

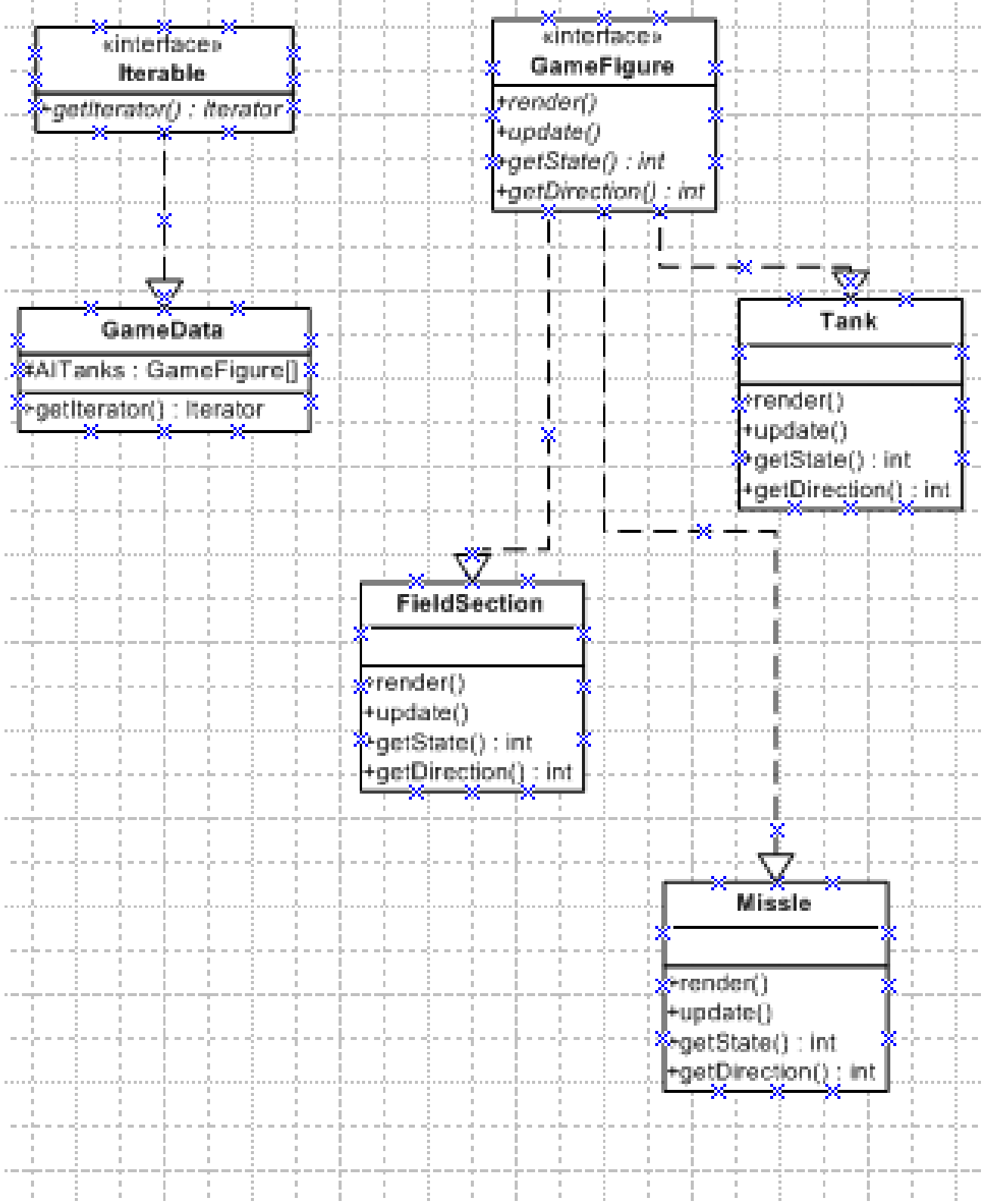
I. Game Description

I would like to create a single player arcade game to be played against the computer. The game involving different levels of mazes that must be navigated in order to progress in the game. The player will be represented by the UI as a tank, of which the user could pick the color, and possibly other visual and behavioral traits. As each level is completed, a score for the level is calculated and stored. Also, each level gets progressively more difficult with each passing level. There should also be enemies to stand in the way of the players progress, which can shoot at the player and do damage. These will also be tank game pieces, but will be managed by an AI that can move the piece around the board and target the player when they are within range. Scores are calculated based upon the speed at which the maze is traversed, and how much damage was taken. Bonus points can also be awarded if the player is able to destroy any of the AI controlled units. If an AI tank is destroyed, it will need to wait a certain amount of time, and then will be respawned onto the field at a specified point. If the player completes all of the game levels, and obtains a high score, that score may be recorded along with their initials to be displayed on a high scores screen later.

II. Features

- Game levels that can be loaded from an XML file. For instance each level would be contained in a different file, as the game progresses, a new field/level is loaded to the screen. This allows the game to be very dynamic and new levels to be easily added.
- A human controlled game piece whose input comes from the keyboard. Inputs would include movement in four cardinal directions on a 2 dimensional plane, as well as firing controls that can fire weapons at AI opponents.
- AI components of the game which act as opponents to the player. These will take the form of other tanks on the game field. They need to be able to navigate whatever field is loaded, and target the player (not each other) when they player is within range. These units must either be avoided or destroyed.
- Multiple levels, each of which become increasingly difficult, and track the users score at each level.
- Sound effects to be played when a weapon is fired, or a game figure is destroyed.
- User defined options for game difficulty, game piece appearance, sound effects, etc. Default values that can be held in/saved to a configuration file.
- A rich user interface which displays the field and players via animation so that the user has a visual interaction with the game.

III. UML Diagram



IV. Polymorphism

Every figure in the game (field sections, player units, AI units, missiles) should implement the interface GameFigure, which defines all the methods required by the GameData and Animator classes to keep their states up to date and to render the items on the screen. The GameData would also implement the iterator pattern to return an iterator for accessing the AI tank units in the game, for the use of the AI class to manage the computer controlled tank units on the screen.

V. Design Patterns

A. Iterator Pattern

The GameData class implements the iterator pattern, being the concrete iterable object. This returns an iterator for traversing the AI controlled GameFigure objects.

B. Strategy Pattern

The AI Tank will be the concrete context that implements different strategies based on the difficulty of the game.