

*IGameContext* – абстрактный класс, в который включаются IGameObject списком.

*IGameObject* – абстрактный класс, который включает базовую функциональность: идентификатор, список компонентов IComponent.

*IComponent* *– ­*имеет состояния. Через состояния компонентов определяется состояние объекта. Каждый такт объекты опрашиваются и формируется общее состояние объектов, то есть заполняется структура GameObjectState и отправляется клиенту, где потом применяется к объекту с определенным id.

*CGameContext* и *SGameContext* – реализация IGameContext.

**На стороне сервера**:

*StaticObject* – статический объект, который имеет координаты, не умеет передвигаться.

*SteeringBehaviours* – управляющее поведение (поиск пути, evoid, flue, pursuit). Требует DynamicObject.

*DynamicObject* – содержит базовую механику «как» нужно передвигаться (скорость, положение и т.д.). Map\_*Graph* – делит карту на секторы. В каждом секторе храниться список SGameObject`ов.

**На стороне клиента:**

*Renderer* – компонент, отвечающий за «отрисовку» объектов. На клиент поступает команда «создать объект» и создается объект с этим компонентом, в .xml просматривается какой файл отвечает за этот тип объекта. Открывается файл и парсистся: какая модель, какие текстуры.

*Terrain* – отвечает за отрисовку земли, что б не казалось, что объекты висят в воздухе. Есть поля: плоскость и текстура.

*Camera* – игровая камера, реагирует на определенные клавиши и мышку. Выпускает «луч». По этому «лучу» ищется точка пересечения с плоскостью Terrain`а.

А вот с Terrain и Camera проблема. Есть два варианта:

1. Сделать Terrain и Camera наследником от *IComponent* и добавлять к объектам.
2. Сделать отдельными классами-наследниками *IGameObject.*

К этим двум классам будет постоянное обращение, так как у Terrain есть ссылка на плоскость (нулевой уровень), а у камеры «луч», только так можно узнать точку пространства. То есть при выделение, указании места, куда идти персонажу будут использоваться эти два класса.

Для этого можно их хранить отдельными указателями в CGameContext на клиенте. Если сделать 2 вариант, то нарушиться общая компонентная структура.