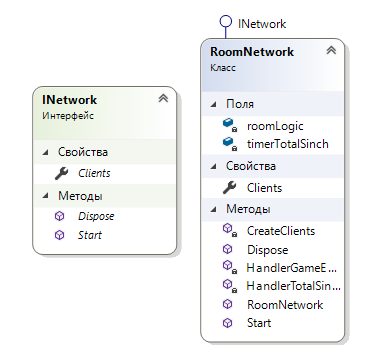
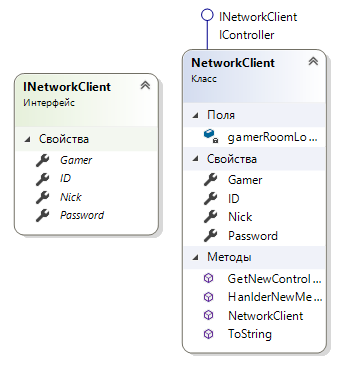


Игровая комната состоит из двух классов:

NetworkRoom – обеспечивает прием и передачу сообщений пользователю. Он передает общее состояние комнаты и события, которые произошли ней. Сообщения пришедшие от пользователя комната передает в комнату, классу игрока, который совершил некоторые действия.



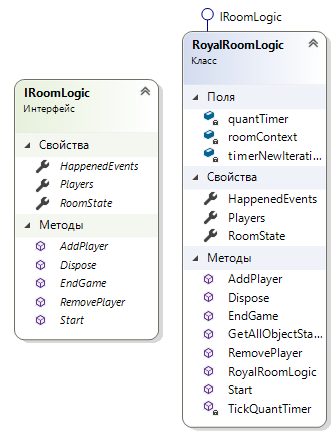
Clients – список объектов NetworkClient



LogicRoom – ничего не знает о сетевой части, организует логику работу игры.

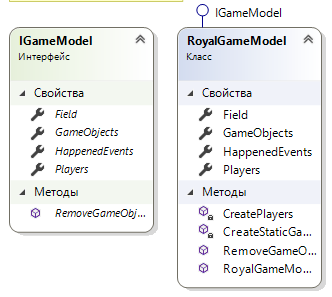
RoomState – возвращает объект описывающий состояние игровой карты

HappenedEvents – возвращает список содержащий объекты описывающие события произошедшие в игре (представлена ObservalableCollection)

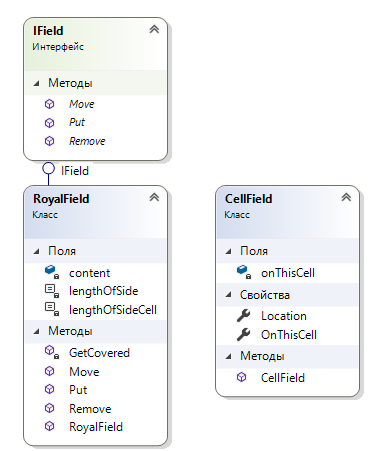


Players – список игроков в комнате. Игрока представлен объектом Gamer

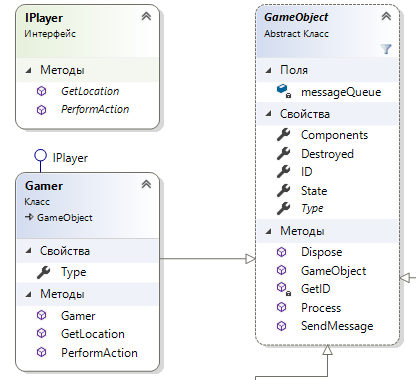
RoomContext – модель игры



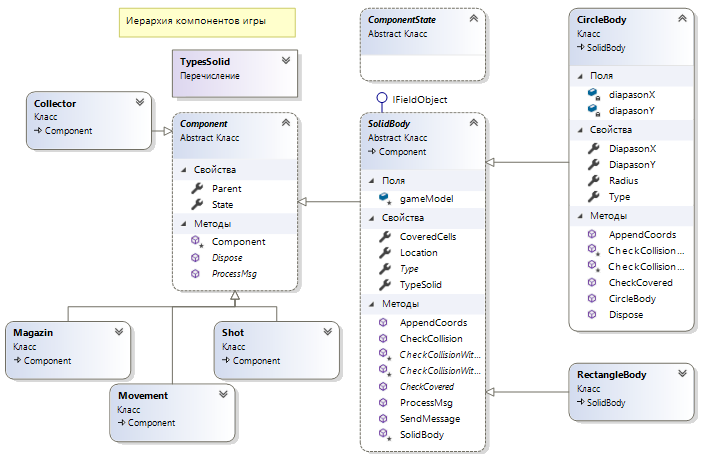
Содержит в себе игровую карту Field, список игровых объектов, список событий(действий) произошедших на карте и список игроков.

