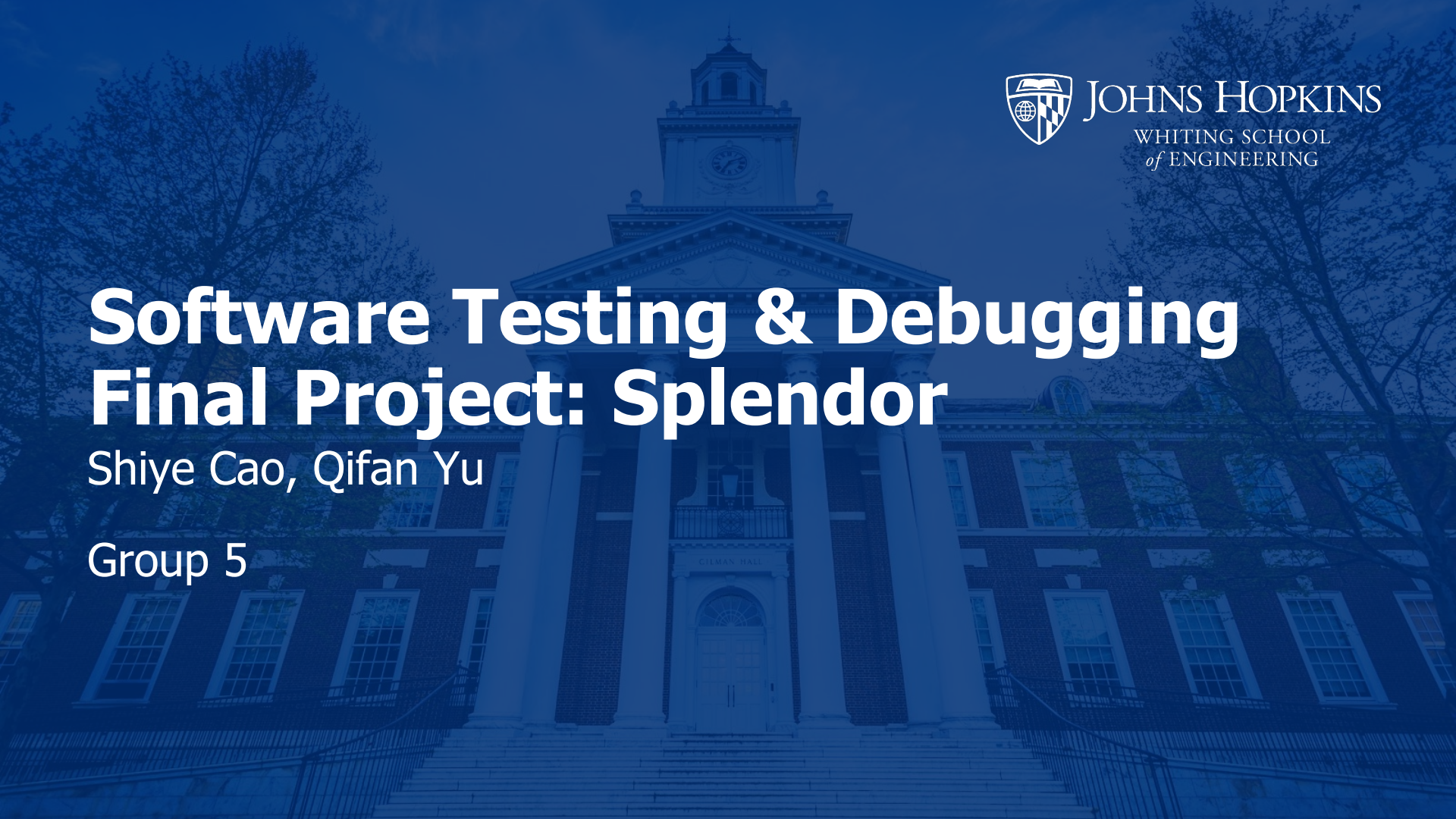


JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Software Testing & Debugging Final Project: Splendor

