

Test Script Documentation

The following document contains a summary of the test cases, sample inputs, and results of the test scripts. The document is separated into six sections: View, Card Classes, Deck Classes, Core Classes, Space Classes, and Appendices. For Section A, sets of screenshots of a game with two players and a game with three players are compiled for the View test documentation.

The documentation for the model test scripts are summarized into individual tables. Sample inputs and outputs that are too long or detailed to fit in the table are included as appendices, while the entries of the fifth column (Expected Result/Updated Object Status) for such cases contain a description of the expected result rather than the detailed result itself.

Moreover, the following legend is used to distinguish between “hard-coded” inputs and previous object statuses in the test scripts and user inputs:

Inputs encoded in blue font signify inputs and passed parameters within the test script

Inputs encoded in black font signify user-entered inputs

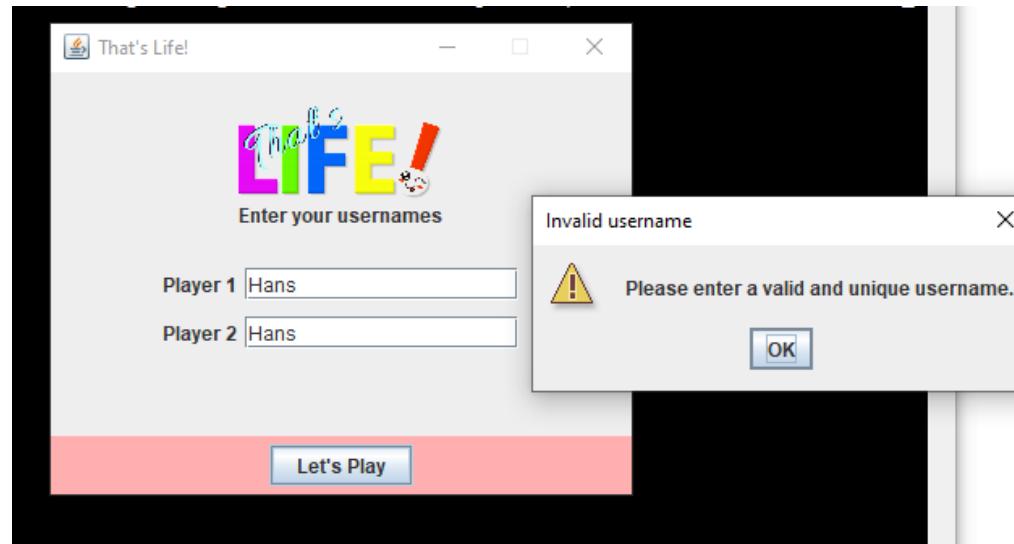
A. View

1. Game with two players



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



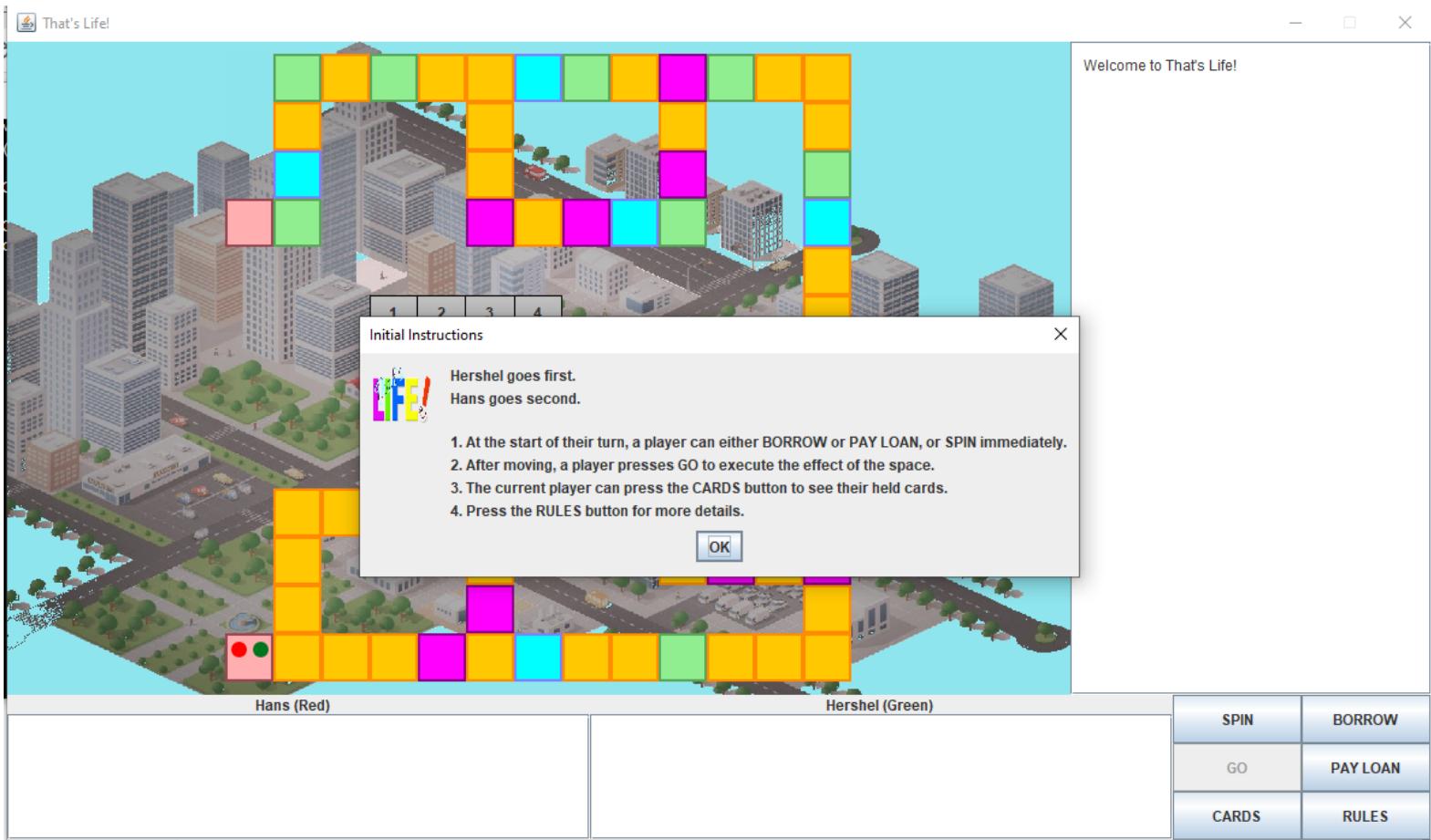
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



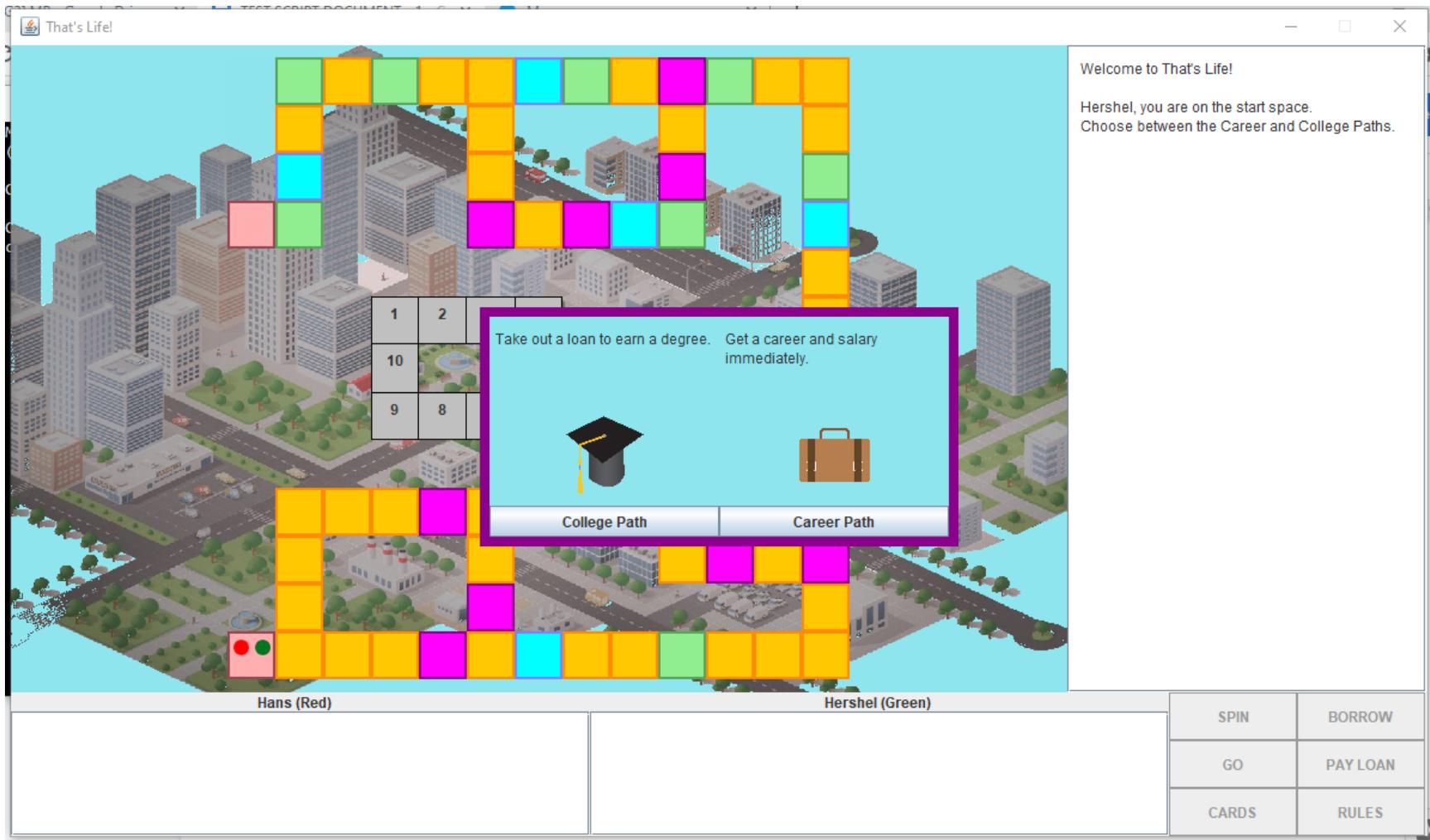
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



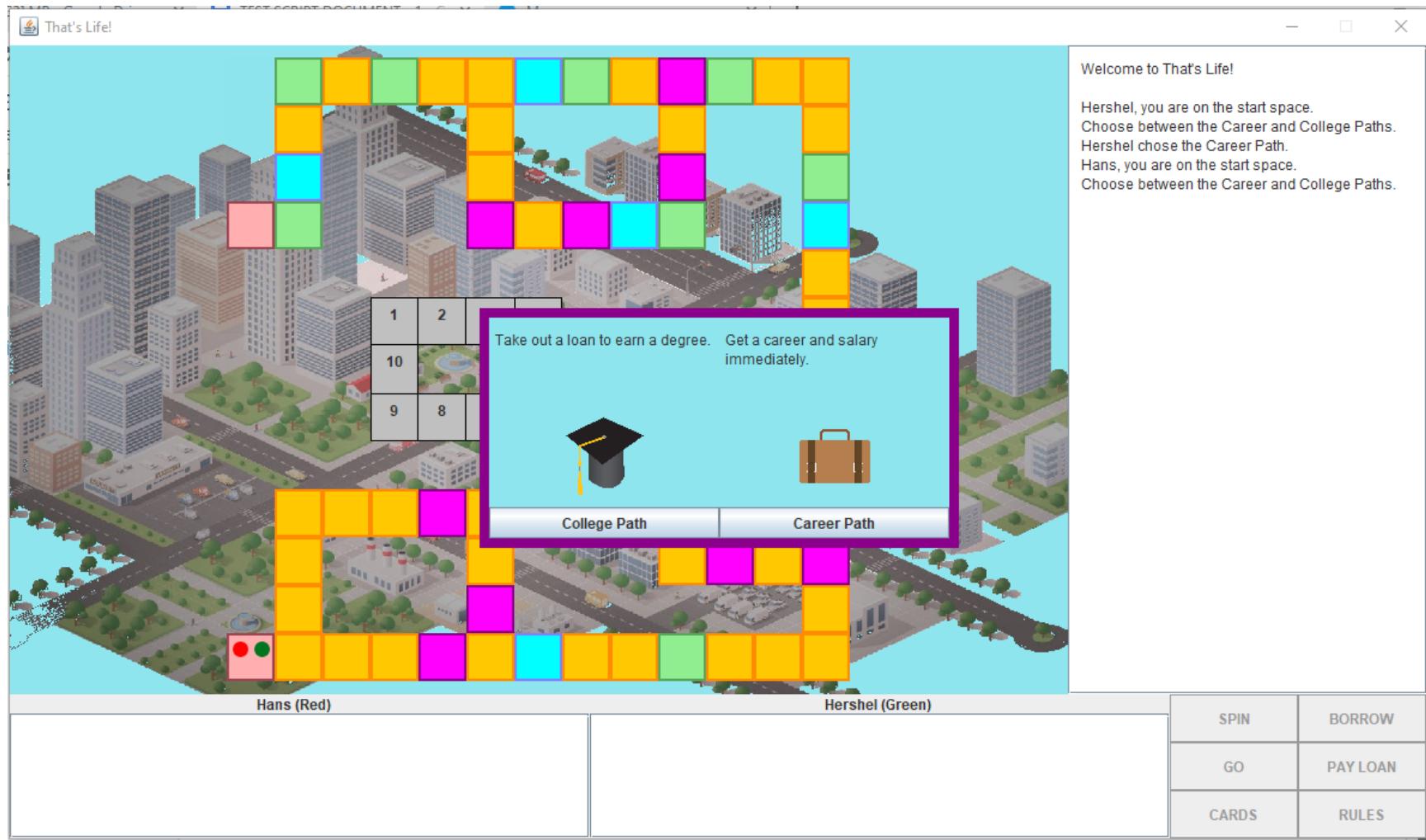
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



(NOTE: Subsequent instances of the wheel spin dialog box appearing are no longer included in this documentation. However, the dialog box appears every time the number wheel is spun.)

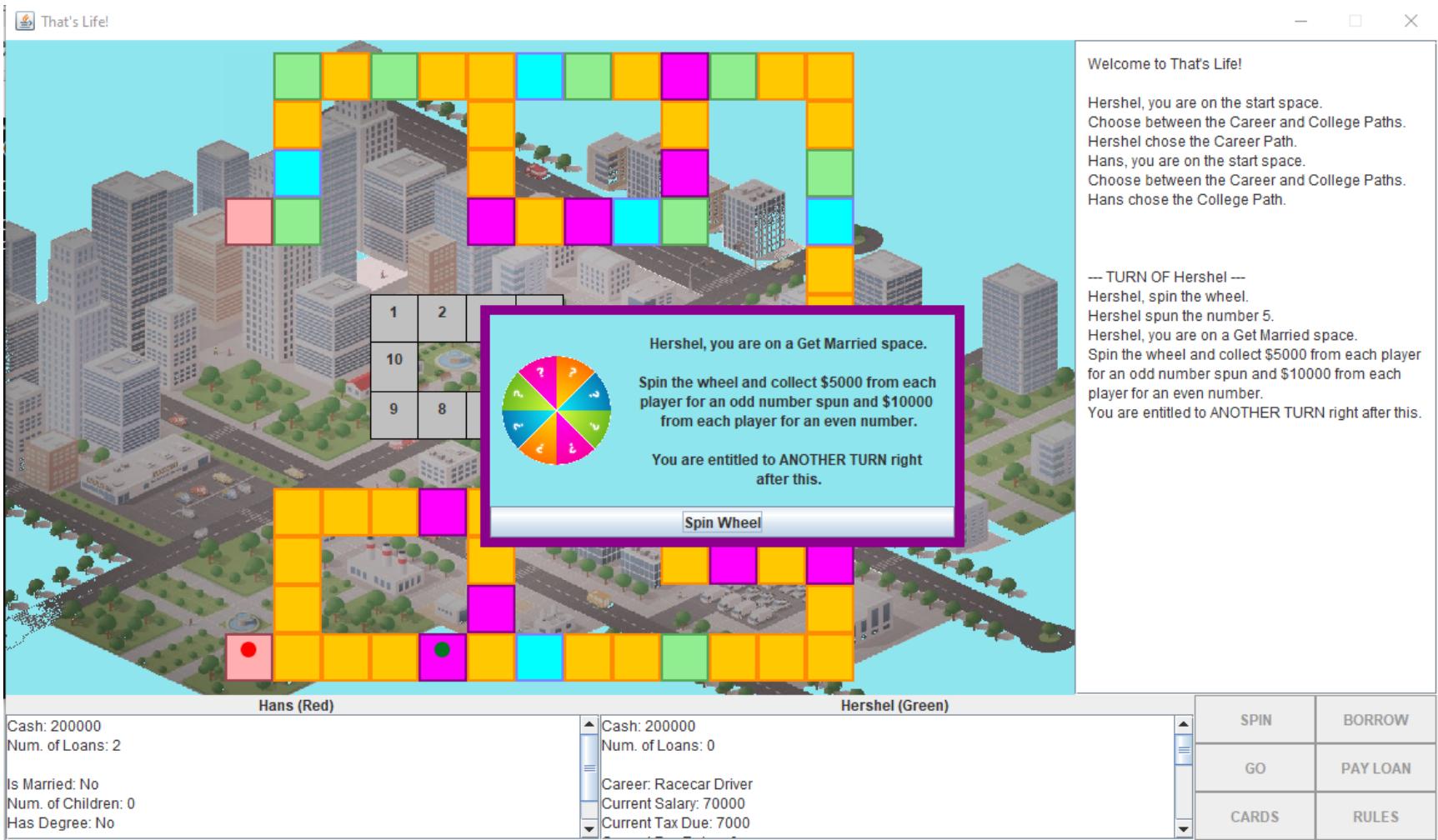
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



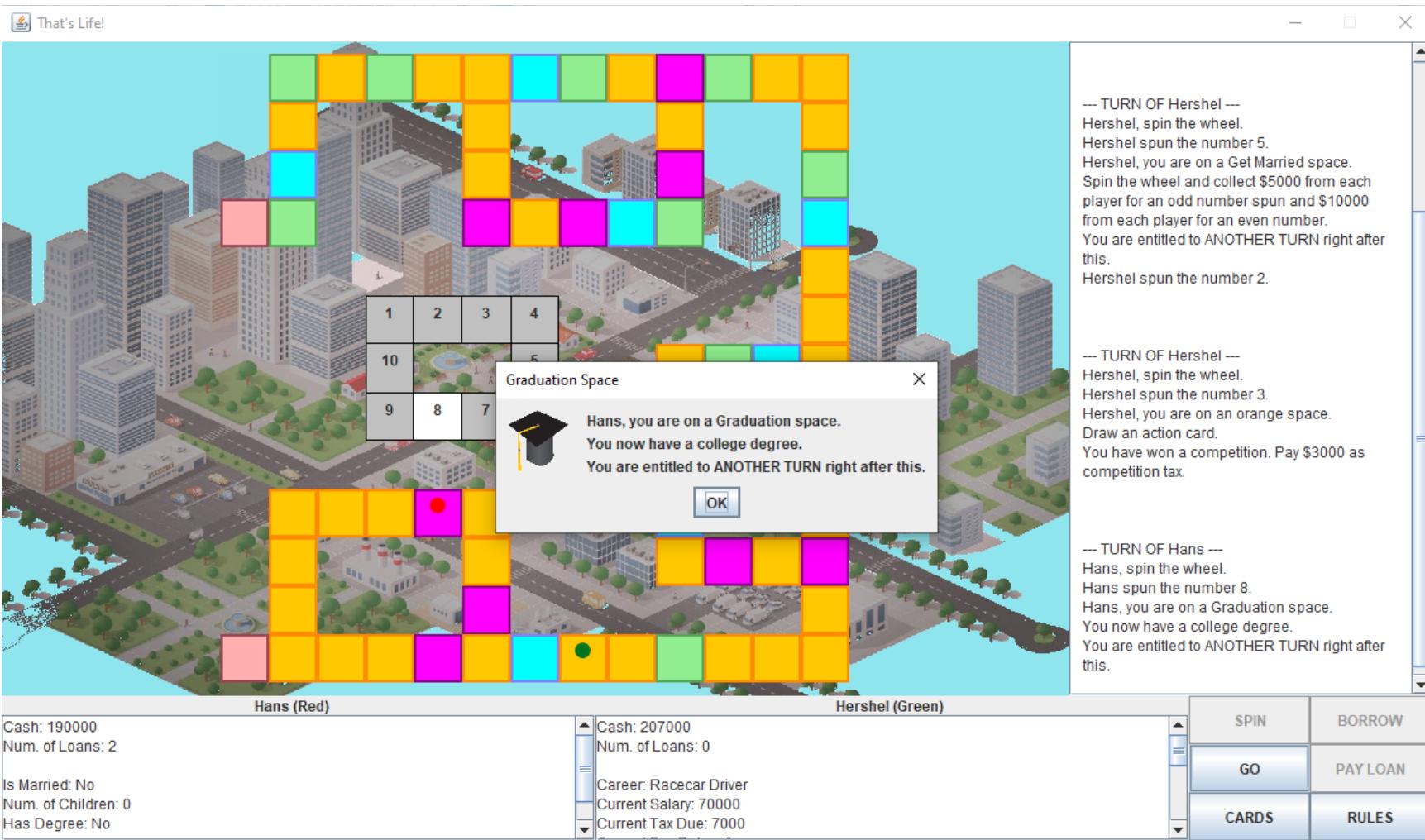
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



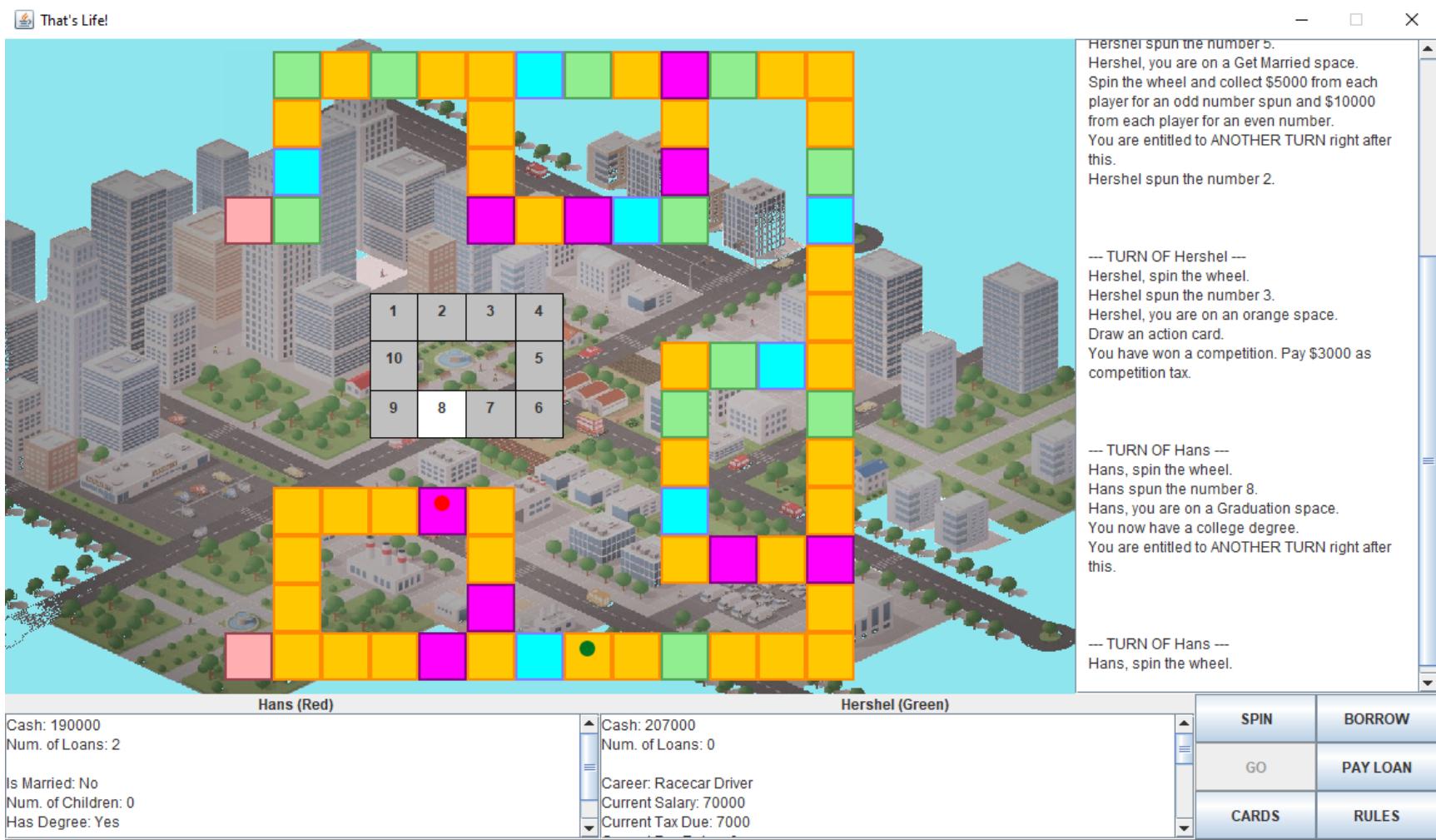
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



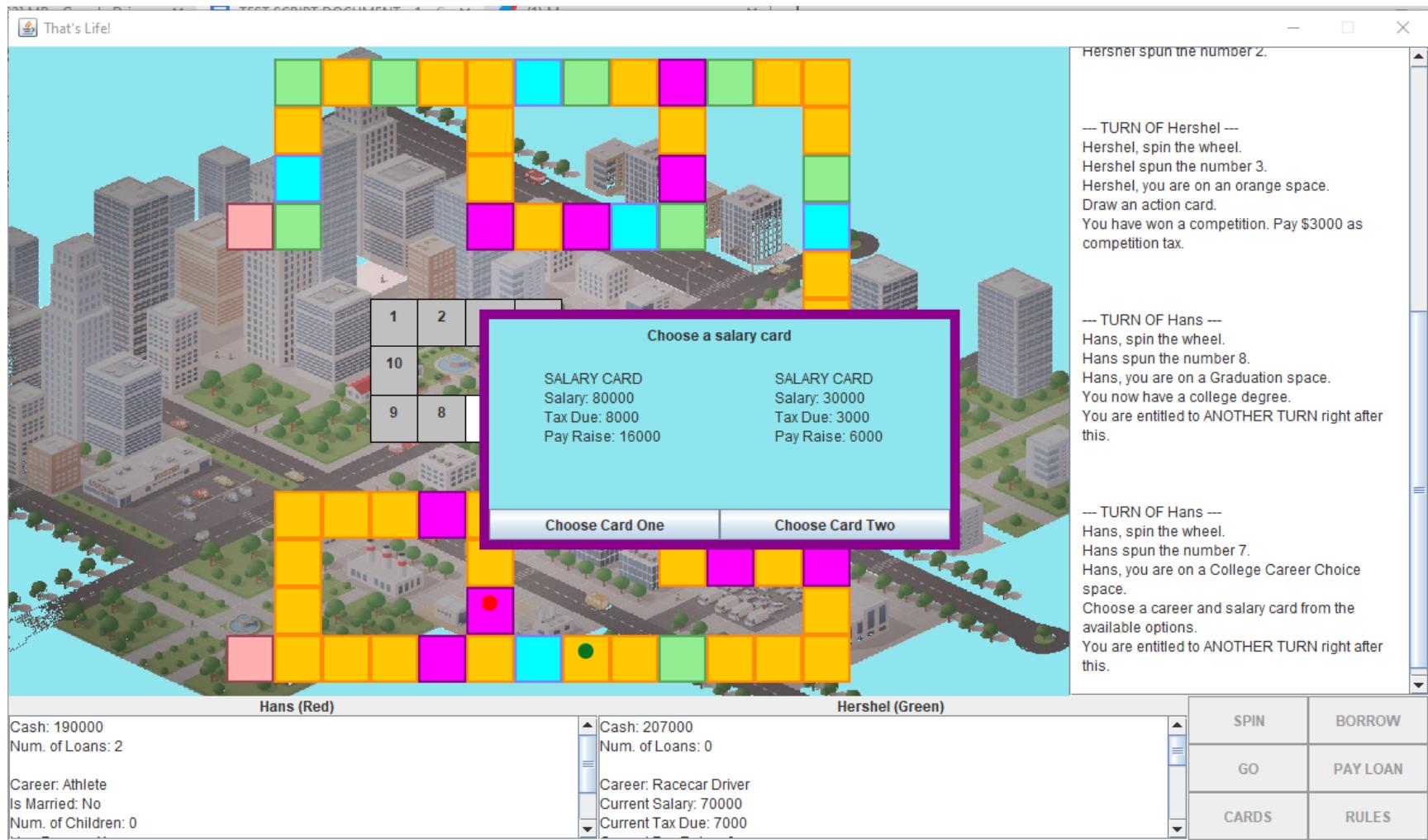
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



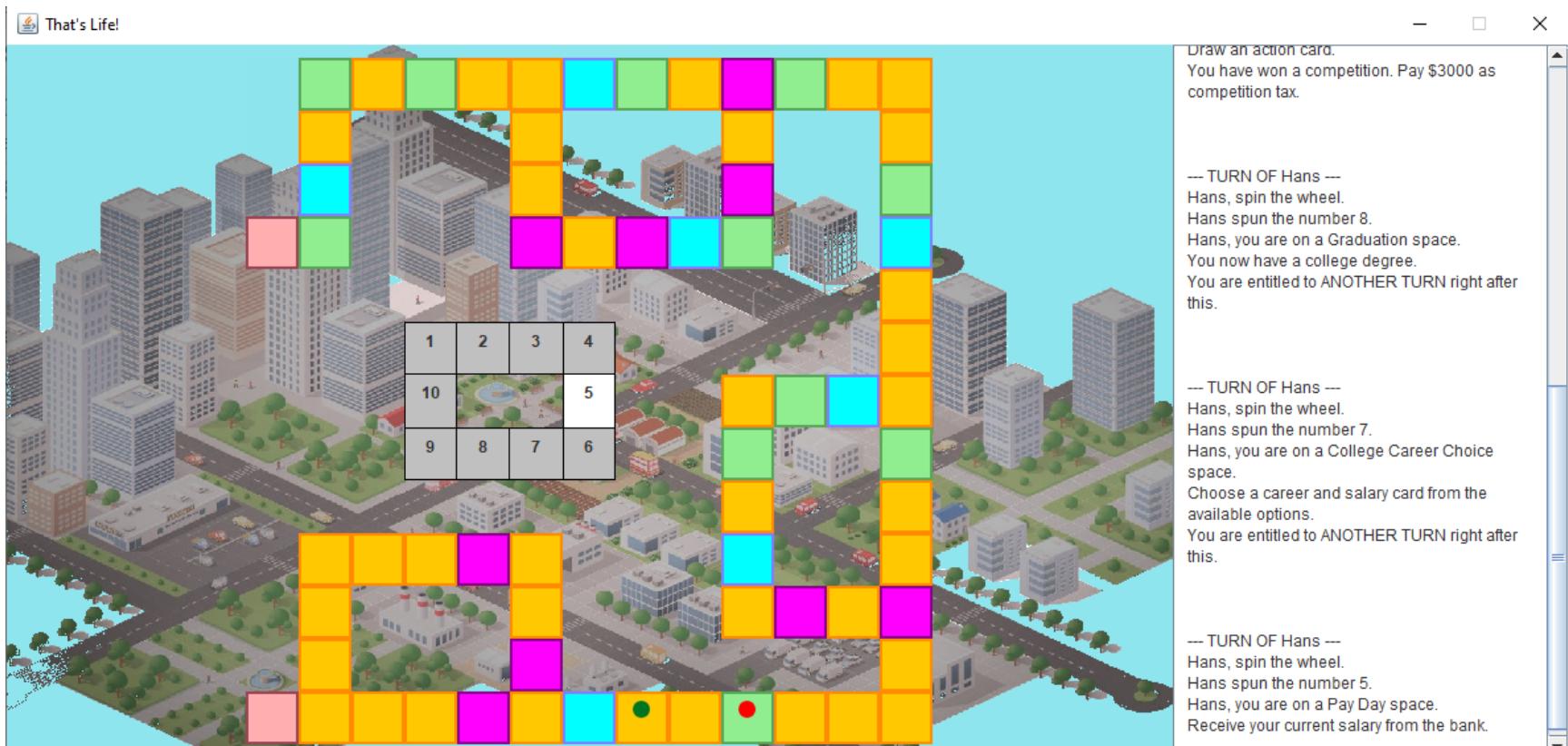
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



Hans (Red)

Cash: 190000
Num. of Loans: 2

Career: Athlete
Current Salary: 80000
Current Tax Due: 8000

Hershel (Green)

Cash: 207000
Num. of Loans: 0

Career: Racecar Driver
Current Salary: 70000
Current Tax Due: 7000

SPIN	BORROW
GO	PAY LOAN
CARDS	RULES

Draw an action card.
You have won a competition. Pay \$3000 as competition tax.

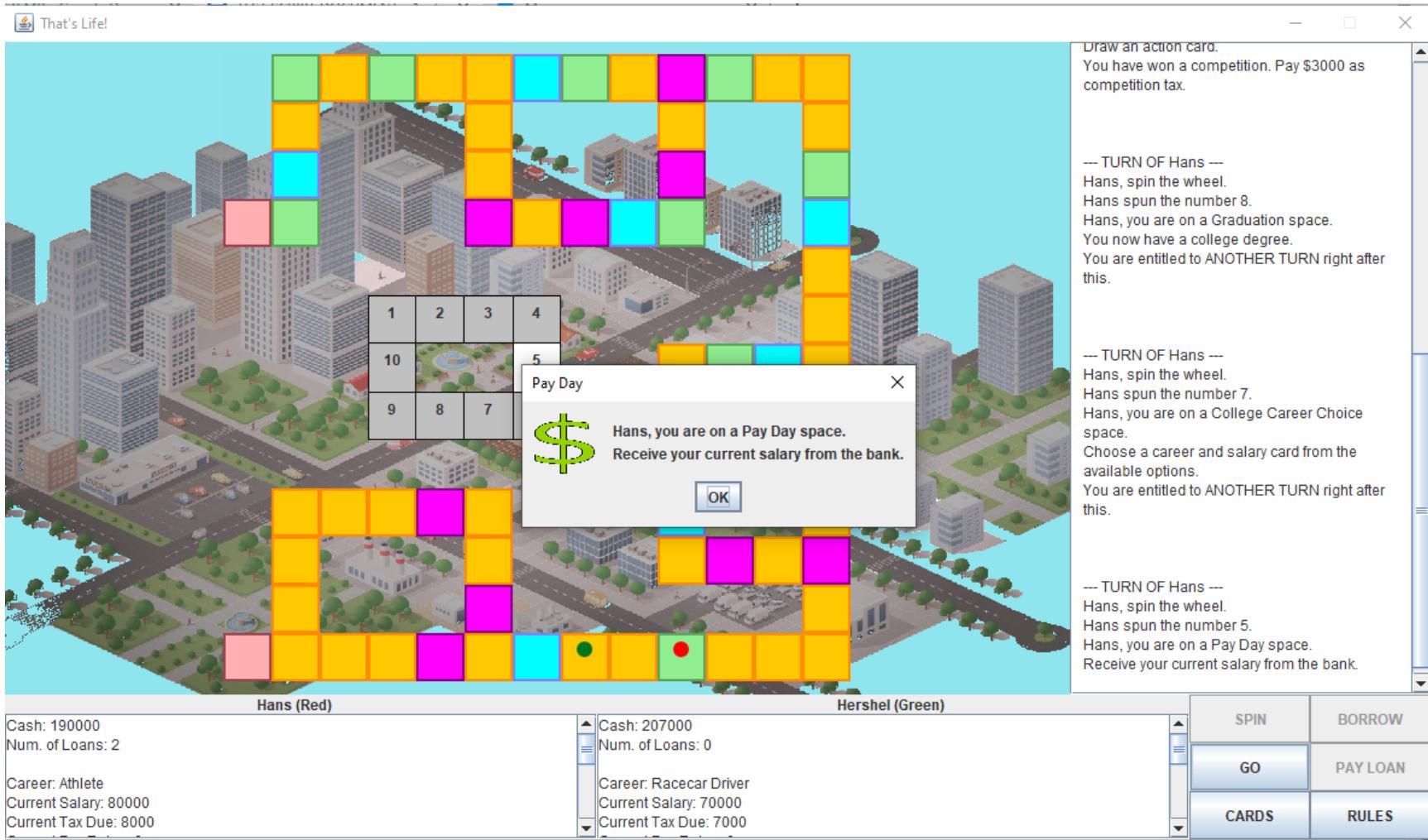
-- TURN OF Hans --
Hans, spin the wheel.
Hans spun the number 8.
Hans, you are on a Graduation space.
You now have a college degree.
You are entitled to ANOTHER TURN right after this.

-- TURN OF Hans --
Hans, spin the wheel.
Hans spun the number 7.
Hans, you are on a College Career Choice space.
Choose a career and salary card from the available options.
You are entitled to ANOTHER TURN right after this.

-- TURN OF Hans --
Hans, spin the wheel.
Hans spun the number 5.
Hans, you are on a Pay Day space.
Receive your current salary from the bank.

