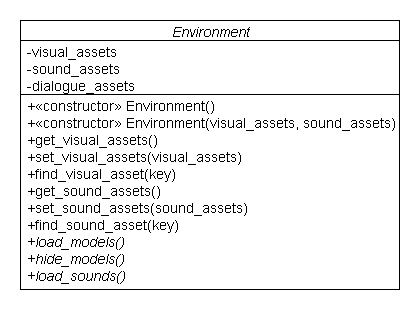
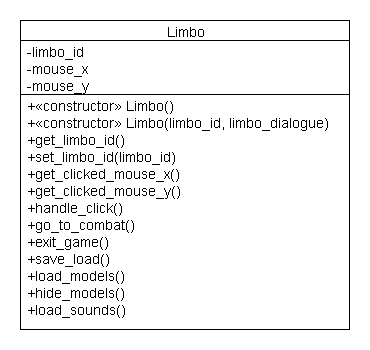
**Class Descriptions**

**Environment**

This class is an abstract super class to Limbo and Combat. Its main purpose is to hold all the visual assets, sound assets, and dialogue assets needed when instantiating Combat or Limbo. 

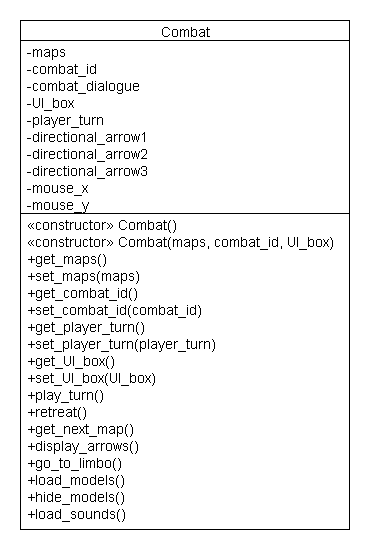
**Limbo**

This class extends Environment and will display anything that might appear in the limbo menu, as well as, start any sound file to be played. The Limbo class will also handle basic menu features of any game, such as exit, saving and loading.



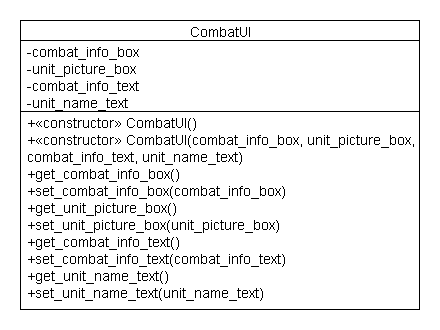
**Combat**

This class extends Environment and is very similar to Limbo. The difference lies in that what the Limbo class holds for the limbo menu the Combat class holds for the combat screens.



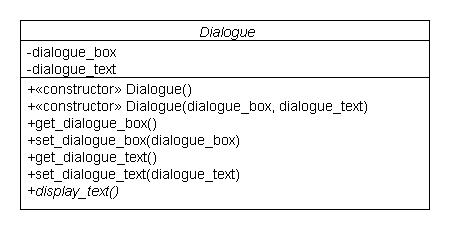
**CombatUI**

This class will hold all the user interface elements of combat. The class Combat will hold a CombatUI instance.



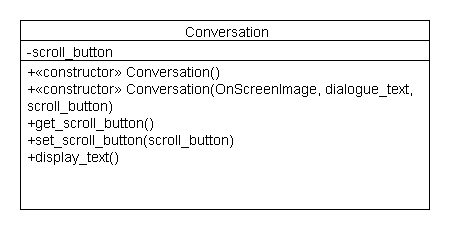
**Dialogue**

Dialogue is an abstract super class to Conversation and Option\_Box and it will hold the dialogue box and dialogue text of any dialogue that may occur between characters. There are two types of dialogues that can occur in the game.



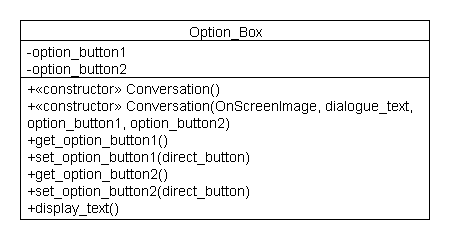
**Conversation**

The first type of dialogue is a regular conversation between characters. Conversation, thus, extends Dialogue. Its main purpose is to display the regular dialogue box and text the player will encounter the most.



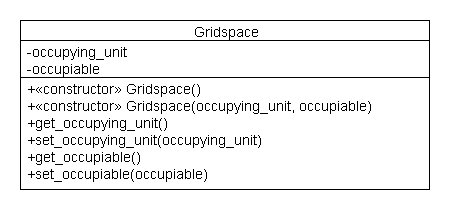
**Option\_Box**

The second type of dialogue is the type of dialogue where the user is allowed to pick in which direction he/she would like to take the dialogue. The Option\_Box class handles that type of dialogue, by extending the class Dialogue.



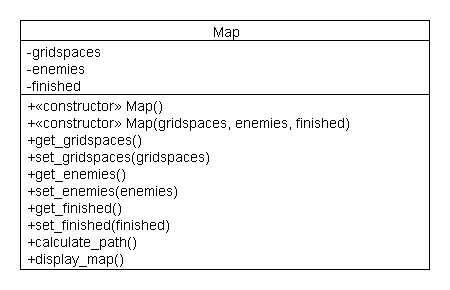
**Gridspace**

The Gridspace class will act as a hexagon in our combat screen. Its main purpose is to know which unit occupies it and if it is occupiable.



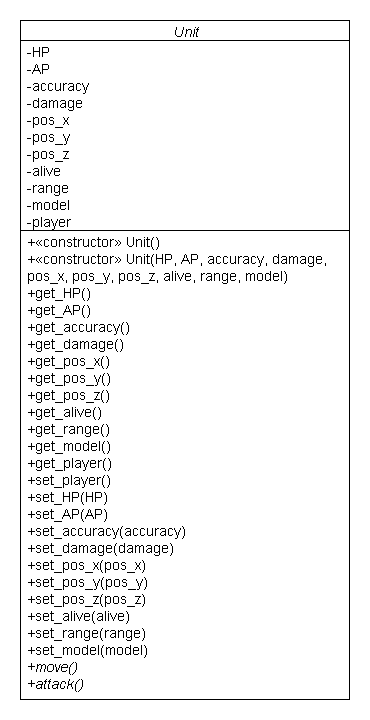
**Map**

The Map class is composed of a collection of Gridspace classes and Enemy classes. The Map class’ main job is to know when the current map has been finished and display the collection of “Gridspace”s.



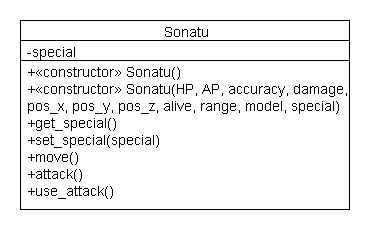
**Unit**

Unit is an abstract super class to the classes Sonatu, Enemy and Queen. The Unit class holds all the attributes of any unit the player might encounter in the combat screen. It has two abstract methods, move() and attack(), both of which are unique to which type of unit.



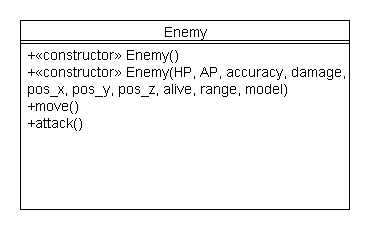
**Sonatu**

Sonatu extends Unit and acts as the main player’s ship in the game. It differs from a general Unit because of the special attack granted to the player.



**Enemy**

The Enemy class defines a regular enemy. The Enemy class doesn’t do much, but construct the enemy and create a unique move() and attack() methods.



**Queen**

The Queen class is much like the Enemy class; however, because the queen has all the attacks as the player’s ship, the Queen’s move() and attack() methods are different.