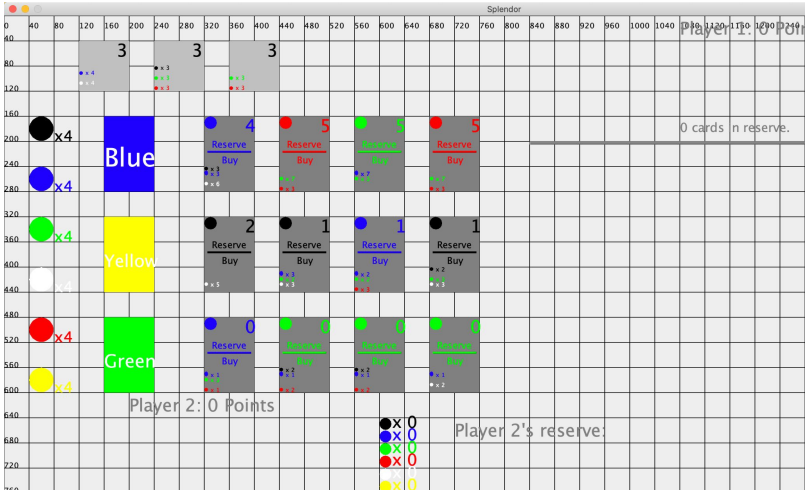
Shiye Cao, Qifan Yu

Group 5

SUT: Splendor

- Splendor: 2-4 player resource management card game
  - Designed by Marc André & Pascal Quidault
  - User: Designed for players 9+
- SUT: Computer implementation of Splendor game
  - 1 Textual Interface: Terminal
  - 1 Graphical Interface: JFrame
- Free players from physical cards and tokens
- Created by GitHub User z1103246
- 100% Java, 3473 lines of Java Code

SUT: Splendor



Graphical Interface

```
It is Player 1's turn!
Player 1, here is what you currently have:


***** Player 1 0 Point(s) *****
Tokens: Onyx x0 ,Sapphire x0 ,Emerlad x0 ,Diamond x0 ,Rubby x0 ,Gold x0

**** Cards ****

**** Nobles ****


**** Cards in Reserve ****


Player 1, what do you wish to do this round?

1. Take 2 tokens of the same color from the table(may only be performed if there is 4 or more tokens in the color of your chosing)
2. Take 1 token from each of 3 different types of tokens(excluding gold)
3. Reserve a card on the table and receive one gold token(keep in mind you may only have three reserved cards maximum. Also, you will not
  recieve any gold token if there is non on the table)
4. Reserve an unknown card from the top of a chosen deck
5. Buy a card from your reserved cards
6. Buy a card on the table
```

Textual Interface

Testing Plan

White Box Testing	**100% method coverage** **90%+ branch coverage**
Black Box Testing	**100% rules coverage**
Mutation Testing	**80% mutation coverage**

Test Results: White Box Testing

- Over 300 test cases covering:
  100% method coverage
  99% statement coverage
  94% branch coverage overall & >90% branch coverage on every method
- This is because sometimes some branches are unreachable

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
Game		99%		97%	3	107	1	187	0	46	0	1
Player		99%		91%	18	156	1	389	0	47	0	1
Tokens		97%		90%	1	12	1	22	0	7	0	1
Cards		100%		100%	0	34	0	74	0	17	0	1
Noble		100%		100%	0	29	0	59	0	14	0	1
Total	8 of 3,382	99%	22 of 414	94%	22	338	3	731	0	131	0	5

Example: Unreachable Code

In Tokens class:

unreachable

statement &

branch due to error

Results in:

97% statement coverage

90% branch coverage

```
public Color returnColor()
{
    if(color.equalsIgnoreCase("Onyx")){
        return Color.BLACK;
    }else if(color.equalsIgnoreCase("Emerald")){
        return Color.GREEN;
    }else if(color.equalsIgnoreCase("Diamond")){
        return Color.WHITE;
    }else if(color.equalsIgnoreCase("Rubby")){
        return Color.RED;
    }else if(color.equalsIgnoreCase("Sapphire")){
        return Color.BLUE;
    }else{
        return Color.yellow;
    }
}
```

Example: Unreachable Class

```
    public boolean checkActionFive(Player p, int rol, int col) throws PlayerDoesNotHaveEnoughResourcesToBuySelectedCard
, InvalidCardsSelectionOfCardsInReserve, InvalidCardsSelectionOfCardsOnTable
    {
        if(rol-1 <= 2 && col-1 <= 3 && rol-1 >= 0 && col-1 >= 0){
            if(p.canBuy(cardsOnTable[rol-1][col-1])){
                //System.out.println("Hi");
                return true;
            }else if(p.canBuyWithYellow(cardsOnTable[rol-1][col-1])){
                //System.out.println("Hello");
                return true;
            }else{
                throw new PlayerDoesNotHaveEnoughResourcesToBuySelectedCard();
```

Only instance where this exception exists in the code.