The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



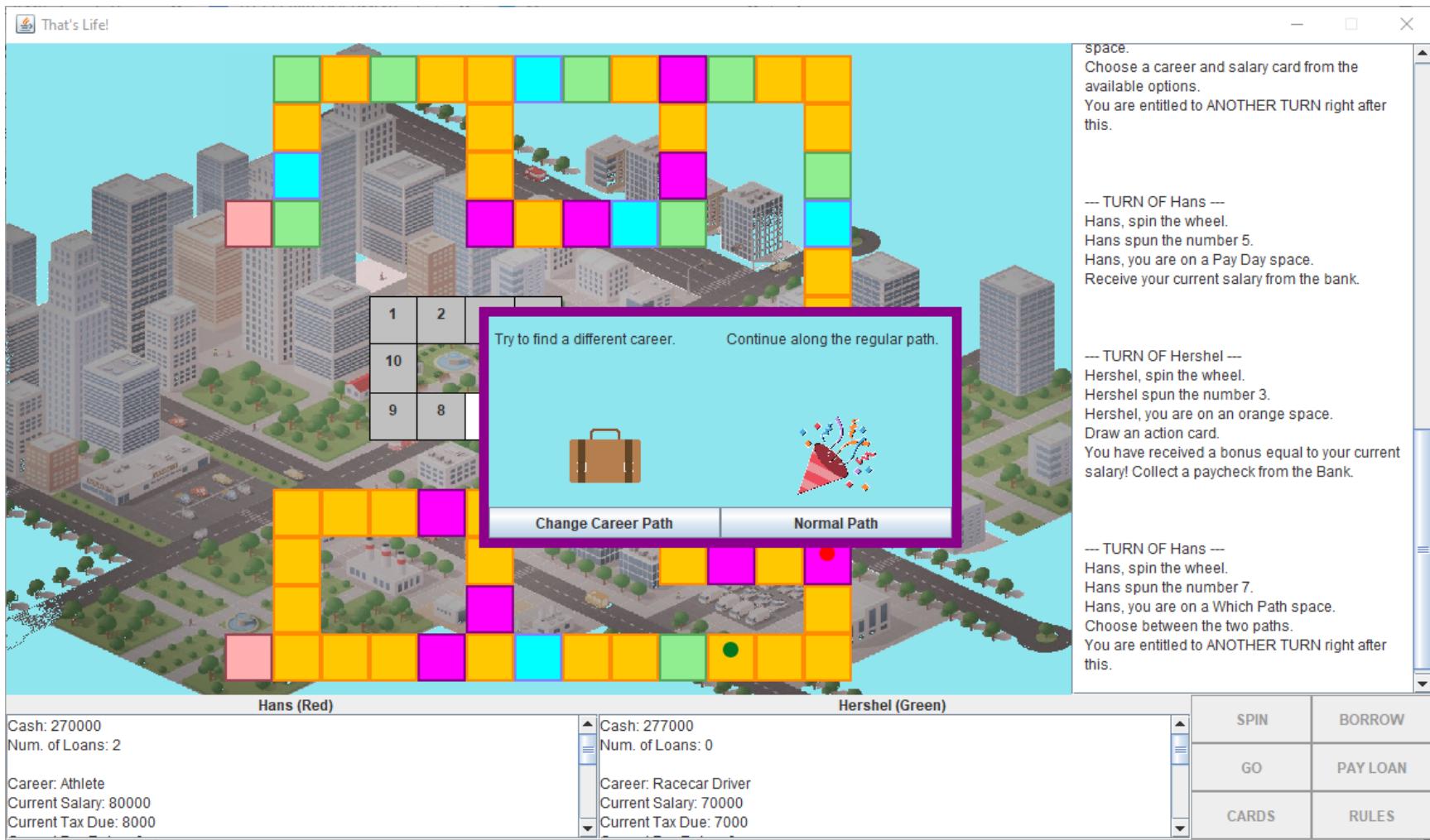
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS

That's Life!

Hans spun the number 7.
Hans, you are on a Which Path space.
Choose between the two paths.
You are entitled to ANOTHER TURN right after this.
Hans chose the Change Career Path.

-- TURN OF Hans --
Hans, spin the wheel.
Hans spun the number 1.
Hans, you are on an orange space.
Draw an action card.
You decided to travel for your vacation. You spent \$70000 on expenses.

-- TURN OF Hershel --
Hershel, spin the wheel.
Hershel spun the number 8.
Hershel, you are on a Which Path space.
Choose between the two paths.
You are entitled to ANOTHER TURN right after this.
Hershel chose the Change Career Path.

-- TURN OF Hershel --
Hershel, spin the wheel.

SPIN	BORROW
GO	PAY LOAN
CARDS	RULES

Hans (Red)

Cash: 200000
Num. of Loans: 2

Career: Athlete
Current Salary: 80000
Current Tax Due: 8000

Hershel (Green)

Cash: 277000
Num. of Loans: 0

Career: Racecar Driver
Current Salary: 70000
Current Tax Due: 7000

The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS

That's Life!

Hans chose the Change Career Path.

-- TURN OF Hans --
Hans, spin the wheel.
Hans spun the number 1.
Hans, you are on an orange space.
Draw an action card.
You decided to travel for your vacation. You spent \$70000 on expenses.

-- TURN OF Hershel --
Hershel, spin the wheel.
Hershel spun the number 8.
Hershel, you are on a Which Path space.
Choose between the two paths.
You are entitled to ANOTHER TURN right after this.
Hershel chose the Change Career Path.

-- TURN OF Hershel --
Hershel, spin the wheel.
Hershel spun the number 8.
Hershel, you are on a Job Search space.
Choose a career and salary card from the available options.
You are entitled to ANOTHER TURN right after this.

SPIN	BORROW
GO	PAY LOAN
CARDS	RULES

Hans (Red)

Cash: 200000
Num. of Loans: 2

Career: Athlete
Current Salary: 80000
Current Tax Due: 8000

Hershel (Green)

Cash: 277000
Num. of Loans: 0

Career: Racecar Driver
Current Salary: 70000
Current Tax Due: 7000

The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS

That's Life!

1	2	3	4
10			5
9	8	7	6

Hans (Red)

Hershel (Green)

SPIN	BORROW
GO	PAY LOAN
CARDS	RULES

The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



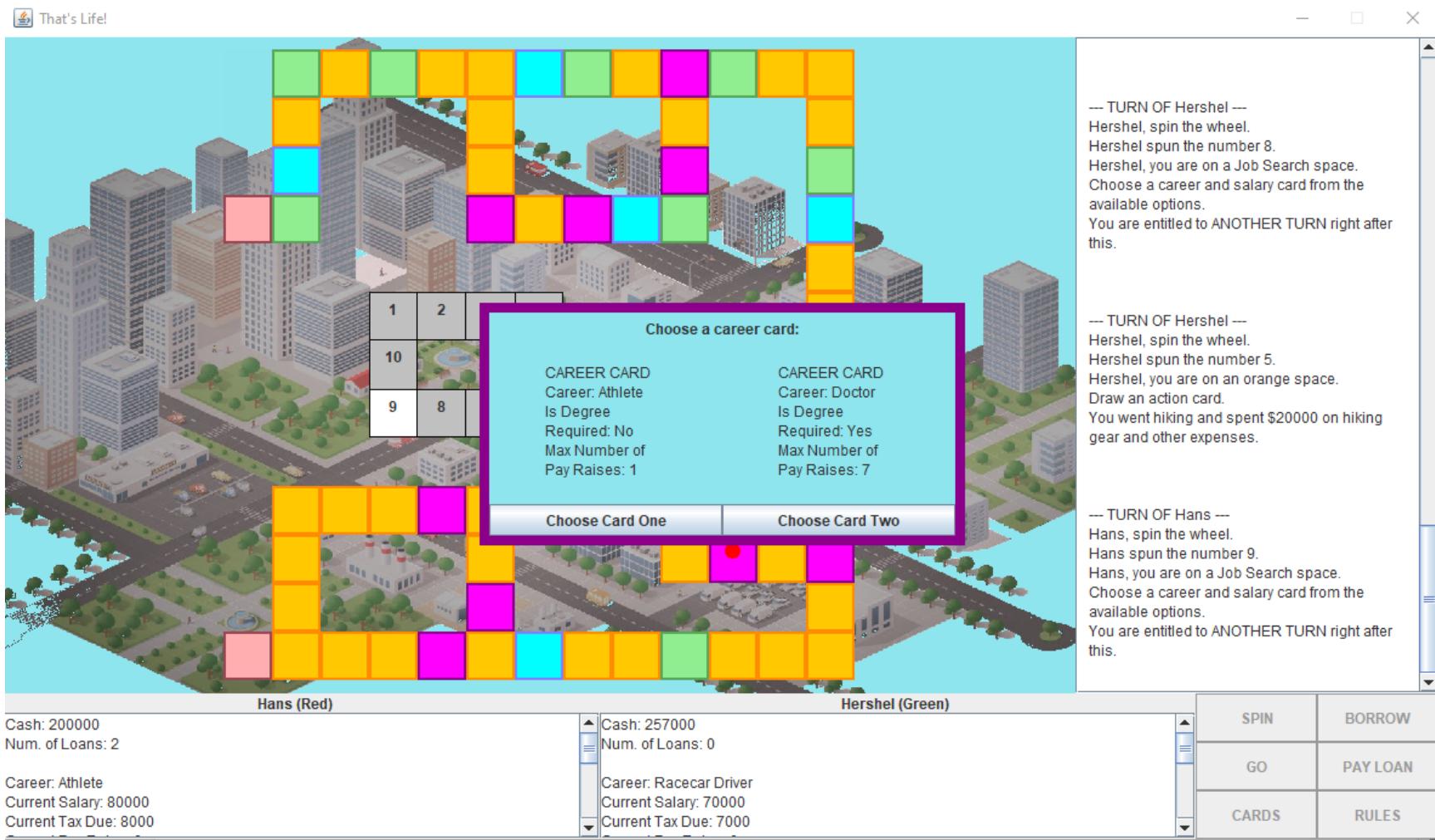
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



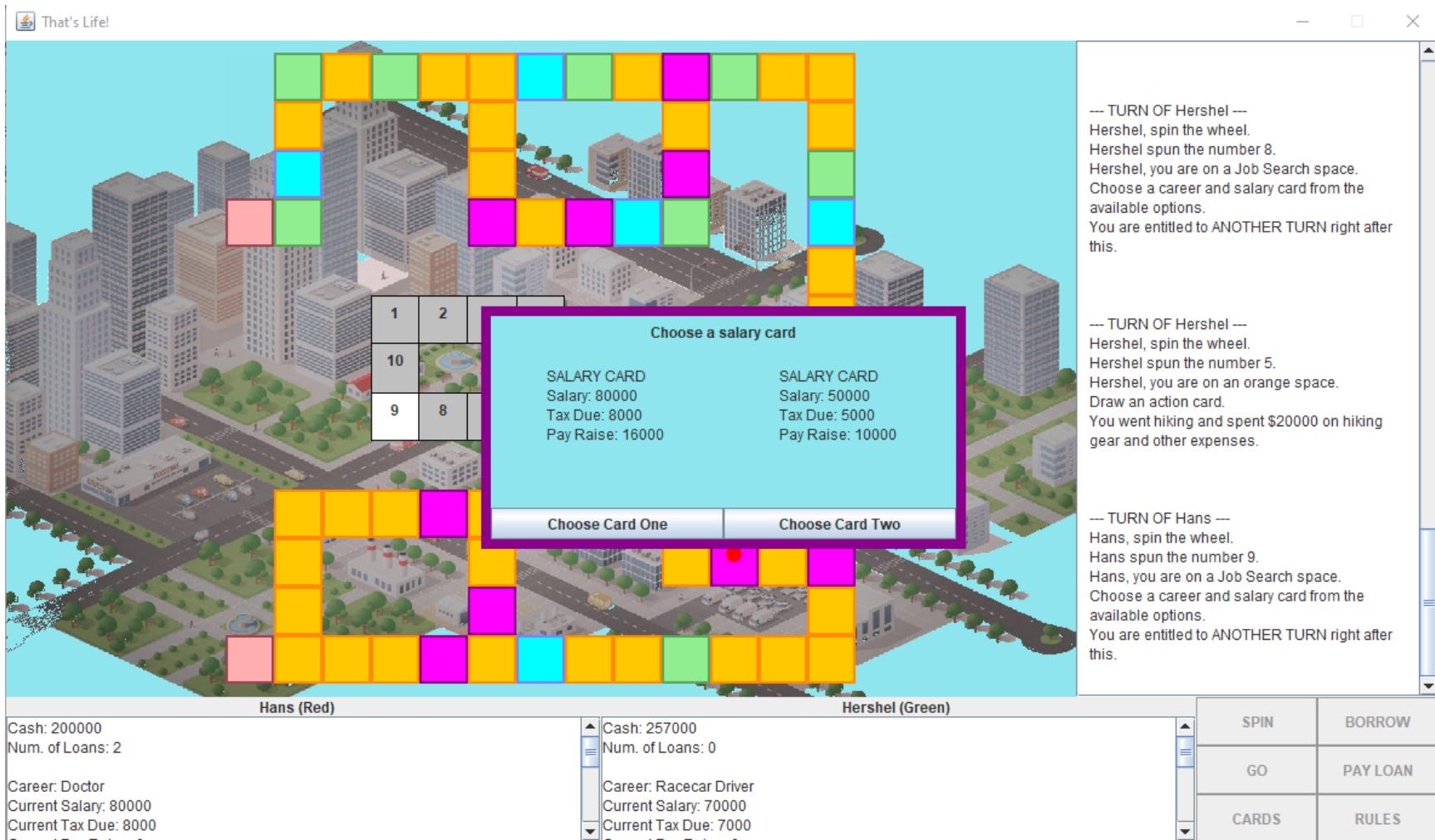
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS

That's Life!

Hans (Red)

- Cash: 130000
- Num. of Loans: 2
- Career: Doctor
- Current Salary: 80000
- Current Tax Due: 8000

Hershel (Green)

- Cash: 257000
- Num. of Loans: 0
- Career: Racecar Driver
- Current Salary: 70000
- Current Tax Due: 7000

SPIN	BORROW
GO	PAY LOAN
CARDS	RULES

gear and other expenses.

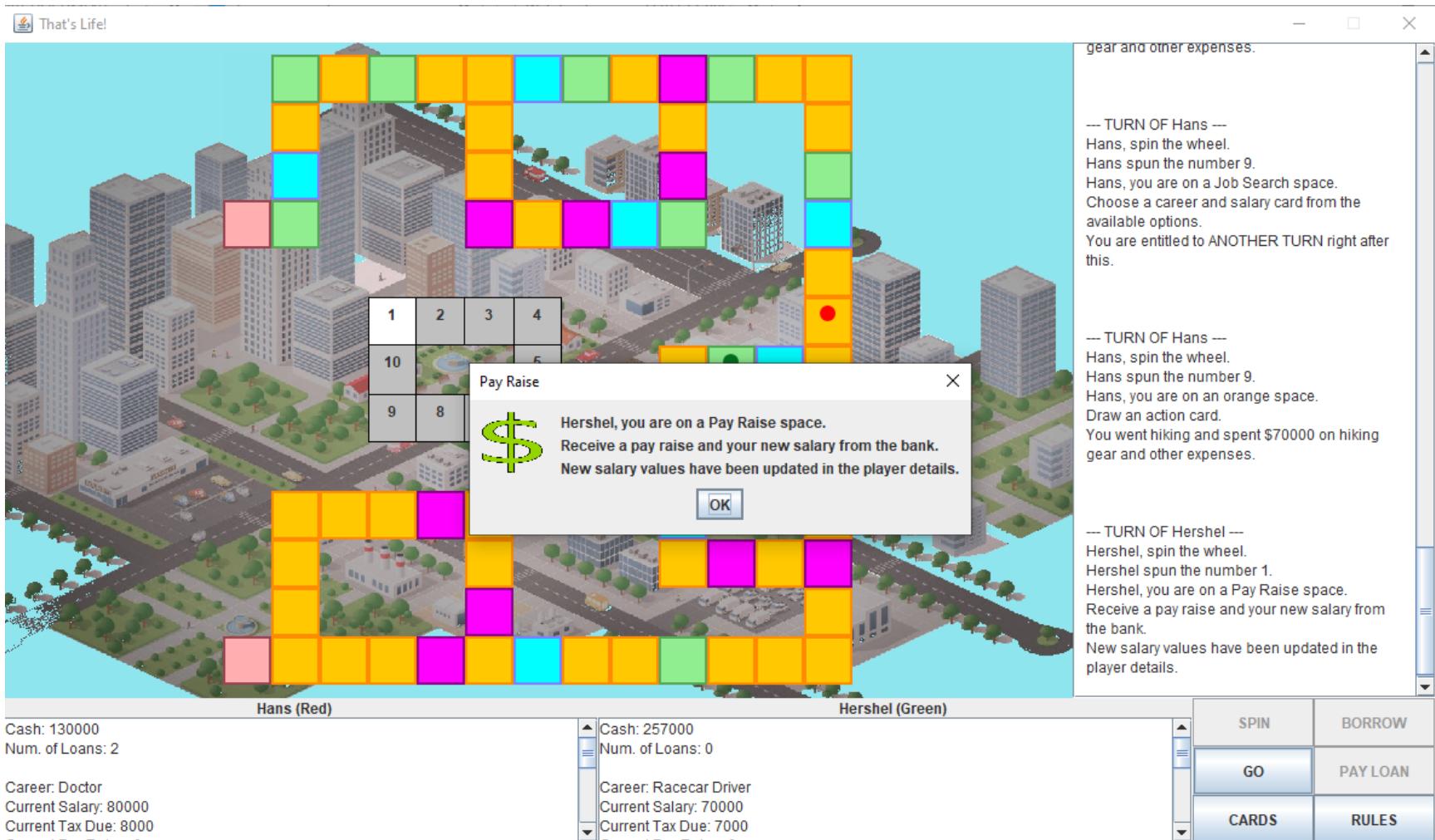
-- TURN OF Hans --
Hans, spin the wheel.
Hans spun the number 9.
Hans, you are on a Job Search space.
Choose a career and salary card from the available options.
You are entitled to ANOTHER TURN right after this.

-- TURN OF Hans --
Hans, spin the wheel.
Hans spun the number 9.
Hans, you are on an orange space.
Draw an action card.
You went hiking and spent \$70000 on hiking gear and other expenses.

-- TURN OF Hershel --
Hershel, spin the wheel.
Hershel spun the number 1.
Hershel, you are on a Pay Raise space.
Receive a pay raise and your new salary from the bank.
New salary values have been updated in the player details.

The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



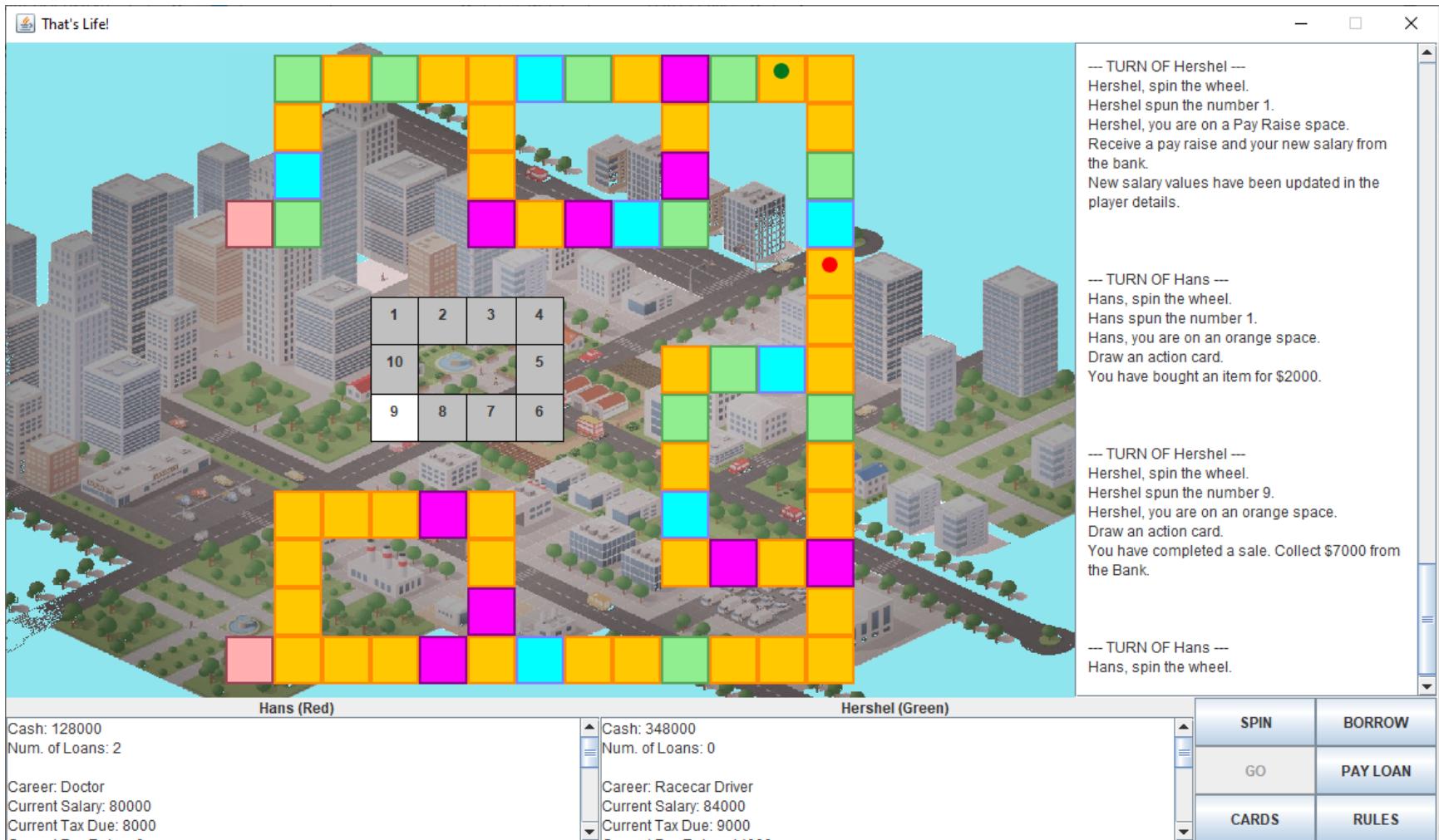
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



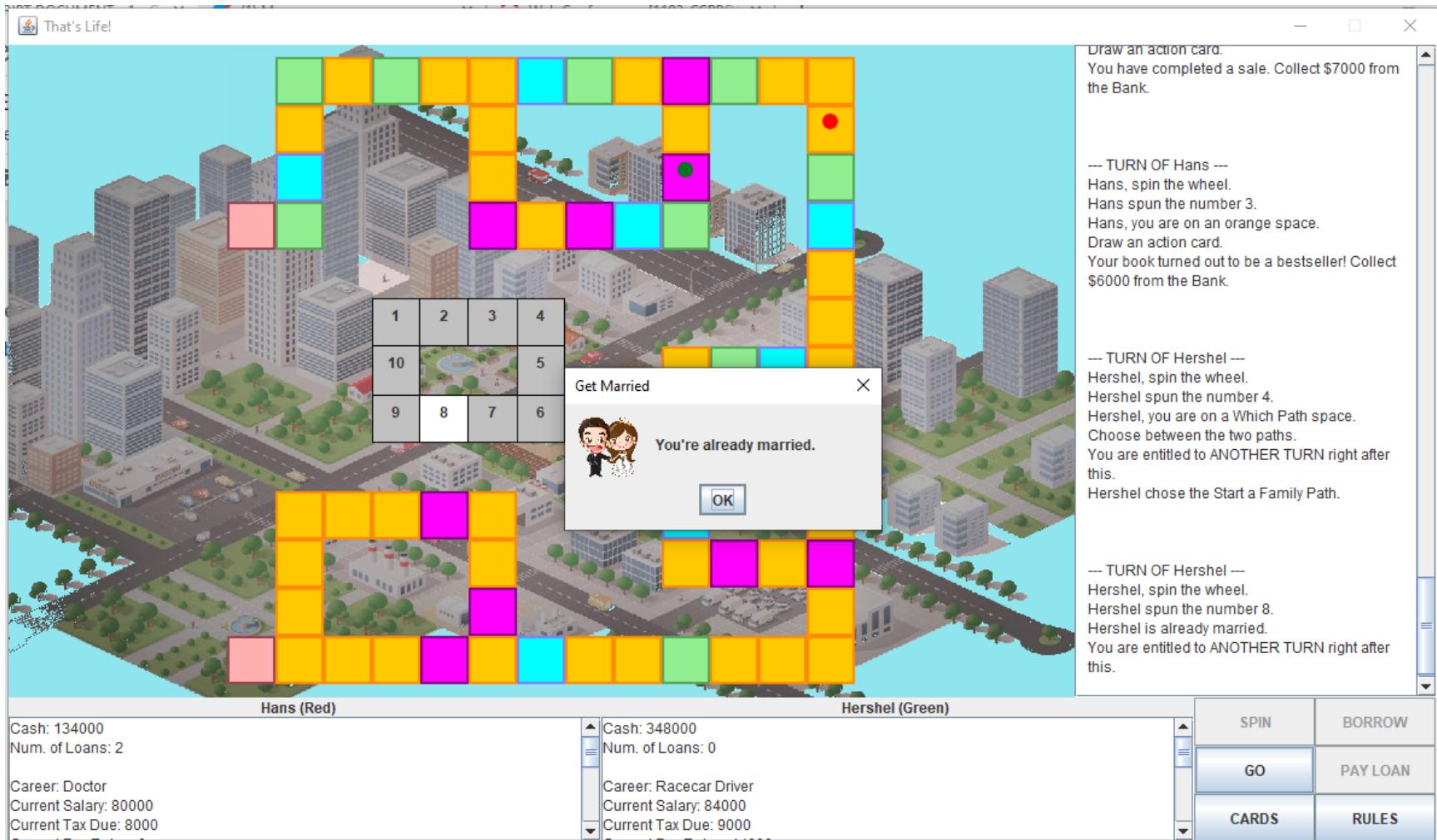
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS

That's Life!

Your book turned out to be a bestseller! Collect \$6000 from the Bank.

— TURN OF Hershel —
Hershel, spin the wheel.
Hershel spun the number 4.
Hershel, you are on a Which Path space.
Choose between the two paths.
You are entitled to ANOTHER TURN right after this.
Hershel chose the Start a Family Path.

— TURN OF Hershel —
Hershel, spin the wheel.
Hershel spun the number 8.
Hershel is already married.
You are entitled to ANOTHER TURN right after this.

— TURN OF Hershel —
Hershel, spin the wheel.
Hershel spun the number 8.
Hershel, you are on a Buy A House space.
Choose a house to buy from the available options.
You are entitled to ANOTHER TURN right after this.

SPIN	BORROW
GO	PAY LOAN
CARDS	RULES

Hans (Red)

Cash: 134000
Num. of Loans: 2

Career: Doctor
Current Salary: 80000
Current Tax Due: 8000

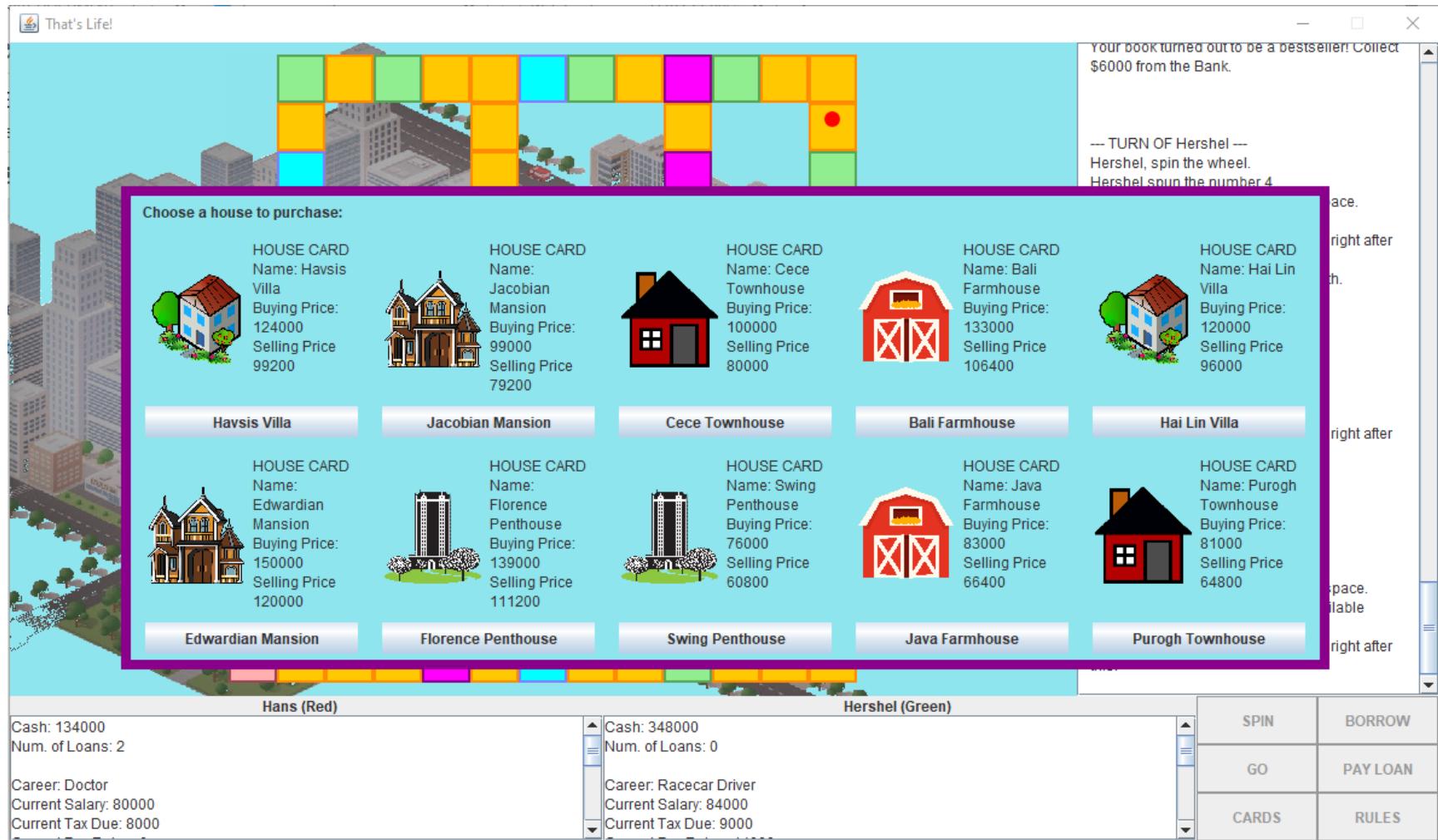
Hershel (Green)

Cash: 348000
Num. of Loans: 0

Career: Racecar Driver
Current Salary: 84000
Current Tax Due: 9000

The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



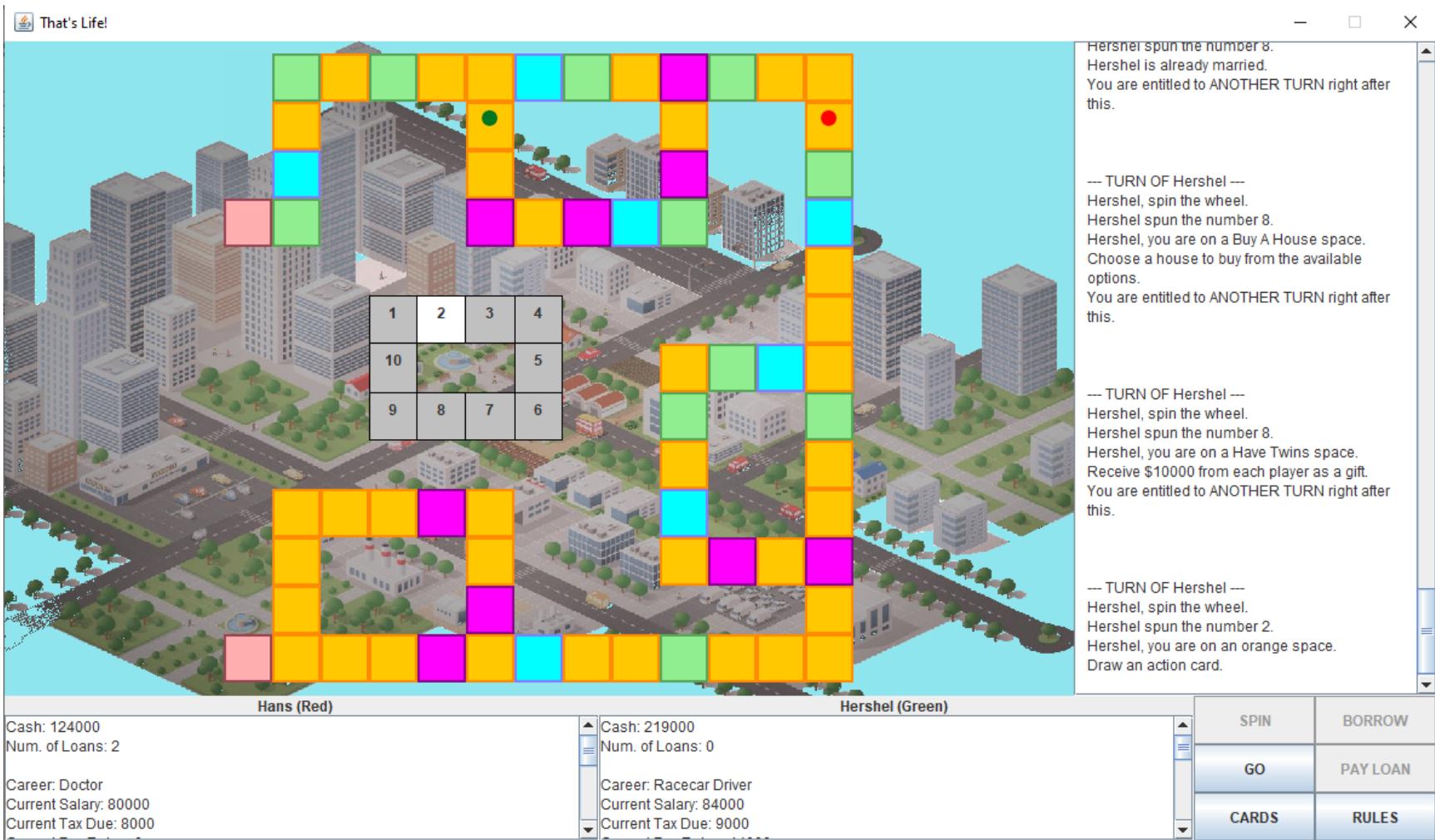
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



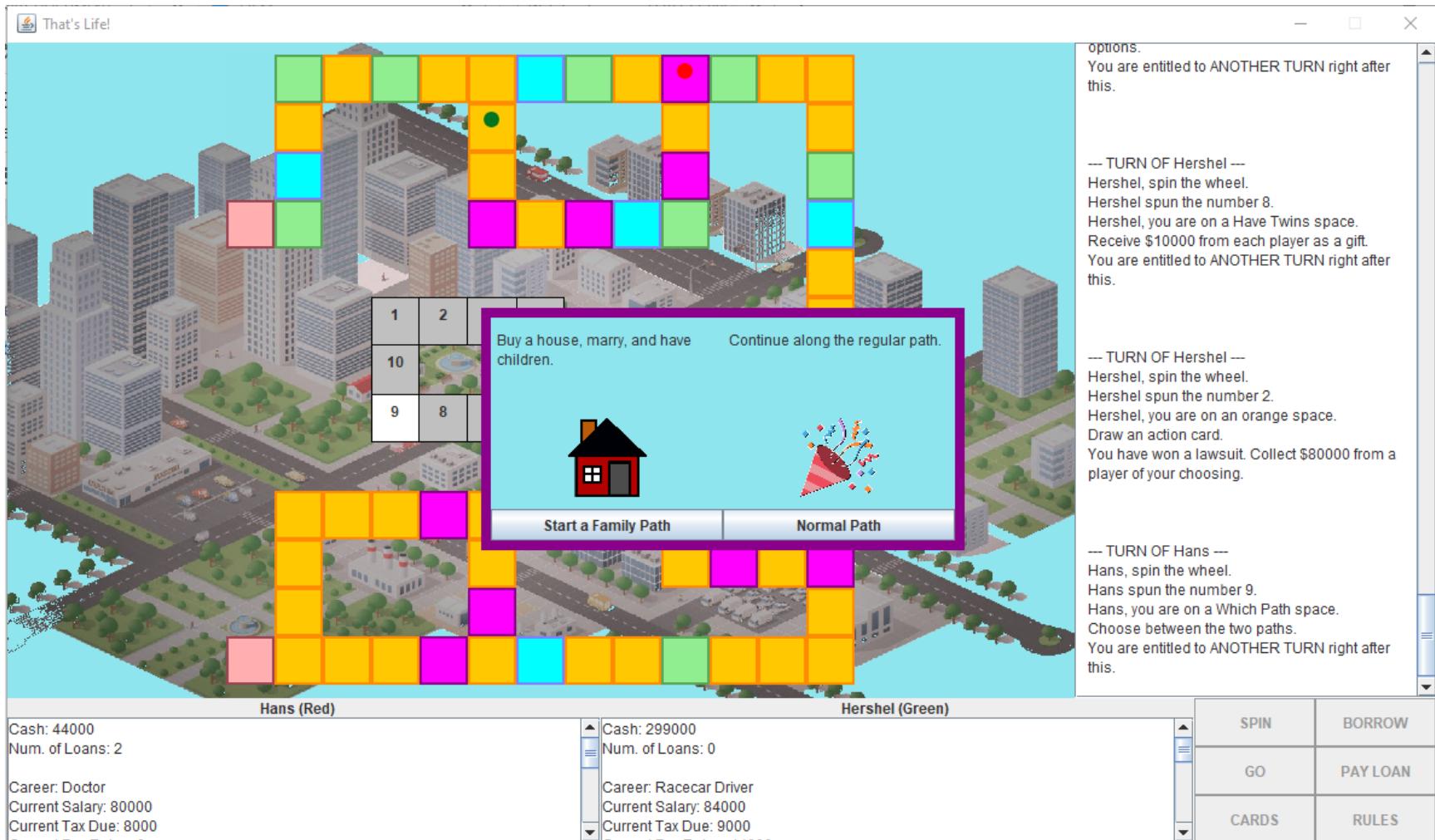
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



