1 Состав и обязанности разработчиков

- Волков В.Д.
  - построение основной структуры программы
  - механизмы обработки событий игры на глубинном, не GUI, уровне
  - часть вспомогательных функций
- Иртикеев М.Н.
  - обработка событий игры на GUI уровне
  - создание интерфейса для корректной работы игры
- Гуняшов Н.Н.
  - разработка режима сетевой игры
  - создание игрового редактора юнитов

2 Постановка задачи

В качестве объекта нашей работы мы решили выбрать стратегию реального времени. Основными критериями нашего выбора являются:

- Возможность объединить множество аспектов программирования в рамках одной программы.
- Возможность работать в коллективе (одно из важнейших качеств программиста), при этом не испытывая проблем с разделением труда и совместной разработкой.
- Приятное и не требующее особых навыков тестирование.
- Возможность почти неограниченного усложнения функционала (в случае наличия времени можно сделать свой игровой движок, искусственный интеллект или игру по сети и т.д.).
- Легко найти тестеров, что несомненно может быть важно для некоторых аспектов (в частности игре по сети).

3 Методы разработки

Основными методами, выбранными нами при разработке являются **классы** и **наследование**. Также как технологические решения можно выделить большое количество элементов типа **std::list**, которые используются для быстрого удаления элементов, а невозможность быстрого доступа по индексу(итератору) компенсируется частым использованием **указателей**. Использование ссылок оказалось затруднительным в связи с невозможностью std::vector<&объект> и std::list<&объект>. Все **юниты**, **здания** и **прочие типы объектов** являются объектами, а не классами, что позволяет сделать удобный **редактор**, в котором пользователям можно будет создавать свои игровые единицы. Прототипы игровых единиц хранятся в **std::vector** прототипов для юнитов, зданий и прочих классов в классе Game, откуда для нужд игроков будут делаться играбельные "слепки".

Также можно отметить что многие алгоритмы в игре будут связаны с **графами** и поэтому будет много **обходов графов в ширину**, **определение кратчайшего расстояния между вершинами** и прочих алгоритмов из теории графов.

4 Выбор языка разработки

Выбор пал на язык c++, так как он даёт:

- возможность работать как на высоком, так и на низком уровне
- понятен большинству программистов
- позволяет легко реализовать возможности ООП
- имеет множество полезных для разработки данного проекта библиотек
- имеет высокую скорость работы по сравнению со многими другими языками
- является компилирующимся языком, что усилит быстродействие игры

5 Выбор других средств разработки

- операционные системы: GNU/Linux Arch, Windows 10
- система контроля версий: Git
- система компьютерной верстки: TeX (LaTeX)

6 Структура

Создана посредством импорта в umbrello5 текущего кода

* методы в классах могут меняться в процессе разработки, также вероятно изменятся некоторые не очевидные названия