Exceptions

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
InvalidCardsSelectionOfCardsInReserve		0%		n/a	1	1	1	1	1	1	1	1
InvalidTokensSelection		100%		n/a	0	1	0	1	0	1	0	1
PlayerDoesNotHaveSelectedCardInReserve		100%		n/a	0	1	0	1	0	1	0	1
PlayerDoesNotHaveEnoughResourcesToBuySelectedCard		100%		n/a	0	1	0	1	0	1	0	1
SelectedDeckRanOutOfCards		100%		n/a	0	1	0	1	0	1	0	1
NotEnoughTokensOnTable		100%		n/a	0	1	0	1	0	1	0	1
PlayerAlreadyHoldsThreeReserveCards		100%		n/a	0	1	0	1	0	1	0	1
InvalidCardsSelectionOfCardsOnTable		100%		n/a	0	1	0	1	0	1	0	1
Total	3 of 24	87%	0 of 0	n/a	1	8	1	8	1	8	1	8

Test Results: Mutation Testing

Used Pit

- Generated mutation tests for classes
- Achieved 84% mutation coverage

Not all mutations can be killed

Example:

```
}else if(color.equalsIgnoreCase("Rubby")){
        return Color.RED;
```

```
Mutation: replaced return value with null
```

Pit Test Coverage Report

Package Summary

Classes

Number of Classes	Line Coverage		Mutation Coverage	
5	100%	728/731	84%	456/544

Breakdown by Class

Name	Line Coverage		Mutation Coverage	
Cards.java	100%	74/74	90%	37/41
Game.java	99%	186/187	91%	151/166
Noble.java	100%	59/59	100%	35/35
Player.java	100%	388/389	76%	218/286
Tokens.java	95%	21/22	94%	15/16