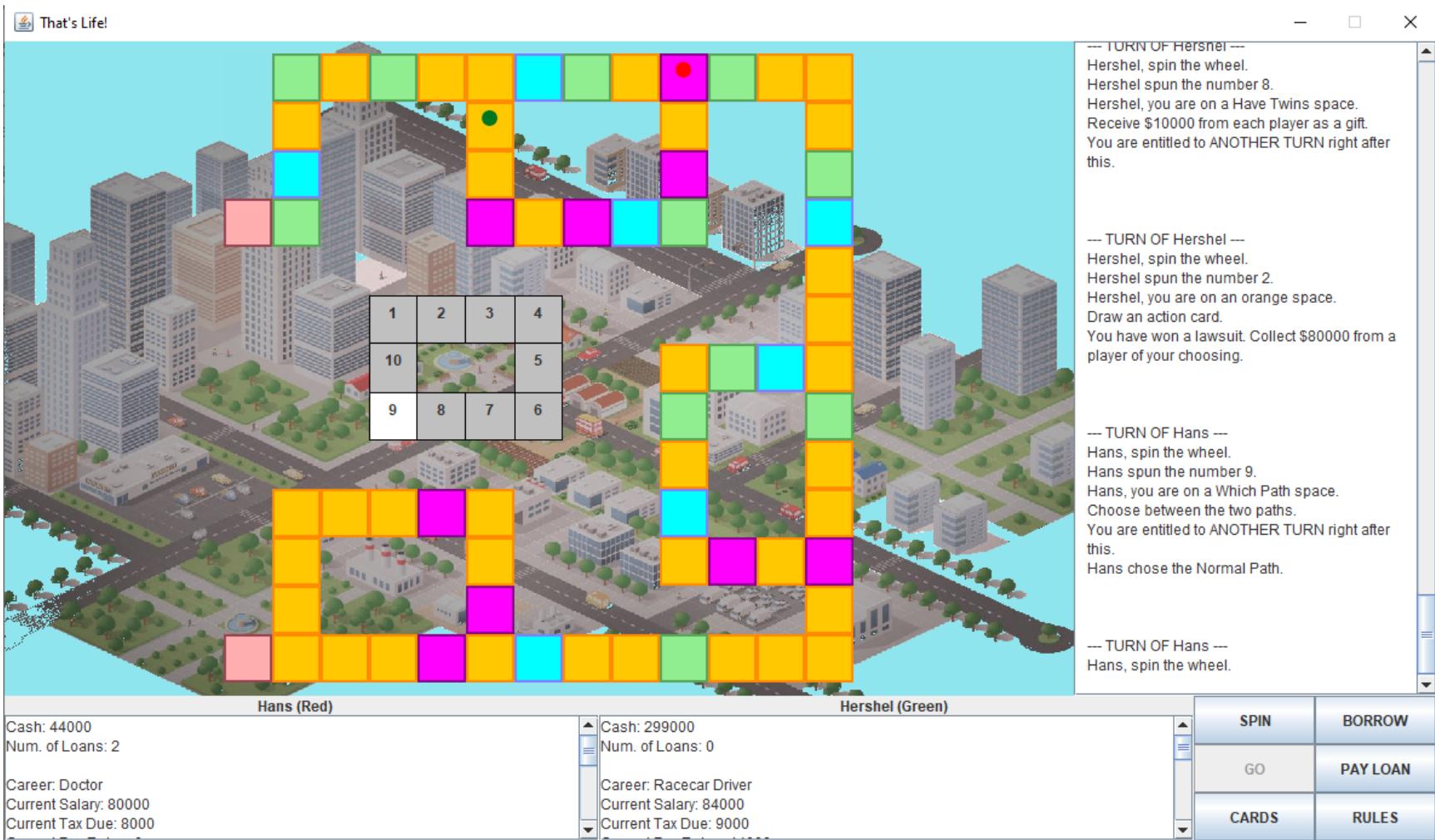
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



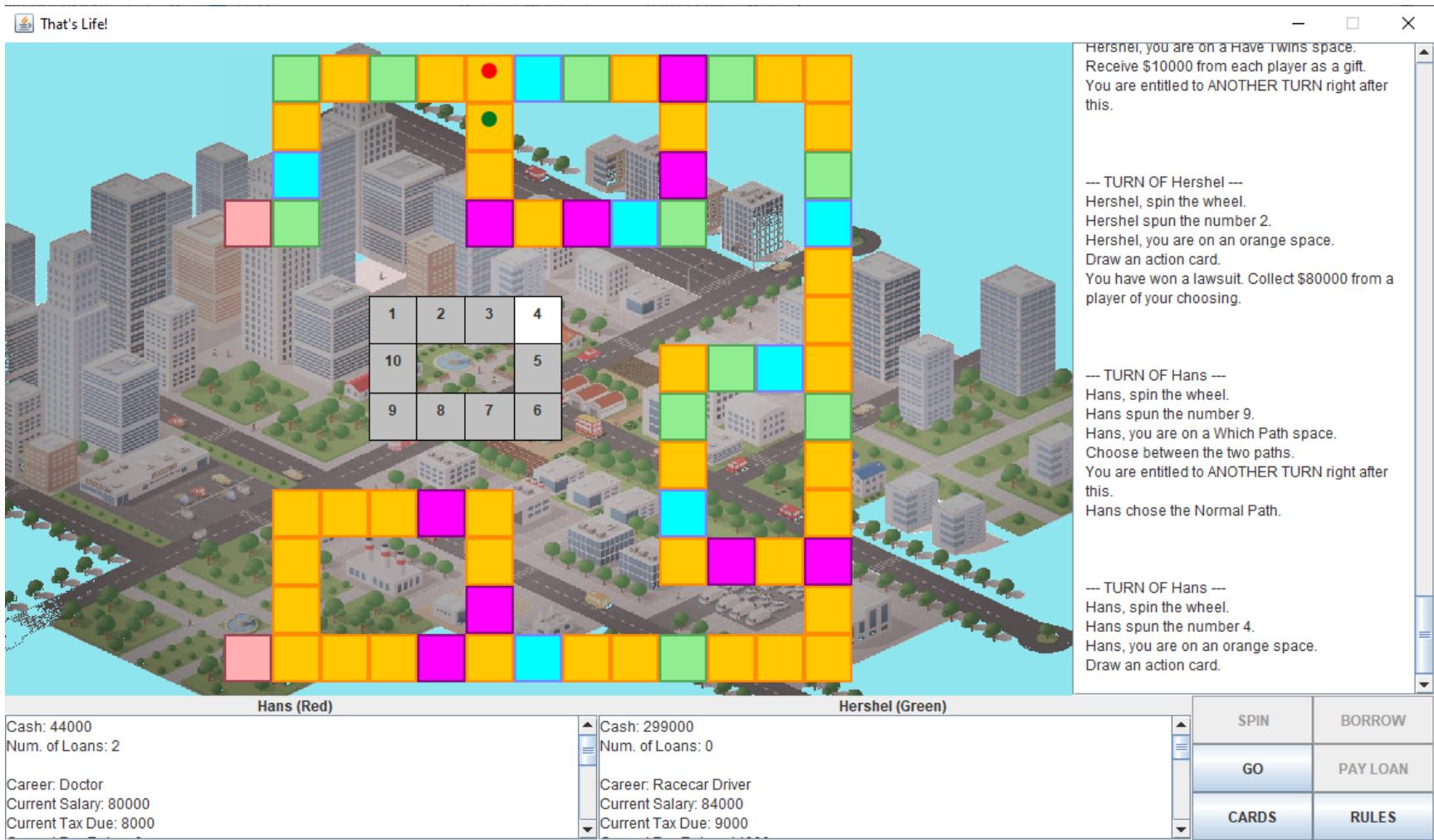
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



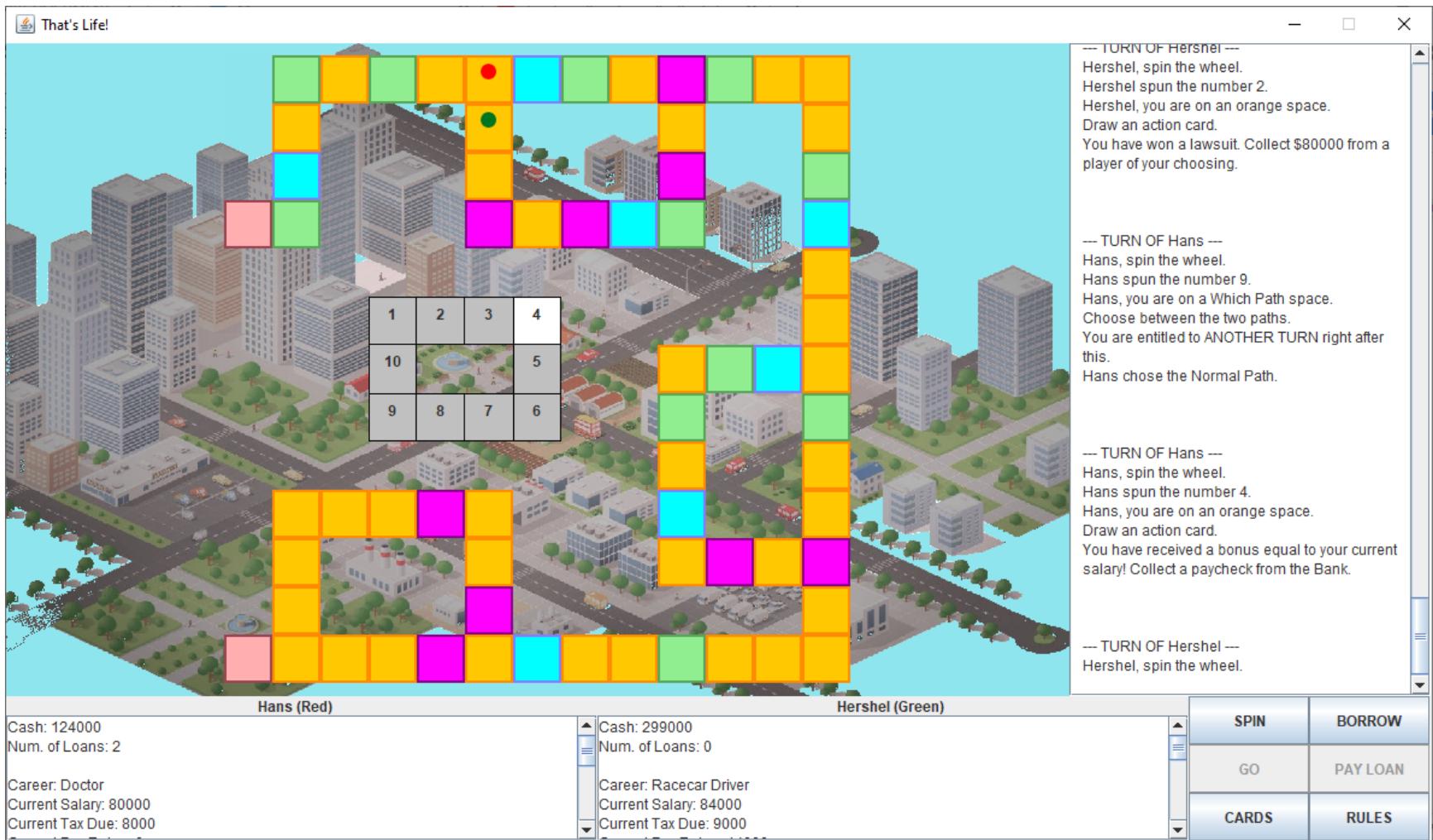
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



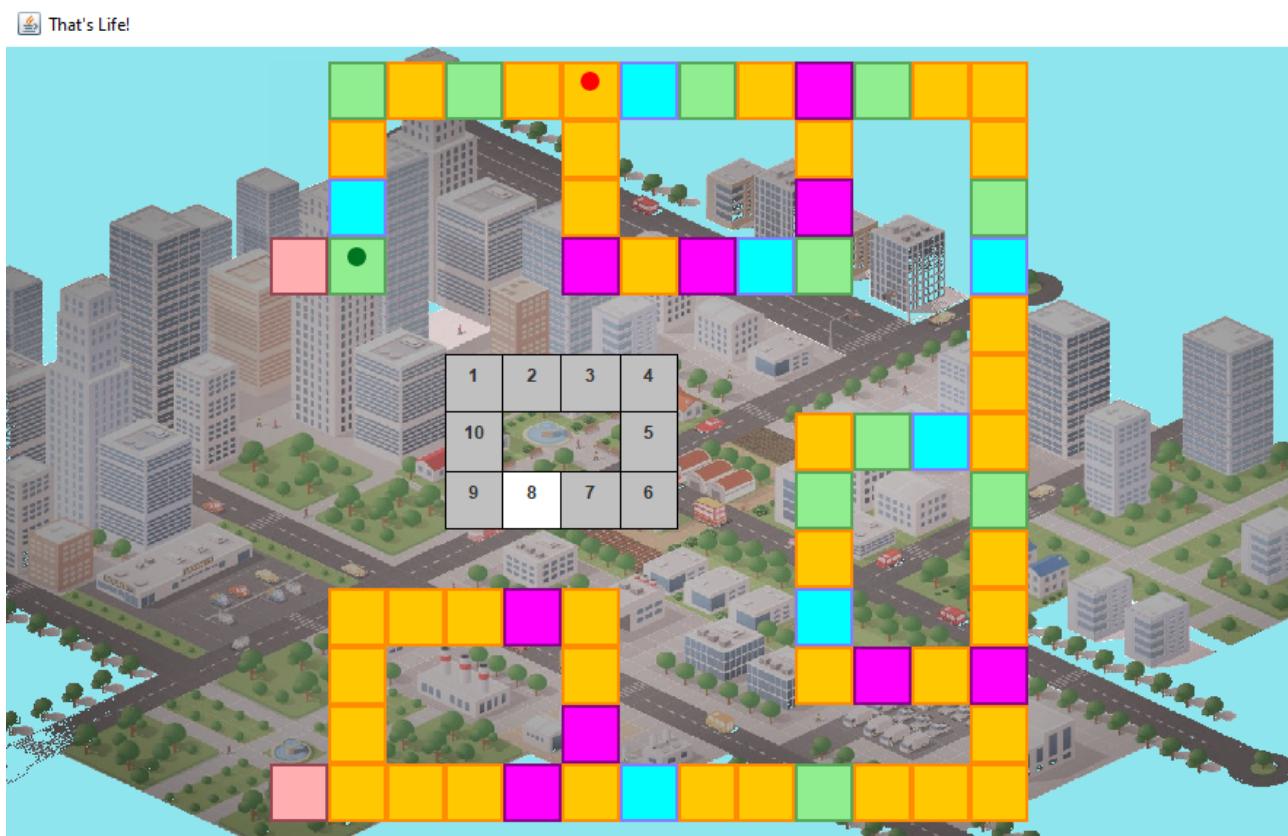
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



Hans (Red)

- Cash: 124000
- Num. of Loans: 2
- Career: Doctor
- Current Salary: 80000
- Current Tax Due: 8000

Hershel (Green)

- Cash: 299000
- Num. of Loans: 0
- Career: Racecar Driver
- Current Salary: 84000
- Current Tax Due: 9000

Game Actions:

- Hershel, you are on an orange space.
- Draw an action card.
- You have won a lawsuit. Collect \$80000 from a player of your choosing.

-- TURN OF Hans --

Hans, spin the wheel.

Hans spun the number 9.

Hans, you are on a Which Path space.

Choose between the two paths.

You are entitled to ANOTHER TURN right after this.

Hans chose the Normal Path.

-- TURN OF Hans --

Hans, spin the wheel.

Hans spun the number 4.

Hans, you are on an orange space.

Draw an action card.

You have received a bonus equal to your current salary! Collect a paycheck from the Bank.

-- TURN OF Hershel --

Hershel, spin the wheel.

Hershel spun the number 8.

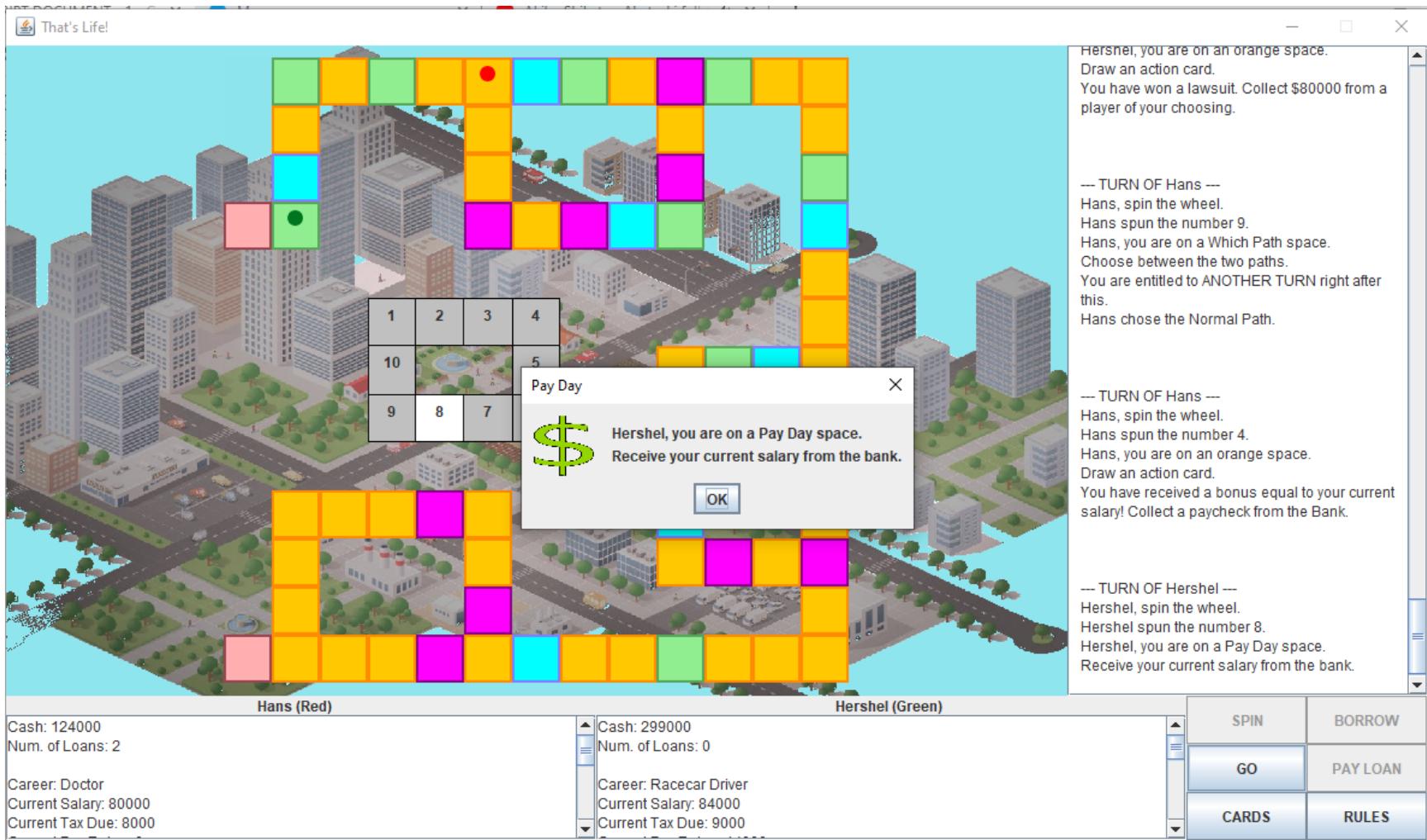
Hershel, you are on a Pay Day space.

Receive your current salary from the bank.

SPIN	BORROW
GO	PAY LOAN
CARDS	RULES

The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



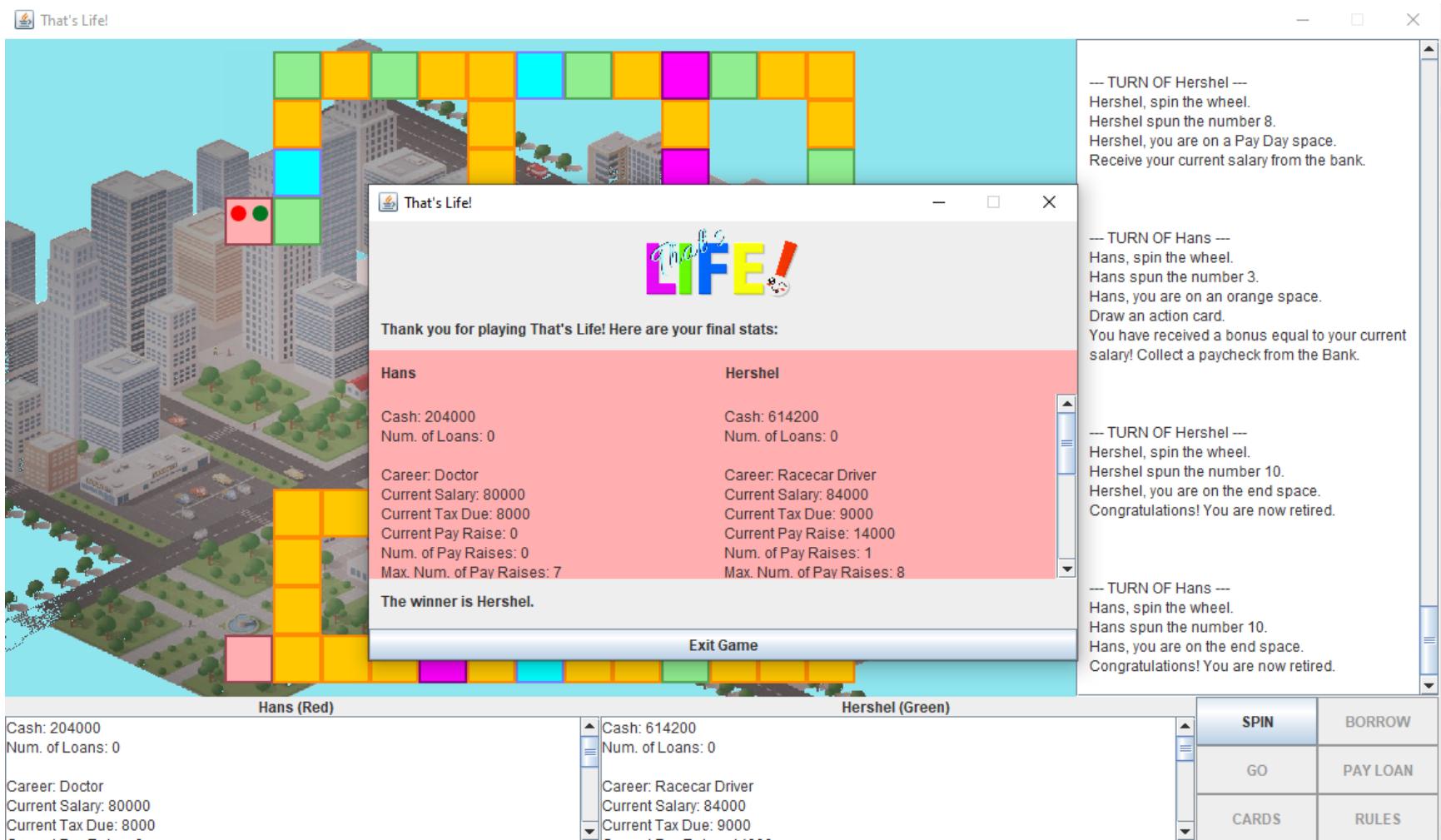
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS

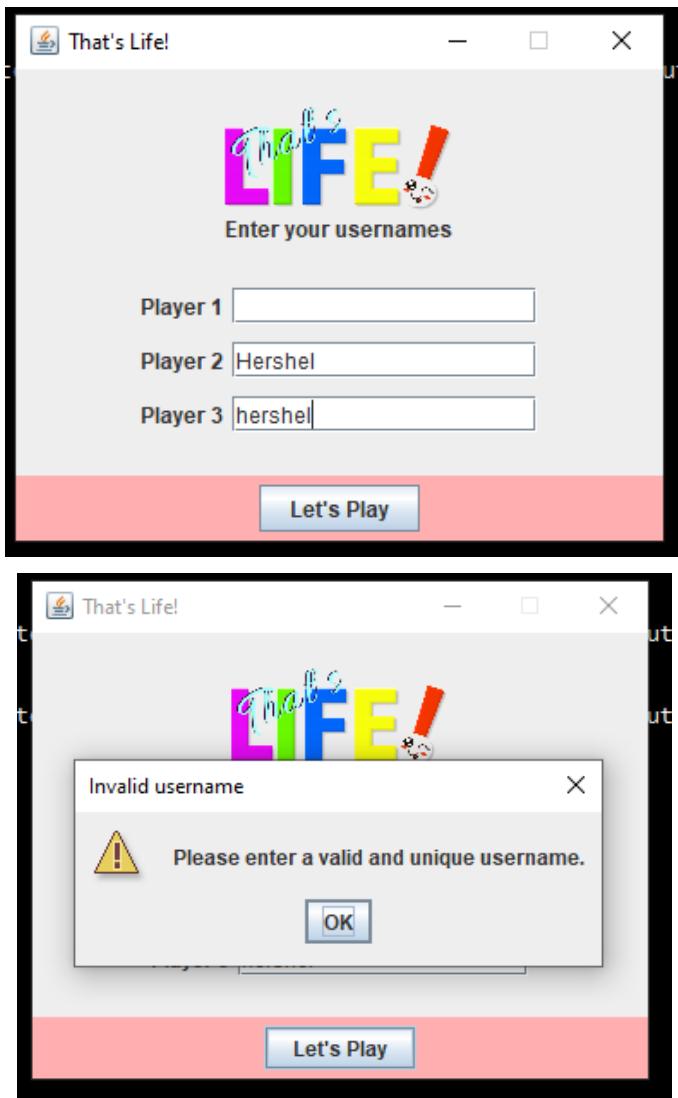
2. Game with three players

For the documentation of this run, the pop up windows that do not require user inputs (such as the dialog boxes for some of the action cards and the payday spaces) are no longer included in this document.



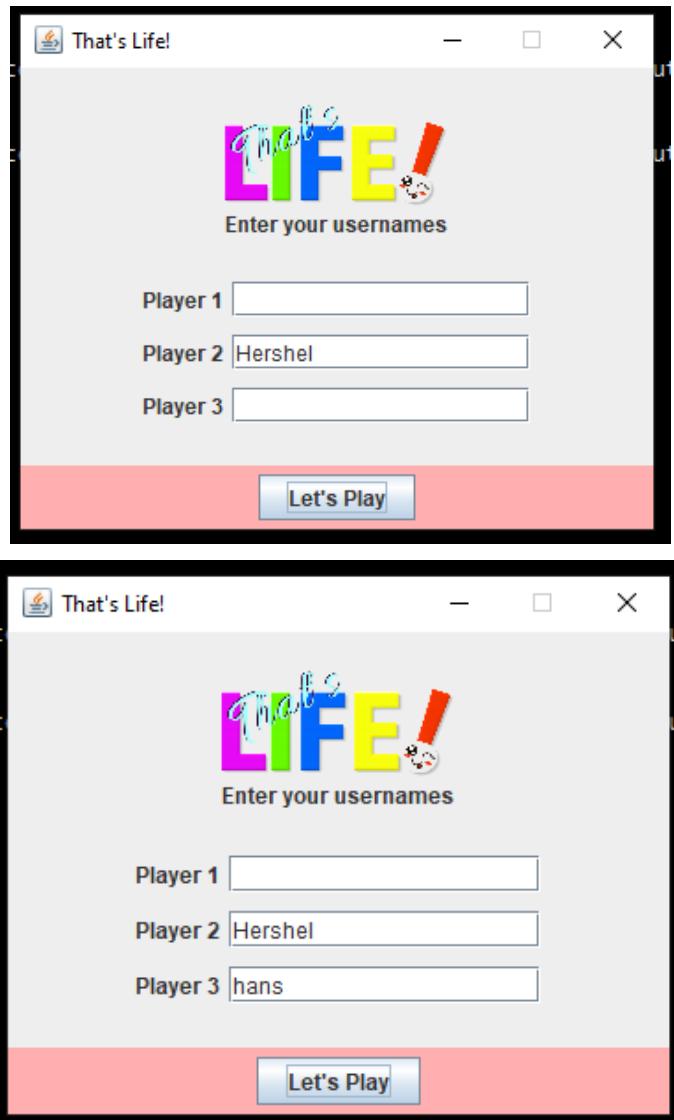
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



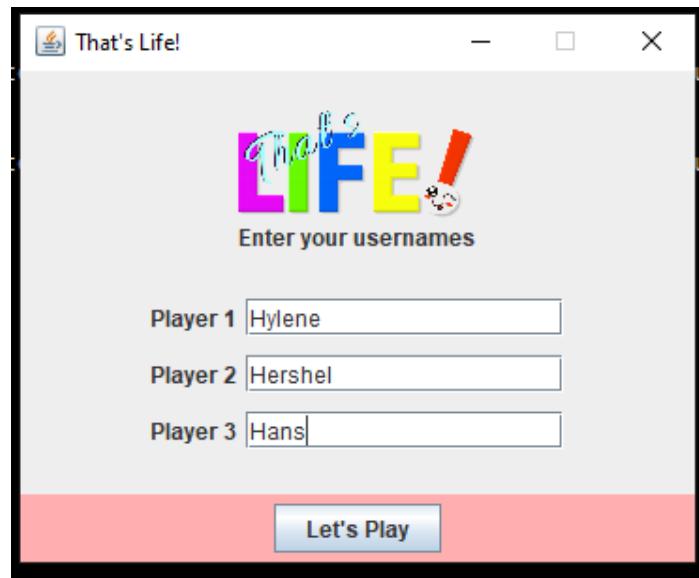
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



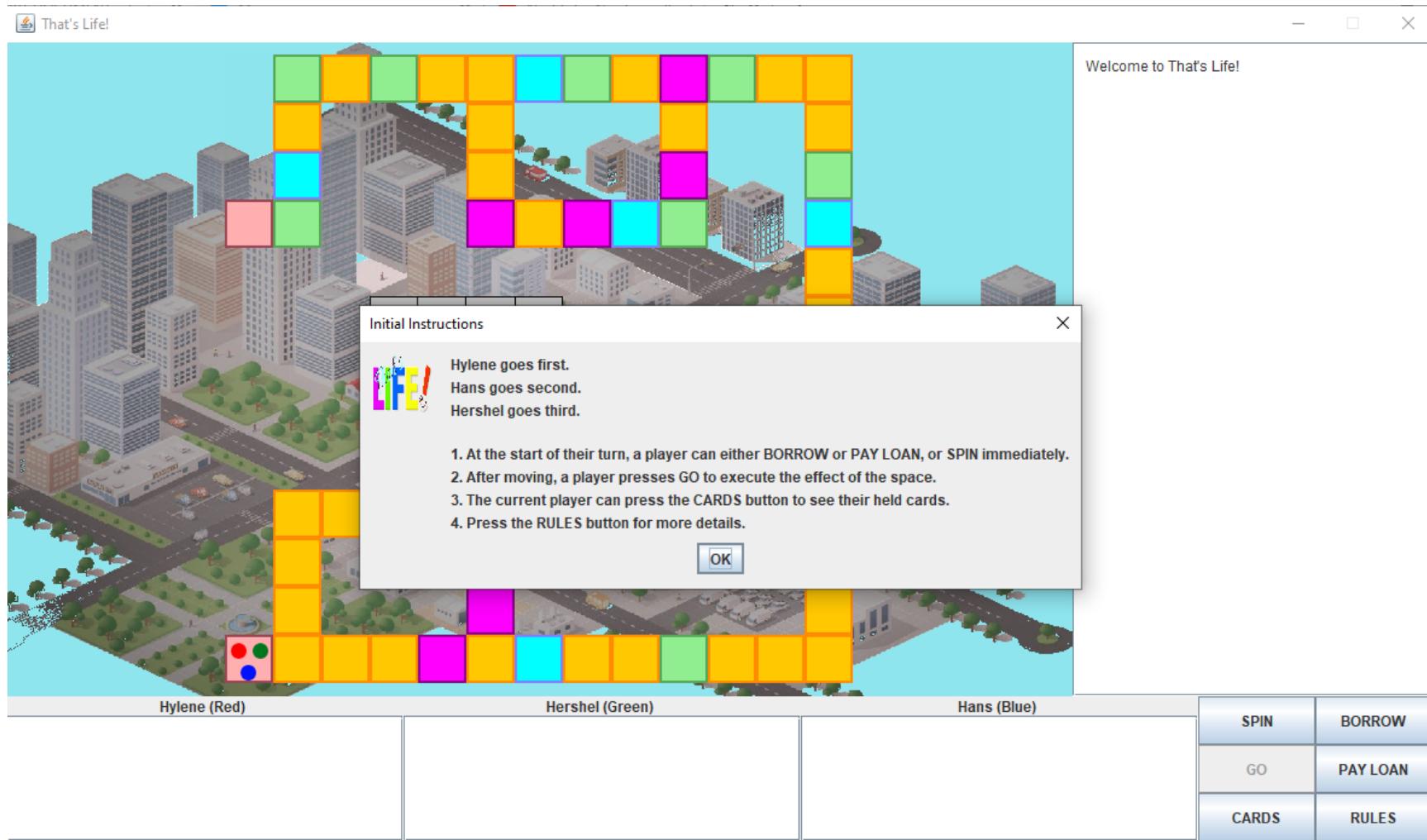
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



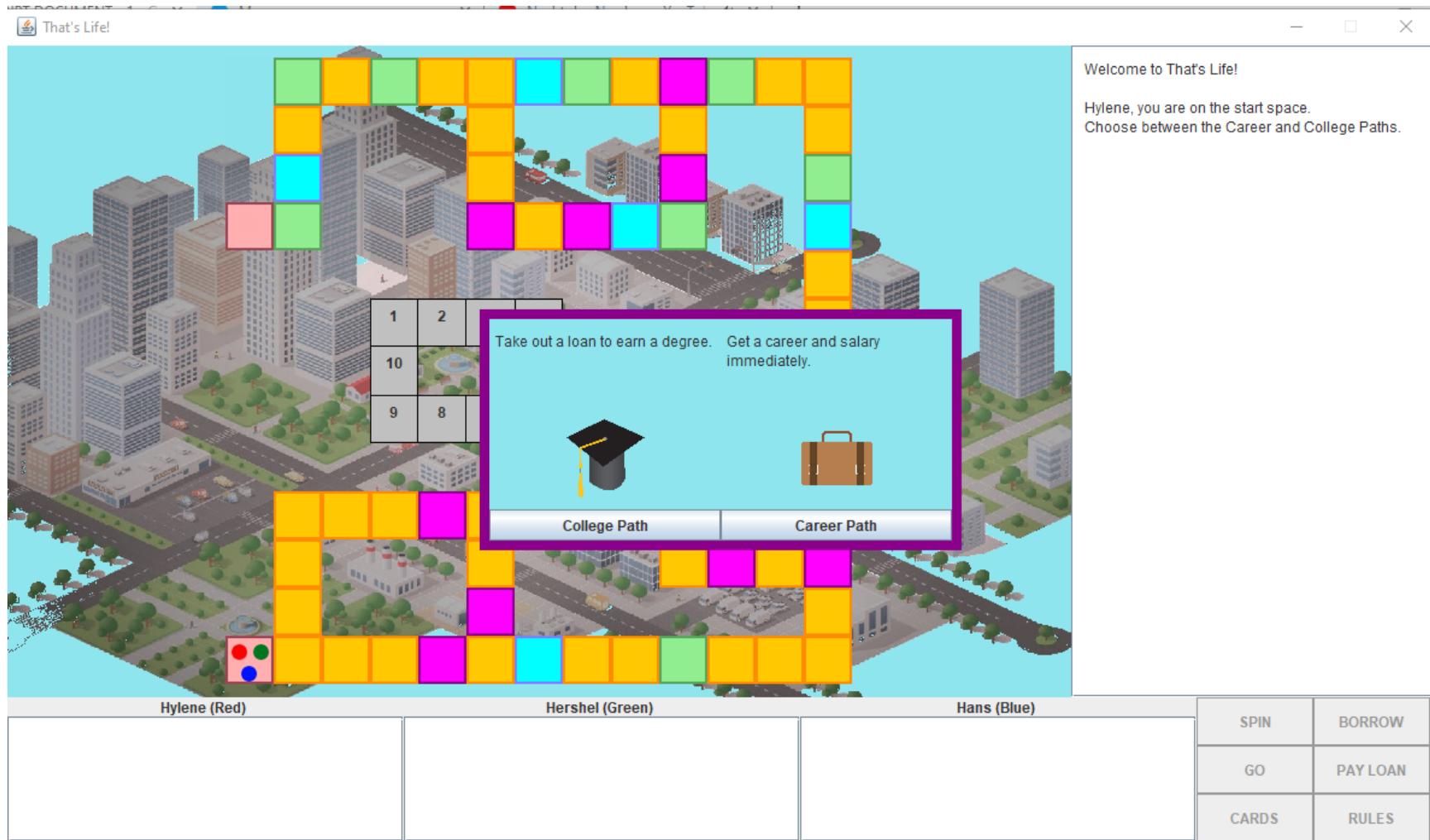
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



