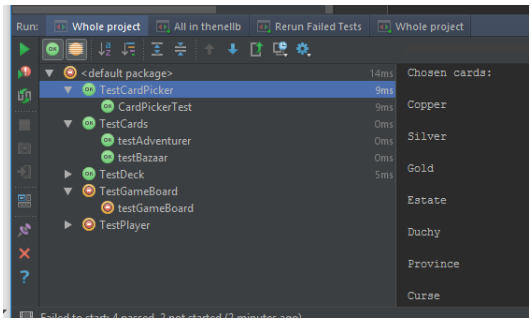
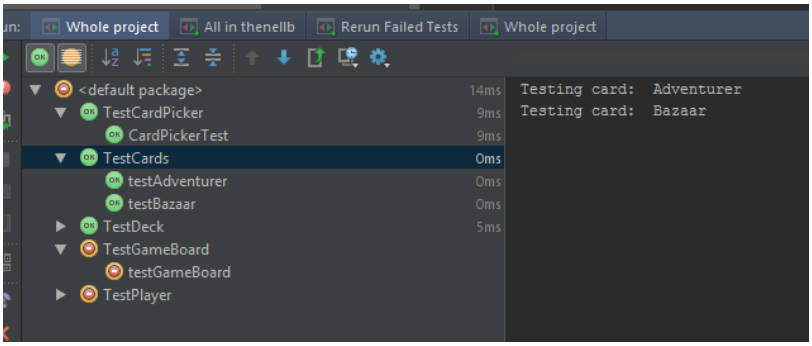
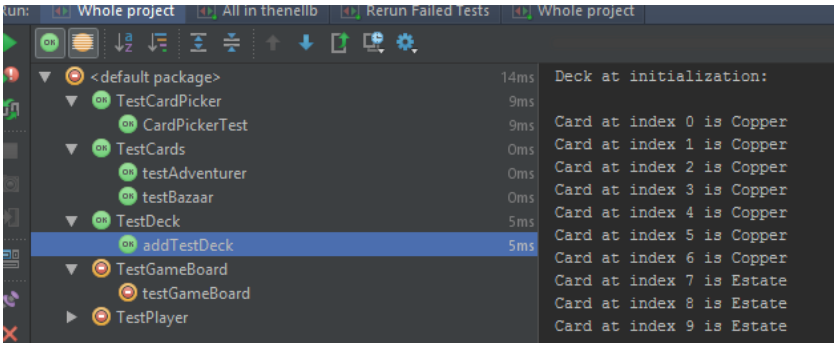
Tests







Output of final test:

Deck at initialization:

Card at index 0 is Copper

Card at index 1 is Copper

Card at index 2 is Copper

Card at index 3 is Copper

Card at index 4 is Copper

Card at index 5 is Copper

Card at index 6 is Copper

Card at index 7 is Estate

Card at index 8 is Estate

Card at index 9 is Estate

Deck after being shuffled:

Index inside loop: 0

Size of discard: 10

Index inside loop: 1

Size of discard: 10

Index inside loop: 2

Size of discard: 10

Index inside loop: 3

Size of discard: 10

Index inside loop: 4

Size of discard: 10

Index inside loop: 5

Size of discard: 10

Index inside loop: 6

Size of discard: 10

Index inside loop: 7

Size of discard: 10

Index inside loop: 8

Size of discard: 10

Index inside loop: 9

Size of discard: 10

Size of discard: 10

Size of deck: 10

Card at index 0 is Copper

Card at index 1 is Copper

Card at index 2 is Copper

Card at index 3 is Estate

Card at index 4 is Copper

Card at index 5 is Copper

Card at index 6 is Estate

Card at index 7 is Copper

Card at index 8 is Estate

Card at index 9 is Copper

Index inside loop: 0

Size of discard: 10

Index inside loop: 1

Size of discard: 10

Index inside loop: 2

Size of discard: 10

Index inside loop: 3

Size of discard: 10

Index inside loop: 4

Other tests were not working because of user input being required.

c. My choice of unit tests simply initialized all different levels of classes. The ones that could went into the functions and made sure everything worked there. Most of my tests I couldn't implement due to lack of automation. I included only a couple of these bad tests to show this.

d. I found a bug where sometimes a card would go to the wrong player.

e. I think the unit tests covered most of the functions on my source code. Unless by source code, you mean the broken C that was given to us, then no, I didn't write tests for code that doesn't compile.

f. Pseudocode

Adding in a script that would just input "2" in most standard in buffers would cover 90% of the user prompts.