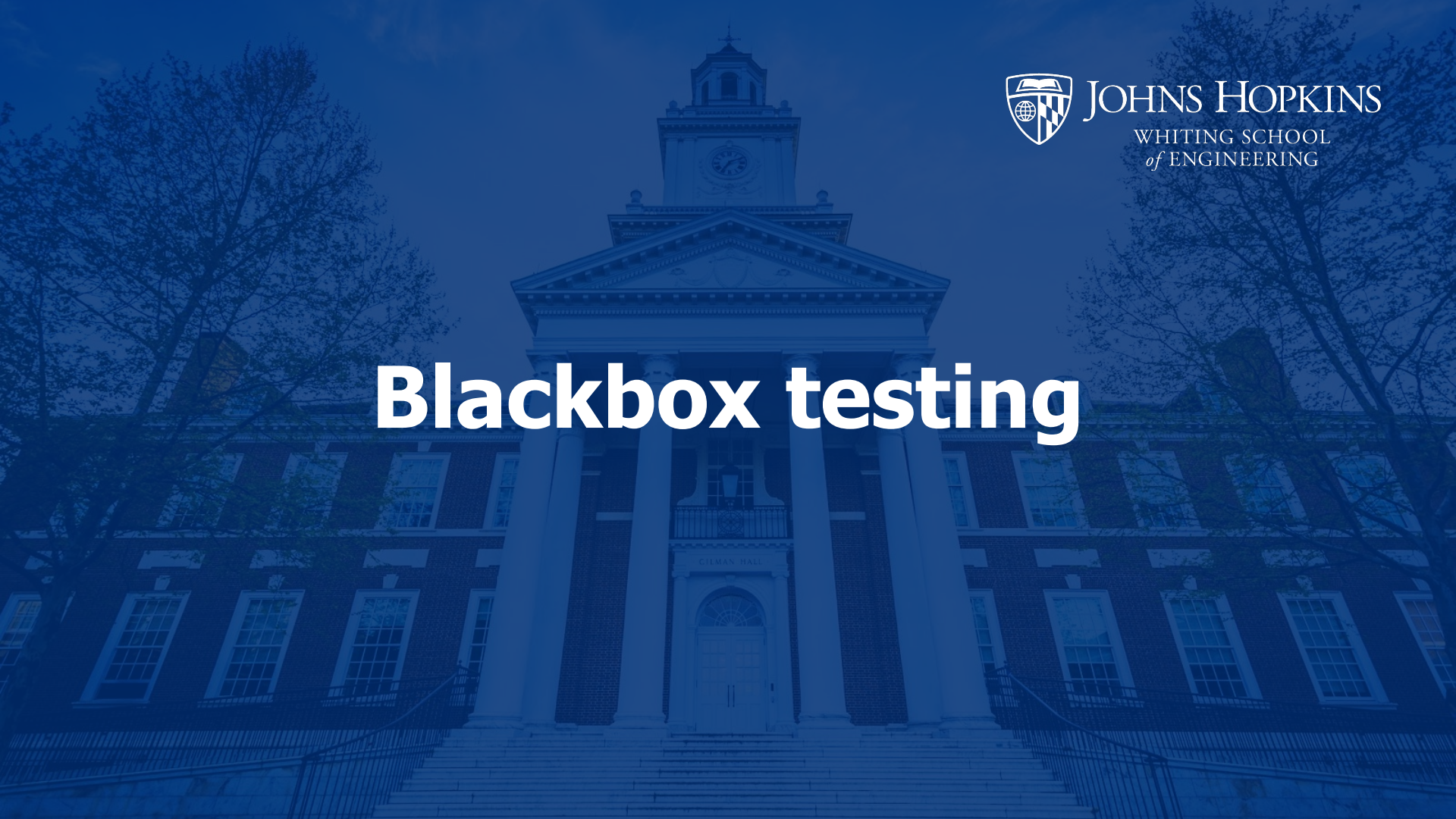
Report generated by PIT 1.5.2

Select Faults Detected from WhiteBox Testing

1. The program miss-spells all instances “ruby” as “rubby”
2. Token class, Card class, Noble class and Game class constructors does not check for invalid arguments
3. Token class’ returnColor returns all non-token colors + “ruby” as yellow
4. CompareTo make sure that the two objects have at least one common out of resource, name, and point
5. Game class’s differentColors contains a for-loop indexing error
6. Game class’s win always returns true regardless if someone has won
7. Game’s Check action Two always returns true
8. Game’s CheckActionFive does not throw error when input column 4 (col should be <=3)
9. Player’s ContainsAll does not check for the counts of each color token
10. Player’s RemoveAll always returns true

Missing Documentation & Lacks Comments & Argument Validation in Code

For a complete list of faults detected see https://github.com/Sally-14/SplendorTest



JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Blackbox testing

Game rules: Basic Instruction works!

On their turn, a player must choose to perform only one of the following four actions. **Equivalence partitioning**

- **Take 3 gem tokens of different colors. 100% coverage**

  Two-play player 1 choose 3 gem, player 2 choose 3 gem. Three-player choose 3 gem. Three-player choose 3 gem.

- **Take 2 gem tokens of the same color. This action is only possible if there are at least 4 tokens of the chosen color left when the player takes them**.

  Two-play player 1 choose 3 gem, player 2 choose 3 gem. Three-player player 1 choose 3 gem

- **Reserve 1 development card and take 1 gold token (joker).**

  Two-play choose reserve development card from card deck or from the table, Three-play choose reserve development card from card deck or from the table,

  Four-play choose reserve development card from card deck or from the table

- **Purchase 1 face-up development card from the middle of the table or a previously reserved one**

  Two-play buy reserved cards in decks (green, yellow, blue) or card on the table, Three-play buy reserved cards in the deck or card on the table, Four-play buy reserved cards in the deck or card on the table

JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Demo

Test Results: Black Box Testing

- 100% coverage of the rules from the Splendor rule book

Input: For the first round player 1 chooses the option 4 to reserve a card from the blue card deck.

Which card do you wish to reserve?

1. Blue Card 1 *** 2. Blue Card 2 *** 3. Blue Card 3

You cannot know which Jeweries required for each card?