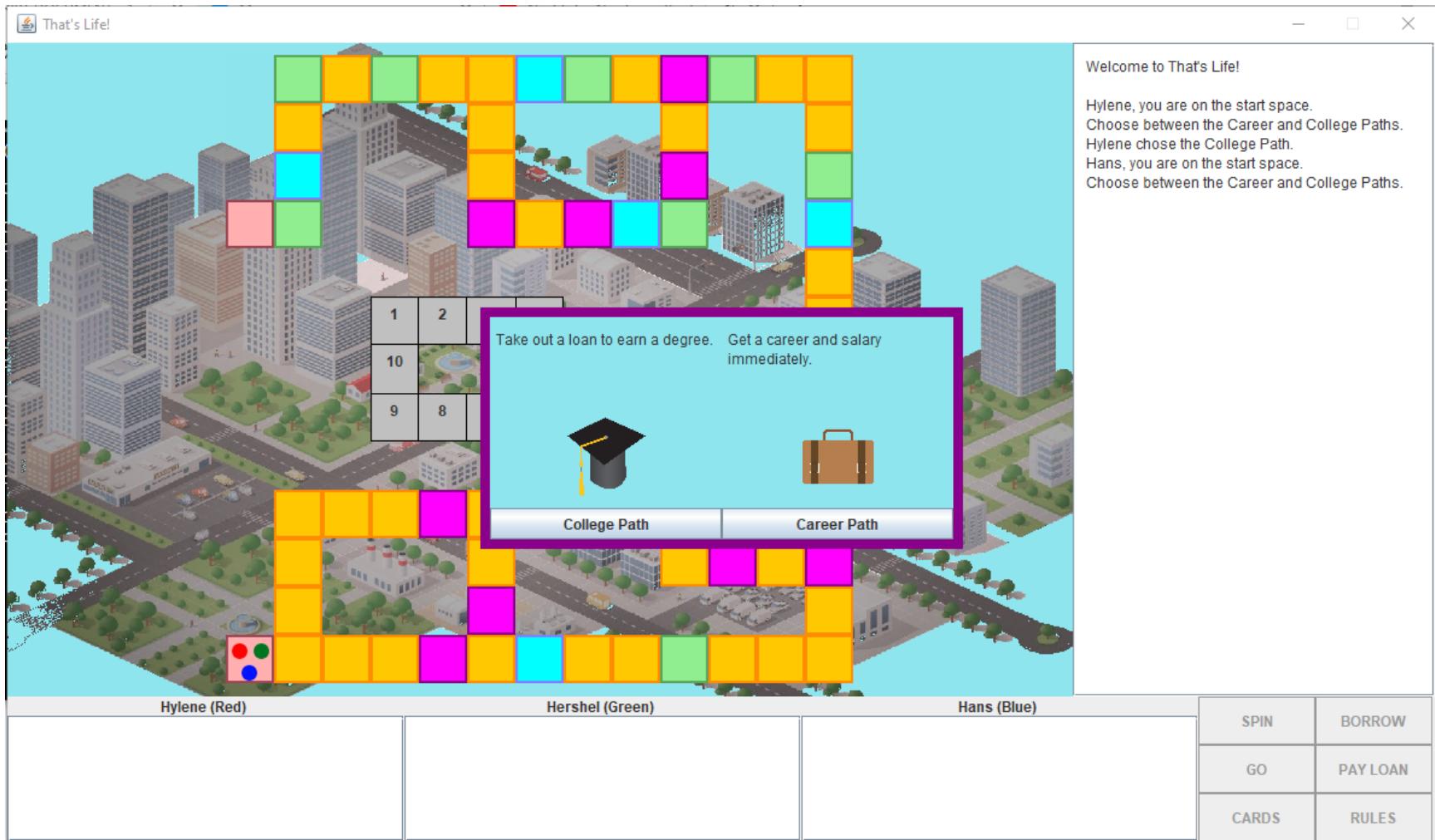
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



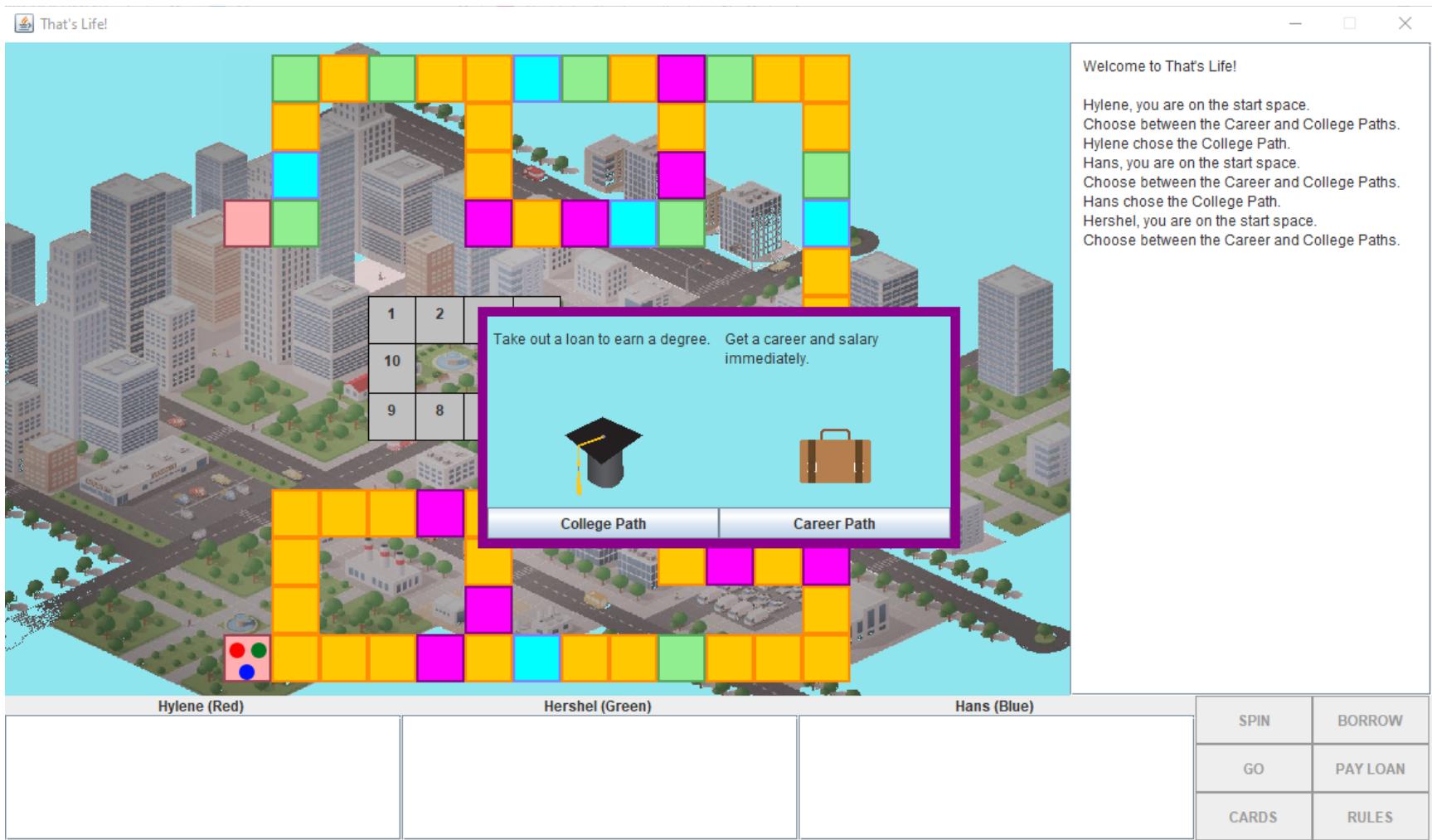
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



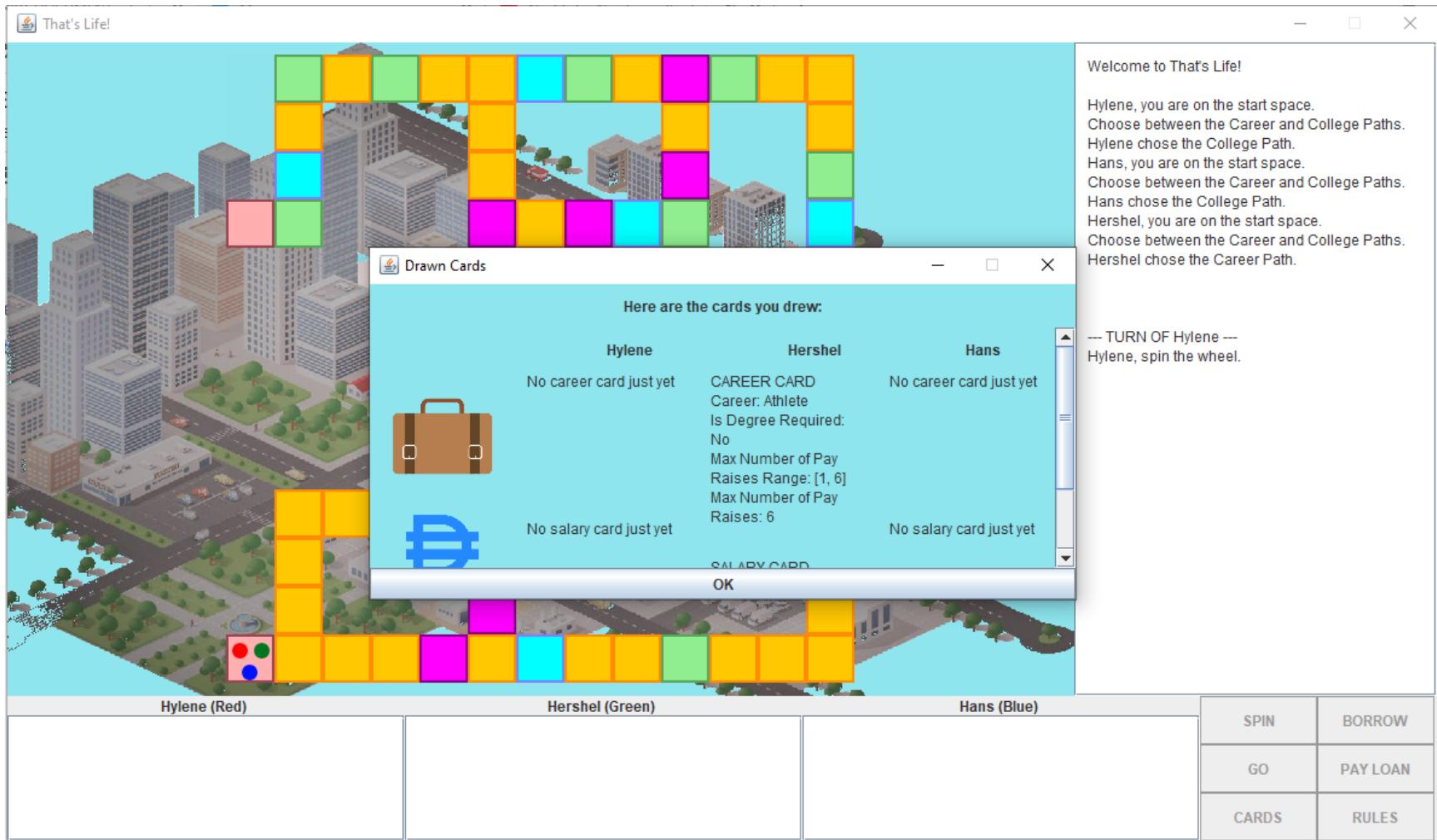
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS

Borrowing loans (using the BORROW button on the main screen)



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS

That's Life!

Hylene (Red)

- Cash: 200000
- Num. of Loans: 2
- Career: Lawyer
- Current Salary: 90000
- Current Tax Due: 9000

Hershel (Green)

- Cash: 200000
- Num. of Loans: 0
- Career: Athlete
- Current Salary: 40000
- Current Tax Due: 4000

Hans (Blue)

- Cash: 200000
- Num. of Loans: 2
- Is Married: No
- Num. of Children: 0
- Has Degree: No

Paths.
Hans chose the College Path.
Hershel, you are on the start space.
Choose between the Career and College Paths.
Hershel chose the Career Path.

-- TURN OF Hylene --
Hylene, spin the wheel.
Hylene spun the number 8.
Hylene, you are on a Graduation space.
You now have a college degree.
You are entitled to ANOTHER TURN right after this.

-- TURN OF Hylene --
Hylene, spin the wheel.
Hylene spun the number 8.
Hylene, you are on a College Career Choice space.
Choose a career and salary card from the available options.
You are entitled to ANOTHER TURN right after this.

-- TURN OF Hylene --
Hylene, spin the wheel.

SPIN	BORROW
GO	PAY LOAN
CARDS	RULES

The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS

Paying loans (using the PAY LOAN button on the main screen)



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



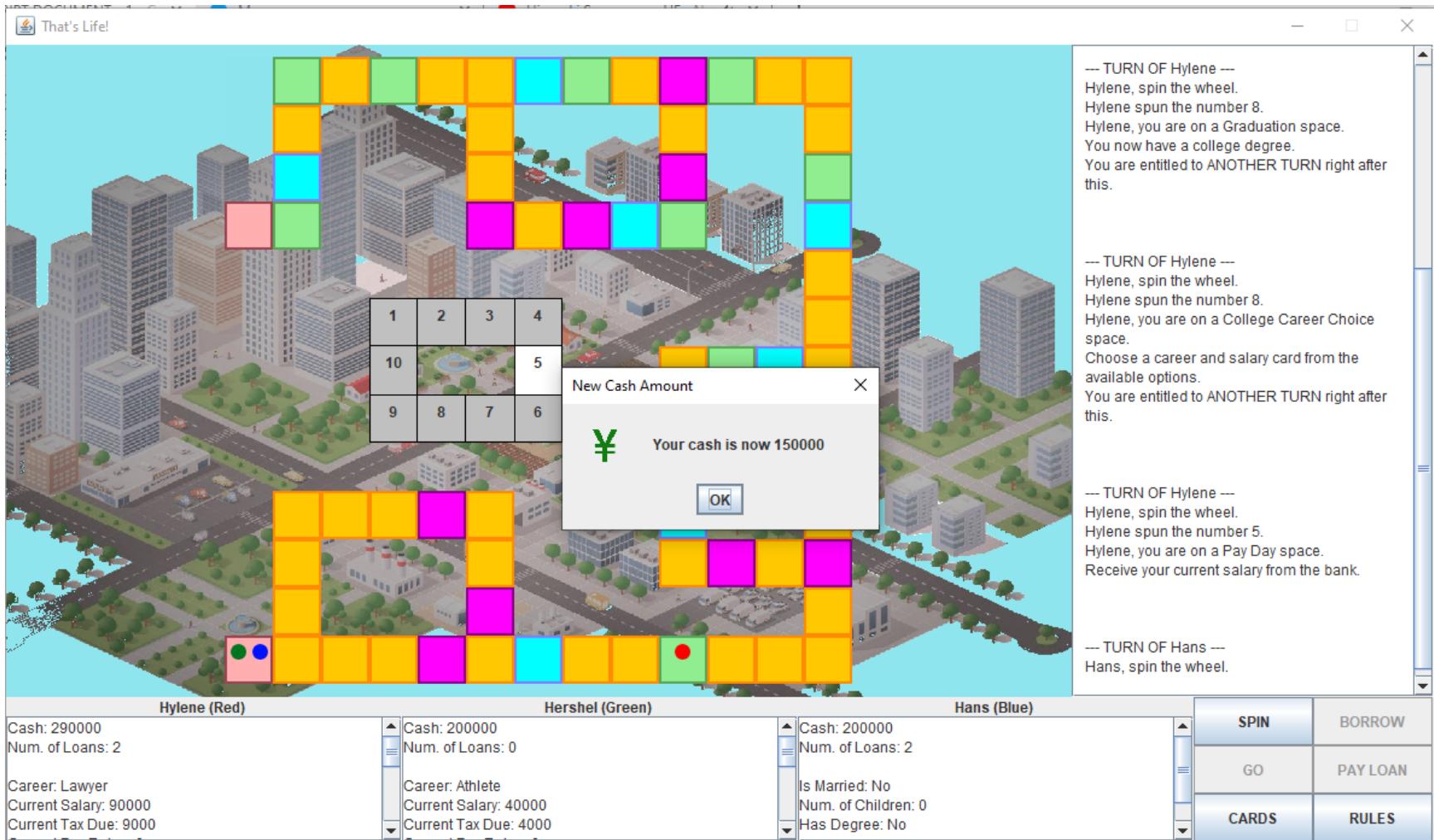
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS

That's Life!

Hershel, you are on an orange space.
Draw an action card.
You received charity benefits for setting up a school. Collect \$60000 from the Bank.

-- TURN OF Hylene --
Hylene, spin the wheel.
Hylene spun the number 10.
Hylene, you are on a Which Path space.
Choose between the two paths.
You are entitled to ANOTHER TURN right after this.
Hylene chose the Normal Path.

-- TURN OF Hylene --
Hylene, spin the wheel.
Hylene spun the number 8.
Hylene, you are on a Pay Day space.
Receive your current salary from the bank.

-- TURN OF Hans --
Hans, spin the wheel.
Hans spun the number 8.
Hans, you are on a Which Path space.
Choose between the two paths.
You are entitled to ANOTHER TURN right after this.

SPIN	BORROW
GO	PAY LOAN
CARDS	RULES

The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS

That's Life!

Hans chose the Change Career Path.

-- TURN OF Hans --
Hans, spin the wheel.
Hans spun the number 3.
Hans, you are on a Job Search space.
Choose a career and salary card from the available options.
You are entitled to ANOTHER TURN right after this.

-- TURN OF Hans --
Hans, spin the wheel.
Hans spun the number 4.
Hans, you are on a Pay Day space.
Receive your current salary from the bank.

-- TURN OF Hershel --
Hershel, spin the wheel.
Hershel spun the number 9.
Hershel, you are on a Get Married space.
Spin the wheel and collect \$5000 from each player for an odd number spun and \$10000 from each player for an even number.
You are entitled to ANOTHER TURN right after this.

SPIN	BORROW
GO	PAY LOAN
CARDS	RULES

Hylene (Red)
Cash: 380000
Num. of Loans: 2

Career: Lawyer
Current Salary: 90000
Current Tax Due: 9000

Hershel (Green)
Cash: 260000
Num. of Loans: 0

Career: Athlete
Current Salary: 40000
Current Tax Due: 4000

Hans (Blue)
Cash: 219000
Num. of Loans: 0

Career: Doctor
Current Salary: 70000
Current Tax Due: 7000

The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



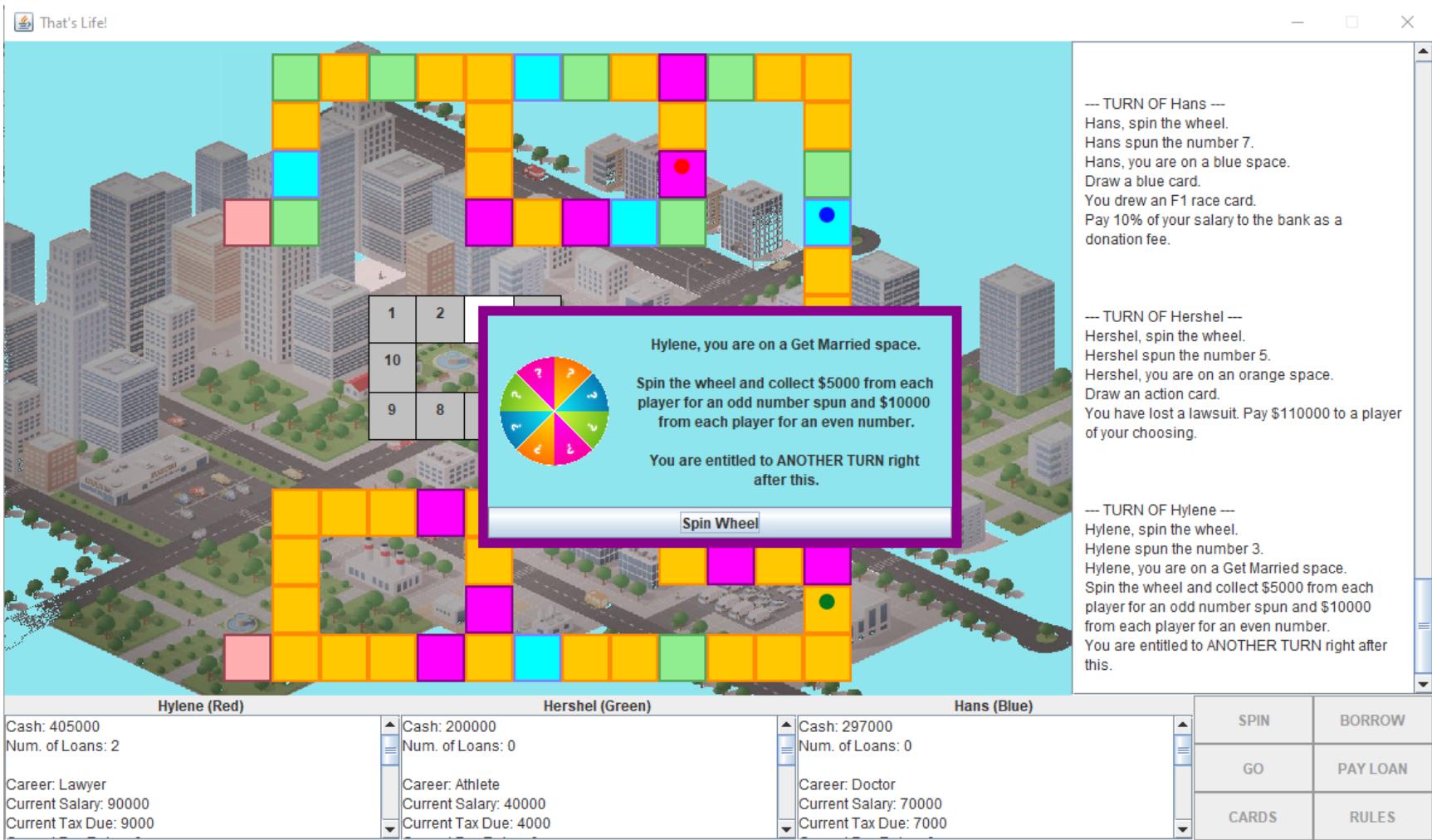
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



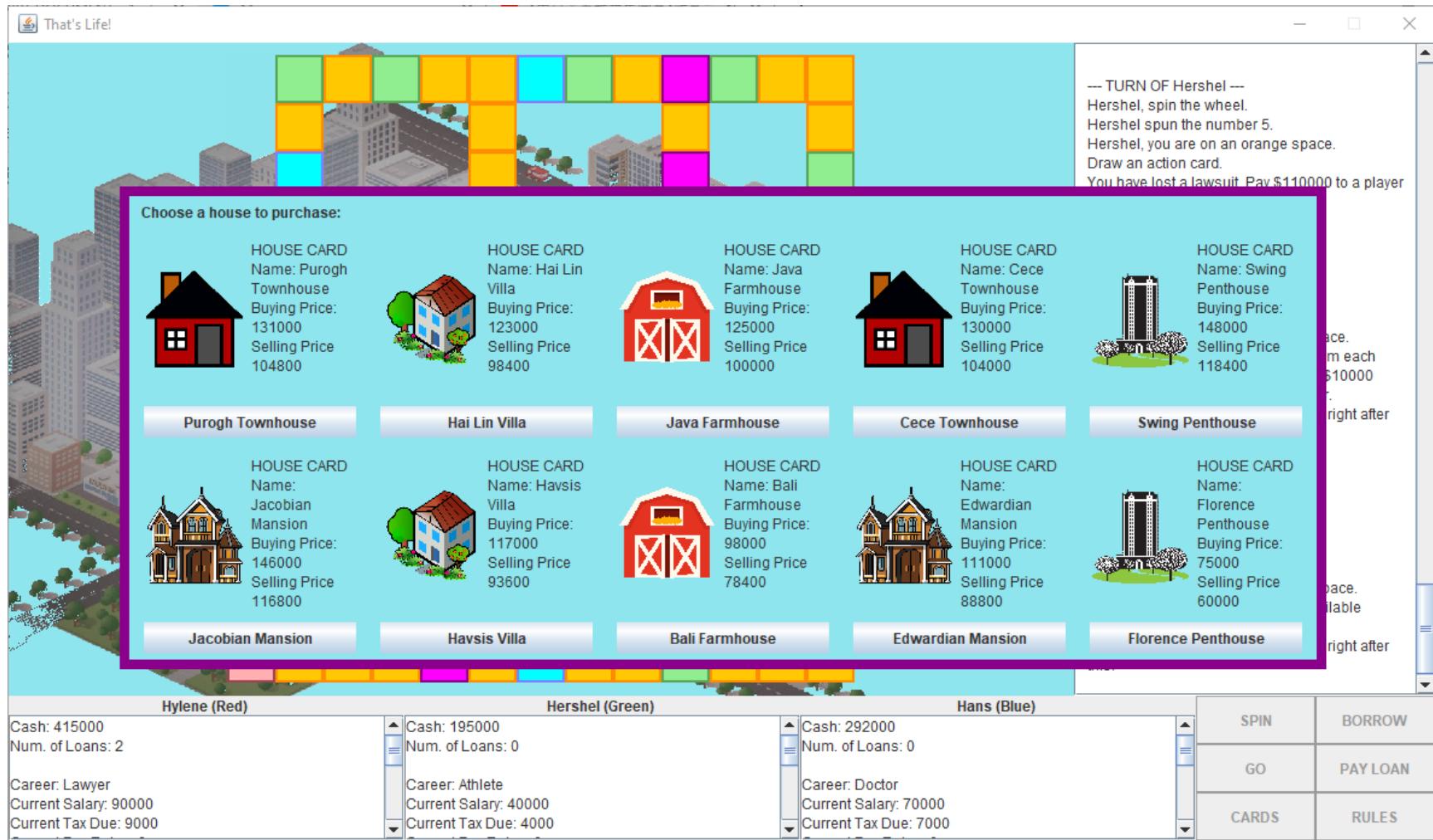
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



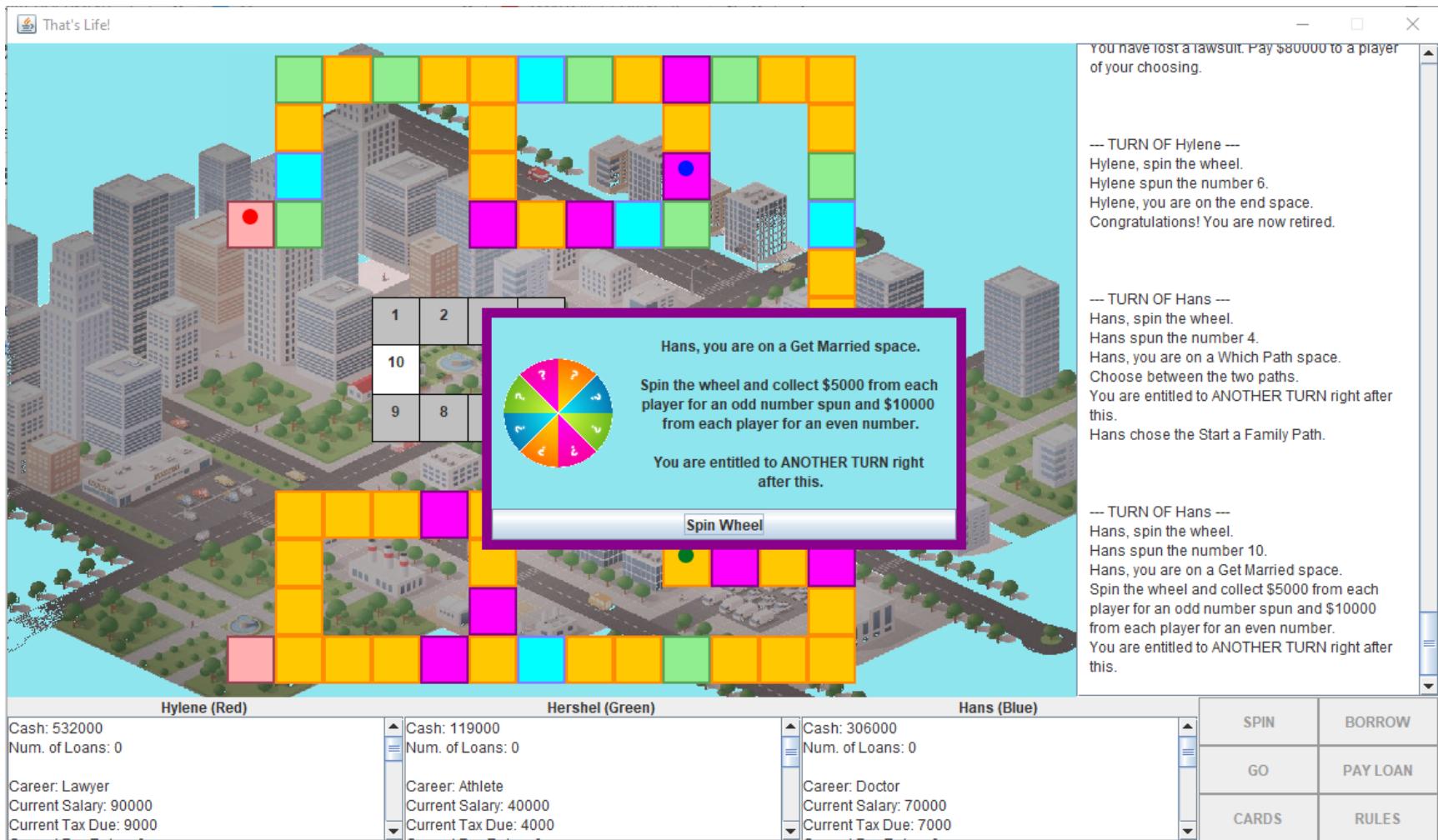
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



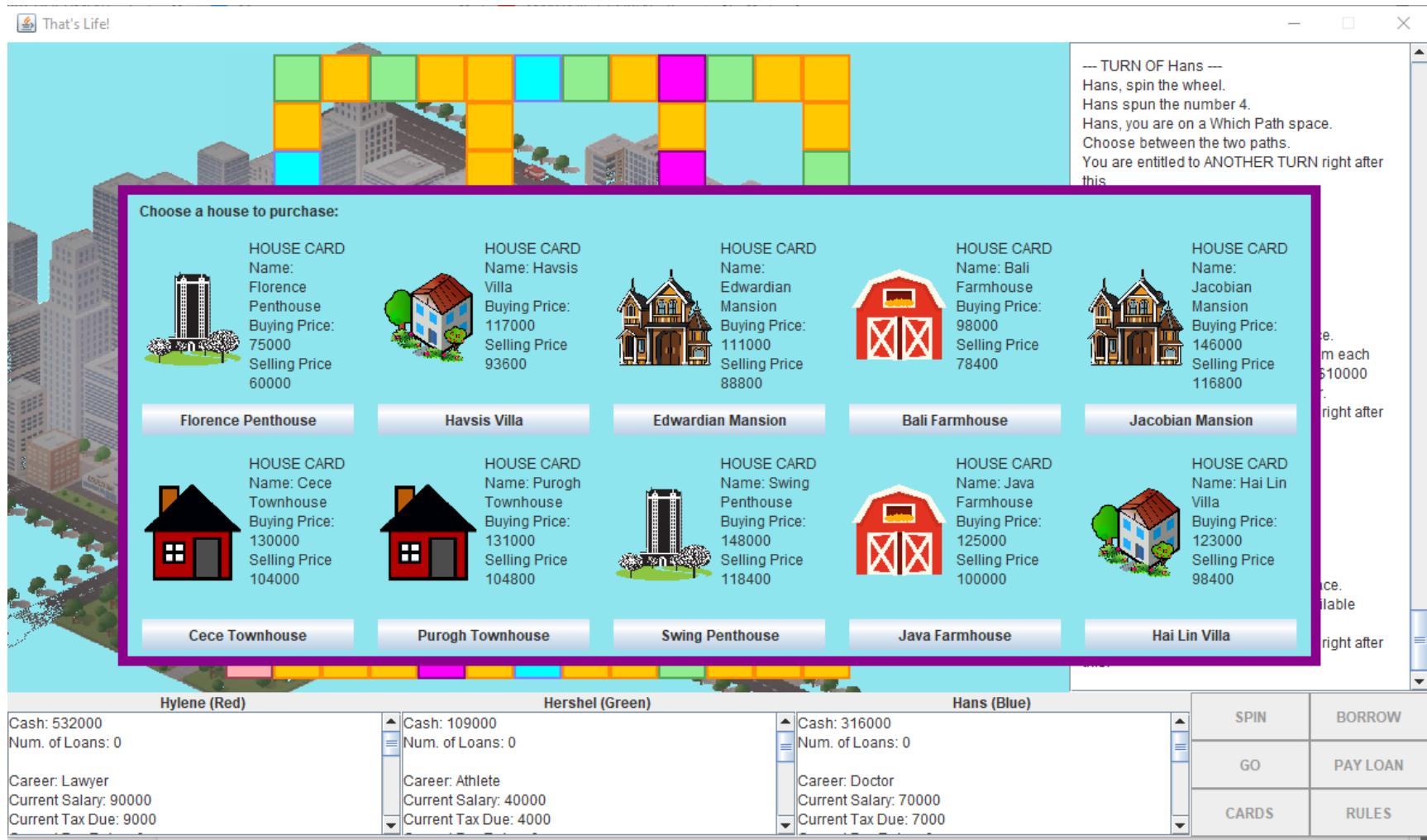
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS

As the player who purchased a house earlier has already retired, their house is returned to the house card deck and made available for the remaining players

The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS

Retired players are no longer affected by the succeeding events of the game

The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

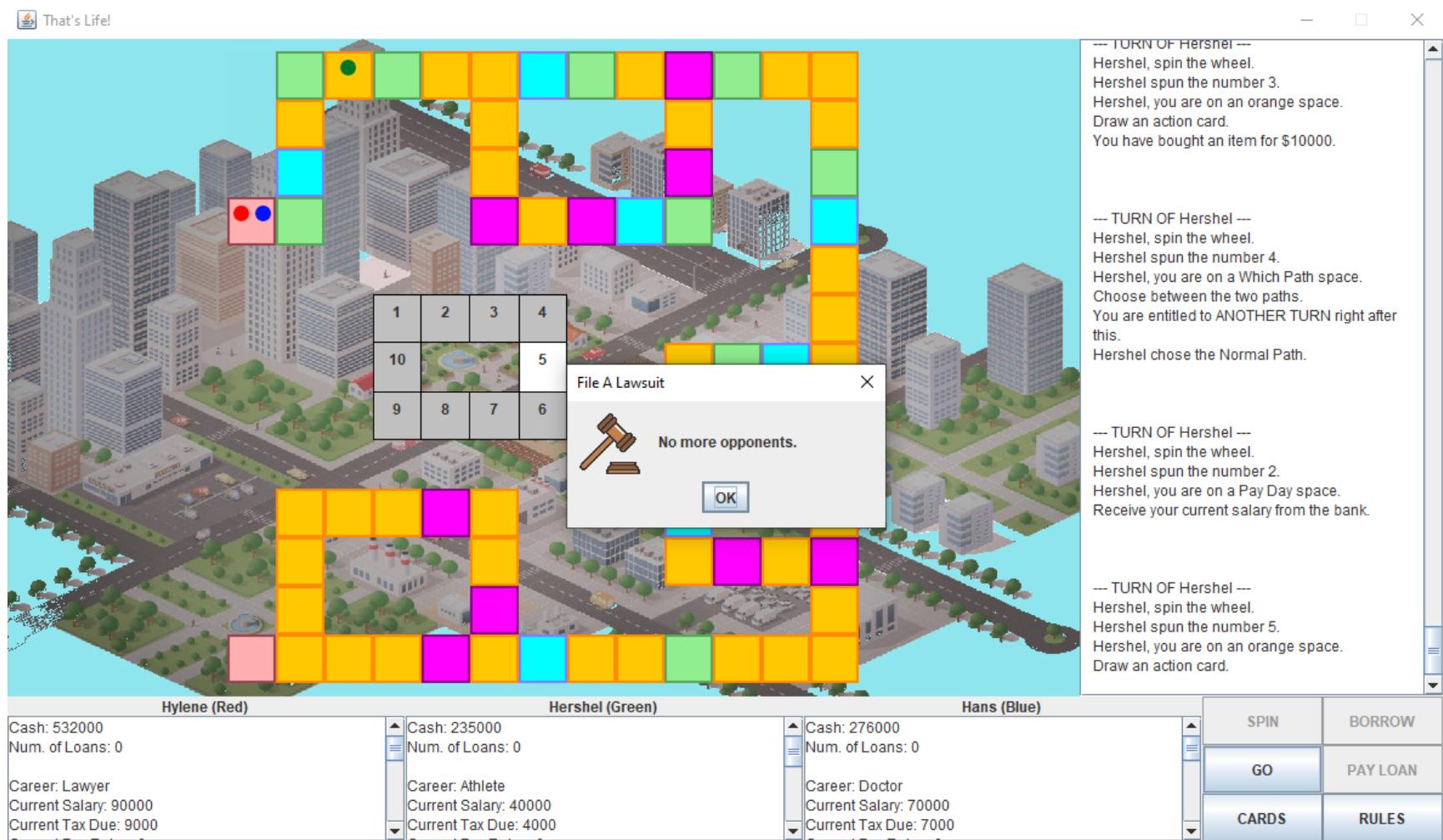
PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS

As all other players have retired, cards such as Lawsuit and File A Lawsuit can no longer take effect.



