

A: Unit tests:

Tests included coverage of the creation of card objects. Cards are valid and contain information relevant to cards within the game. Since instructions didn't include necessity of extensive test coverage I focused on developing the game more thoroughly. If more test coverage is preferred please contact me at the address found within the README

B Actual outcomes of executed tests

Card objects confirmed valid and accessible through array, applicable to hand, deck, discard pile, and card piles.

C Choice of input

Input for program was chosen through number commands 1-n. Validity checks were utilized throughout. Input for test suite were functions `assertTrue()` returning boolean.

D bugs

Since the instructions required bugs to be present, 5 bugs within the program are as follows:

- Special actions for cards do not activate
- platinum cards if in great number do not correctly add up in display
- computer cheats to get extra Victory points
- choosing card within 100 but beyond hand size accesses an empty card of no value
- Error upon string input instead of number commands

E commentary

The tests honestly cover very little of the actual program, since they were added as an afterthought after in depth dynamic testing was used throughout development. The program had an enourmous amounts of sysouts and variable checks littered throughout it's functions before being cleaned up to be turned in, and these tests ensured the program runs smoothly. Official test coverage through Junit only asserts that card assignment is accurate and runs correctly. All other functions were tested through dynamic testing previously.

F auto test system

```
Method(pass in requirements){  
  Assert true(requirements are met)  
}
```