```
GameMap ----Dependency----> Player ----Association----> Action

GameMap ----Dependency----> Toad

Player --Dependency--> Toad (1, 1)

Toad --Association--> Application (1, 1)
```

Design Rationale
Player needs to create multiple objects of Action to then display the different options for purchasing from Toad. To check to see if Player is within range of Toad to create these actions, Player must have access to GameMap to call methods to check for the positions around the Player. This is why it is a dependency because the object of GameMap is already passed in and only the methods are used.

We had a couple oversights in our last UML diagram by not including GameMap and Action as we lacked the understanding of how we would implement the Display functions for Player. Having Action and GameMap is crucial for our design.