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



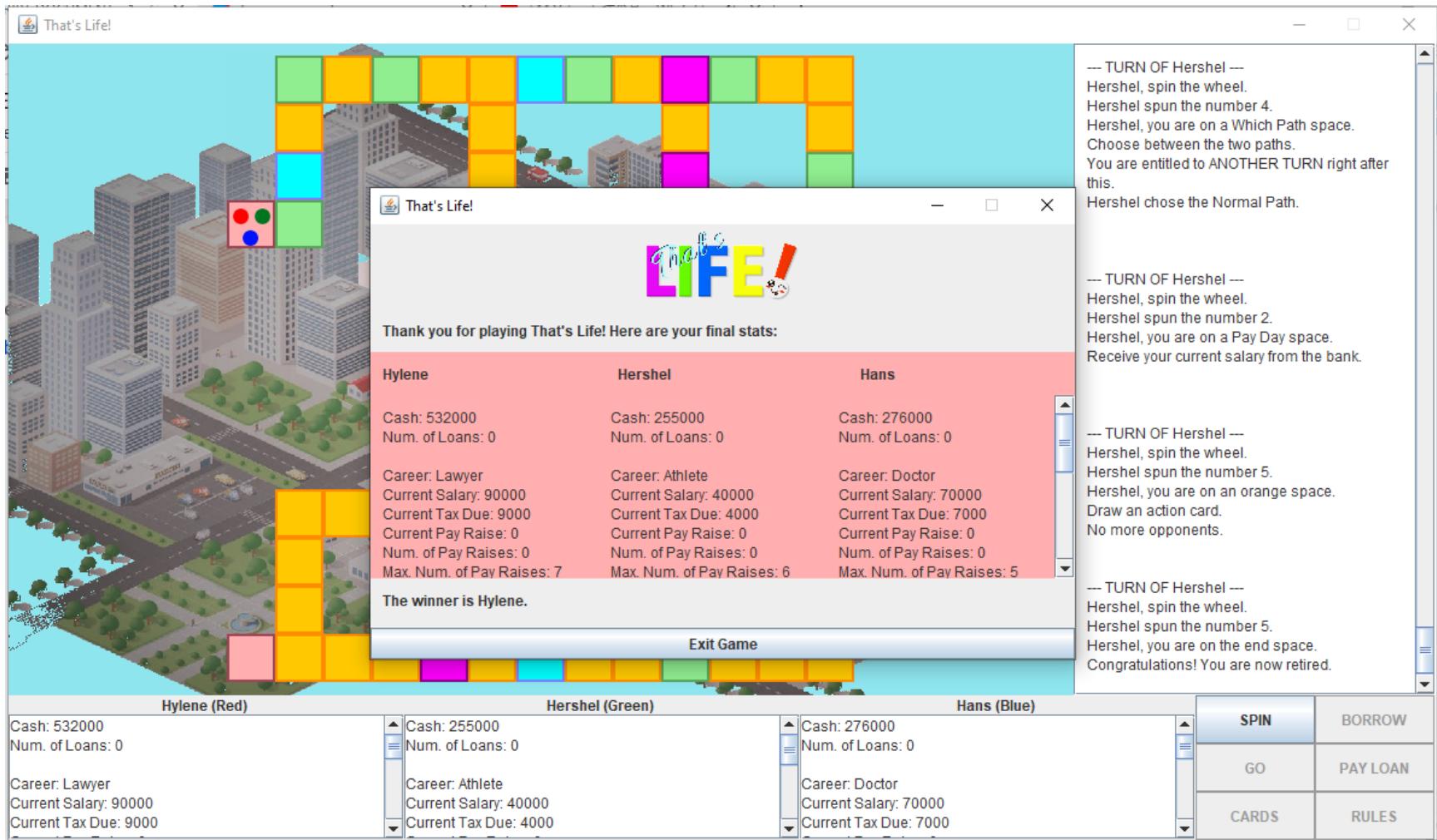
The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS



The tested components involved in this game action performed as expected

PASSED TEST CASE ANALYSIS

B. Card Classes

1. BonusPayday

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The effect of BonusPayday is executed on p1.	Stats before BonusPayday execution: p1: Cash = 200000 p1: Salary = 10000	Stats after BonusPayday execution: p1: Cash = 210000	Stats after BonusPayday execution: p1: Cash = 210000	P
	2	The effect of BonusPayday is executed on p2.	Stats before BonusPayday execution: p2: Cash = 200000 p2: Salary = 90000	Stats after BonusPayday execution: p2: Cash = 290000	Stats after BonusPayday execution: p2: Cash = 290000	P
	3	The effect of BonusPayday is executed on p3.	Stats before BonusPayday execution: p3: Cash = 200000 p3: Salary = 90000	Stats after BonusPayday execution: p3: Cash = 290000	Stats after BonusPayday execution: p3: Cash = 290000	P
getInfo	1	The getInfo() method of c1 is called.	N/A	You have received a bonus equal to your current salary! Collect a paycheck from the Bank.	You have received a bonus equal to your current salary! Collect a paycheck from the Bank.	P
toString	1	The BonusPayday object is printed (using the toString() method).	N/A	ACTION CARD Name: Bonus Pay Day Amount Collected from Bank: Player salary	ACTION CARD Name: Bonus Pay Day Amount Collected from Bank: Player salary	P

2. SellAnItem

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F

execute	1	The effect of SellAnItem is executed on p1 (item price is randomly generated).	Stats before SellAnItem execution: p1: Cash = 200000 c1: Item Price = 9000	Stats after SellAnItem execution: p1: Cash = 209000	Stats after SellAnItem execution: p1: Cash = 209000	P
	2	The effect of SellAnItem is executed on p1 (item price is randomly generated).	Stats before SellAnItem execution: p1: Cash = 209000 c2: Item Price = 4000	Stats after SellAnItem execution: p1: Cash = 213000	Stats after SellAnItem execution: p1: Cash = 213000	P
	3	The effect of SellAnItem is executed on p1 (item price is randomly generated).	Stats before SellAnItem execution: p1: Cash = 213000 c3: Item Price = 8000	Stats after SellAnItem execution: p1: Cash = 221000	Stats after SellAnItem execution: p1: Cash = 221000	P
getInfo	1	The getInfo() method of c1 is called.	N/A	You have completed a sale. Collect 9000 from the bank.	You have completed a sale. Collect 9000 from the bank.	P
	2	The getInfo() method of c2 is called.	N/A	You have completed a sale. Collect 4000 from the bank.	You have completed a sale. Collect 4000 from the bank.	P
	3	The getInfo() method of c3 is called.	N/A	You have completed a sale. Collect 8000 from the bank.	You have completed a sale. Collect 8000 from the bank.	P
toString	1	The SellAnItem c1 object is printed (using the toString() method).	N/A	ACTION CARD Name: Sell an Item Amount Collected from Bank: 9000	ACTION CARD Name: Sell an Item Amount Collected from Bank: 9000	P
	2	The SellAnItem c2 object is printed (using the toString() method).	N/A	ACTION CARD Name: Sell an Item Amount Collected from Bank: 4000	ACTION CARD Name: Sell an Item Amount Collected from Bank: 4000	P
	3	The SellAnItem c3 object is printed	N/A	ACTION CARD Name: Sell an Item	ACTION CARD Name: Sell an Item	P

		(using the <code>toString()</code> method).		Amount Collected from Bank: 8000	Amount Collected from Bank: 8000	
--	--	---	--	----------------------------------	----------------------------------	--

3. SetupSchool

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The effect of <code>SetupSchool</code> is executed on <code>p1</code> (charity benefit is randomly generated).	Stats before <code>SetupSchool</code> execution: <code>p1: Cash = 200000</code> <code>c1: Charity Benefit = 50000</code>	Stats after <code>SetupSchool</code> execution: <code>p1: Cash = 250000</code>	Stats after <code>SetupSchool</code> execution: <code>p1: Cash = 250000</code>	P
	2	The effect of <code>SetupSchool</code> is executed on <code>p1</code> (charity benefit is randomly generated).	Stats before <code>SetupSchool</code> execution: <code>p1: Cash = 250000</code> <code>c2: Charity Benefit = 80000</code>	Stats after <code>SetupSchool</code> execution: <code>p1: Cash = 330000</code>	Stats after <code>SetupSchool</code> execution: <code>p1: Cash = 330000</code>	P
	3	The effect of <code>SetupSchool</code> is executed on <code>p1</code> (charity benefit is randomly generated).	Stats before <code>SetupSchool</code> execution: <code>p1: Cash = 330000</code> <code>c3: Charity Benefit = 90000</code>	Stats after <code>SetupSchool</code> execution: <code>p1: Cash = 420000</code>	Stats after <code>SetupSchool</code> execution: <code>p1: Cash = 420000</code>	P
getInfo	1	The <code>getInfo()</code> method of <code>c1</code> is called.	N/A	You received charity benefits for setting up a school. Collect 50000 from the bank.	You received charity benefits for setting up a school. Collect 50000 from the bank.	P
	2	The <code>getInfo()</code> method of <code>c2</code> is called.	N/A	You received charity benefits for setting up a school. Collect 80000 from the bank.	You received charity benefits for setting up a school. Collect 80000 from the bank.	P

	3	The getInfo() method of c3 is called.	N/A	You received charity benefits for setting up a school. Collect 90000 from the bank.	You received charity benefits for setting up a school. Collect 90000 from the bank.	P
toString	1	The SetupSchool object c1 is printed (using the toString() method).	N/A	ACTION CARD Name: Setup School Amount Collected from Bank: 50000	ACTION CARD Name: Setup School Amount Collected from Bank: 50000	P
	2	The SetupSchool object c2 is printed (using the toString() method).	N/A	ACTION CARD Name: Setup School Amount Collected from Bank: 80000	ACTION CARD Name: Setup School Amount Collected from Bank: 80000	P
	3	The SetupSchool object c3 is printed (using the toString() method).	N/A	ACTION CARD Name: Setup School Amount Collected from Bank: 90000	ACTION CARD Name: Setup School Amount Collected from Bank: 90000	P

4. TaxRefund

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The effect of TaxRefund is executed on p1.	Stats before TaxRefund execution: p1: Cash = 200000 p1: Tax Due = 6000	Stats after TaxRefund execution: p1: Cash = 206000	Stats after TaxRefund execution: p1: Cash = 206000	P
	2	The effect of TaxRefund is executed on p2.	Stats before TaxRefund execution: p2: Cash = 200000 p2: Tax Due = 4000	Stats after TaxRefund execution: p2: Cash = 204000	Stats after TaxRefund execution: p2: Cash = 204000	P
	3	The effect of TaxRefund is executed on p3.	Stats before TaxRefund execution: p3: Cash = 200000	Stats after TaxRefund execution: p3: Cash = 202000	Stats after TaxRefund execution: p3: Cash = 202000	P

			p3: Tax Due = 2000			
getInfo	1	The getInfo() method of c1 is called.	N/A	You received a tax refund! Collect your tax due from the bank.	You received a tax refund! Collect your tax due from the bank.	P
toString	1	The TaxRefund object is printed (using the toString() method).	N/A	ACTION CARD Name: Tax Refund Amount Collected from Bank: Player tax	ACTION CARD Name: Tax Refund Amount Collected from Bank: Player tax	P

5. WriteABook

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The effect of WriteABook is executed on p1 (book payment is randomly generated).	Stats before WriteABook execution: p1: Cash = 200000 c1: Book Payment = 8000	Stats after WriteABook execution: p1: Cash = 208000	Stats after WriteABook execution: p1: Cash = 208000	P
	2	The effect of WriteABook is executed on p1 (book payment is randomly generated).	Stats before WriteABook execution: p1: Cash = 208000 c2: Book Payment = 19000	Stats after WriteABook execution: p1: Cash = 227000	Stats after WriteABook execution: p1: Cash = 227000	P
	3	The effect of WriteABook is executed on p1 (book	Stats before WriteABook execution: p1: Cash = 227000 c3: Book Payment = 8000	Stats after WriteABook execution: p1: Cash = 235000	Stats after WriteABook execution: p1: Cash = 235000	P

		payment is randomly generated).				
getInfo	1	The getInfo() method of c1 is called.	N/A	Your book turned out to be a bestseller! Collect 8000 from the bank.	Your book turned out to be a bestseller! Collect 8000 from the bank.	P
	2	The getInfo() method of c2 is called.	N/A	Your book turned out to be a bestseller! Collect 19000 from the bank.	Your book turned out to be a bestseller! Collect 19000 from the bank.	P
	3	The getInfo() method of c3 is called.	N/A	Your book turned out to be a bestseller! Collect 8000 from the bank.	Your book turned out to be a bestseller! Collect 8000 from the bank.	P
toString	1	The WriteABook object c1 is printed (using the toString() method).	N/A	ACTION CARD Name: Write a Book Amount Collected from Bank: 8000	ACTION CARD Name: Write a Book Amount Collected from Bank: 8000	P
	2	The WriteABook object c2 is printed (using the toString() method).	N/A	ACTION CARD Name: Write a Book Amount Collected from Bank: 19000	ACTION CARD Name: Write a Book Amount Collected from Bank: 19000	P
	3	The WriteABook object c3 is printed (using the toString() method).	N/A	ACTION CARD Name: Write a Book Amount Collected from Bank: 8000	ACTION CARD Name: Write a Book Amount Collected from Bank: 8000	P

6. FileALawsuit

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The effect of FileALawsuit is executed on p1	Stats before FileALawsuit execution: p1: Cash = 200000 p2: Cash = 200000	Stats after FileALawsuit execution: p1: Cash = 340000 p2: Cash = 60000	Stats after FileALawsuit execution: p1: Cash = 340000 p2: Cash = 60000	P

		(amount is randomly generated).	p2: Number of Loans = 0 c1: Amount = 140000	p2: Number of Loans = 0	p2: Number of Loans = 0	
	2	The effect of FileALawsuit is executed on p1 (amount is randomly generated).	Stats before FileALawsuit execution: p1: Cash = 340000 p2: Cash = 60000 p2: Number of Loans = 0 c2: Amount = 150000	Stats after FileALawsuit execution: p1: Cash = 490000 p2: Cash = 10000 p2: Number of Loans = 5	Stats after FileALawsuit execution: p1: Cash = 490000 p2: Cash = 10000 p2: Number of Loans = 5	P
	3	The effect of FileALawsuit is executed on p1 (amount is randomly generated).	Stats before FileALawsuit execution: p1: Cash = 490000 p2: Cash = 10000 p2: Number of Loans = 5 c3: Amount = 140000	Stats after FileALawsuit execution: p1: Cash = 630000 p2: Cash = 10000 p2: Number of Loans = 12	Stats after FileALawsuit execution: p1: Cash = 630000 p2: Cash = 10000 p2: Number of Loans = 12	P
getInfo	1	The getInfo() method of c1 is called.	N/A	You have won a lawsuit. Collect 140000 from a player of your choosing.	You have won a lawsuit. Collect 140000 from a player of your choosing.	P
	2	The getInfo() method of c2 is called.	N/A	You have won a lawsuit. Collect 150000 from a player of your choosing.	You have won a lawsuit. Collect 150000 from a player of your choosing.	P
	3	The getInfo() method of c3 is called.	N/A	You have won a lawsuit. Collect 140000 from a player of your choosing.	You have won a lawsuit. Collect 140000 from a player of your choosing.	P
toString	1	The FileALawsuit object c1 is printed (using the toString() method).	N/A	ACTION CARD Name: File a Lawsuit Amount Collected from Other Player: 140000 (from chosen player)	ACTION CARD Name: File a Lawsuit Amount Collected from Other Player: 140000 (from chosen player)	P
	2	The FileALawsuit object c2 is printed (using the toString() method).	N/A	ACTION CARD Name: File a Lawsuit Amount Collected from Other Player: 150000 (from chosen player)	ACTION CARD Name: File a Lawsuit Amount Collected from Other Player: 150000 (from chosen player)	P

	3	The FileALawsuit object c3 is printed (using the <code>toString()</code> method).	N/A	ACTION CARD Name: File a Lawsuit Amount Collected from Other Player: 140000 (from chosen player)	ACTION CARD Name: File a Lawsuit Amount Collected from Other Player: 140000 (from chosen player)	P
--	---	---	-----	---	---	---

7. ItsYourBirthday

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The effect of ItsYourBirthday is executed on p1 (amount is randomly generated).	Stats before ItsYourBirthday execution: p1: Cash = 200000 p2: Cash = 200000 p2: Number of Loans = 0 p3: Cash = 200000 p3: Number of Loans = 0 c1: Amount = 15000	Stats after ItsYourBirthday execution: p1: Cash = 230000 p2: Cash = 185000 p2: Number of Loans = 0 p3: Cash = 185000 p3: Number of Loans = 0	Stats after ItsYourBirthday execution: p1: Cash = 230000 p2: Cash = 185000 p2: Number of Loans = 0 p3: Cash = 185000 p3: Number of Loans = 0	P
	2	The effect of ItsYourBirthday is executed on p1 (amount is randomly generated).	Stats before ItsYourBirthday execution: p1: Cash = 230000 p2: Cash = 185000 p2: Number of Loans = 0 p3: Cash = 185000 p3: Number of Loans = 0 c2: Amount = 17000	Stats after ItsYourBirthday execution: p1: Cash = 264000 p2: Cash = 168000 p2: Number of Loans = 0 p3: Cash = 168000 p3: Number of Loans = 0	Stats after ItsYourBirthday execution: p1: Cash = 264000 p2: Cash = 168000 p2: Number of Loans = 0 p3: Cash = 168000 p3: Number of Loans = 0	P

	3	The effect of ItsYourBirthday is executed on p1 (amount is randomly generated).	<p>Stats before ItsYourBirthday execution:</p> <p>p1: Cash = 264000 p2: Cash = 168000 p2: Number of Loans = 0 p3: Cash = 168000 p3: Number of Loans = 0 c3: Amount = 12000</p>	<p>Stats after ItsYourBirthday execution:</p> <p>p1: Cash = 288000 p2: Cash = 156000 p2: Number of Loans = 0 p3: Cash = 156000 p3: Number of Loans = 0</p>	<p>Stats after ItsYourBirthday execution:</p> <p>p1: Cash = 288000 p2: Cash = 156000 p2: Number of Loans = 0 p3: Cash = 156000 p3: Number of Loans = 0</p>	P
getInfo	1	The getInfo() method of c1 is called.	N/A	<p>It's your birthday! Collect 15000 from every other player as a gift.</p>	<p>It's your birthday! Collect 15000 from every other player as a gift.</p>	P
	2	The getInfo() method of c2 is called.	N/A	<p>It's your birthday! Collect 17000 from every other player as a gift.</p>	<p>It's your birthday! Collect 17000 from every other player as a gift.</p>	P
	3	The getInfo() method of c3 is called.	N/A	<p>It's your birthday! Collect 12000 from every other player as a gift.</p>	<p>It's your birthday! Collect 12000 from every other player as a gift.</p>	P
toString	1	The ItsYourBirthday object c1 is printed (using the toString() method).	N/A	<p>ACTION CARD Name: It's Your Birthday Amount Collected from Other Player: 15000 (from every player)</p>	<p>ACTION CARD Name: It's Your Birthday Amount Collected from Other Player: 15000 (from every player)</p>	P
	2	The ItsYourBirthday object c2 is printed (using the toString() method).	N/A	<p>ACTION CARD Name: It's Your Birthday Amount Collected from Other Player: 17000 (from every player)</p>	<p>ACTION CARD Name: It's Your Birthday Amount Collected from Other Player: 17000 (from every player)</p>	P
	3	The ItsYourBirthday object c3 is printed (using the toString() method).	N/A	<p>ACTION CARD Name: It's Your Birthday</p>	<p>ACTION CARD Name: It's Your Birthday</p>	P

				Amount Collected from Other Player: 12000 (from every player)	Amount Collected from Other Player: 12000 (from every player)	
--	--	--	--	--	--	--

8. BuyAnItem

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The effect of BuyAnItem is executed on p1 (item price is randomly generated).	Stats before BuyAnItem execution: p1: Cash = 200000 c1: Item Price = 1000	Stats after BuyAnItem execution: p1: Cash = 199000	Stats after BuyAnItem execution: p1: Cash = 199000	P
	2	The effect of BuyAnItem is executed on p1 (item price is randomly generated).	Stats before BuyAnItem execution: p1: Cash = 199000 c2: Item Price = 3000	Stats after BuyAnItem execution: p1: Cash = 196000	Stats after BuyAnItem execution: p1: Cash = 196000	P
	3	The effect of BuyAnItem is executed on p1 (item price is randomly generated).	Stats before BuyAnItem execution: p1: Cash = 196000 c3: Item Price = 10000	Stats before BuyAnItem execution: p1: Cash = 196000 c3: Item Price = 10000	Stats before BuyAnItem execution: p1: Cash = 196000 c3: Item Price = 10000	P
getInfo	1	The getInfo() method of c1 is called.	N/A	You have bought an item for 1000.	You have bought an item for 1000.	P
	2	The getInfo() method of c2 is called.	N/A	You have bought an item for 3000.	You have bought an item for 3000.	P
	3	The getInfo() method of c3 is called.	N/A	You have bought an item for 10000.	You have bought an item for 10000.	P
toString	1	The BuyAnItem object c1 is printed	N/A	ACTION CARD Name: Buy an Item	ACTION CARD Name: Buy an Item	P

		(using the <code>toString()</code> method).		Amount Paid to Bank: 1000	Amount Paid to Bank: 1000	
	2	The BuyAnItem object c2 is printed (using the <code>toString()</code> method).	N/A	ACTION CARD Name: Buy an Item Amount Paid to Bank: 3000	ACTION CARD Name: Buy an Item Amount Paid to Bank: 3000	P
	3	The BuyAnItem object c3 is printed (using the <code>toString()</code> method).	N/A	ACTION CARD Name: Buy an Item Amount Paid to Bank: 10000	ACTION CARD Name: Buy an Item Amount Paid to Bank: 10000	P

9. Hiking

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The effect of Hiking is executed on p1 (hiking expenses is randomly generated).	Stats before Hiking execution: <code>p1: Cash = 200000</code> <code>c1: Hiking Expenses = 10000</code>	Stats after Hiking execution: <code>p1: Cash = 190000</code>	Stats after Hiking execution: <code>p1: Cash = 190000</code>	P
	2	The effect of Hiking is executed on p1 (hiking expenses is randomly generated).	Stats before Hiking execution: <code>p1: Cash = 190000</code> <code>c2: Hiking Expenses = 40000</code>	Stats after Hiking execution: <code>p1: Cash = 150000</code>	Stats after Hiking execution: <code>p1: Cash = 150000</code>	P
	3	The effect of Hiking is executed on p1 (hiking expenses is randomly generated).	Stats before Hiking execution: <code>p1: Cash = 150000</code> <code>c3: Hiking Expenses = 40000</code>	Stats after Hiking execution: <code>p1: Cash = 110000</code>	Stats after Hiking execution: <code>p1: Cash = 110000</code>	P
getInfo	1	The <code>getInfo()</code> method of c1 is called.	N/A	You went hiking and spent 10000 on hiking	You went hiking and spent 10000 on hiking	P

				gear and other expenses.	gear and other expenses.	
	2	The getInfo() method of c2 is called.	N/A	You went hiking and spent 40000 on hiking gear and other expenses.	You went hiking and spent 40000 on hiking gear and other expenses.	P
	3	The getInfo() method of c3 is called.	N/A	You went hiking and spent 40000 on hiking gear and other expenses.	You went hiking and spent 40000 on hiking gear and other expenses.	P
toString	1	The Hiking object c1 is printed (using the toString() method).	N/A	ACTION CARD Name: Hiking Amount Paid to Bank: 10000	ACTION CARD Name: Hiking Amount Paid to Bank: 10000	P
	2	The Hiking object c2 is printed (using the toString() method).	N/A	ACTION CARD Name: Hiking Amount Paid to Bank: 40000	ACTION CARD Name: Hiking Amount Paid to Bank: 40000	P
	3	The Hiking object c3 is printed (using the toString() method).	N/A	ACTION CARD Name: Hiking Amount Paid to Bank: 40000	ACTION CARD Name: Hiking Amount Paid to Bank: 40000	P

10. TrafficViolation

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The effect of TrafficViolation is executed on p1 (fine)	Stats before TrafficViolation execution: p1: Cash = 200000 c1: Fine = 2000	Stats after TrafficViolation execution: p1: Cash = 198000	Stats after TrafficViolation execution: p1: Cash = 198000	P

		is randomly generated).				
	2	The effect of TrafficViolation is executed on p1 (fine is randomly generated).	Stats before TrafficViolation execution: p1: Cash = 198000 c2: Fine = 2000	Stats after TrafficViolation execution: p1: Cash = 196000	Stats after TrafficViolation execution: p1: Cash = 196000	P
	3	The effect of TrafficViolation is executed on p1 (fine is randomly generated).	Stats before TrafficViolation execution: p1: Cash = 196000 c3: Fine = 5000	Stats after TrafficViolation execution: p1: Cash = 191000	Stats after TrafficViolation execution: p1: Cash = 191000	P
getInfo	1	The getInfo() method of c1 is called.	N/A	You committed a traffic violation. Pay a fine of 2000.	You committed a traffic violation. Pay a fine of 2000.	P
	2	The getInfo() method of c2 is called.	N/A	You committed a traffic violation. Pay a fine of 2000.	You committed a traffic violation. Pay a fine of 2000.	P
	3	The getInfo() method of c3 is called.	N/A	You committed a traffic violation. Pay a fine of 5000.	You committed a traffic violation. Pay a fine of 5000.	P
toString	1	The TrafficViolation object c1 is printed (using the toString() method).	N/A	ACTION CARD Name: Traffic Violation Amount Paid to Bank: 2000	ACTION CARD Name: Traffic Violation Amount Paid to Bank: 2000	P
	2	The TrafficViolation object c2 is printed (using the toString() method).	N/A	ACTION CARD Name: Traffic Violation Amount Paid to Bank: 2000	ACTION CARD Name: Traffic Violation Amount Paid to Bank: 2000	P
	3	The TrafficViolation object c3 is printed (using the toString() method).	N/A	ACTION CARD Name: Traffic Violation Amount Paid to Bank: 5000	ACTION CARD Name: Traffic Violation Amount Paid to Bank: 5000	P

11. VisitAPlace

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The effect of VisitAPlace is executed on p1 (expenses is randomly generated).	Stats before VisitAPlace execution: p1: Cash = 200000 c1: Expenses = 40000	Stats after VisitAPlace execution: p1: Cash = 160000	Stats after VisitAPlace execution: p1: Cash = 160000	P
	2	The effect of VisitAPlace is executed on p1 (expenses is randomly generated).	Stats before VisitAPlace execution: p1: Cash = 160000 c2: Expenses = 30000	Stats after VisitAPlace execution: p1: Cash = 130000	Stats after VisitAPlace execution: p1: Cash = 130000	P
	3	The effect of VisitAPlace is executed on p1 (expenses is randomly generated).	Stats before VisitAPlace execution: p1: Cash = 130000 c3: Expenses = 80000	Stats after VisitAPlace execution: p1: Cash = 50000	Stats after VisitAPlace execution: p1: Cash = 50000	P
getInfo	1	The getInfo() method of c1 is called.	N/A	You decided to travel for your vacation. You spent 40000 on expenses.	You decided to travel for your vacation. You spent 40000 on expenses.	P
	2	The getInfo() method of c2 is called.	N/A	You decided to travel for your vacation. You spent 30000 on expenses.	You decided to travel for your vacation. You spent 30000 on expenses.	P
	3	The getInfo() method of c3 is called.	N/A	You decided to travel for your vacation. You spent 80000 on expenses.	You decided to travel for your vacation. You spent 80000 on expenses.	P
toString	1	The VisitAPlace object c1 is printed	N/A	ACTION CARD Name: Visit a Place	ACTION CARD Name: Visit a Place	P

		(using the <code>toString()</code> method).		Amount Paid to Bank: 40000	Amount Paid to Bank: 40000	
	2	The VisitAPlace object c2 is printed (using the <code>toString()</code> method).	N/A	ACTION CARD Name: Visit a Place Amount Paid to Bank: 30000	ACTION CARD Name: Visit a Place Amount Paid to Bank: 30000	P
	3	The VisitAPlace object c3 is printed (using the <code>toString()</code> method).	N/A	ACTION CARD Name: Visit a Place Amount Paid to Bank: 80000	ACTION CARD Name: Visit a Place Amount Paid to Bank: 80000	P

12. WatchAShow

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The effect of WatchAShow is executed on p1 (ticket price is randomly generated).	Stats before WatchAShow execution: <code>p1: Cash = 200000</code> <code>c1: Ticket Price = 5000</code>	Stats after WatchAShow execution: <code>p1: Cash = 195000</code>	Stats after WatchAShow execution: <code>p1: Cash = 195000</code>	P
	2	The effect of WatchAShow is executed on p1 (ticket price is randomly generated).	Stats before WatchAShow execution: <code>p1: Cash = 195000</code> <code>c2: Ticket Price = 7000</code>	Stats after WatchAShow execution: <code>p1: Cash = 188000</code>	Stats after WatchAShow execution: <code>p1: Cash = 188000</code>	P
	3	The effect of WatchAShow is executed on p1 (ticket price is randomly generated).	Stats before WatchAShow execution: <code>p1: Cash = 188000</code> <code>c3: Ticket Price = 6000</code>	Stats after WatchAShow execution: <code>p1: Cash = 182000</code>	Stats after WatchAShow execution: <code>p1: Cash = 182000</code>	P

getInfo	1	The getInfo() method of c1 is called.	N/A	You bought a ticket to a show for 5000.	You bought a ticket to a show for 5000.	P
	2	The getInfo() method of c2 is called.	N/A	You bought a ticket to a show for 7000.	You bought a ticket to a show for 7000.	P
	3	The getInfo() method of c3 is called.	N/A	You bought a ticket to a show for 6000.	You bought a ticket to a show for 6000.	P
toString	1	The WatchAShow object c1 is printed (using the toString() method).	N/A	ACTION CARD Name: Watch a Show Amount Paid to Bank: 5000	ACTION CARD Name: Watch a Show Amount Paid to Bank: 5000	P
	2	The WatchAShow object c2 is printed (using the toString() method).	N/A	ACTION CARD Name: Watch a Show Amount Paid to Bank: 7000	ACTION CARD Name: Watch a Show Amount Paid to Bank: 7000	P
	3	The WatchAShow object c3 is printed (using the toString() method).	N/A	ACTION CARD Name: Watch a Show Amount Paid to Bank: 6000	ACTION CARD Name: Watch a Show Amount Paid to Bank: 6000	P

13. WinACompetition

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The effect of WinACompetition is executed on p1 (competition tax is randomly generated).	Stats before WinACompetition execution: p1: Cash = 200000 c1: Competition Tax = 2000	Stats after WinACompetition execution: p1: Cash = 198000	Stats after WinACompetition execution: p1: Cash = 198000	P
	2	The effect of WinACompetition is executed on p1	Stats before WinACompetition execution: p1: Cash = 198000	Stats after WinACompetition execution:	Stats after WinACompetition execution:	P

		(competition tax is randomly generated).	c2: Competition Tax = 4000	p1: Cash = 194000	p1: Cash = 194000	
	3	The effect of WinACompetition is executed on p1 (competition tax is randomly generated).	Stats before WinACompetition execution: p1: Cash = 194000 c3: Competition Tax = 5000	Stats after WinACompetition execution: p1: Cash = 189000	Stats after WinACompetition execution: p1: Cash = 189000	P
getInfo	1	The getInfo() method of c1 is called.	N/A	You have won a competition. Pay 2000 as competition tax.	You have won a competition. Pay 2000 as competition tax.	P
	2	The getInfo() method of c2 is called.	N/A	You have won a competition. Pay 4000 as competition tax.	You have won a competition. Pay 4000 as competition tax.	P
	3	The getInfo() method of c3 is called.	N/A	You have won a competition. Pay 5000 as competition tax.	You have won a competition. Pay 5000 as competition tax.	P
toString	1	The WinACompetition object c1 is printed (using the toString() method).	N/A	ACTION CARD Name: Win a Competition Amount Paid to Bank: 2000	ACTION CARD Name: Win a Competition Amount Paid to Bank: 2000	P
	2	The WinACompetition object c2 is printed (using the toString() method).	N/A	ACTION CARD Name: Win a Competition Amount Paid to Bank: 4000	ACTION CARD Name: Win a Competition Amount Paid to Bank: 4000	P
	3	The WinACompetition object c3 is printed (using the toString() method).	N/A	ACTION CARD Name: Win a Competition Amount Paid to Bank: 5000	ACTION CARD Name: Win a Competition Amount Paid to Bank: 5000	P

14. ChristmasBonus

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The effect of ChristmasBonus is executed on p1 (amount is randomly generated).	Stats before ChristmasBonus execution: p1: Cash = 200000 p1: Number of Loans = 0 p2: Cash = 200000 p3: Cash = 200000 c1: Amount = 16000	Stats after ItsYourBirthday execution: p1: Cash = 168000 p1: Number of Loans = 0 p2: Cash = 216000 p3: Cash = 216000	Stats after ItsYourBirthday execution: p1: Cash = 168000 p1: Number of Loans = 0 p2: Cash = 216000 p3: Cash = 216000	P
	2	The effect of ChristmasBonus is executed on p1 (amount is randomly generated).	Stats before ChristmasBonus execution: p1: Cash = 168000 p1: Number of Loans = 0 p2: Cash = 216000 p3: Cash = 216000 c2: Amount = 10000	Stats after ItsYourBirthday execution: p1: Cash = 148000 p1: Number of Loans = 0 p2: Cash = 226000 p3: Cash = 226000	Stats after ItsYourBirthday execution: p1: Cash = 148000 p1: Number of Loans = 0 p2: Cash = 226000 p3: Cash = 226000	P
	3	The effect of ChristmasBonus is executed on p1 (amount is randomly generated).	Stats before ChristmasBonus execution: p1: Cash = 148000 p1: Number of Loans = 0 p2: Cash = 226000 p3: Cash = 226000 c3: Amount = 8000	Stats after ItsYourBirthday execution: p1: Cash = 132000 p1: Number of Loans = 0 p2: Cash = 234000 p3: Cash = 234000	Stats after ItsYourBirthday execution: p1: Cash = 132000 p1: Number of Loans = 0 p2: Cash = 234000 p3: Cash = 234000	P
getInfo	1	The getInfo() method of c1 is called.	N/A	Happy holidays! Pay 16000 to every other player as a Christmas gift.	Happy holidays! Pay 16000 to every other player as a Christmas gift.	P
	2	The getInfo() method of c2 is called.	N/A	Happy holidays! Pay 10000 to every other player as a Christmas gift.	Happy holidays! Pay 10000 to every other player as a Christmas gift.	P

	3	The getInfo() method of c3 is called.	N/A	Happy holidays! Pay 8000 to every other player as a Christmas gift.	Happy holidays! Pay 8000 to every other player as a Christmas gift.	P
toString	1	The ChristmasBonus object c1 is printed (using the toString() method).	N/A	ACTION CARD Name: Christmas Bonus Amount Paid to Other Player: 16000 (to every player)	ACTION CARD Name: Christmas Bonus Amount Paid to Other Player: 16000 (to every player)	P
	2	The ChristmasBonus object c2 is printed (using the toString() method).	N/A	ACTION CARD Name: Christmas Bonus Amount Paid to Other Player: 10000 (to every player)	ACTION CARD Name: Christmas Bonus Amount Paid to Other Player: 10000 (to every player)	P
	3	The ChristmasBonus object c3 is printed (using the toString() method).	N/A	ACTION CARD Name: Christmas Bonus Amount Paid to Other Player: 8000 (to every player)	ACTION CARD Name: Christmas Bonus Amount Paid to Other Player: 8000 (to every player)	P

15. Lawsuit

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The effect of Lawsuit is executed on p1 (amount is randomly generated).	Stats before Lawsuit execution: p1: Cash = 200000 p1: Number of Loans = 0 p2: Cash = 200000 c1: Amount = 120000	Stats after Lawsuit execution: p1: Cash = 80000 p1: Number of Loans = 0 p2: Cash = 320000	Stats after Lawsuit execution: p1: Cash = 80000 p1: Number of Loans = 0 p2: Cash = 320000	P
	2	The effect of Lawsuit is executed on p1	Stats before Lawsuit execution:	Stats after Lawsuit execution:	Stats after Lawsuit execution:	P

		(amount is randomly generated).	<p1: cash="80000<br/">p1: Number of Loans = 0 p2: Cash = 320000 c2: Amount = 150000</p1:>	p1: Cash = 10000 p1: Number of Loans = 4 p2: Cash = 470000	p1: Cash = 10000 p1: Number of Loans = 4 p2: Cash = 470000	
	3	The effect of Lawsuit is executed on p1 (amount is randomly generated).	Stats before Lawsuit execution: <p1: cash="10000<br/">p1: Number of Loans = 4 p2: Cash = 470000 c3: Amount = 90000</p1:>	Stats after Lawsuit execution: <p1: cash="0<br/">p1: Number of Loans = 8 p2: Cash = 560000</p1:>	Stats after Lawsuit execution: <p1: cash="0<br/">p1: Number of Loans = 8 p2: Cash = 560000</p1:>	P
getInfo	1	The getInfo() method of c1 is called.	N/A	You have lost a lawsuit. Pay 120000 to a player of your choosing.	You have lost a lawsuit. Pay 120000 to a player of your choosing.	P
	2	The getInfo() method of c2 is called.	N/A	You have lost a lawsuit. Pay 150000 to a player of your choosing.	You have lost a lawsuit. Pay 150000 to a player of your choosing.	P
	3	The getInfo() method of c3 is called.	N/A	You have lost a lawsuit. Pay 90000 to a player of your choosing.	You have lost a lawsuit. Pay 90000 to a player of your choosing.	P
toString	1	The Lawsuit object c1 is printed (using the toString() method).	N/A	ACTION CARD Name: Lawsuit Amount Paid to Other Player: 120000 (to chosen player)	ACTION CARD Name: Lawsuit Amount Paid to Other Player: 120000 (to chosen player)	P
	2	The Lawsuit object c2 is printed (using the toString() method).	N/A	ACTION CARD Name: Lawsuit Amount Paid to Other Player: 150000 (to chosen player)	ACTION CARD Name: Lawsuit Amount Paid to Other Player: 150000 (to chosen player)	P
	3	The Lawsuit object c3 is printed (using the toString() method).	N/A	ACTION CARD Name: Lawsuit Amount Paid to Other Player: 90000 (to chosen player)	ACTION CARD Name: Lawsuit Amount Paid to Other Player: 90000 (to chosen player)	P

