1. Firstly you would need to let the user choose how many players are playing. If there were 4 players playing, then the decks would have to be increased in size relative to the number of players playing. E.G. 4 players with 10 cards each instead of 2 players with 5 cards each. The rules of the game would be similar, each player picks a card and player 1 chooses an attribute. A score would be held for each ‘round’ (rotation through the players), the player with the highest score at the end of the round takes all the cards. This would be repeated until one player has all the cards.

2. Factory Pattern. An argument could be made that the prototype pattern would be a better solution in this context since the cards are fundamentally the same. However, the Factory Pattern will work fine.

3. (This is a new concept to me, so this answer isn’t great) The Service Locator abstracts dependencies by using a central registry which returns information on request. By contrast to this, dependency injection creates the service and passes it to the client. For an application such as this, a Service Locator will work fine, as it is a small-scale application (this might be hugely wrong but it’s my interpretation of it).

4. There are a couple of ways to improve the gameplay however they are limited due to the nature of the game:

1. Add more attributes
2. Add different branded biscuits (e.g. McVities vs Tesco’s own brand)