API Documentation

Group Houston

Command design pattern is used for this game. The objects use the command and observable-observer design patterns to determine what happens when objects meet with each other. Singleton design patterns are used when possible.

1 View

We have a 1000×500 game map and a restart button on the canvas.

2 Controller

Controller package includes PacWorldController which will process GET and POST requests and return the JSON representation of all the concrete objects in the PacWorld. The controller creates the dispatch adapter and defines the REST end points

The PacWorldController will load and update objects in the PacWorld. It also a reset and clear game options. Meanwhile, it posts the direction information for the Pacman. In addition, it has an endpoint to get the canvas dimension from the front end.

3 Model

The model package includes: DispatchAdapter, command, strategy and objects.

3.1 DispatchAdapter

DispatchAdapter communicates with the model and the view. It contains observer lists and parameters for the game includes:

- Point dims for the dimension of game map
- static int score to record the score
- static int lives to record the lives
- static int fruitTimer to record the cool down time to generate the fruit
- static int afraidTimer for the time that Pacman can eat ghosts

- static int[][] map to construct the game map.

Methods in the DispatchAdapter include: loadObject, resetGame, updateGame, switchDirection, canvasDims:

- DispatchAdapter is the constructor, it calls initializeMap.
- setCanvasDims set the dimension of the game map. It has one parameter, Point dims.
- initializeGame initializes the game. It will load the game map, initialize the Pacman, ghosts, foods, and timers. It does not take any parameter and it has no returns.
- updatePacWorld updates the game canvas. It takes no parameters and has no returns.
- switchDirection switches the direction of the Pacman. It has one parameter, String body, which is the keyboard information from the front end.

3.2 Command

The cmd package has an interface IGameObjectCmd, which has an execute method. Every concrete command implements this interface.

3.3 Strategy

The strategy package has two interfaces:

- IUpdateStrategy for single object strategies
- IInteractStrategy for interaction strategies between objects

IUpdateStrategy has two methods:

- getName return the name of the strategy in a string. It has no parameters.
- update applies the strategy to an object. It has one parameter AGameObject context and no returns.

IInteractStrategy has two methods:

- getName returns the name of the strategy in a string. It has no parameters.
- interact will apply the interact strategy of the src object to the dest object. It takes two parameters, the first is the src object, the second is the dest object. It has no returns.

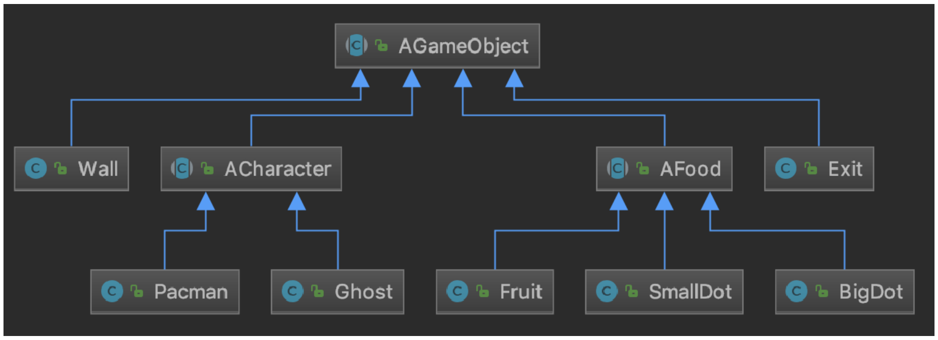

Figure 1: Objects

3.4 Objects

The object package has an abstract class AGameObject. Two abstract classes ACharater and AFood extend AGameObject. Two concrete objects Wall and Exit extend AGameObject. Their relationship is shown in figure 1.

The AGameObject has a constructor AGameObject, and four methods: getLocation, setLocation, getType, getSize and update.

- AGameObject is the constructor, it has three parameters: Point location, String type, int size.
- getLocation returns the location of the object in a Point. It has no parameters.
- setLocation set the location of the object. It has a parameter Point location and it has no returns.
- getType returns the type of the object in String.
- getSize returns the size of the object in int.
- update updates the state of the object using strategies associated with the object.

An abstract class ACharacter extends AGameObject for moving objects. Two concrete objects Pacman and Ghost extend ACharacter. The ACharacter has a constructor ACharacter and six methods: getInteractStrategy, getUpdateStrategy, setInteractStrategy, setUpdateStrategy, getVel, setVel.

- ACharacter is the constructor. It has six parameters: Point loc, String type, Point vel, IUpdateStrategy updateStrategy, IInteractStrategy interactStrategy, int size.
- getInteractStrategy returns the interact strategy. It has no parameters.
- getUpdateStrategy returns the update strategy. It has no parameters.

- setInteractStrategy sets the interact strategy for the ACharacter object. It has one parameter IInteractStrategy interactStrategy and has no returns.
- setUpdateStrategy sets the update strategy for the ACharactor object. It has one parameter IUpdateStrategy updateStrategy and has no returns.
- getVel returns the velocity of the object in a Point. It has no parameters.
- setVel set the velocity of the object. It has one parameter Point vel and has no returns.

An abstract object AFood extends AGameObject for different food. It does not have any new methods. Three concrete objects SmallDot, BigDot and Fruit extend AFood.