

Стрелок

HP=10, КЭ=4, Обычный стрелок. Никуда не ходит, только стреляет и иногда ожидает. Самая сильная атака на средней дистанции, плох на дальней и близкой, правда мало жизни. Иногда может восстановить своё здоровье. Урон: 3-15.

```
calcHit(D) = {
      if (D<=3) return rangernd(5,15);
      else if (D<10) return rangernd(3,5);
      return rangernd(3,5);
}
```

Штурмовик

hp = 30, КЭ=6 Обычный штурмовик. Всегда двигается на Вас, сражается только врукопашную. Иногда может в ярости перекинуть вас в дальний конец карты. Среднее количество жизни и низкая атака. Удар:

```
calcHit(D) = {
if (D==0) return rangernd(2,4);
}
```

Огневик

hp = 10, gen = 5. Обычный огневик. Бродит туда-сюда и периодически отстреливается. Может выпустить нефтяной шар, который будет гореть на жертве какое-то время

Логика работы:

```
//с вероятностью 30% меняем тактику ходить в одну сторону, другую или стрелять
      if ( rangernd(1,10) <= 3 ) {
         if (self._is_move == true) {
            self._is_move = nil;
         }
         else {
            self._is_move = true;
            self._curr_dir = rangernd(MOVE_DIR_FORWARD,MOVE_DIR_BACKWARD);
         }
      }

      if (self._is_move) {
         //если уперлись в тупик, то идём обратно
         if ( (self._curr_dir==MOVE_DIR_FORWARD) && (self.pos == loc._field_size) )
self._curr_dir = MOVE_DIR_BACKWARD;
         else if ( (self._curr_dir==MOVE_DIR_BACKWARD) && (self.pos == 0) )
self._curr_dir = MOVE_DIR_FORWARD;
      }

      //Стреляем огненным шаром с нефтяной плёнкой
      if ( rangernd(1,10) <= 3 && (global.is_easy==nil) ) {
         "<br><<ZAG(self,&sdesc)>> выпустил по вам огненный шар из нефти!<br>";
         if (!FireBallLong.isIn(Me)) FireBallLong.moveInto(Me);
         //обновляем урон от шара
         FireBallLong._numFireTimeLeft = FireBallLong._maxFireTimeLeft;
         //выполняем урон сразу же
         FireBallLong.preBattleAny(Me);
      }
    ...
   calcHit(/*int*/D) = {
      if (D<=3) return rangernd(8,12);
      else if (D<6) return rangernd(5,8);
      return rangernd(0,1);
   }
```

Босс уровня - взбесившаяся приставка, типа денди.

Не спрашивай почему такой, народ очень хотел.

HP=200, есть ближняя и дальняя атака. Атакует периодически, но мощно, всякими мелкими запчастями. Иногда может телепортиться в другую часть карты. Аналог пылесос из предыдущей части:

```
//Логика - всегда следуем за персонажем, находимся рядом

      //Обновляем атакующее состояние - до выстрела.
      self._curr_state += 1;
      //Сообщаем о его действиях
      if (self._curr_state == 1) "Пылесборщик втягивает в себя окружающую пыль и
мелкие предметы.";
      else if (self._curr_state == 3) { "Сборщик переключает режимы, готовится к
выдуву."; play_sound('vacuum.ogg');}
```