16. Accountant

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
getInfo	1	The getInfo() method of career1 is called (max number of pay raises is randomly generated).	N/A	CAREER CARD Career: Accountant Is Degree Required: Yes Max Number of Pay Raises: 5	CAREER CARD Career: Accountant Is Degree Required: Yes Max Number of Pay Raises: 5	P
	2	The getInfo() method of career2 is called (max number of pay raises is randomly generated).	N/A	CAREER CARD Career: Accountant Is Degree Required: Yes Max Number of Pay Raises: 4	CAREER CARD Career: Accountant Is Degree Required: Yes Max Number of Pay Raises: 4	P
	3	The getInfo() method of career3 is called (max number of pay raises is randomly generated).	N/A	CAREER CARD Career: Accountant Is Degree Required: Yes Max Number of Pay Raises: 7	CAREER CARD Career: Accountant Is Degree Required: Yes Max Number of Pay Raises: 7	P
toString	1	The Accountant object career1 is printed (using the toString() method).	N/A	CAREER CARD Is Degree Required: true Max Pay Raise Range: [4, 7] Max Pay Raise: 5 Career: Accountant	CAREER CARD Is Degree Required: true Max Pay Raise Range: [4, 7] Max Pay Raise: 5 Career: Accountant	P
	2	The Accountant object career2 is printed (using the toString() method).	N/A	CAREER CARD Is Degree Required: true Max Pay Raise Range: [4, 7] Max Pay Raise: 4	CAREER CARD Is Degree Required: true Max Pay Raise Range: [4, 7] Max Pay Raise: 4	P

				Career: Accountant	Career: Accountant	
	3	The Accountant object career3 is printed (using the <code>toString()</code> method).	N/A	CAREER CARD Is Degree Required: true Max Pay Raise Range: [4, 7] Max Pay Raise: 7 Career: Accountant	CAREER CARD Is Degree Required: true Max Pay Raise Range: [4, 7] Max Pay Raise: 7 Career: Accountant	P

17. Athlete

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
getInfo	1	The getInfo() method of career1 is called (max number of pay raises is randomly generated).	N/A	CAREER CARD Career: Athlete Is Degree Required: No Max Number of Pay Raises: 5	CAREER CARD Career: Athlete Is Degree Required: No Max Number of Pay Raises: 5	P
	2	The getInfo() method of career2 is called (max number of pay raises is randomly generated).	N/A	CAREER CARD Career: Athlete Is Degree Required: No Max Number of Pay Raises: 2	CAREER CARD Career: Athlete Is Degree Required: No Max Number of Pay Raises: 2	P
	3	The getInfo() method of career3 is called (max number of pay raises is randomly generated).	N/A	CAREER CARD Career: Athlete Is Degree Required: No Max Number of Pay Raises: 1	CAREER CARD Career: Athlete Is Degree Required: No Max Number of Pay Raises: 1	P
toString	1	The Athlete object career1 is printed (using the <code>toString()</code> method).	N/A	CAREER CARD Is Degree Required: false Max Pay Raise Range: [1, 6]	CAREER CARD Is Degree Required: false Max Pay Raise Range: [1, 6]	P

				Max Pay Raise: 5 Career: Athlete	Max Pay Raise: 5 Career: Athlete	
	2	The Athlete object career2 is printed (using the <code>toString()</code> method).	N/A	CAREER CARD Is Degree Required: false Max Pay Raise Range: [1, 6] Max Pay Raise: 2 Career: Athlete	CAREER CARD Is Degree Required: false Max Pay Raise Range: [1, 6] Max Pay Raise: 2 Career: Athlete	P
	3	The Athlete object career3 is printed (using the <code>toString()</code> method).	N/A	CAREER CARD Is Degree Required: false Max Pay Raise Range: [1, 6] Max Pay Raise: 1 Career: Athlete	CAREER CARD Is Degree Required: false Max Pay Raise Range: [1, 6] Max Pay Raise: 1 Career: Athlete	P

18. ComputerConsultant

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
getInfo	1	The <code>getInfo()</code> method of career1 is called (max number of pay raises is randomly generated).	N/A	CAREER CARD Career: Computer Consultant Is Degree Required: Yes Max Number of Pay Raises: 3	CAREER CARD Career: Computer Consultant Is Degree Required: Yes Max Number of Pay Raises: 3	P
	2	The <code>getInfo()</code> method of career2 is called (max number of pay raises is randomly generated).	N/A	CAREER CARD Career: Computer Consultant Is Degree Required: Yes Max Number of Pay Raises: 6	CAREER CARD Career: Computer Consultant Is Degree Required: Yes Max Number of Pay Raises: 6	P

	3	The getInfo() method of career3 is called (max number of pay raises is randomly generated).	N/A	CAREER CARD Career: Computer Consultant Is Degree Required: Yes Max Number of Pay Raises: 6	CAREER CARD Career: Computer Consultant Is Degree Required: Yes Max Number of Pay Raises: 6	P
toString	1	The ComputerConsultant object career1 is printed (using the toString() method).	N/A	CAREER CARD Is Degree Required: true Max Pay Raise Range: [3, 7] Max Pay Raise: 3 Career: Computer Consultant	CAREER CARD Is Degree Required: true Max Pay Raise Range: [3, 7] Max Pay Raise: 3 Career: Computer Consultant	P
	2	The ComputerConsultant object career2 is printed (using the toString() method).	N/A	CAREER CARD Is Degree Required: true Max Pay Raise Range: [3, 7] Max Pay Raise: 6 Career: Computer Consultant	CAREER CARD Is Degree Required: true Max Pay Raise Range: [3, 7] Max Pay Raise: 6 Career: Computer Consultant	P
	3	The ComputerConsultant object career3 is printed (using the toString() method).	N/A	CAREER CARD Is Degree Required: true Max Pay Raise Range: [3, 7] Max Pay Raise: 6 Career: Computer Consultant	CAREER CARD Is Degree Required: true Max Pay Raise Range: [3, 7] Max Pay Raise: 6 Career: Computer Consultant	P

19. Doctor

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F

getInfo	1	The getInfo() method of career1 is called (max number of pay raises is randomly generated).	N/A	CAREER CARD Career: Doctor Is Degree Required: Yes Max Number of Pay Raises: 5	CAREER CARD Career: Doctor Is Degree Required: Yes Max Number of Pay Raises: 5	P
	2	The getInfo() method of career2 is called (max number of pay raises is randomly generated).	N/A	CAREER CARD Career: Doctor Is Degree Required: Yes Max Number of Pay Raises: 6	CAREER CARD Career: Doctor Is Degree Required: Yes Max Number of Pay Raises: 6	P
	3	The getInfo() method of career3 is called (max number of pay raises is randomly generated).	N/A	CAREER CARD Career: Doctor Is Degree Required: Yes Max Number of Pay Raises: 7	CAREER CARD Career: Doctor Is Degree Required: Yes Max Number of Pay Raises: 7	P
toString	1	The Doctor object career1 is printed (using the toString() method).	N/A	CAREER CARD Is Degree Required: true Max Pay Raise Range: [5, 8] Max Pay Raise: 5 Career: Doctor	CAREER CARD Is Degree Required: true Max Pay Raise Range: [5, 8] Max Pay Raise: 5 Career: Doctor	P
	2	The Doctor object career2 is printed (using the toString() method).	N/A	CAREER CARD Is Degree Required: true Max Pay Raise Range: [5, 8] Max Pay Raise: 6 Career: Doctor	CAREER CARD Is Degree Required: true Max Pay Raise Range: [5, 8] Max Pay Raise: 6 Career: Doctor	P
	3	The Doctor object career3 is printed (using the toString() method).	N/A	CAREER CARD Is Degree Required: true Max Pay Raise Range: [5, 8] Max Pay Raise: 7 Career: Doctor	CAREER CARD Is Degree Required: true Max Pay Raise Range: [5, 8] Max Pay Raise: 7 Career: Doctor	P

20. Lawyer

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
getInfo	1	The getInfo() method of career1 is called (max number of pay raises is randomly generated).	N/A	CAREER CARD Career: Lawyer Is Degree Required: Yes Max Number of Pay Raises: 5	CAREER CARD Career: Lawyer Is Degree Required: Yes Max Number of Pay Raises: 5	P
	2	The getInfo() method of career2 is called (max number of pay raises is randomly generated).	N/A	CAREER CARD Career: Lawyer Is Degree Required: Yes Max Number of Pay Raises: 8	CAREER CARD Career: Lawyer Is Degree Required: Yes Max Number of Pay Raises: 8	P
	3	The getInfo() method of career3 is called (max number of pay raises is randomly generated).	N/A	CAREER CARD Career: Lawyer Is Degree Required: Yes Max Number of Pay Raises: 7	CAREER CARD Career: Lawyer Is Degree Required: Yes Max Number of Pay Raises: 7	P
toString	1	The Lawyer object career1 is printed (using the toString() method).	N/A	CAREER CARD Is Degree Required: true Max Pay Raise Range: [5, 8] Max Pay Raise: 5 Career: Lawyer	CAREER CARD Is Degree Required: true Max Pay Raise Range: [5, 8] Max Pay Raise: 5 Career: Lawyer	P
	2	The Lawyer object career2 is printed (using the toString() method).	N/A	CAREER CARD Is Degree Required: true Max Pay Raise Range: [5, 8] Max Pay Raise: 8 Career: Lawyer	CAREER CARD Is Degree Required: true Max Pay Raise Range: [5, 8] Max Pay Raise: 8 Career: Lawyer	P
	3	The Lawyer object career3 is printed	N/A	CAREER CARD Is Degree Required: true	CAREER CARD Is Degree Required: true	P

		(using the <code>toString()</code> method).		Max Pay Raise Range: [5, 8] Max Pay Raise: 7 Career: Lawyer	Max Pay Raise Range: [5, 8] Max Pay Raise: 7 Career: Lawyer	
--	--	---	--	---	---	--

21. RacecarDriver

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
getInfo	1	The <code>getInfo()</code> method of career1 is called (max number of pay raises is randomly generated).	N/A	CAREER CARD Career: Racecar Driver Is Degree Required: No Max Number of Pay Raises: 4	CAREER CARD Career: Racecar Driver Is Degree Required: No Max Number of Pay Raises: 4	P
	2	The <code>getInfo()</code> method of career2 is called (max number of pay raises is randomly generated).	N/A	CAREER CARD Career: Racecar Driver Is Degree Required: No Max Number of Pay Raises: 4	CAREER CARD Career: Racecar Driver Is Degree Required: No Max Number of Pay Raises: 4	P
	3	The <code>getInfo()</code> method of career3 is called (max number of pay raises is randomly generated).	N/A	CAREER CARD Career: Racecar Driver Is Degree Required: No Max Number of Pay Raises: 7	CAREER CARD Career: Racecar Driver Is Degree Required: No Max Number of Pay Raises: 7	P
toString	1	The RacecarDriver object career1 is printed (using the <code>toString()</code> method).	N/A	CAREER CARD Is Degree Required: false Max Pay Raise Range: [2, 8] Max Pay Raise: 4 Career: Racecar Driver	CAREER CARD Is Degree Required: false Max Pay Raise Range: [2, 8] Max Pay Raise: 4 Career: Racecar Driver	P

	2	The RacecarDriver object career2 is printed (using the toString() method).	N/A	CAREER CARD Is Degree Required: false Max Pay Raise Range: [2, 8] Max Pay Raise: 4 Career: Racecar Driver	CAREER CARD Is Degree Required: false Max Pay Raise Range: [2, 8] Max Pay Raise: 4 Career: Racecar Driver	P
	3	The RacecarDriver object career3 is printed (using the toString() method).	N/A	CAREER CARD Is Degree Required: false Max Pay Raise Range: [2, 8] Max Pay Raise: 7 Career: Racecar Driver	CAREER CARD Is Degree Required: false Max Pay Raise Range: [2, 8] Max Pay Raise: 7 Career: Racecar Driver	P

22. Server

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
getInfo	1	The getInfo() method of career1 is called (max number of pay raises is randomly generated).	N/A	CAREER CARD Career: Server Is Degree Required: No Max Number of Pay Raises: 1	CAREER CARD Career: Server Is Degree Required: No Max Number of Pay Raises: 1	P
	2	The getInfo() method of career2 is called (max number of pay raises is randomly generated).	N/A	CAREER CARD Career: Server Is Degree Required: No Max Number of Pay Raises: 4	CAREER CARD Career: Server Is Degree Required: No Max Number of Pay Raises: 4	P
	3	The getInfo() method of career3 is called (max number of pay raises is randomly generated).	N/A	CAREER CARD Career: Server Is Degree Required: No Max Number of Pay Raises: 1	CAREER CARD Career: Server Is Degree Required: No Max Number of Pay Raises: 1	P

toString	1	The Server object career1 is printed (using the <code>toString()</code> method).	N/A	CAREER CARD Is Degree Required: false Max Pay Raise Range: [1, 4] Max Pay Raise: 1 Career: Server	CAREER CARD Is Degree Required: false Max Pay Raise Range: [1, 4] Max Pay Raise: 1 Career: Server	P
	2	The Server object career2 is printed (using the <code>toString()</code> method).	N/A	CAREER CARD Is Degree Required: false Max Pay Raise Range: [1, 4] Max Pay Raise: 4 Career: Server	CAREER CARD Is Degree Required: false Max Pay Raise Range: [1, 4] Max Pay Raise: 4 Career: Server	P
	3	The Server object career3 is printed (using the <code>toString()</code> method).	N/A	CAREER CARD Is Degree Required: false Max Pay Raise Range: [1, 4] Max Pay Raise: 1 Career: Server	CAREER CARD Is Degree Required: false Max Pay Raise Range: [1, 4] Max Pay Raise: 1 Career: Server	P

23. SalaryCard

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
getInfo	1	The <code>getInfo()</code> method of salary1 is called (salary is randomly incremented).	N/A	SALARY CARD Salary: 30000 Tax Due: 3000 Pay Raise Increment: 6000	SALARY CARD Salary: 30000 Tax Due: 3000 Pay Raise Increment: 6000	P
	2	The <code>getInfo()</code> method of salary2 is called (salary is randomly incremented).	N/A	SALARY CARD Salary: 30000 Tax Due: 3000	SALARY CARD Salary: 30000 Tax Due: 3000	P

				Pay Raise Increment: 6000	Pay Raise Increment: 6000	
	3	The getInfo() method of salary3 is called (salary is randomly incremented).	N/A	SALARY CARD Salary: 50000 Tax Due: 5000 Pay Raise Increment: 10000	SALARY CARD Salary: 50000 Tax Due: 5000 Pay Raise Increment: 10000	P
toString	1	The SalaryCard object salary1 is printed (using the toString() method).	N/A	SALARY CARD Salary: 30000 Tax Due: 3000 Pay Raise : 6000	SALARY CARD Salary: 30000 Tax Due: 3000 Pay Raise : 6000	P
	2	The SalaryCard object salary2 is printed (using the toString() method).	N/A	SALARY CARD Salary: 30000 Tax Due: 3000 Pay Raise : 6000	SALARY CARD Salary: 30000 Tax Due: 3000 Pay Raise : 6000	P
	3	The SalaryCard object salary3 is printed (using the toString() method).	N/A	SALARY CARD Salary: 50000 Tax Due: 5000 Pay Raise : 10000	SALARY CARD Salary: 50000 Tax Due: 5000 Pay Raise : 10000	P

24. ComputerRepair

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The effect of ComputerRepair is executed on p1, and p2 is the player who has the Computer Consultant career.	Stats before ComputerRepair execution: p1: Cash = 200000 p2: Cash = 200000	The cash of p1 deceased by either 5000 or 10000, and the cash of p2 will increase by the same amount.	Stats after ComputerRepair execution: p1: Cash = 190000 p2: Cash = 210000	P

	2	The effect of ComputerRepair is executed on p1, and p1 is the player who has the Computer Consultant career.	Stats before ComputerRepair execution: p1: Cash = 190000	Stats after ComputerRepair execution: p1: Cash = 205000	Stats after ComputerRepair execution: p1: Cash = 205000	P
	3	The effect of ComputerRepair is executed on p1, and no player has the Computer Consultant career.	Stats before ComputerRepair execution: p1: Cash = 205000	The cash of p1 deceased by either 5000 or 10000.	Stats after ComputerRepair execution: p1: Cash = 200000	P
spinWheel	1	The spinWheel() method of c1 is called.	N/A	A randomly determined number between 1 and 10.	9	P
	2	The spinWheel() method of c1 is called a second time.	N/A	A randomly determined number between 1 and 10.	2	P
	3	The spinWheel() method of c1 is called a third time.	N/A	A randomly determined number between 1 and 10.	8	P
getInfo	1	The getInfo() method of c1 is called.	N/A	You drew a computer repair card.	You drew a computer repair card.	P
getInfoBank	1	The getInfoBank() method of c1 is called.	N/A	You drew a computer repair card. Spin the wheel to determine the repair cost to be paid to the bank.	You drew a computer repair card. Spin the wheel to determine the repair cost to be paid to the bank.	P
getInfoMatch	1	The getInfoMatch() method of c1 is called.	N/A	You drew a computer repair card. Receive 15000 as a computer consultant.	You drew a computer repair card. Receive 15000 as a computer consultant.	P

getInfoOtherPlayer	1	The getInfoOtherPlayer() method of c1 is called.	N/A	You drew a computer repair card. Spin the wheel to determine the repair cost to be paid to the computer consultant.	You drew a computer repair card. Spin the wheel to determine the repair cost to be paid to the computer consultant.	P
toString	1	The ComputerRepair object c1 is printed (using the toString() method).	N/A	BLUE CARD Career Match: Computer Consultant Name: Computer Repair Received if Career Match: 15000 Paid to Other Player (or Bank): 10000 (if odd) 5000 (if even)	BLUE CARD Career Match: Computer Consultant Name: Computer Repair Received if Career Match: 15000 Paid to Other Player (or Bank): 10000 (if odd) 5000 (if even)	P

25. F1Race

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The effect of F1Race is executed on p1, and p2 is the player who has the Racecar Driver career.	Stats before F1Race execution: p1: Cash = 200000 p1: Salary = 80000 p2: Cash = 200000	Stats after F1Race execution: p1: Cash = 192000 p2: Cash = 208000	Stats after F1Race execution: p1: Cash = 192000 p2: Cash = 208000	P
	2	The effect of F1Race is executed on p1, and p1 is the player who has the Racecar Driver career.	Stats before F1Race execution: p1: Cash = 192000	Stats after F1Race execution: p1: Cash = 207000	Stats after F1Race execution: p1: Cash = 207000	P

	3	The effect of F1Race is executed on p1, and no player has the Racecar Driver career.	Stats before F1Race execution: p1: Cash = 207000 p1: Salary = 80000	Stats after F1Race execution: p1: Cash = 199000	Stats after F1Race execution: p1: Cash = 199000	P
getInfo	1	The getInfo() method of c1 is called.	N/A	You drew an F1 race card.	You drew an F1 race card.	P
getInfoBank	1	The getInfoBank() method of c1 is called.	N/A	You drew an F1 race card. Pay 10% of your salary to the bank as a donation fee.	You drew an F1 race card. Pay 10% of your salary to the bank as a donation fee.	P
getInfoMatch	1	The getInfoMatch() method of c1 is called.	N/A	You drew an F1 race card. Receive 15000 as a racecar driver.	You drew an F1 race card. Receive 15000 as a racecar driver.	P
getInfoOtherPlayer	1	The getInfoOtherPlayer() method of c1 is called.	N/A	You drew an F1 race card. Pay 10% of your salary to the racecar driver as a donation fee.	You drew an F1 race card. Pay 10% of your salary to the racecar driver as a donation fee.	P
toString	1	The F1Race object c1 is printed (using the toString() method).	N/A	BLUE CARD Career Match: Racecar Driver Name: F1 Race Received if Career Match: 15000 Paid to Other Player (or Bank): 0.1 of salary	BLUE CARD Career Match: Racecar Driver Name: F1 Race Received if Career Match: 15000 Paid to Other Player (or Bank): 0.1 of salary	P

26. LawsuitBlue

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
getMachineRandNum	1	The getMachineRandNum() method of c1 is called.	N/A	A randomly generated multiple of 10000 between 50000 and 150000 inclusive.	120000	P
	2	The getMachineRandNum() method of c2 is called.	N/A	A randomly generated multiple of 10000 between 50000 and 150000 inclusive.	80000	P
	3	The getMachineRandNum() method of c3 is called.	N/A	A randomly generated multiple of 10000 between 50000 and 150000 inclusive.	130000	P
execute	1	The effect of LawsuitBlue is executed on p1, and p2 is the player who has the Lawyer career.	Stats before LawsuitBlue execution: p1: Cash = 200000 p2: Cash = 200000 Amount = 70000	Stats after LawsuitBlue execution: p1: Cash = 130000 p2: Cash = 270000	Stats after LawsuitBlue execution: p1: Cash = 130000 p2: Cash = 270000	P
	2	The effect of LawsuitBlue is executed on p1, and p1 is the player who has the Lawyer career.	Stats before LawsuitBlue execution: p1: Cash = 130000	Stats after LawsuitBlue execution: p1: Cash = 145000	Stats after LawsuitBlue execution: p1: Cash = 145000	P
	3	The effect of LawsuitBlue is executed on p1, and no player has the Lawyer career.	Stats before LawsuitBlue execution: p1: Cash = 145000 Amount = 70000	Stats after LawsuitBlue execution: p1: Cash = 75000	Stats after LawsuitBlue execution: p1: Cash = 75000	P
getInfo	1	The getInfo() method of c1 is called.	N/A	You drew a Lawsuit (Blue) card.	You drew a Lawsuit (Blue) card.	P

getInfoBank	1	The getInfoBank() method of c1 is called.	N/A	You drew a Lawsuit (Blue) card. Pay 70000 to the bank for legal fees.	You drew a Lawsuit (Blue) card. Pay 70000 to the bank for legal fees.	P
getInfoMatch	1	The getInfoMatch() method of c1 is called.	N/A	You drew a Lawsuit (Blue) card. Receive 15000 as a lawyer.	You drew a Lawsuit (Blue) card. Receive 15000 as a lawyer.	P
getInfoOtherPlayer	1	The getInfoOtherPlayer() method of c1 is called.	N/A	You drew a Lawsuit (Blue) card. Pay 70000 to the lawyer for legal fees.	You drew a Lawsuit (Blue) card. Pay 70000 to the lawyer for legal fees.	P
toString	1	The LawsuitBlue object c1 is printed (using the toString() method).	N/A	BLUE CARD Career Match: Lawyer Name: Lawsuit (Blue Card) Received if Career Match: 15000 Paid to Other Player (or Bank): 70000	BLUE CARD Career Match: Lawyer Name: Lawsuit (Blue Card) Received if Career Match: 15000 Paid to Other Player (or Bank): 70000	P

27. SalaryTaxDue

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The effect of SalaryTaxDue is executed on p1, and p2 is the player who has the Accountant career.	Stats before SalaryTaxDue execution: p1: Cash = 200000 p1: Tax Due = 6000 p2: Cash = 200000	Stats after SalaryTaxDue execution: p1: Cash = 194000 p2: Cash = 206000	Stats after SalaryTaxDue execution: p1: Cash = 194000 p2: Cash = 206000	P

	2	The effect of SalaryTaxDue is executed on p1, and p1 is the player who has the Accountant career.	Stats before SalaryTaxDue execution: p1: Cash = 194000	Stats after SalaryTaxDue execution: p1: Cash = 209000	Stats after SalaryTaxDue execution: p1: Cash = 209000	P
	3	The effect of SalaryTaxDue is executed on p1, and no player has the Accountant career.	Stats before SalaryTaxDue execution: p1: Cash = 209000 p1: Tax Due = 6000	Stats after SalaryTaxDue execution: p1: Cash = 203000	Stats after SalaryTaxDue execution: p1: Cash = 203000	P
getInfo	1	The getInfo() method of c1 is called.	N/A	You drew a salary tax due card.	You drew a salary tax due card.	P
getInfoBank	1	The getInfoBank() method of c1 is called.	N/A	You drew a salary tax due card. Pay your current tax due to the bank.	You drew a salary tax due card. Pay your current tax due to the bank.	P
getInfoMatch	1	The getInfoMatch() method of c1 is called.	N/A	You drew a salary tax due card. Receive 15000 as an accountant.	You drew a salary tax due card. Receive 15000 as an accountant.	P
getInfoOtherPlayer	1	The getInfoOtherPlayer() method of c1 is called.	N/A	You drew a salary tax due card. Pay your current tax due to the accountant.	You drew a salary tax due card. Pay your current tax due to the accountant.	P
toString	1	The SalaryTaxDue object c1 is printed (using the toString() method).	N/A	BLUE CARD Career Match: Accountant Name: Lawsuit (Blue Card) Received if Career Match: 15000 Paid to Other Player (or Bank): Tax Due	BLUE CARD Career Match: Accountant Name: Lawsuit (Blue Card) Received if Career Match: 15000 Paid to Other Player (or Bank): Tax Due	P

28. SkiAccident

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The effect of SkiAccident is executed on p1, and p2 is the player who has the Doctor career.	Stats before SkiAccident execution: p1: Cash = 200000 p2: Cash = 200000	Stats after SkiAccident execution: p1: Cash = 190000 p2: Cash = 210000	Stats after SkiAccident execution: p1: Cash = 190000 p2: Cash = 210000	P
	2	The effect of SkiAccident is executed on p1, and p1 is the player who has the Doctor career.	Stats before SkiAccident execution: p1: Cash = 190000	Stats after SkiAccident execution: p1: Cash = 205000	Stats after SkiAccident execution: p1: Cash = 205000	P
	3	The effect of SkiAccident is executed on p1, and no player has the Doctor career.	Stats before SkiAccident execution: p1: Cash = 205000	Stats after SkiAccident execution: p1: Cash = 195000	Stats after SkiAccident execution: p1: Cash = 195000	P
getInfo	1	The getInfo() method of c1 is called.	N/A	You drew a ski accident card.	You drew a ski accident card.	P
getInfoBank	1	The getInfoBank() method of c1 is called.	N/A	You drew a ski accident card. Pay 10000 to the bank for treatment fees.	You drew a ski accident card. Pay 10000 to the bank for treatment fees.	P
getInfoMatch	1	The getInfoMatch() method of c1 is called.	N/A	You drew a ski accident card.	You drew a ski accident card.	P

				Receive 15000 as a doctor.	Receive 15000 as a doctor.	
getInfoOtherPlayer	1	The getInfoOtherPlayer() method of c1 is called.	N/A	You drew a ski accident card. Pay 10000 to the doctor for treatment fees.	You drew a ski accident card. Pay 10000 to the doctor for treatment fees.	P
toString	1	The SkiAccident object c1 is printed (using the toString() method).	N/A	BLUE CARD Career Match: Doctor Name: Ski Accident Received if Career Match: 15000 Paid to Other Player (or Bank): 10000	BLUE CARD Career Match: Doctor Name: Ski Accident Received if Career Match: 15000 Paid to Other Player (or Bank): 10000	P

29. TipTheServer

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The effect of TipTheServer is executed on p1, and p2 is the player who has the Server career.	Stats before TipTheServer execution: p1: Cash = 200000 p2: Cash = 200000	The cash of p1 will decrease by the random amount (between 1000 and 10000 inclusive) determined using the number wheel and the cash of p2 will increase by the same amount.	Stats after TipTheServer execution: p1: Cash = 193000 p2: Cash = 207000	P
	2	The effect of TipTheServer is executed on p1, and p1 is the player who has the Server career.	Stats before TipTheServer execution: p1: Cash = 193000	Stats after TipTheServer execution: p1: Cash = 208000	Stats after TipTheServer execution: p1: Cash = 208000	P

	3	The effect of TipTheServer is executed on p1, and no player has the Server career.	Stats before TipTheServer execution: p1: Cash = 208000	The cash of p1 will decrease by the random amount (between 1000 and 10000 inclusive) determined using the number wheel.	Stats after TipTheServer execution: p1: Cash = 201000	P
spinWheel	1	The spinWheel() method of c1 is called.	N/A	A randomly determined number between 1 and 10.	4	P
	2	The spinWheel() method of c1 is called a second time.	N/A	A randomly determined number between 1 and 10.	5	P
	3	The spinWheel() method of c1 is called a third time.	N/A	A randomly determined number between 1 and 10.	9	P
getInfo	1	The getInfo() method of c1 is called.	N/A	You drew a tip the server card.	You drew a tip the server card.	P
getInfoBank	1	The getInfoBank() method of c1 is called.	N/A	You drew a tip the server card. Spin the wheel to determine the tip to be paid to the bank.	You drew a tip the server card. Spin the wheel to determine the tip to be paid to the bank.	P
getInfoMatch	1	The getInfoMatch() method of c1 is called.	N/A	You drew a tip the server card. Receive 15000 as a server.	You drew a tip the server card. Receive 15000 as a server.	P
getInfoOtherPlayer	1	The getInfoOtherPlayer() method of c1 is called.	N/A	You drew a tip the server card. Spin the wheel to determine the tip to be paid to the server.	You drew a tip the server card. Spin the wheel to determine the tip to be paid to the server.	P
toString	1	The TipTheServer object c1 is printed	N/A	BLUE CARD Career Match: Server Name: Tip the Server	BLUE CARD Career Match: Server Name: Tip the Server	P

		(using the <code>toString()</code> method).		Received if Career Match: 15000 Paid to Other Player (or Bank): Randomly spun number * 1000	Received if Career Match: 15000 Paid to Other Player (or Bank): Randomly spun number * 1000	
--	--	---	--	--	--	--

30. WorldCup

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The effect of WorldCup is executed on p1, and p2 is the player who has the Athlete career.	Stats before WorldCup execution: p1: Cash = 200000 p2: Cash = 200000	Stats after WorldCup execution: p1: Cash = 190000 p2: Cash = 210000	Stats after WorldCup execution: p1: Cash = 190000 p2: Cash = 210000	P
	2	The effect of WorldCup is executed on p1, and p1 is the player who has the Athlete career.	Stats before WorldCup execution: p1: Cash = 190000	Stats after WorldCup execution: p1: Cash = 205000	Stats after WorldCup execution: p1: Cash = 205000	P
	3	The effect of WorldCup is executed on p1, and no player has the Athlete career.	Stats before WorldCup execution: p1: Cash = 205000	Stats after WorldCup execution: p1: Cash = 195000	Stats after WorldCup execution: p1: Cash = 195000	P
getInfo	1	The <code>getInfo()</code> method of c1 is called.	N/A	You drew a world cup card.	You drew a world cup card.	P
getInfoBank	1	The <code>getInfoBank()</code> method of c1 is called.	N/A	You drew a world cup card.	You drew a world cup card.	P

				Pay 5000 for each player to the bank for training costs.	Pay 5000 for each player to the bank for training costs.	
getInfoMatch	1	The getInfoMatch() method of c1 is called.	N/A	You drew a world cup card. Receive 15000 as an athlete.	You drew a world cup card. Receive 15000 as an athlete.	P
getInfoOtherPlayer	1	The getInfoOtherPlayer() method of c1 is called.	N/A	You drew a world cup card. Pay 5000 for each player to the athlete for training costs.	You drew a world cup card. Pay 5000 for each player to the athlete for training costs.	P
toString	1	The WorldCup object c1 is printed (using the toString() method).	N/A	BLUE CARD Career Match: Athlete Name: World Cup Received if Career Match: 15000 Paid to Other Player (or Bank): Number of players * 5000	BLUE CARD Career Match: Athlete Name: World Cup Received if Career Match: 15000 Paid to Other Player (or Bank): Number of players * 5000	P

31. HouseCard

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
getMachineRandNum	1	The getMachineRandNum() method of house1 is called.	N/A	A randomly generated number between 75 and 150 inclusive.	111	P
	2	The getMachineRandNum	N/A	A randomly generated number between 75 and 150 inclusive.	75	P

		() method of house2 is called.				
	3	The getMachineRandNum () method of house3 is called.	N/A	A randomly generated number between 75 and 150 inclusive.	91	P
getInfo	1	The getInfo() method of house1 is called.	N/A	The string "You bought " + getName() + " for " + getBuyingPriceHouse() + ".", where getName() refers to the name of the house card and getBuyingPriceHouse() refers to the randomly generated house price.	You bought Java Farmhouse for 81000.	P
	2	The getInfo() method of house2 is called.	N/A	The string "You bought " + getName() + " for " + getBuyingPriceHouse() + ".", where getName() refers to the name of the house card and getBuyingPriceHouse() refers to the randomly generated house price.	You bought Bali Farmhouse for 126000.	P
	3	The getInfo() method of house3 is called.	N/A	The string "You bought " + getName() + " for " + getBuyingPriceHouse() + ".", where getName() refers to the name of the house card and getBuyingPriceHouse() refers to the randomly generated house price.	You bought Swing Penthouse for 90000.	P

toString	1	The HouseCard object house1 is printed (using the <code>toString()</code> method).	N/A	The name, buying price, and selling price (mortgage) of the house card.	HOUSE CARD Name: Java Farmhouse Buying Price: 81000 Selling Price 64800	P
	2	The HouseCard object house2 is printed (using the <code>toString()</code> method).	N/A	The name, buying price, and selling price (mortgage) of the house card.	HOUSE CARD Name: Bali Farmhouse Buying Price: 126000 Selling Price 100800	P
	3	The HouseCard object house3 is printed (using the <code>toString()</code> method).	N/A	The name, buying price, and selling price (mortgage) of the house card.	HOUSE CARD Name: Swing Penthouse Buying Price: 90000 Selling Price 72000	P
equals	1	house1 is compared to house2 using the <code>equals()</code> method.	House 1: Price = 81000 House 2: Price = 126000	false	false	P
	2	house1 is compared to house3 using the <code>equals()</code> method.	House 1: Price = 81000 House 3: Price = 90000	false	false	P
	3	house2 is compared to house3 using the <code>equals()</code> method.	House 2: Price = 126000 House 3: Price = 90000	false	false	P

C. Deck Classes

1. ActionCardDeck

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
generate	1	The generate() method of actionDeck1 is called (card types per category are randomly generated).	N/A	50 cards generated using the following scheme: 20 "Collect From the Bank" cards, 20 "Pay the Bank" cards, 5 "Pay the Player" cards, and 5 "Collect From the Player" cards. The individual card types under these classifications are randomly generated.	See Appendix B-1-01	P
	2	The generate() method of actionDeck2 is called (card types per category are randomly generated).	N/A	50 cards generated using the following scheme: 20 "Collect From the Bank" cards, 20 "Pay the Bank" cards, 5 "Pay the Player" cards, and 5 "Collect From the Player" cards. The individual card types under these classifications are randomly generated.	See Appendix B-1-02	P

	3	The generate() method of actionDeck3 is called (card types per category are randomly generated).	N/A	50 cards generated using the following scheme: 20 "Collect From the Bank" cards, 20 "Pay the Bank" cards, 5 "Pay the Player" cards, and 5 "Collect From the Player" cards. The individual card types under these classifications are randomly generated.	See Appendix B-1-03	P
drawFromDeck	1	One card is drawn from actionDeck1 (originally containing 50 cards).	See Appendix B-1-01	actionDeck1 will contain 49 elements, and pile will contain one element.	See Appendix-B-1-04	P
	2	One card is drawn from actionDeck1 (originally containing 1 card).	ActionCardDeck containing one element (not displayed in the test script).	actionDeck1 will contain 0 elements, and pile will contain 50 elements.	See Appendix-B-1-05	P
	3	One card is drawn from actionDeck1 (originally containing 0 cards).	See Appendix B-1-05	actionDeck1 will contain 49 elements, and pile will contain 1 element.	See Appendix-B-1-06	P
getMachineRandNum	1	getMachineRandNum() is called (return value is randomly generated).	N/A	A randomly generated non-negative integer (used for generating the action cards).	842138257	P
	2	getMachineRandNum() is called (return value is randomly generated).	N/A	A randomly generated non-negative integer (used for generating the action cards).	1253556252	P

	3	getMachineRandNum () is called (return value is randomly generated).	N/A	A randomly generated non-negative integer (used for generating the action cards).	1853838182	P
--	---	--	-----	---	------------	---

2. CareerCardDeck

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
generate	1	The generate() method of careerDeck1 is called (max number of pay raises of the career cards are randomly generated).	N/A	7 cards generated using the following scheme: one each of the Lawyer, Accountant, ComputerConsultant, Doctor, Server, RacecarDriver, and Athlete career cards.	See Appendix B-2-01	P
	2	The generate() method of careerDeck1 is called (max number of pay raises of the career cards are randomly generated).	N/A	7 cards generated using the following scheme: one each of the Lawyer, Accountant, ComputerConsultant, Doctor, Server, RacecarDriver, and Athlete career cards.	See Appendix B-2-02	P
	3	The generate() method of careerDeck1 is called (max number of pay raises of the career cards are randomly generated).	N/A	7 cards generated using the following scheme: one each of the Lawyer, Accountant, ComputerConsultant, Doctor, Server,	See Appendix B-2-03	P

				RacecarDriver, and Athlete career cards.		
drawFromDeck	1	One card is drawn from careerDeck1 (originally containing 7 cards).	See Appendix B-2-01	careerDeck1 will contain 6 elements.	See Appendix-B-2-04	P
	2	One card is drawn from careerDeck1 (originally containing 1 card).	CareerCardDeck containing one element (not displayed in the test script).	Career Card Deck Num. of Cards (Original): 7 Num. of Cards (Current): 0	Career Card Deck Num. of Cards (Original): 7 Num. of Cards (Current): 0	P
	3	One card is drawn from careerDeck1 (originally containing 0 cards).	Career Card Deck Num. of Cards (Original): 7 Num. of Cards (Current): 0	Career Card Deck Num. of Cards (Original): 7 Num. of Cards (Current): 0	Career Card Deck Num. of Cards (Original): 7 Num. of Cards (Current): 0	P

3. SalaryCardDeck

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
generate	1	The generate() method of salaryDeck1 is called (salaries are randomly generated).	N/A	10 cards with varying salary amounts. The tax dues are 10% of the salaries, and the pay raise increments are 20% of the salaries.	See Appendix B-3-01	P
	2	The generate() method of salaryDeck2 is called	N/A	10 cards with varying salary amounts. The tax dues are 10% of the	See Appendix B-3-02	P

