

Task 0

- We chose to use smart pointers because once out of scope, the pointer automatically deletes itself
- The copy constructor creates a 1-1 copy of the object, which can be resource heavy. The move constructor moves the resources into the heap which prevents unnecessary copying
- Not only does it use more memory than is necessary, in the case of the card game there is only 1 of each card, so if you create a copy, there are now duplicates
- Since the destructors for smart pointers are automatically called once out of scope, it is easier for the programmer to make use of them, while for objects, you have to be careful not to accidentally copy the object

GameWithObjects

```
Huey 1's turn: Choose choice
```

```
1 0 0
```

```
1: Draw cards from deck
```

```
2: Remove all cards and start with 5 next turn
```

```
3: Place card in stack
```

```
Are you done with your turn(y/n): y
```

```
Huey 1 wins!
```

```
Deleting PersonWithObjectsDewey 3
```

```
Deleting PersonWithObjectsLouie 2
```

```
Deleting PersonWithObjectsHuey 1
```

```
End
```

GameWithSmartPointers

```
Huey 1's turn: Choose choice
```

```
1 0 0
```

```
1: Draw cards from deck
```

```
2: Remove all cards and start with 5 next turn
```

```
3: Place card in stack
```

Are you done with your turn(y/n): y

=====

Huey 1 wins!

=====

End

user@user:~/Desktop/Data-Structures/Lab03/build\$