Architecture, 29/9 2012

Model

The GameModel keeps most of its game data in containers called -Manager, such as BulletManager and EnemyManager. It also has a Player and an EnemySpawner.

The EnemyManager has a list of Enemies, and those have much in common with the Player class. The biggest difference is that Player is controlled by the actual player and an Enemy is controlled by the game.

A Player has an Itembar which contains Weapons, and a Player also has a PlayerShip which is an Entity that is shown on the screen.

An Enemy has a single Weapon and an EnemyShip, which is an Entity.

All objects that appear in the game extends the class AbstractEntity. These are bullets, ships and items that can be picked up.

The EnemySpawner has WaveLists which describes when waves of enemies should start. A Wave has a number of enemies, and the enemies in the wave fly in a pattern, such as a V pattern, or maybe they fly in a horizontal line.

Controller & view

MainActivity extends AndEngine's SimpleBaseGameActivity and it has a number of scenes, which extends AndEngine's Scene class. The scenes we currently have are MainMenuScene and GameScene. Each scene also has a controller.

The controller we have worked the most on so far is the GameController. It manages the GameScene, it also has a reference to the GameModel, and to regularly update the model and the scene it gets help from a class called GameLoop which implements AndEngine's interface IUpdateHandler.