		(salaries are randomly generated).		salaries, and the pay raise increments are 20% of the salaries.		
	3	The generate() method of salaryDeck3 is called (salaries are randomly generated).	N/A	10 cards with varying salary amounts. The tax dues are 10% of the salaries, and the pay raise increments are 20% of the salaries.	See Appendix B-3-03	P
drawFromDeck	1	One card is drawn from salaryDeck1 (originally containing 10 cards).	See Appendix B-3-01	salaryDeck1 will contain 9 elements.	See Appendix-B-3-04	P
	2	One card is drawn from salaryDeck1 (originally containing 1 card).	SalaryCardDeck containing one element (not displayed in the test script).	Salary Card Deck Num. of Cards (Original): 10 Num. of Cards (Current): 0	Salary Card Deck Num. of Cards (Original): 10 Num. of Cards (Current): 0	P
	3	One card is drawn from salaryDeck1 (originally containing 0 cards).	Salary Card Deck Num. of Cards (Original): 10 Num. of Cards (Current): 0	Salary Card Deck Num. of Cards (Original): 10 Num. of Cards (Current): 0	Salary Card Deck Num. of Cards (Original): 10 Num. of Cards (Current): 0	P

4. BlueCardDeck

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
generate	1	The generate() method of deck1 is called.	N/A	Seven generated blue cards, one for each type as specified in the machine problem specifications.	See Appendix B-4-01	P
drawFromDeck	1	The drawFromDeck() method of deck1 is called.	BlueCardDeck containing 7 elements (not displayed in the test script).	The topmost card of the blue card deck. The deck is shuffled afterwards.	See Appendix-B-4-02	P
	2	The drawFromDeck() method of deck1 is called a second time.	See Appendix B-4-02	The topmost card of the blue card deck. The deck is shuffled afterwards.	See Appendix B-4-03	P
	3	The drawFromDeck() method of deck1 is called a third time.	See Appendix B-4-03	The topmost card of the blue card deck. The deck is shuffled afterwards.	See Appendix B-4-04	P

5. HouseCardDeck

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
generate	1	The generate() method of houseDeck1 is called.	N/A	10 house cards with varying names and buying and selling amounts.	See Appendix B-5-01	P

drawFromDeck	1	The first card is drawn from houseDeck1 (originally containing 10 cards).	HouseCardDeck containing 10 elements (not displayed in the test script).	The first card of the house card deck.	See Appendix-B-5-02	P
	2	The second card is drawn from houseDeck1 (originally containing 9 cards).	See Appendix B-5-02	The second card of the house card deck.	See Appendix B-5-03	P
	3	The second card is drawn from houseDeck1 (originally containing 8 cards).	See Appendix B-5-03	The third card of the house card deck.	See Appendix B-5-04	P

D. Core Classes

1. NumberWheel

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
spin	1	The number wheel is spun five times (corresponding to five test cases).	N/A	5 randomly generated numbers between 1 and 10 inclusive. The outputs of each spin should be random.	Spin 1: 9 Spin 2: 4 Spin 3: 5 Spin 4: 6 Spin 5: 6	P
toString	1	The number wheel is printed (using the overridden toString() method).	N/A	Wheel: 1 2 3 4 5 6 7 8 9 10	Wheel: 1 2 3 4 5 6 7 8 9 10	P

2. DiscardPile

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
add	1	c1 is added to the discard pile.	A DiscardPile object with 0 elements (not displayed in the test script).	Number of cards: 1 Cards: [Bonus Pay Day]	Number of cards: 1 Cards: [Bonus Pay Day]	P
	2	c2, c3, and c4 are added to the discard pile.	Number of cards: 1 Cards: [Bonus Pay Day]	Number of cards: 4 Cards: [Bonus Pay Day, File a Lawsuit, Buy an Item, Christmas Bonus]	Number of cards: 4 Cards: [Bonus Pay Day, File a Lawsuit, Buy an Item, Christmas Bonus]	P
	3	c5 (a second instance of BonusPayDay) is	Number of cards: 4	Number of cards: 5	Number of cards: 5	P

		added to the discard pile.	Cards: [Bonus Pay Day, File a Lawsuit, Buy an Item, Christmas Bonus]	Cards: [Bonus Pay Day, File a Lawsuit, Buy an Item, Christmas Bonus, Bonus Pay Day]	Cards: [Bonus Pay Day, File a Lawsuit, Buy an Item, Christmas Bonus, Bonus Pay Day]	
returnCardsTo	1	The cards in pile are returned to deck (pile originally contains 5 elements).	Number of cards: 5 Cards: [Bonus Pay Day, File a Lawsuit, Buy an Item, Christmas Bonus, Bonus Pay Day]	Cards in the deck: [Buy an Item, Bonus Pay Day, File a Lawsuit, Bonus Pay Day, Christmas Bonus] Cards in the discard pile: []	Cards in the deck: [Buy an Item, Bonus Pay Day, File a Lawsuit, Bonus Pay Day, Christmas Bonus] Cards in the discard pile: []	P
	2	The cards in pile are returned to deck (pile originally contains 0 elements).	Number of cards: 0 Cards: []	Cards in the deck: [] Cards in the discard pile: []	Cards in the deck: [] Cards in the discard pile: []	P
toString	1	The DiscardPile object is printed (using the <code>toString()</code> method).	Number of cards: 5 Cards: [Bonus Pay Day, File a Lawsuit, Buy an Item, Christmas Bonus, Bonus Pay Day]	The number of elements in pile (5) and the names and effects of the individual action cards.	See Appendix C-2-01	P

3. Bank

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
lendLoan	1	p1 takes out one bank loan.	p1: Cash = 200000 p1: Number of Loans = 0	p1: Cash = 220000 p1: Number of Loans = 1	p1: Cash = 220000 p1: Number of Loans = 1	P

	2	p2 takes out a negative number of bank loans.	p2: Cash = 200000 p2: Number of Loans = 0	p2: Cash = 200000 p2: Number of Loans = 0	p2: Cash = 200000 p2: Number of Loans = 0	P
	3	p3 takes out fifty bank loans.	p3: Cash = 200000 p3: Number of Loans = 0	p3: Cash = 1200000 p3: Number of Loans = 50	p3: Cash = 1200000 p3: Number of Loans = 50	P
receivePayment	1	p1 pays off all their loans (one loan).	p1: Cash = 220000 p1: Number of Loans = 1	p1: Cash = 195000 p1: Number of Loans = 0	p1: Cash = 195000 p1: Number of Loans = 0	P
	2	p2 pays off all more loans than they have (zero loans).	p2: Cash = 200000 p2: Number of Loans = 0	p2: Cash = 200000 p2: Number of Loans = 0	p2: Cash = 200000 p2: Number of Loans = 0	P
	3	p3 pays off more loans than they have enough cash for (50 loans).	p3: Cash = 1200000 p3: Number of Loans = 50	p3: Cash = 1200000 p3: Number of Loans = 50	p3: Cash = 1200000 p3: Number of Loans = 50	P
giveCash	1	1000 is given to p1.	p1: Cash = 195000	p1: Cash = 196000	p1: Cash = 196000	P
	2	0 is given to p2.	p2: Cash = 200000	p2: Cash = 200000	p2: Cash = 200000	P
	3	-10000 is given to p3.	p3: Cash = 1200000	p3: Cash = 1200000	p3: Cash = 1200000	P
receiveCash	1	4000 is received from p1.	p1: Cash = 196000	p1: Cash = 192000	p1: Cash = 192000	P
	2	0 is received from p2.	p2: Cash = 200000	p2: Cash = 200000	p2: Cash = 200000	P
	3	-14000 is received from p3.	p3: Cash = 1200000	p3: Cash = 1200000	p3: Cash = 1200000	P
toString	1	The Bank object is printed (using the overridden <code>toString()</code> method).	N/A	BANK 0 0 50	BANK 0 0 50	P

4. Player

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
spinWheel	1	The spinWheel() method of p1 is called (return value is randomly generated).	N/A	A randomly generated number between 1 and 10 inclusive.	p1 Spin: 2	P
	2	The spinWheel() method of p2 is called (return value is randomly generated).	N/A	A randomly generated number between 1 and 10 inclusive.	p2 Spin: 8	P
	3	The spinWheel() method of p2 is called (return value is randomly generated).	N/A	A randomly generated number between 1 and 10 inclusive.	p3 Spin: 2	P
move	1	p1 moves three spaces from their previous space.	p1: Board Index = 0	p1: Board Index = 3	p1: Board Index = 3	P
	2	p1 moves ten spaces from their previous space.	p1: Board Index = 3	p1: Board Index = 4	p1: Board Index = 4	P
graduate	1	The graduate() method of p1 is called.	p1: hasDegree = false	p1: hasDegree = true	p1: hasDegree = true	P

	2	The graduate() method of p2 is called.	p2: hasDegree = false	p2: hasDegree = true	p2: hasDegree = true	P
	3	The graduate() method of p3 is called.	p3: hasDegree = false	p3: hasDegree = true	p3: hasDegree = true	P
marry	1	The marry() method of p1 is called.	p1: isMarried = false	p1: isMarried = true	p1: isMarried = true	P
	2	The marry() method of p1 is called again.	p1: isMarried = true	p1: isMarried = true	p1: isMarried = true	P
haveBaby	1	The haveBaby() method of p1 is called.	p1: Number of Children = 0	p1: Number of Children = 1	p1: Number of Children = 1	P
haveTwins	1	The haveTwins() method of p2 is called.	p2: Number of Children = 0	p2: Number of Children = 2	p2: Number of Children = 2	P
drawFromCareer Deck	1	The drawFromCareer Deck method of p1 is called.	N/A	A valid career card (following the MP specifications). careerDeck will have six elements.	See Appendix C-4-01	P
	2	The drawFromCareer Deck method of p2 is called.	N/A	A valid and unique career card (following the MP specifications). careerDeck will have five elements.	See Appendix C-4-02	P
	3	The drawFromCareer Deck method of p3 is called.	N/A	A valid and unique career card (following the MP specifications).	See Appendix C-4-03	P

				careerDeck will have four elements.		
drawFromSalaryDeck	1	The drawFromSalaryDeck method of p1 is called.	N/A	A valid salary card (following the MP specifications). salaryDeck will have nine elements.	See Appendix C-4-04	P
	2	The drawFromSalaryDeck method of p2 is called.	N/A	A valid and unique salary card (following the MP specifications). salaryDeck will have eight elements.	See Appendix C-4-05	P
	3	The drawFromSalaryDeck method of p3 is called.	N/A	A valid and unique salary card (following the MP specifications). salaryDeck will have seven elements.	See Appendix C-4-06	P
drawFromActionDeck	1	The drawFromActionDeck method of p1 is called.	N/A	A valid action card (following the MP specifications). actionDeck will have 49 elements. pile will have 1 one element.	See Appendix C-4-07	P
	2	The drawFromActionDeck method of p2 is called.	N/A	A valid action card (following the MP specifications). actionDeck will have 48 elements. pile will have 2 elements.	See Appendix C-4-08	P

	3	The drawFromAction Deck method of p3 is called.	N/A	A valid action card (following the MP specifications). actionDeck will have 47 elements. pile will have 3 elements.	See Appendix C-4-09	P
drawFromHouse Deck	1	The drawFromHouse Deck method of p1 is called.	N/A	A valid house card. houseDeck will have 9 remaining elements.	See Appendix C-4-10	P
drawFromBlueDeck	1	The drawFromBlueDeck method of p1 is called.	N/A	A valid blue card following the MP specifications. blueDeck will still have 7 elements.	See Appendix C-4-11	P
keepCareerCard	1	p1 keeps a career card from the CareerCard deck.	N/A	p1: Career = Server p1: Num Pay Raises = 4 CAREER CARD Is Degree Required: false Max Number of Pay Raises Range: [1, 4] Max Number of Pay Raises: 4 Career: Server	p1: Career = Server p1: Num Pay Raises = 4 CAREER CARD Is Degree Required: false Max Number of Pay Raises Range: [1, 4] Max Number of Pay Raises: 4 Career: Server	P
	2	p2 keeps a career card from the CareerCard deck.	N/A	p2: Career = Athlete p2: Num Pay Raises = 3 CAREER CARD Is Degree Required: false	p2: Career = Athlete p2: Num Pay Raises = 3 CAREER CARD Is Degree Required: false	P

				Max Number of Pay Raises Range: [1, 6] Max Number of Pay Raises: 3 Career: Athlete	Max Number of Pay Raises Range: [1, 6] Max Number of Pay Raises: 3 Career: Athlete	
	3	p3 keeps a career card from the CareerCard deck.	N/A	p3: Career = Accountant p3: Num Pay Raises = 4 CAREER CARD Is Degree Required: true Max Number of Pay Raises Range: [4, 7] Max Number of Pay Raises: 4 Career: Accountant	p3: Career = Accountant p3: Num Pay Raises = 4 CAREER CARD Is Degree Required: true Max Number of Pay Raises Range: [4, 7] Max Number of Pay Raises: 4 Career: Accountant	P
keepSalaryCard	1	p1 keeps a salary card from the SalaryCard deck.	N/A	p1: Salary = 80000 p1: Tax Due = 8000 p1: Pay Raise = 16000 SALARY CARD Salary: 80000 Tax Due: 8000 Pay Raise: 16000	p1: Salary = 80000 p1: Tax Due = 8000 p1: Pay Raise = 16000 SALARY CARD Salary: 80000 Tax Due: 8000 Pay Raise: 16000	P
	2	p2 keeps a salary card from the SalaryCard deck.	N/A	p2: Salary = 60000 p2: Tax Due = 6000 p2: Pay Raise = 12000 SALARY CARD Salary: 60000 Tax Due: 6000 Pay Raise: 12000	p2: Salary = 60000 p2: Tax Due = 6000 p2: Pay Raise = 12000 SALARY CARD Salary: 60000 Tax Due: 6000 Pay Raise: 12000	P

	3	p3 keeps a salary card from the SalaryCard deck.	N/A	p3: Salary = 70000 p3: Tax Due = 7000 p3: Pay Raise = 14000 SALARY CARD Salary: 70000 Tax Due: 7000 Pay Raise: 14000	p3: Salary = 70000 p3: Tax Due = 7000 p3: Pay Raise = 14000 SALARY CARD Salary: 70000 Tax Due: 7000 Pay Raise: 14000	P
keepHouseCard	1	p1 keeps a house card from the HouseCard deck.	p1: Cash = 200000	The house details of p1 will be initialized and their cash will decrease by the specified buying price of the house.	p1: Cash = 113000 p1: House = Bali Farmhouse	P
returnCard	1	The career card of p2 is returned to the deck.	CAREER CARD Career: Server Is Degree Required: No Max Number of Pay Raises Range: [1, 4] Max Number of Pay Raises: 2	The returned card is added to the career card deck.	CAREER CARD Career: Server Is Degree Required: No Max Number of Pay Raises Range: [1, 4] Max Number of Pay Raises: 2 CAREER CARD Career: Athlete Is Degree Required: No Max Number of Pay Raises Range: [1, 6] Max Number of Pay Raises: 5	P

returnCardToBottom	1	The career card of p3 is returned to the bottom of the deck.	CAREER CARD Career: Doctor Is Degree Required: Yes Max Number of Pay Raises Range: [5, 8] Max Number of Pay Raises: 7	The returned card is added to the bottom of the career card deck.	CAREER CARD Career: Server Is Degree Required: No Max Number of Pay Raises Range: [1, 4] Max Number of Pay Raises: 2 CAREER CARD Career: Athlete Is Degree Required: No Max Number of Pay Raises Range: [1, 6] Max Number of Pay Raises: 5 CAREER CARD Career: Doctor Is Degree Required: Yes Max Number of Pay Raises Range: [5, 8] Max Number of Pay Raises: 7	P
executeActionCard	1	p1 executes a BonusPayday card.	p1: Cash = 200000 p1: Salary = 80000	You have received a bonus equal to your current salary! Collect a paycheck from the Bank. p1: Cash = 280000	You have received a bonus equal to your current salary! Collect a paycheck from the Bank. p1: Cash = 280000	P

	2	p2 executes a SellAnItem card.	p2: Cash = 280000	You have completed a sale. Collect 7000 from the Bank. Item Price = 7000 p2: Cash = 207000	You have completed a sale. Collect 7000 from the Bank. Item Price = 7000 p2: Cash = 207000	P
	3	p3 executes a TaxRefund card.	p3: Cash = 200000 p3: Tax Due = 7000	You received a tax refund! Collect your tax due from the Bank. p3: Cash = 207000	You received a tax refund! Collect your tax due from the Bank. p3: Cash = 207000	P
	4	p1 executes a BuyAnItem card.	p1: Cash = 280000	You have bought an item for 8000. Item Price = 8000 p1: Cash = 272000	You have bought an item for 8000. Item Price = 8000 p1: Cash = 272000	P
	5	p2 executes a TrafficViolation card.	p2: Cash = 207000	You committed a traffic violation. Pay a fine of 2000. Fine = 2000 p2: Cash = 205000	You committed a traffic violation. Pay a fine of 2000. Fine = 2000 p2: Cash = 205000	P
	6	p3 executes a WinACompetition card.	p3: Cash = 207000	You have won a competition. Pay 5000 as competition tax. Competition Tax = 5000 p3: Cash = 202000	You have won a competition. Pay 5000 as competition tax. Competition Tax = 5000 p3: Cash = 202000	P

	7	p1 executes a Lawsuit card.	<p>p1: Cash = 272000</p> <p>p2: Cash = 205000</p>	You have lost a lawsuit. Pay 60000 to a player of your choosing. Amount = 60000 p1: Cash = 212000 p2: Cash = 265000	You have lost a lawsuit. Pay 60000 to a player of your choosing. Amount = 60000 p1: Cash = 212000 p2: Cash = 265000	P
	8	p2 executes a ChristmasBonus card.	<p>p2: Cash = 265000</p> <p>p1: Cash = 212000</p> <p>p3: Cash = 202000</p>	Happy holidays! Pay 14000 to every other player as a Christmas gift. Amount = 14000 p2: Cash = 237000 p1: Cash = 226000 p3: Cash = 216000	Happy holidays! Pay 14000 to every other player as a Christmas gift. Amount = 14000 p2: Cash = 237000 p1: Cash = 226000 p3: Cash = 216000	P
	9	p3 executes a FileALawsuit card.	<p>p3: Cash = 216000</p> <p>p1: Cash = 226000</p>	You have won a lawsuit. Collect 50000 from a player of your choosing. Amount = 50000 p3: Cash = 266000 p1: Cash = 176000	You have won a lawsuit. Collect 50000 from a player of your choosing. Amount = 50000 p3: Cash = 266000 p1: Cash = 176000	P
	10	p1 executes a ItsYourBirthday card.	<p>p1: Cash = 176000</p> <p>p2: Cash = 237000</p> <p>p3: Cash = 266000</p>	It's your birthday! Collect 6000 from every other player as a gift. Amount = 6000 p1: Cash = 188000 p2: Cash = 231000	It's your birthday! Collect 6000 from every other player as a gift. Amount = 6000 p1: Cash = 188000 p2: Cash = 231000	P

				p3: Cash = 260000	p3: Cash = 260000	
executeBlueCard	1	p1 executes a LawsuitBlue card.	p1: Cash = 1023000 Card Amount = 90000	p1: Cash = 933000	p1: Cash = 933000	P
	2	p1 executes a SalaryTaxDue card.	p1: Cash = 933000 p1: Tax Due = 4000	p1: Cash = 929000	p1: Cash = 929000	P
	3	p1 executes a TipTheServer card.	p1: Cash = 929000	p1: Cash = 928000	p1: Cash = 928000	P
	4	p1 executes a SkiAccident card.	p1: Cash = 928000	p1: Cash = 918000	p1: Cash = 918000	P
	5	p1 executes a ComputerRepair card.	p1: Cash = 918000	p1: Cash = 913000	p1: Cash = 913000	P
	6	p1 executes a WorldCup card.	p1: Cash = 913000	p1: Cash = 898000	p1: Cash = 898000	P
	7	p1 executes an F1Race card.	p1: Cash = 898000 p1: Salary = 40000	p1: Cash = 894000	p1: Cash = 894000	P
borrowLoan	1	p1 takes out 3 loans.	p1: Cash = 188000 p1: NumLoans = 0	p1 takes out three loans. p1: Cash = 248000 p1: NumLoans = 3	p1 takes out three loans. p1: Cash = 248000 p1: NumLoans = 3	P
	2	p2 takes out 0 loans.	p2: Cash = 231000 p2: NumLoans = 0	p2 takes out zero loans. p2: Cash = 231000 p2: NumLoans = 0	p2 takes out zero loans. p2: Cash = 231000 p2: NumLoans = 0	P
	3	p3 takes out -5 loans.	p3: Cash = 260000 p3: NumLoans = 0	p3 takes out -5 loans. p3: Cash = 260000 p3: NumLoans = 0	p3 takes out -5 loans. p3: Cash = 260000 p3: NumLoans = 0	P
payLoan	1	p1 pays 3 loans.	p1: Cash = 248000 p1: NumLoans = 3	p1 pays 3 loans. p1: Cash = 173000 p1: NumLoans = 0	p1 pays 3 loans. p1: Cash = 173000 p1: NumLoans = 0	P
	2	p2 pays 0 loans.	p2: Cash = 231000 p2: NumLoans = 0	p2 pays 0 loans. p2: Cash = 231000	p2 pays 0 loans. p2: Cash = 231000	P

				p2: NumLoans = 0	p2: NumLoans = 0	
	3	p3 pays 1000000 loans.	p3: Cash = 260000 p3: NumLoans = 0	p3 pays 1000000 loans. p3: Cash = 260000 p3: NumLoans = 0	p3 pays 1000000 loans. p3: Cash = 260000 p3: NumLoans = 0	P
receiveCash	1	p1 receives 1000.	p1: Cash = 173000	p1 receives 1000. p1: Cash = 174000	p1 receives 1000. p1: Cash = 174000	P
	2	p2 receives 0.	p2: Cash = 231000	p2 receives 0. p2: Cash = 231000	p2 receives 0. p2: Cash = 231000	P
	3	p3 receives -1000.	p3: Cash = 260000	p3 receives -1000. p3: Cash = 260000	p3 receives -1000. p3: Cash = 260000	P
payCashToPlayer	1	p1 pays p2 1000.	p1: Cash = 174000 p2: Cash = 231000	p1 pays p2 1000. p1: Cash = 173000 p2: Cash = 232000	p1 pays p2 1000. p1: Cash = 173000 p2: Cash = 232000	P
	2	p2 pays p3 0.	p2: Cash = 232000 p3: Cash = 260000	p2 pays p3 0. p2: Cash = 232000 p3: Cash = 260000	p2 pays p3 0. p2: Cash = 232000 p3: Cash = 260000	P
	3	p3 pays p1 double p3's cash on hand.	p3: Cash = 260000 p3: Num Loans = 0 p1: Cash = 173000	p3 pays p1 double p3's cash on hand p3: Cash = 0 p3: Num Loans = 13 p1: Cash = 693000	p3 pays p1 double p3's cash on hand p3: Cash = 0 p3: Num Loans = 13 p1: Cash = 693000	P
collectCashFromBank	1	p1 collects -1000.	p1: Cash = 693000	p1 collects -1000. p1: Cash = 693000	p1 collects -1000. p1: Cash = 693000	P
	2	p2 collects 0.	p2: Cash = 232000	p2 collects 0. p2: Cash = 232000	p2 collects 0. p2: Cash = 232000	P
	3	p3 collects 100000.	p3: Cash = 0	p3 collects 100000. p3: Cash = 100000	p3 collects 100000. p3: Cash = 100000	P
payCashToBank	1	p1 pays more than p1's cash on hand.	p1: Cash = 693000 p1: Num Loans = 0	p1 pays more than p1's cash on hand. p1: Cash = 0	p1 pays more than p1's cash on hand. p1: Cash = 0	P

				p1: Num Loans = 1	p1: Num Loans = 1	
	2	p2 pays 0.	p2: Cash = 232000	p2 pays 0. p2: Cash = 232000	p2 pays 0. p2: Cash = 232000	P
	3	p3 pays -1000.	p3: Cash = 100000	p3 pays -1000. p3: Cash = 100000	p3 pays -1000. p3: Cash = 100000	P
increaseCashBy	1	p1 increases cash by 5000.	p1: Cash = 0	p1 increases cash by 5000. p1: Cash = 5000	p1 increases cash by 5000. p1: Cash = 5000	P
	2	p2 increases cash by 0.	p2: Cash = 232000	p2 increases cash by 0. p2: Cash = 232000	p2 increases cash by 0. p2: Cash = 232000	P
	3	p3 increases cash by -1000.	p3: Cash = 100000	p3 increases cash by -1000. p3: Cash = 100000	p3 increases cash by -1000. p3: Cash = 100000	P
decreaseCashBy	1	p1 decreases cash by 1000.	p1: Cash = 5000	p1 decreases cash by 1000. p1: Cash = 4000	p1 decreases cash by 1000. p1: Cash = 4000	P
	2	p2 decreases cash by -5000.	p2: Cash = 232000	p2 decreases cash by -5000. p2: Cash = 232000	p2 decreases cash by -5000. p2: Cash = 232000	P
	3	p3 decreases cash by more than p3's cash on hand.	p3: Cash = 100000 p3: Num Loans = 12	p3 decreases cash by more than p3's cash on hand. p3: Num Loans = 13	p3 decreases cash by more than p3's cash on hand. p3: Num Loans = 13	P
increaseNumLoans By	1	p1 increases their number of loans by 5.	p1: Num Loans = 1	p1 increases their number of loans by 5. p1: Num Loans = 6	p1 increases their number of loans by 5. p1: Num Loans = 6	P
	2	p2 increases their number of loans by 0.	p2: Num Loans = 0	p2 increases their number of loans by 0. p2: Num Loans = 0	p2 increases their number of loans by 0. p2: Num Loans = 0	P

	3	p3 increases their number of loans by -1.	p3: Num Loans = 13	p3 increases their number of loans by -1. p3: Num Loans = 13	p3 increases their number of loans by -1. p3: Num Loans = 13	P
decreaseNumLoansBy	1	p1 decreases their number of loans by 5.	p1: Num Loans = 6	p1 decreases their number of loans by 5. p1: Num Loans = 1	p1 decreases their number of loans by 5. p1: Num Loans = 1	P
	2	p2 decreases their number of loans by 0.	p2: Num Loans = 0	p2 decreases their number of loans by 0. p2: Num Loans = 0	p2 decreases their number of loans by 0. p2: Num Loans = 0	P
	3	p3 decreases their number of loans by -1.	p3: Num Loans = 13	p3 decreases their number of loans by -1. p3: Num Loans = 13	p3 decreases their number of loans by -1. p3: Num Loans = 13	P
sellHouseToBank	1	p1 sells their house to the bank.	p1: Cash = 4000 p1: House Selling Price = 84800	p1: Cash = 88800	p1: Cash = 88800	P
	2	p2 sells their house to the bank (but p2 has no house).	p2: Cash = 199000	p2: Cash = 199000	p2: Cash = 199000	P
receivePayRaise	1	p1 receives a pay raise.	p1: Salary = 50000 p1: Pay Raise Amount = 10000 p1: Current Pay Raises = 0 p1: Maximum Pay Raises = 7	p1: Salary = 60000	p1: Salary = 60000	P
	2	p1 receives a pay raise (p1 is already at the maximum number of pay raises).	p1: Salary = 120000 p1: Pay Raise Amount = 10000 p1: Current Pay Raises = 7 p1: Maximum Pay Raises = 7	p1: Salary = 120000	p1: Salary = 120000	P

retire	1	p1 is retired (set as the first player to retire).	<p1: cash="151200<br/">p1: Loans = 7 p1: Children = 1 p1: House Selling Price = 73600</p1:> p1: Loans = 0	p1: Cash = 159800 p1: Loans = 0	P
isValidName	1	The string "Mark" is checked for validity by p1.	Mark	true	true
	2	The string " " is checked for validity by p1.	(one space key)	false	false
	3	The string " Hello" is checked for validity by p1.	Hello (preceded by a space key)	true	true
toString	1	The Player object p1 is printed (using the overridden toString() method).	N/A	Cash: 159800 Num. of Loans: 0 Career: Computer Consultant Current Salary: 66000 Current Tax Due: 15000 Current Pay Raise: 36000 Num. of Pay Raises: 6 Max. Num. of Pay Raises: 6 Increase per Pay Raise: 6000	Cash: 159800 Num. of Loans: 0 Career: Computer Consultant Current Salary: 66000 Current Tax Due: 15000 Current Pay Raise: 36000 Num. of Pay Raises: 6 Max. Num. of Pay Raises: 6 Increase per Pay Raise: 6000

			House (Name): Florence Penthouse Buying Price: 92000 Selling Price: 73600 Is Married: Yes Num. of Children: 1 Has Degree: Yes	House (Name): Florence Penthouse Buying Price: 92000 Selling Price: 73600 Is Married: Yes Num. of Children: 1 Has Degree: Yes		
	2	The Player object p2 is printed (using the overridden toString() method).	N/A	Cash: 294000 Num. of Loans: 0 Career: Doctor Current Salary: 90000 Current Tax Due: 9000 Current Pay Raise: 0 Num. of Pay Raises: 0 Max. Num. of Pay Raises: 7 Increase per Pay Raise: 18000 Is Married: No Num. of Children: 2 Has Degree: Yes	Cash: 294000 Num. of Loans: 0 Career: Doctor Current Salary: 90000 Current Tax Due: 9000 Current Pay Raise: 0 Num. of Pay Raises: 0 Max. Num. of Pay Raises: 7 Increase per Pay Raise: 18000 Is Married: No Num. of Children: 2 Has Degree: Yes	P
	3	The Player object p3 is printed (using the overridden toString() method).	N/A	Cash: 110000 Num. of Loans: 17 Career: Athlete Current Salary: 80000 Current Tax Due: 8000	Cash: 110000 Num. of Loans: 17 Career: Athlete Current Salary: 80000 Current Tax Due: 8000	P

			Current Pay Raise: 0 Num. of Pay Raises: 0 Max. Num. of Pay Raises: 3 Increase per Pay Raise: 16000 Is Married: No Num. of Children: 0 Has Degree: Yes	Current Pay Raise: 0 Num. of Pay Raises: 0 Max. Num. of Pay Raises: 3 Increase per Pay Raise: 16000 Is Married: No Num. of Children: 0 Has Degree: Yes	
--	--	--	--	--	--

5. Board

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
isOrangeSpace	1	The index of the start space is passed as a parameter.	0	false	false	P
	2	The index of an orange space is passed as a parameter.	1	true	true	P
isBlueSpace	1	The index of the start space is passed as a parameter.	0	false	false	P
	2	The index of a blue space is passed as a parameter.	15	true	true	P

isGreenSpace	1	The index of the start space is passed as a parameter.	0	false	false	P
	2	The index of a green space is passed as a parameter.	18	true	true	P
isMagentaSpace	1	The index of the start space is passed as a parameter.	0	false	false	P
	2	The index of a magenta space is passed as a parameter.	4	true	true	P
getMachineRandNum	1	The getMachineRandNum() method of this object is called.	N/A	Either the number 1 or 2	1	P

6. GameMaster

The results of the methods of this class cannot be displayed in isolation. Specifically, the GameMaster class methods alter the attributes of the players in the game, but there exists no method for getting the Player objects, as this would violate the design principles of the program. Thus, the methods of this class were tested by running the GUI, sample runs of which are included in Section A of this document.

E. Space Classes

1. PayDay

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The execute() method of space1 is called, and p1 has the salary card s1.	p1: Cash = 200000 p1: Salary = 20000	p1: Cash = 220000	p1: Cash = 220000	P
	2	The execute() method of space1 is called, and p1 has the salary card s2.	p1: Cash = 220000 p1: Salary = 20000	p1: Cash = 240000	p1: Cash = 240000	P
	3	The execute() method of space1 is called, and p1 has the salary card s3.	p1: Cash = 240000 p1: Salary = 40000	p1: Cash = 280000	p1: Cash = 280000	P
getInfo	1	The getInfo() method of space1 is called with p1 passed as the parameter.	N/A	Sample Name, you are on a Pay Day space. Receive your current salary from the bank.	Sample Name, you are on a Pay Day space. Receive your current salary from the bank.	P

2. PayRaise

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The execute() method of space1 is	p1: Cash = 200000 p1: Salary = 20000	p1: Cash = 224000 p1: New Salary = 24000	p1: Cash = 224000 p1: New Salary = 24000	P

		called, and p1 has the salary card s1.	p1: Pay Raise Amount = 4000			
	2	The execute() method of space1 is called a second time.	p1: Cash = 224000 p1: Salary = 24000 p1: Pay Raise Amount = 4000	p1: Cash = 252000 p1: New Salary = 28000	p1: Cash = 252000 p1: New Salary = 28000	P
	3	The execute() method of space1 is called a third time.	p1: Cash = 252000 p1: Salary = 28000 p1: Pay Raise Amount = 4000	p1: Cash = 284000 p1: New Salary = 32000	p1: Cash = 284000 p1: New Salary = 32000	P
getInfo	1	The getInfo() method of space1 is called with p1 passed as the parameter.	N/A	Sample Name, you are on a Pay Raise space. Receive a pay raise and your new salary from the bank. New salary values have been updated in the player details.	Sample Name, you are on a Pay Raise space. Receive a pay raise and your new salary from the bank. New salary values have been updated in the player details.	P

3. BuyAHouse

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
getHouseCards	1	The getHouseCards() method of space1 is called.	N/A	The list of cards available in the house card deck.	See Appendix D-3-01	P
execute	1	The execute() method of space1 is called, and p1 chooses to keep the first house card in the list.	p1: House = null	p1 keeps the first house card.	p1: New House = HOUSE CARD Name: Bali Farmhouse Buying Price: 91000 Selling Price 72800	P

	2	The execute() method of space1 is called, and p1 chooses to keep the second house card in the list.	p1: House = HOUSE CARD Name: Bali Farmhouse Buying Price: 91000 Selling Price 72800	p1 has a new house card	p1: New House = HOUSE CARD Name: Java Farmhouse Buying Price: 87000 Selling Price 69600	P
	3	The execute() method of space1 is called, and p1 chooses to keep the first house card in the list.	p1: House = HOUSE CARD Name: Java Farmhouse Buying Price: 87000 Selling Price 69600	p1 has a new house card	p1: New House = HOUSE CARD Name: Cece Townhouse Buying Price: 80000 Selling Price 64000	P
getInfo	1	The getInfo() method of space1 is called with p1 passed as the parameter.	N/A	Sample Name, you are on a Buy A House space. Choose a house to buy from the available options. You are entitled to ANOTHER TURN right after this.	Sample Name, you are on a Buy A House space. Choose a house to buy from the available options. You are entitled to ANOTHER TURN right after this.	P

4. CollegeCareerChoice

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
drawCareerCards	1	The drawCareerCards() method of space1 is called.	N/A	Two career cards drawn from the career card deck.	CAREER CARD Career: Lawyer Is Degree Required: Yes Max Number of Pay Raises Range: [5, 8] Max Number of Pay Raises: 5	P

				CAREER CARD Career: Computer Consultant Is Degree Required: Yes Max Number of Pay Raises Range: [3, 7] Max Number of Pay Raises: 6	
drawSalaryCards	1	The drawSalaryCards() method of space1 is called.	N/A	Two salary cards drawn from the salary card deck. SALARY CARD Salary: 80000 Tax Due: 8000 Pay Raise: 16000 SALARY CARD Salary: 50000 Tax Due: 5000 Pay Raise: 10000	P
executeCareer	1	The executeCareer() method of space1 is called, and p1 chooses to keep the first career card in the list.	N/A (no kept career card yet)	p1: New Career = Lawyer	P
executeSalary	1	The executeSalary() method of space1 is called, and p1 chooses to keep the first salary card in the list.	N/A (no kept salary card yet)	p1: New Salary = 80000	P
getInfo	1	The getInfo() method of space1 is called with p1 passed as the parameter.	N/A	Sample Name, you are on a College Career Choice space.	P

				Choose a career and salary card from the available options. You are entitled to ANOTHER TURN right after this.	Choose a career and salary card from the available options. You are entitled to ANOTHER TURN right after this.	
--	--	--	--	---	---	--

5. GetMarried

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The execute() method of space1 is called, with an odd random number passed.	p1: Cash = 200000 p1: Status = false p2: Cash = 200000 p3: Cash = 200000	p1: New Cash = 210000 p1: Status = true p2: New Cash = 195000 p3: New Cash = 195000	p1: New Cash = 210000 p1: Status = true p2: New Cash = 195000 p3: New Cash = 195000	P
	2	The execute() method of space1 is called, with an even random number passed.	p2: Cash = 195000 p2: Status = false p1: Cash = 210000 p3: Cash = 195000	p2: New Cash = 215000 p2: Status = true p1: New Cash = 200000 p3: New Cash = 185000	p2: New Cash = 215000 p2: Status = true p1: New Cash = 200000 p3: New Cash = 185000	P
	3	The execute() method of space1 is called, with p1 already married.	p1: Cash = 200000 p1: Status = true	p1: New Cash = 200000 p1: Status = true	p1: New Cash = 200000 p1: Status = true	P
spinWheel	1	The spinWheel() method of space1 is called.	N/A	A randomly generated number between 1 and 10 inclusive.	8	P
	2	The spinWheel() method of space1 is called a second time.	N/A	A randomly generated number between 1 and 10 inclusive.	7	P