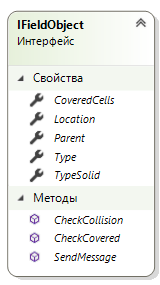


Каждый GameObject состоит их компонентов, каждый из которых реализует часть его логики



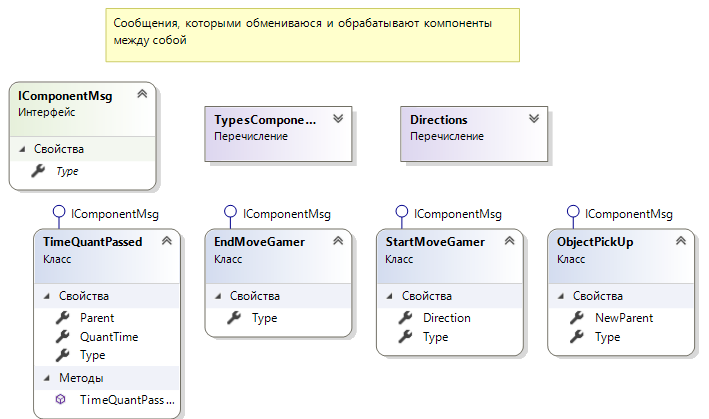
Примеры компонентов:



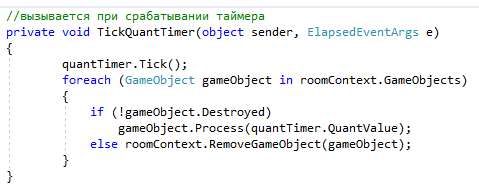


Взаимодействие между компонентами осуществляется с помощью сообщений

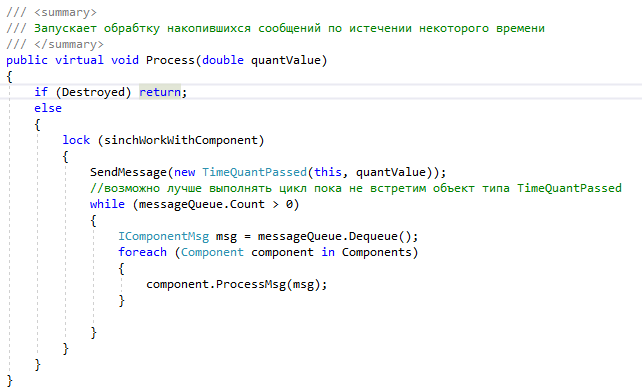
Примеры сообщений:



В объекте RoyalRoomLogic находится таймер *timerNewIteration*, который каждые 50 миллисекунд вызывает метод (в отдельным потоке):



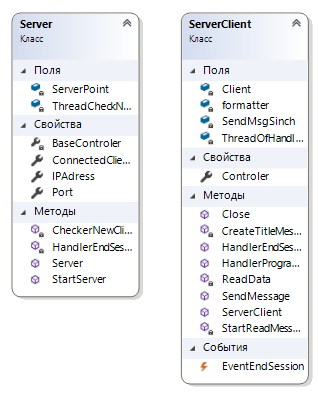
В свою очередь каждый объект запускает обработчик сообщений пришедших этому объекту у каждого своего компонента:



Решение проблемы слишком большого количества объектов:

* Введение у каждого игрового объекта Status – может быть active или static. Если объект static он не вызывается при каждом срабатывании таймера. Для обработки сообщений используется ObservalableCollection которую мы инкапсулируем в собственном классе ObservalableQueue – есть только два метода положить в конец очереди и извлечь из начала очереди. То есть обработка сообщения начинается сразу после получения сообщения (но, если только объект находится в состоянии static).
* Можно создать отдельную коллекцию для объектов, которые должны начинать обрабатывать свои сообщения при каждом срабатывании таймера.

В компоненте NetworkRoom также есть timerTotalSinch, который вызывается каждые 3 секунды и отправляет всем активными игрокам общее состояние карты(для полной синхронизации) также происходит обработка сообщений произошедших в игре по мере их поступления.

Работа уровня передачи и поддержания соединения:

