The AI:

The Enemy and GA are really self explanatory. (GAs aren't really so difficult to understand or program, they're just rough to bug test. I simply mashed together some ideas from the book's GA section, and some of my own ideas to make it, and got excellent results after I killed the bugs). The not so self explanatory AI is the player AI. It's really only an FSM and motion blending, but I use them to good effect to get complex behavior out of the player bot.

So here's how the thing works:

I will cover the motion blending first because it needs to be understood before the FSM I build can be. The AI bot sees the same screen the player would (with a little pre-knowledge of where the enemies are coming from, the way any half way experienced player would.), and used that exclusively. The bot sees each object and calculates his distance to it. There is then for each object 2 virtual forces exerted on the player. An attraction and a repulsion. For each object type, the player has a set of weights characterizing his interaction with that object. The force of attraction and repulsion are given by:

$$F=\frac{k}{d^p}$$

Where d is the distance to the object, k and p are variables held by the ai player. Note that this is basically coulomb attraction. Also note that p and k are both real numbers. And if p is negative then the force becomes proportional to the distance rather than inversely proportional. Finally, note that since each object type has it's own collection of variables for attraction AND repulsion, they can be set in such a way that the player may be drawn but kept at a certain distance, if, say attraction is linear but repulsion is 1 / d^2.

The objects to which the bot is attracted and repulsed are: bullets, items, game line, and enemy line. The game line is a line a certain fraction up the screen (given in the player variables). The enemy line is a line below the currently targeted enemy. Once the ai has looked at each object and created a net force, it simply calculates the nearest of the big eight directions (up, down, left, right, and the four diagonals), and presses the correct keys!

The FSM has 3 states: panic, item, and normal. Normal is the default that is used in any situation. Item is used whenever there is an item on the screen to pick up. Panic is used whenever a bullet comes close to the ship. The ONLY DIFFERENCE between the states is the set of attraction/repulsion variables used!

So the complex behavior comes from: The robust (slightly modified) coulomb equation, the fact that each object has both attraction and repulsion, and finally the ability to totally change gears in different situations.