# ­­GAME 440 Scrum Meeting Report

|  |  |
| --- | --- |
| Team: Server (pretty heads) | Date: Mar-06-2014 |
| Phase: Six | Due Date: Mar-13-2014 |

|  |  |
| --- | --- |
| Team Member | Work Description |
| Rohun Banerji | Added GameManager class to hold the running games (GameManager.h/cpp)  Added PlayerBase class to represent the 2 team bases and updated GameModel to include 2 of them  (PlayerBase.h/cpp)  Updated GameModel UML to include GameManager and PlayerBase (GameModel.class.violet.html)  Added Command\_CreateNewGame (Command\_CreateNewGame.h)  Added Tester method for Command\_CreateNewGame (Test\_Command\_CreateNewGame() in ServerTester.cpp)  Added Command\_UpdateMinions (Command\_UpdateMinions.h)  Added Tester method for Command\_UpdateMinions (Test\_Command\_UpdateMinions() in ServerTester.cpp)  Added id variable to the Entity class that Minion and Tower derive from (Entity.h)  Added BoxColliders to Minions and PlayerBase to run Collision Tests (Minion.h/cpp) |
| Alex McCann |  |
| Chris Devlieger | Implemented functionality to GameStart() method in TeamLobby.h/cpp and added necessary fields/changes elsewhere for GameStart() |
| Patrick Barahona-Griffiths |  |
| Jordan Kjaer | Created PacketFactory (PacketFactory.h/cpp) Takes the place of every command and workitem. Easy to use for executing the command and for sending commands to whomever needs them. Created classes to match the Protocols found in Protocol.h  I know the idea was for Server/Client to create their own implementation of Packetfactory but I think it should just be one, especially since it can be used for easy sending as well, so I included the Client stuff too. |
| Justin Kan |  |
| Wayne Gauthier |  |

­­­­­­­