Line refers to the number of line in the a2.py (377 lines) submitted to blackboard

Line 13

class Player(object):

In python 3, we can delete (object)

Line 43-46, line 230-233

```
        if self.get_deck().get_amount() == 0:
            return True
        else:
            return False
```

We can change all this to:

return self.get_deck().get_amount() == 0

Line 89-96

```
        for card in self.get_deck().get_cards():
            if card.get_pickup_amount() == 4:
                self.get_deck().get_cards().remove(card)
                return card
            elif card.matches(putdown_pile.top()):
                self.get_deck().get_cards().remove(card)
                putdown_pile.add_card(card)
                return card
```

We can change all this to:

```
for card in self.get_deck().get_cards():
    if card.matches(putdown_pile.top()):
        self.get_deck().get_cards().remove(card)
        return card
```

This is because gui know whether card is black card or normal card, and card will be automatically put on the putdown pile.

Line 245-251, line 273-279, line 301-307, line 334-340, line 364-370

```
    def __str__(self):
        """Returns the string representation of the card."""
        return "Card({0}, {1})".format(self._number, self._colour)

    def __repr__(self):
        """Returns the string representation of the card."""
        return "Card({0}, {1})".format(self._number, self._colour)
```

We can change all this to:

```
    def __str__(self):
        """Returns the string representation of the card."""
        return repr(self)

    def __repr__(self):
```

```
        """Returns the string representation of the card."""
        return "{0}({1}, {2})".format(self.__class__.__name__, self._number, self._colour)
```

- If you call __str__, __str__ will automatically call __repr__, notice when we return repr(self), we don’t need double underscore.
- self.__class__.__name__ can get the name, so we just need to write str and repr in parent class