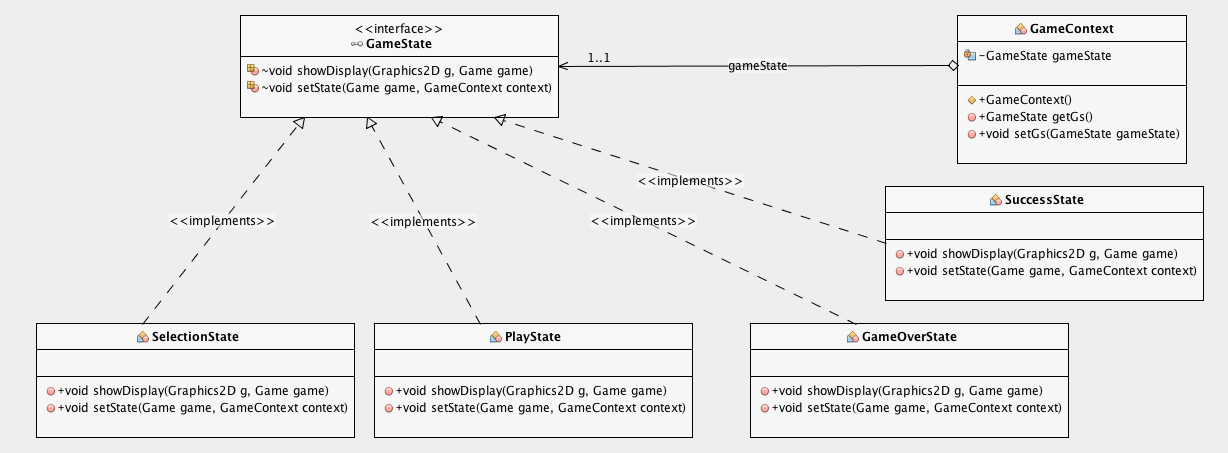
**Pattern Descriptions**

1. **State Pattern**

****

**Motivation**

State pattern is a design pattern that handles different behaviors needing to change dynamically at runtime.

**Implementation**

Pacman has 4 different states such as: Selection, Play, Success and Game Over.

All the state concrete classes have the same showDisplay() function that is implemented by the GameState interface. This function plots the necessary graphs in order to show different screens changing during the game. GameContext is created for containing as well as changing the states by using the interface.

**Missions**

Selection state shows the welcome message created with the awt library of Java.Play state has been used after choosing the board of the game. Drawing the components of the game such as : ghosts, walls, fruits etc. takes place in play state. Success state prints the success message after passing all the levels that contain different conditions. Game over state runs only if Pac-Man has no lives remaining.

Consequently, the flexibility of adding new independent states to the game is improved by utilizing the structure of the State pattern.