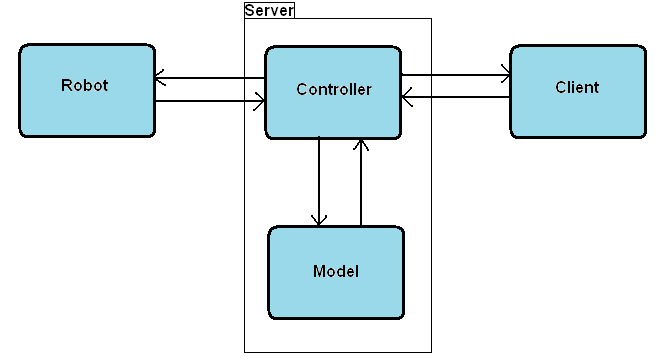
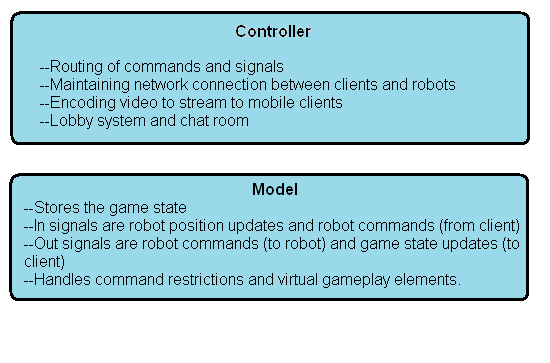
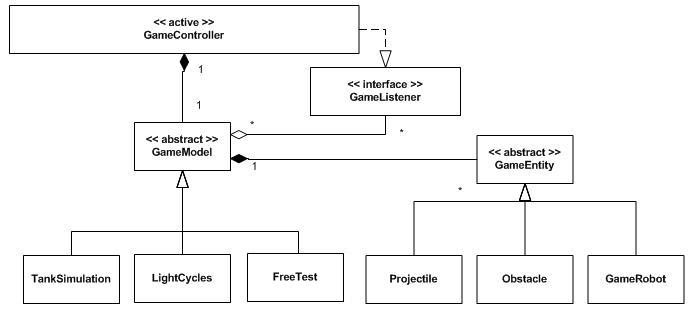
**System Architecture**



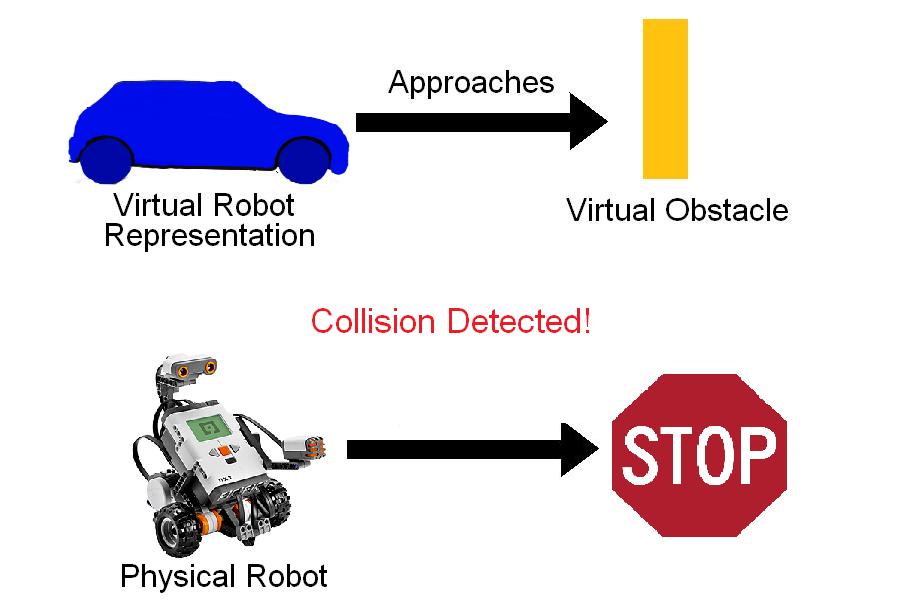
* **Discrete modules with rigidly defined responsibilities.**
* **System Architecture:**



**Server Model Architecture:**



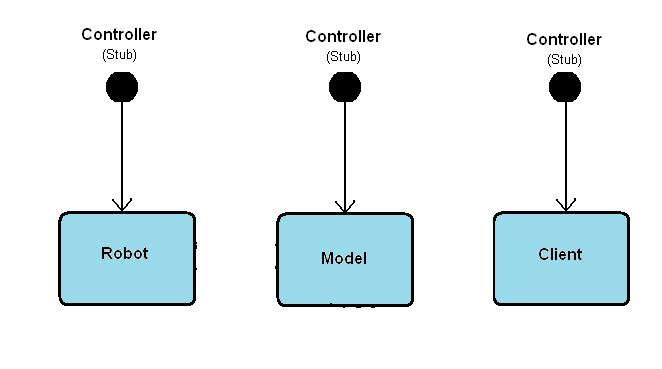
**Function Highlight:**



**Testing Methodology:**

* **Easy to test with unit testing (internal functionality only).**
* **For functionality that depends on signals passed in from other modules, stubbing is required for ease of testing.**
* **No unrelated functionality of stub is needed, only meant for injecting all possible signals for given module to process.**

**Testing Methodology:**



**Testing Methodology:**

* **The Model stub**
  + **A subclass of GameModel class, a game type with no virtual entities and no command restrictions.**
* **The Robot Stub**
  + **Fully functioning software representation of robot. Lejos navigator classes hooked up to simulated motors.**
* **The Client Stub**
  + **A simple client simulator controlled at console command line.**
* **The Controller Stubs**
  + **Simulating controller for each module require different implementations.**
  + **Each stub has different in/out signals to generate.**

**2D Collision Detection:**

* **General polygon based collision detection model**
* **All GameEntities have defined polygon vertices, used in further coordinate geometry calculations**
* **Collision occurs when there is no interval of separation for the projection of two polygons onto every axis of its sides**