Blackbox: Token number not changed

When player 2 choose option 2, select tokens from different Jewelries.

The number of jewelries do not change after selection!

Choose a token to take.

1. Onyx x3 2. Sapphire x3 3. Emerald x3 4. Diamond x4 5. Rubby x4

1

Choose a different token that you have not selected.

1. Onyx x3 2. Sapphire x3 3. Emerald x3 4. Diamond x4 5. Rubby x4

2

Should be Onyx x2

Choose another different token that you have not selected.

1. Onyx x3 2. Sapphire x3 3. Emerald x3 4. Diamond4 5. Rubby x4 6. Gold x3

Should be Sapphire x2

Expected vs What we got

Input: For the first round player 1 chooses the option 2 to select tokens from three different types, Onyx and diamond, and encounter unexpected instructions from the game.

```
Player 1, what do you wish to do this round?

1. Take 2 tokens of the same color from the table(may only be performed if there is 4 or more to
2. Take 1 token from each of 3 different types of tokens(excluding gold)
3. Reserve a card on the table and receive one gold token(keep in mind you may only have three r
4. Reserve an unknown card from the top of a chosen deck
5. Buy a card from your reserved cards
6. Buy a card on the table
2
Choose a token to take.
1. Onyx x3 2. Sapphire x3 3. Emerald x3 4. Diamond x4 5. Rubby x4
1
Choose a different token that you have not selected.
1. Onyx x3 2. Sapphire x3 3. Emerald x3 4. Diamond x4 5. Rubby x4
4
Choose another different token that you have not selected.
1. Onyx x3 2. Sapphire x3 3. Emerald x3 4. Diamond x4 5. Rubby x4
```

Expected

```
Player 1, what do you wish to do this round?

1. Take 2 tokens of the same color from the table(may only be performed if there is 4 or more toke
2. Take 1 token from each of 3 different types of tokens(excluding gold)
3. Reserve a card on the table and receive one gold token(keep in mind you may only have three res
4. Reserve an unknown card from the top of a chosen deck
5. Buy a card from your reserved cards
6. Buy a card on the table
2
Choose a token to take.
1. Onyx x3 2. Sapphire x3 3. Emerald x3 4. Diamond x4 5. Rubby x4
1
Choose a different token that you have not selected.
1. Onyx x3 2. Sapphire x3 3. Emerald x3 4. Diamond x4 5. Rubby x4
4
Choose another different token that you have not selected.
1. Onyx x3 2. Sapphire x3 3. Emerald x3 4. Diamond4 5. Rubby x4 6. Gold x3
```

What we actually got

Blackbox testing: Can choose gold

Choose another different token that you have not selected.

1. Onyx x3
2. Sapphire x3
3. Emerald x3
4. Diamond4
5. Rubby x4
6. Gold x3

Should be Diamond x4 instead of Diamond4

Reserving a card is also the only way to get a gold token (joker). If there is no gold left, you can still reserve a card, but you won't get any gold.

Reserve three+ card

For test case: testReserveThreeCard()

Player 2 reserve three card: one Green, oneYellow, and one Blue, and would like to reserve the fourth card

According to the rule Players may not have more than three reserved cards in hand, and the only way to get rid of a card is to buy it.

Instead of showing: **No, you should not reserve more card.**

The result is: **The deck you chose ran out of cards. Please choose another one**

Blackbox testing: Game that never ends

When Player 2 already have three cards, and they accidentally choose option 4. They could never end the game!

The following conversation will show up no matter what input you give to it:

The deck you chose ran out of cards. Please choose another one.

Which deck do you wish to reserve card from?

1. Blue Deck 2. Yellow Deck 3. Green Deck

Blackbox testing: Game never ends II

In the round 4, Player 1 choose the option 2, select three different color token, player 1 choose 1 onyx but there is no other Jewelries left.

The following conversation will show up no matter what input you give to it:

Sorry. There is not enough tokens on the table. Please re-enter what you want.

Please choose the tokens you want to take.

1. Onyx x2 2. Sapphire x0 3. Emerald x0 4. Diamond x0 5. Rubby x0

Lesson-learned

JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Thank you!

Open to Q & A