	3	The spinWheel() method of space1 is called a third time.	N/A	A randomly generated number between 1 and 10 inclusive.	5	
getInfo	1	The getInfo() method of space1 is called with p3 passed as the parameter.	N/A	Player 3, you are on a Get Married space. Spin the wheel and collect \$5000 from each player for an odd number spun and \$10000 from each player for an even number. You are entitled to ANOTHER TURN right after this.	Player 3, you are on a Get Married space. Spin the wheel and collect \$5000 from each player for an odd number spun and \$10000 from each player for an even number. You are entitled to ANOTHER TURN right after this.	P
	2	The getInfo() method of space1 is called with p1 passed as the parameter.	N/A	Sample Name is already married. You are entitled to ANOTHER TURN right after this.	Sample Name is already married. You are entitled to ANOTHER TURN right after this.	P
getInfoDisplay	1	The getInfoDisplay() method of space1 is called with p3 passed as the parameter.	N/A	Player 3, you are on a Get Married space. Spin the wheel and collect \$5000 from each player for an odd number spun and \$10000 from each player for an even number. You are entitled to ANOTHER TURN right after this.	Player 3, you are on a Get Married space. Spin the wheel and collect \$5000 from each player for an odd number spun and \$10000 from each player for an even number. You are entitled to ANOTHER TURN right after this.	P
	2	The getInfoDisplay() method of space1 is	N/A	Sample Name is already married.	Sample Name is already married.	P

		called with p1 passed as the parameter.		You are entitled to ANOTHER TURN right after this.	You are entitled to ANOTHER TURN right after this.	
--	--	---	--	--	--	--

6. GraduationSpace

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The execute() method of space1 is called, with p1 passed as a parameter.	p1: Degree = false	p1: New Degree = true	p1: New Degree = true	P
getInfo	1	The getInfo() method of space1 is called with p1 passed as the parameter.	N/A	Sample Name, you are on a Graduation space. You now have a college degree. You are entitled to ANOTHER TURN right after this.	Sample Name, you are on a Graduation space. You now have a college degree. You are entitled to ANOTHER TURN right after this.	P

7. HaveBaby

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The execute() method of space1 is called, with p1 (unmarried) passed as a parameter.	p1: Status = false p1: Children = 0	p1: New Children = 0	p1: New Children = 0	P

	2	The execute() method of space1 is called, with p1 (married) passed as a parameter.	<p1: status="true<br/">p1: Children = 0 p1: Cash = 210000</p1:>	p1: New Children = 1 p1: New Cash = 220000	p1: New Children = 1 p1: New Cash = 220000	P
getInfo	1	The getInfo() method of space1 is called with p1 passed as the parameter.	N/A	Sample Name, you are on a Have Baby space. Receive \$5000 from each player as a gift. You are entitled to ANOTHER TURN right after this.	Sample Name, you are on a Have Baby space. Receive \$5000 from each player as a gift. You are entitled to ANOTHER TURN right after this.	P
	2	The getInfo() method of space1 is called with p2 passed as the parameter.	N/A	Player 2 is not married. You are entitled to ANOTHER TURN right after this.	Player 2 is not married. You are entitled to ANOTHER TURN right after this.	P

8. HaveTwins

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The execute() method of space1 is called, with p1 (unmarried) passed as a parameter.	<p1: status="false<br/">p1: Children = 0</p1:>	p1: New Children = 0	p1: New Children = 0	P
	2	The execute() method of space1 is called, with p1 (married) passed as a parameter.	<p1: status="true<br/">p1: Children = 0 p1: Cash = 210000</p1:>	p1: New Children = 2 p1: New Cash = 230000	p1: New Children = 2 p1: New Cash = 230000	P

getInfo	1	The getInfo() method of space1 is called with p1 passed as the parameter.	N/A	Sample Name, you are on a Have Twins space. Receive \$10000 from each player as a gift. You are entitled to ANOTHER TURN right after this.	Sample Name, you are on a Have Twins space. Receive \$10000 from each player as a gift. You are entitled to ANOTHER TURN right after this.	P
	2	The getInfo() method of space1 is called with p2 passed as the parameter.	N/A	Player 2 is not married. You are entitled to ANOTHER TURN right after this.	Player 2 is not married. You are entitled to ANOTHER TURN right after this.	P

9. JobSearch

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
drawCareerCard	1	The drawCareerCard() method of space1 is called.	p1: Current career card CAREER CARD Career: Accountant Is Degree Required: Yes Max Number of Pay Raises Range: [4, 7] Max Number of Pay Raises: 4	A career card drawn from the career card deck and the player's current career card.	CAREER CARD Career: Accountant Is Degree Required: Yes Max Number of Pay Raises Range: [4, 7] Max Number of Pay Raises: 4 CAREER CARD Career: Racecar Driver Is Degree Required: No Max Number of Pay Raises Range: [2, 8] Max Number of Pay Raises: 5	P

drawSalaryCard	1	The drawSalaryCard() method of space1 is called.	p1: Current salary card SALARY CARD Salary: 70000 Tax Due: 7000 Pay Raise: 14000	A salary card drawn from the salary card deck and the player's current salary card.	SALARY CARD Salary: 70000 Tax Due: 7000 Pay Raise: 14000 SALARY CARD Salary: 80000 Tax Due: 8000 Pay Raise: 16000	P
executeCareer	1	The executeCareer() method of space1 is called, and p1 chooses to keep the first career card in the list.	p1: Current career card CAREER CARD Career: Accountant Is Degree Required: Yes Max Number of Pay Raises Range: [4, 7] Max Number of Pay Raises: 4	p1: New Career = Accountant	p1: New Career = Accountant	P
executeSalary	1	The executeSalary() method of space1 is called, and p1 chooses to keep the first salary card in the list.	p1: Current salary card SALARY CARD Salary: 70000 Tax Due: 7000 Pay Raise: 14000	p1 keeps their new salary card	p1: New Salary = 80000	P
getInfo	1	The getInfo() method of space1 is called with p1 passed as the parameter.	N/A	Sample Name, you are on a Job Search space. Choose a career and salary card from the available options. You are entitled to ANOTHER TURN right after this.	Sample Name, you are on a Job Search space. Choose a career and salary card from the available options. You are entitled to ANOTHER TURN right after this.	P

10. WhichPath

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The execute() method of space1 is called, and p1 chooses the Change Career Path.	Current next index = 1	New next index = 2	New next index = 2	P
	2	The execute() method of space1 is called, and p1 chooses the Normal Path.	Current next index = 2	New next index = 1	New next index = 1	P
	3	The execute() method of space1 is called, and p1 chooses the Start A Family Path.	Current next index = 1	New next index = 2	New next index = 2	P
getInfo	1	The getInfo() method of space1 is called with p1 passed as the parameter.	N/A	Sample Name, you are on a Which Path space. Choose between the two paths. You are entitled to ANOTHER TURN right after this.	Sample Name, you are on a Which Path space. Choose between the two paths. You are entitled to ANOTHER TURN right after this.	P

11. BlueSpace

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
drawBlueCard	1	The drawBlueCard() method of space1 is called.	N/A (no drawn blue card)	A drawn blue card	BLUE CARD Career Match: Accountant Name: Salary Tax Due Received if Career Match: 15000 Paid to Other Player (or Bank): Tax Due	P
getInfo	1	The getInfo() method of space1 is called with p1 passed as the parameter.	N/A	Sample Name, you are on a blue space. Draw a blue card.	Sample Name, you are on a blue space. Draw a blue card.	P

12. EndSpace

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The execute() method of space1 is called.	p1: Cash = 200000	p1: New Cash = 300000	p1: New Cash = 300000	P
getInfo	1	The getInfo() method of space1 is called with p1 passed as the parameter.	N/A	Sample Name, you are on the end space. Congratulations! You are now retired.	Sample Name, you are on the end space. Congratulations! You are now retired.	P

13. OrangeSpace

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
drawActionCard	1	The drawActionCard() method of space1 is called.	N/A (no drawn action card)	A drawn action card	ACTION CARD Name: Write a Book Amount Collected from the Bank: 8000	P
getInfo	1	The getInfo() method of space1 is called with p1 passed as the parameter.	N/A	Sample Name, you are on an orange space. Draw an action card.	Sample Name, you are on an orange space. Draw an action card.	P

14. StartSpace

Method Name	#	Test Case Description	Sample Input/ Previous Object Status	Expected Result/Updated Object Status	Actual Result/Current Object Status	P/F
execute	1	The execute() method of space1 is called, and p1 chooses the Career Path.	Current next index = 1	New next index = 1	New next index = 1	P
	2	The execute() method of space1 is called, and p1 chooses the College Path.	Current next index = 1	New next index = 5	New next index = 5	P
getInfo	1	The getInfo() method of space1 is called with p1 passed as the parameter.	N/A	Sample Name, you are on the start space.	Sample Name, you are on the start space.	P

				Choose between the Career and College Paths.	Choose between the Career and College Paths.	
--	--	--	--	--	--	--

F. Appendices

B-1-01

ACTION CARD

Name: Setup School

Amount Collected from Bank: 120000

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Tax Refund

Amount Collected from Bank: Player tax

ACTION CARD

Name: Tax Refund

Amount Collected from Bank: Player tax

ACTION CARD

Name: Setup School

Amount Collected from Bank: 90000

ACTION CARD

Name: Setup School

Amount Collected from Bank: 140000

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 9000

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Sell an Item

Amount Collected from Bank: 7000

ACTION CARD

Name: Setup School

Amount Collected from Bank: 50000

ACTION CARD

Name: Setup School

Amount Collected from Bank: 120000

ACTION CARD

Name: Sell an Item

Amount Collected from Bank: 8000

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 6000

ACTION CARD

Name: Setup School

Amount Collected from Bank: 150000

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 6000

ACTION CARD

Name: Tax Refund

Amount Collected from Bank: Player tax

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 5000

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 11000

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 3000

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 2000

ACTION CARD

Name: Win a Competition

Amount Paid to Bank: 5000

ACTION CARD

Name: Win a Competition

Amount Paid to Bank: 5000

ACTION CARD

Name: Visit a Place

Amount Paid to Bank: 60000

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 10000

ACTION CARD

Name: Hiking

Amount Paid to Bank: 40000

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 8000

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 2000

ACTION CARD

Name: Watch a Show

Amount Paid to Bank: 3000

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 1000

ACTION CARD

Name: Hiking

Amount Paid to Bank: 40000

ACTION CARD

Name: Visit a Place

Amount Paid to Bank: 50000

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 6000

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 4000

ACTION CARD

Name: Win a Competition

Amount Paid to Bank: 1000

ACTION CARD

Name: Watch a Show

Amount Paid to Bank: 3000

ACTION CARD

Name: Watch a Show

Amount Paid to Bank: 6000

ACTION CARD

Name: Visit a Place

Amount Paid to Bank: 70000

ACTION CARD

Name: Visit a Place

Amount Paid to Bank: 30000

ACTION CARD

Name: Christmas Bonus

Amount Paid to Other Player: 16000 (to every player)

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 90000 (to chosen player)

ACTION CARD

Name: Christmas Bonus

Amount Paid to Other Player: 17000 (to every player)

ACTION CARD

Name: Christmas Bonus

Amount Paid to Other Player: 16000 (to every player)

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 100000 (to chosen player)

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 12000 (from every player)

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 17000 (from every player)

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 6000 (from every player)

ACTION CARD

Name: File a Lawsuit

Amount Collected from Other Player: 60000 (from chosen player)

ACTION CARD

Name: File a Lawsuit

Amount Collected from Other Player: 140000 (from chosen player)

B-1-02

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Setup School

Amount Collected from Bank: 90000

ACTION CARD

Name: Setup School

Amount Collected from Bank: 90000

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Setup School

Amount Collected from Bank: 50000

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 18000

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Sell an Item

Amount Collected from Bank: 6000

ACTION CARD

Name: Tax Refund

Amount Collected from Bank: Player tax

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 15000

ACTION CARD

Name: Tax Refund

Amount Collected from Bank: Player tax

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 5000

ACTION CARD

Name: Setup School

Amount Collected from Bank: 60000

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 11000

ACTION CARD

Name: Sell an Item

Amount Collected from Bank: 5000

ACTION CARD

Name: Tax Refund

Amount Collected from Bank: Player tax

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Visit a Place

Amount Paid to Bank: 80000

ACTION CARD

Name: Hiking

Amount Paid to Bank: 40000

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 10000

ACTION CARD

Name: Visit a Place

Amount Paid to Bank: 20000

ACTION CARD

Name: Hiking

Amount Paid to Bank: 50000

ACTION CARD

Name: Watch a Show

Amount Paid to Bank: 5000

ACTION CARD

Name: Win a Competition

Amount Paid to Bank: 3000

ACTION CARD

Name: Win a Competition

Amount Paid to Bank: 2000

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 1000

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 10000

ACTION CARD

Name: Hiking

Amount Paid to Bank: 50000

ACTION CARD

Name: Visit a Place

Amount Paid to Bank: 80000

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 1000

ACTION CARD

Name: Watch a Show

Amount Paid to Bank: 1000

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 2000

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 4000

ACTION CARD

Name: Win a Competition

Amount Paid to Bank: 4000

ACTION CARD

Name: Win a Competition

Amount Paid to Bank: 4000

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 8000

ACTION CARD

Name: Visit a Place

Amount Paid to Bank: 50000

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 150000 (to chosen player)

ACTION CARD

Name: Christmas Bonus

Amount Paid to Other Player: 6000 (to every player)

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 70000 (to chosen player)

ACTION CARD

Name: Christmas Bonus

Amount Paid to Other Player: 17000 (to every player)

ACTION CARD

Name: Christmas Bonus

Amount Paid to Other Player: 7000 (to every player)

ACTION CARD

Name: File a Lawsuit

Amount Collected from Other Player: 80000 (from chosen player)

ACTION CARD

Name: File a Lawsuit

Amount Collected from Other Player: 70000 (from chosen player)

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 11000 (from every player)

ACTION CARD

Name: File a Lawsuit

Amount Collected from Other Player: 80000 (from chosen player)

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 18000 (from every player)

B-1-03

ACTION CARD

Name: Sell an Item

Amount Collected from Bank: 4000

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 9000

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 12000

ACTION CARD

Name: Setup School

Amount Collected from Bank: 90000

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Tax Refund

Amount Collected from Bank: Player tax

ACTION CARD

Name: Setup School

Amount Collected from Bank: 120000

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 5000

ACTION CARD

Name: Tax Refund

Amount Collected from Bank: Player tax

ACTION CARD

Name: Setup School

Amount Collected from Bank: 110000

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Setup School

Amount Collected from Bank: 130000

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Sell an Item

Amount Collected from Bank: 8000

ACTION CARD

Name: Tax Refund

Amount Collected from Bank: Player tax

ACTION CARD

Name: Sell an Item

Amount Collected from Bank: 3000

ACTION CARD

Name: Tax Refund

Amount Collected from Bank: Player tax

ACTION CARD

Name: Setup School

Amount Collected from Bank: 120000

ACTION CARD

Name: Sell an Item

Amount Collected from Bank: 9000

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 5000

ACTION CARD

Name: Win a Competition

Amount Paid to Bank: 5000

ACTION CARD

Name: Win a Competition

Amount Paid to Bank: 1000

ACTION CARD

Name: Watch a Show

Amount Paid to Bank: 1000

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 1000

ACTION CARD

Name: Win a Competition

Amount Paid to Bank: 2000

ACTION CARD

Name: Visit a Place

Amount Paid to Bank: 20000

ACTION CARD

Name: Watch a Show

Amount Paid to Bank: 7000

ACTION CARD

Name: Win a Competition

Amount Paid to Bank: 2000

ACTION CARD

Name: Visit a Place

Amount Paid to Bank: 40000

ACTION CARD

Name: Visit a Place

Amount Paid to Bank: 60000

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 1000

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 5000

ACTION CARD

Name: Hiking

Amount Paid to Bank: 50000

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 2000

ACTION CARD

Name: Hiking

Amount Paid to Bank: 50000

ACTION CARD

Name: Win a Competition

Amount Paid to Bank: 5000

ACTION CARD

Name: Hiking

Amount Paid to Bank: 60000

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 3000

ACTION CARD

Name: Watch a Show

Amount Paid to Bank: 4000

ACTION CARD

Name: Christmas Bonus

Amount Paid to Other Player: 5000 (to every player)

ACTION CARD

Name: Christmas Bonus

Amount Paid to Other Player: 19000 (to every player)

ACTION CARD

Name: Christmas Bonus

Amount Paid to Other Player: 15000 (to every player)

ACTION CARD

Name: Christmas Bonus

Amount Paid to Other Player: 7000 (to every player)

ACTION CARD

Name: Christmas Bonus

Amount Paid to Other Player: 5000 (to every player)

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 16000 (from every player)

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 13000 (from every player)

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 11000 (from every player)

ACTION CARD

Name: File a Lawsuit

Amount Collected from Other Player: 120000 (from chosen player)

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 8000 (from every player)

B-1-04

Action Card Deck

Num. of Cards (Original): 50

Num. of Cards (Current): 49

ACTION CARD

Name: Hiking

Amount Paid to Bank: 20000

ACTION CARD

Name: Visit a Place

Amount Paid to Bank: 20000

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 9000

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 2000

ACTION CARD

Name: Tax Refund

Amount Collected from Bank: Player tax

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 4000

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 2000

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 100000 (to chosen player)

ACTION CARD

Name: Tax Refund

Amount Collected from Bank: Player tax

ACTION CARD

Name: Sell an Item

Amount Collected from Bank: 3000

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 15000 (from every player)

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 2000

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 15000

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 18000

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 7000

ACTION CARD

Name: Hiking

Amount Paid to Bank: 40000

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 17000

ACTION CARD

Name: Setup School

Amount Collected from Bank: 80000

ACTION CARD

Name: Sell an Item

Amount Collected from Bank: 10000

ACTION CARD

Name: Setup School

Amount Collected from Bank: 150000

ACTION CARD

Name: Win a Competition

Amount Paid to Bank: 5000

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 4000

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 15000

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 150000 (to chosen player)

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: File a Lawsuit

Amount Collected from Other Player: 140000 (from chosen player)

ACTION CARD

Name: Setup School

Amount Collected from Bank: 60000

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 10000 (from every player)

ACTION CARD

Name: Win a Competition

Amount Paid to Bank: 2000

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 130000 (to chosen player)

ACTION CARD

Name: Sell an Item

Amount Collected from Bank: 7000

ACTION CARD

Name: Visit a Place

Amount Paid to Bank: 40000

ACTION CARD

Name: Setup School

Amount Collected from Bank: 140000

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Hiking

Amount Paid to Bank: 70000

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 90000 (to chosen player)

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 4000

ACTION CARD

Name: Hiking

Amount Paid to Bank: 40000

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 4000

ACTION CARD

Name: Visit a Place

Amount Paid to Bank: 50000

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 8000

ACTION CARD

Name: File a Lawsuit

Amount Collected from Other Player: 140000 (from chosen player)

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 3000

ACTION CARD

Name: Win a Competition

Amount Paid to Bank: 2000

ACTION CARD

Name: Sell an Item

Amount Collected from Bank: 5000

ACTION CARD

Name: File a Lawsuit

Amount Collected from Other Player: 110000 (from chosen player)

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 2000

Discard Pile

Num. of Cards (Current): 1

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 100000 (to chosen player)

B-1-05

Action Card Deck

Num. of Cards (Original): 50

Num. of Cards (Current): 0

Discard Pile

Num. of Cards (Current): 50

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 2000

ACTION CARD

Name: File a Lawsuit

Amount Collected from Other Player: 110000 (from chosen player)

ACTION CARD

Name: Sell an Item

Amount Collected from Bank: 5000

ACTION CARD

Name: Win a Competition

Amount Paid to Bank: 2000

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 3000

ACTION CARD

Name: File a Lawsuit

Amount Collected from Other Player: 140000 (from chosen player)

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 8000

ACTION CARD

Name: Visit a Place

Amount Paid to Bank: 50000

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 4000

ACTION CARD

Name: Hiking

Amount Paid to Bank: 40000

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 4000

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 90000 (to chosen player)

ACTION CARD

Name: Hiking

Amount Paid to Bank: 70000

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Setup School

Amount Collected from Bank: 140000

ACTION CARD

Name: Visit a Place

Amount Paid to Bank: 40000

ACTION CARD

Name: Sell an Item

Amount Collected from Bank: 7000

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 130000 (to chosen player)

ACTION CARD

Name: Win a Competition

Amount Paid to Bank: 2000

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 10000 (from every player)

ACTION CARD

Name: Setup School

Amount Collected from Bank: 60000

ACTION CARD

Name: File a Lawsuit

Amount Collected from Other Player: 140000 (from chosen player)

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 150000 (to chosen player)

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 15000

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 4000

ACTION CARD

Name: Win a Competition

Amount Paid to Bank: 5000

ACTION CARD

Name: Setup School

Amount Collected from Bank: 150000

ACTION CARD

Name: Sell an Item

Amount Collected from Bank: 10000

ACTION CARD

Name: Setup School

Amount Collected from Bank: 80000

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 17000

ACTION CARD

Name: Hiking

Amount Paid to Bank: 40000

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 7000

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 18000

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 15000

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 2000

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 15000 (from every player)

ACTION CARD

Name: Sell an Item

Amount Collected from Bank: 3000

ACTION CARD

Name: Tax Refund

Amount Collected from Bank: Player tax

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 100000 (to chosen player)

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 2000

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 4000

ACTION CARD

Name: Tax Refund

Amount Collected from Bank: Player tax

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 2000

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 9000

ACTION CARD

Name: Visit a Place

Amount Paid to Bank: 20000

ACTION CARD

Name: Hiking

Amount Paid to Bank: 20000

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 100000 (to chosen player)

B-1-06

Action Card Deck

Num. of Cards (Original): 50

Num. of Cards (Current): 49

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 2000

ACTION CARD

Name: Win a Competition

Amount Paid to Bank: 2000

ACTION CARD

Name: Visit a Place

Amount Paid to Bank: 40000

ACTION CARD

Name: Visit a Place

Amount Paid to Bank: 20000

ACTION CARD

Name: Setup School

Amount Collected from Bank: 60000

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 4000

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 100000 (to chosen player)

ACTION CARD

Name: Tax Refund

Amount Collected from Bank: Player tax

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 15000

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 2000

ACTION CARD

Name: Sell an Item

Amount Collected from Bank: 5000

ACTION CARD

Name: Win a Competition

Amount Paid to Bank: 2000

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 100000 (to chosen player)

ACTION CARD

Name: Sell an Item

Amount Collected from Bank: 10000

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 10000 (from every player)

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 9000

ACTION CARD

Name: Setup School

Amount Collected from Bank: 140000

ACTION CARD

Name: Sell an Item

Amount Collected from Bank: 3000

ACTION CARD

Name: Visit a Place

Amount Paid to Bank: 50000

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 18000

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 3000

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 2000

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 4000

ACTION CARD

Name: Tax Refund

Amount Collected from Bank: Player tax

ACTION CARD

Name: File a Lawsuit

Amount Collected from Other Player: 140000 (from chosen player)

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 17000

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 8000

ACTION CARD

Name: Setup School

Amount Collected from Bank: 80000

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 130000 (to chosen player)

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 7000

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 150000 (to chosen player)

ACTION CARD

Name: Hiking

Amount Paid to Bank: 40000

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Traffic Violation

Amount Paid to Bank: 4000

ACTION CARD

Name: Write a Book

Amount Collected from Bank: 15000

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 4000

ACTION CARD

Name: Hiking

Amount Paid to Bank: 20000

ACTION CARD

Name: File a Lawsuit

Amount Collected from Other Player: 110000 (from chosen player)

ACTION CARD

Name: Hiking

Amount Paid to Bank: 70000

ACTION CARD

Name: Buy an Item

Amount Paid to Bank: 2000

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 15000 (from every player)

ACTION CARD

Name: File a Lawsuit

Amount Collected from Other Player: 140000 (from chosen player)

ACTION CARD

Name: Setup School

Amount Collected from Bank: 150000

ACTION CARD

Name: Win a Competition

Amount Paid to Bank: 5000

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 90000 (to chosen player)

ACTION CARD

Name: Hiking

Amount Paid to Bank: 40000

ACTION CARD

Name: Sell an Item

Amount Collected from Bank: 7000

Discard Pile

Num. of Cards (Current): 1

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

B-2-01

CAREER CARD

Is Degree Required: true

Max Pay Raise Range: [5, 8]

Max Pay Raise: 6

Career: Lawyer

CAREER CARD

Is Degree Required: true

Max Pay Raise Range: [4, 7]

Max Pay Raise: 5

Career: Accountant

CAREER CARD

Is Degree Required: true

Max Pay Raise Range: [3, 7]

Max Pay Raise: 7

Career: Computer Consultant

CAREER CARD

Is Degree Required: true

Max Pay Raise Range: [5, 8]

Max Pay Raise: 5

Career: Doctor

CAREER CARD

Is Degree Required: false

Max Pay Raise Range: [1, 4]

Max Pay Raise: 3

Career: Server

CAREER CARD

Is Degree Required: false

Max Pay Raise Range: [2, 8]

Max Pay Raise: 6

Career: Racecar Driver

CAREER CARD

Is Degree Required: false

Max Pay Raise Range: [1, 6]

Max Pay Raise: 5

Career: Athlete

B-2-02

CAREER CARD

Is Degree Required: true

Max Pay Raise Range: [5, 8]

Max Pay Raise: 5

Career: Lawyer

CAREER CARD

Is Degree Required: true

Max Pay Raise Range: [4, 7]

Max Pay Raise: 5

Career: Accountant

CAREER CARD

Is Degree Required: true

Max Pay Raise Range: [3, 7]

Max Pay Raise: 4

Career: Computer Consultant

CAREER CARD

Is Degree Required: true

Max Pay Raise Range: [5, 8]

Max Pay Raise: 8

Career: Doctor

CAREER CARD

Is Degree Required: false

Max Pay Raise Range: [1, 4]

Max Pay Raise: 1

Career: Server

CAREER CARD

Is Degree Required: false

Max Pay Raise Range: [2, 8]

Max Pay Raise: 3

Career: Racecar Driver

CAREER CARD

Is Degree Required: false

Max Pay Raise Range: [1, 6]

Max Pay Raise: 5

Career: Athlete

B-2-03

CAREER CARD

Is Degree Required: true

Max Pay Raise Range: [5, 8]

Max Pay Raise: 8

Career: Lawyer

CAREER CARD

Is Degree Required: true

Max Pay Raise Range: [4, 7]

Max Pay Raise: 5

Career: Accountant

CAREER CARD

Is Degree Required: true

Max Pay Raise Range: [3, 7]

Max Pay Raise: 4

Career: Computer Consultant

CAREER CARD

Is Degree Required: true

Max Pay Raise Range: [5, 8]

Max Pay Raise: 7

Career: Doctor

CAREER CARD

Is Degree Required: false

Max Pay Raise Range: [1, 4]

Max Pay Raise: 3

Career: Server

CAREER CARD

Is Degree Required: false

Max Pay Raise Range: [2, 8]

Max Pay Raise: 7

Career: Racecar Driver

CAREER CARD

Is Degree Required: false

Max Pay Raise Range: [1, 6]

Max Pay Raise: 1

Career: Athlete

B-2-04

Career Card Deck

Num. of Cards (Original): 7

Num. of Cards (Current): 6

CAREER CARD

Is Degree Required: false

Max Pay Raise Range: [1, 4]

Max Pay Raise: 4

Career: Server

CAREER CARD

Is Degree Required: true

Max Pay Raise Range: [5, 8]

Max Pay Raise: 6

Career: Doctor

CAREER CARD

Is Degree Required: false

Max Pay Raise Range: [1, 6]

Max Pay Raise: 2

Career: Athlete

CAREER CARD

Is Degree Required: true

Max Pay Raise Range: [3, 7]

Max Pay Raise: 3

Career: Computer Consultant

CAREER CARD

Is Degree Required: false

Max Pay Raise Range: [2, 8]

Max Pay Raise: 2

Career: Racecar Driver

CAREER CARD

Is Degree Required: true

Max Pay Raise Range: [4, 7]

Max Pay Raise: 4

Career: Accountant

B-3-01

SALARY CARD

Salary: 50000

Tax Due: 5000

Pay Raise: 10000

SALARY CARD

Salary: 10000

Tax Due: 1000

Pay Raise: 2000

SALARY CARD

Salary: 80000

Tax Due: 8000

Pay Raise: 16000

SALARY CARD

Salary: 70000

Tax Due: 7000

Pay Raise: 14000

SALARY CARD

Salary: 40000

Tax Due: 4000

Pay Raise: 8000

SALARY CARD

Salary: 60000

Tax Due: 6000

Pay Raise: 12000

SALARY CARD

Salary: 20000

Tax Due: 2000

Pay Raise: 4000

SALARY CARD

Salary: 90000

Tax Due: 9000

Pay Raise: 18000

SALARY CARD

Salary: 30000
Tax Due: 3000
Pay Raise: 6000

SALARY CARD
Salary: 100000
Tax Due: 10000
Pay Raise: 20000

B-3-02

SALARY CARD
Salary: 80000
Tax Due: 8000
Pay Raise: 16000

SALARY CARD
Salary: 20000
Tax Due: 2000
Pay Raise: 4000

SALARY CARD
Salary: 30000
Tax Due: 3000
Pay Raise: 6000

SALARY CARD
Salary: 100000
Tax Due: 10000
Pay Raise: 20000

SALARY CARD

Salary: 10000

Tax Due: 1000

Pay Raise: 2000

SALARY CARD

Salary: 60000

Tax Due: 6000

Pay Raise: 12000

SALARY CARD

Salary: 90000

Tax Due: 9000

Pay Raise: 18000

SALARY CARD

Salary: 70000

Tax Due: 7000

Pay Raise: 14000

SALARY CARD

Salary: 40000

Tax Due: 4000

Pay Raise: 8000

SALARY CARD

Salary: 50000

Tax Due: 5000

Pay Raise: 10000

B-3-03

SALARY CARD

Salary: 10000

Tax Due: 1000

Pay Raise: 2000

SALARY CARD

Salary: 90000

Tax Due: 9000

Pay Raise: 18000

SALARY CARD

Salary: 20000

Tax Due: 2000

Pay Raise: 4000

SALARY CARD

Salary: 80000

Tax Due: 8000

Pay Raise: 16000

SALARY CARD

Salary: 100000

Tax Due: 10000

Pay Raise: 20000

SALARY CARD

Salary: 60000

Tax Due: 6000

Pay Raise: 12000

SALARY CARD

Salary: 30000

Tax Due: 3000

Pay Raise: 6000

SALARY CARD

Salary: 70000

Tax Due: 7000

Pay Raise: 14000

SALARY CARD

Salary: 50000

Tax Due: 5000

Pay Raise: 10000

SALARY CARD

Salary: 40000

Tax Due: 4000

Pay Raise: 8000

B-3-04

Salary Card Deck

Num. of Cards (Original): 10

Num. of Cards (Current): 9

SALARY CARD

Salary: 70000

Tax Due: 7000

Pay Raise: 14000

SALARY CARD

Salary: 20000

Tax Due: 2000

Pay Raise: 4000

SALARY CARD

Salary: 80000

Tax Due: 8000

Pay Raise: 16000

SALARY CARD

Salary: 40000

Tax Due: 4000

Pay Raise: 8000

SALARY CARD

Salary: 50000

Tax Due: 5000

Pay Raise: 10000

SALARY CARD

Salary: 10000

Tax Due: 1000

Pay Raise: 2000

SALARY CARD

Salary: 60000

Tax Due: 6000

Pay Raise: 12000

SALARY CARD

Salary: 30000

Tax Due: 3000

Pay Raise: 6000

SALARY CARD

Salary: 100000

Tax Due: 10000

Pay Raise: 20000

B-4-01

[BLUE CARD

Career Match: Lawyer

Name: Lawsuit (Blue Card)

Received if Career Match: 15000

Paid to Other Player (or Bank): 140000

, BLUE CARD

Career Match: Accountant

Name: Lawsuit (Blue Card)

Received if Career Match: 15000

Paid to Other Player (or Bank): Tax Due

, BLUE CARD

Career Match: Server

Name: Tip the Server

Received if Career Match: 15000

Paid to Other Player (or Bank): Randomly spun number * 1000

, BLUE CARD

Career Match: Doctor

Name: Ski Accident

Received if Career Match: 15000
Paid to Other Player (or Bank): 10000
, BLUE CARD
Career Match: Computer Consultant
Name: Computer Repair
Received if Career Match: 15000
Paid to Other Player (or Bank): 10000 (if odd) 5000 (if even)
, BLUE CARD
Career Match: Athlete
Name: World Cup
Received if Career Match: 15000
Paid to Other Player (or Bank): Number of players * 5000
, BLUE CARD
Career Match: Racecar Driver
Name: F1 Race
Received if Career Match: 15000
Paid to Other Player (or Bank): 0.1 of salary
]

B-4-02

Drawn card:
BLUE CARD
Career Match: Racecar Driver
Name: F1 Race
Received if Career Match: 15000
Paid to Other Player (or Bank): 0.1 of salary

Current deck order:
Blue Card Deck
Num. of Cards (Original): 7

Num. of Cards (Current): 7

BLUE CARD

Career Match: Athlete

Name: World Cup

Received if Career Match: 15000

Paid to Other Player (or Bank): Number of players * 5000

BLUE CARD

Career Match: Lawyer

Name: Lawsuit (Blue Card)

Received if Career Match: 15000

Paid to Other Player (or Bank): 100000

BLUE CARD

Career Match: Racecar Driver

Name: F1 Race

Received if Career Match: 15000

Paid to Other Player (or Bank): 0.1 of salary

BLUE CARD

Career Match: Doctor

Name: Ski Accident

Received if Career Match: 15000

Paid to Other Player (or Bank): 10000

BLUE CARD

Career Match: Server

Name: Tip the Server

Received if Career Match: 15000

Paid to Other Player (or Bank): Randomly spun number * 1000

BLUE CARD

Career Match: Accountant

Name: Lawsuit (Blue Card)

Received if Career Match: 15000

Paid to Other Player (or Bank): Tax Due

BLUE CARD

Career Match: Computer Consultant

Name: Computer Repair

Received if Career Match: 15000

Paid to Other Player (or Bank): 10000 (if odd) 5000 (if even)

B-4-03

Drawn card:

BLUE CARD

Career Match: Athlete

Name: World Cup

Received if Career Match: 15000

Paid to Other Player (or Bank): Number of players * 5000

Current deck order:

Blue Card Deck

Num. of Cards (Original): 7

Num. of Cards (Current): 7

BLUE CARD

Career Match: Doctor

Name: Ski Accident

Received if Career Match: 15000

Paid to Other Player (or Bank): 10000

BLUE CARD

Career Match: Computer Consultant

Name: Computer Repair

Received if Career Match: 15000

Paid to Other Player (or Bank): 10000 (if odd) 5000 (if even)

BLUE CARD

Career Match: Athlete

Name: World Cup

Received if Career Match: 15000

Paid to Other Player (or Bank): Number of players * 5000

BLUE CARD

Career Match: Racecar Driver

Name: F1 Race

Received if Career Match: 15000

Paid to Other Player (or Bank): 0.1 of salary

BLUE CARD

Career Match: Server

Name: Tip the Server

Received if Career Match: 15000

Paid to Other Player (or Bank): Randomly spun number * 1000

BLUE CARD

Career Match: Lawyer

Name: Lawsuit (Blue Card)

Received if Career Match: 15000

Paid to Other Player (or Bank): 100000

BLUE CARD

Career Match: Accountant

Name: Lawsuit (Blue Card)

Received if Career Match: 15000

Paid to Other Player (or Bank): Tax Due

B-4-04

Drawn card:

BLUE CARD

Career Match: Doctor

Name: Ski Accident

Received if Career Match: 15000

Paid to Other Player (or Bank): 10000

Current deck order:

Blue Card Deck

Num. of Cards (Original): 7

Num. of Cards (Current): 7

BLUE CARD

Career Match: Doctor

Name: Ski Accident

Received if Career Match: 15000

Paid to Other Player (or Bank): 10000

BLUE CARD

Career Match: Server

Name: Tip the Server

Received if Career Match: 15000

Paid to Other Player (or Bank): Randomly spun number * 1000

BLUE CARD

Career Match: Computer Consultant

Name: Computer Repair

Received if Career Match: 15000

Paid to Other Player (or Bank): 10000 (if odd) 5000 (if even)

BLUE CARD

Career Match: Racecar Driver

Name: F1 Race

Received if Career Match: 15000

Paid to Other Player (or Bank): 0.1 of salary

BLUE CARD

Career Match: Athlete

Name: World Cup

Received if Career Match: 15000

Paid to Other Player (or Bank): Number of players * 5000

BLUE CARD

Career Match: Lawyer

Name: Lawsuit (Blue Card)

Received if Career Match: 15000

Paid to Other Player (or Bank): 100000

BLUE CARD

Career Match: Accountant

Name: Lawsuit (Blue Card)

Received if Career Match: 15000

Paid to Other Player (or Bank): Tax Due

B-5-01

[HOUSE CARD

Name: Florence Penthouse

Buying Price: 107000

Selling Price 85600

, HOUSE CARD

Name: Swing Penthouse

Buying Price: 135000

Selling Price 108000

, HOUSE CARD

Name: Hai Lin Villa

Buying Price: 114000

Selling Price 91200

, HOUSE CARD

Name: Havsis Villa

Buying Price: 133000

Selling Price 106400

, HOUSE CARD

Name: Edwardian Mansion

Buying Price: 75000

Selling Price 60000

, HOUSE CARD

Name: Jacobian Mansion

Buying Price: 98000

Selling Price 78400

, HOUSE CARD

Name: Cece Townhouse

Buying Price: 80000

Selling Price 64000

, HOUSE CARD

Name: Purogh Townhome

Buying Price: 122000

Selling Price 97600

, HOUSE CARD

Name: Java Farmhouse

Buying Price: 146000

Selling Price 116800

, HOUSE CARD

Name: Bali Farmhouse

Buying Price: 101000

Selling Price 80800

]

B-5-02

Drawn card:

HOUSE CARD

Name: Cece Townhouse

Buying Price: 106000

Selling Price 84800

Remaining cards:

House Card Deck

Num. of Cards (Original): 10

Num. of Cards (Current): 9

HOUSE CARD

Name: Hai Lin Villa

Buying Price: 90000

Selling Price 72000

HOUSE CARD

Name: Florence Penthouse

Buying Price: 120000

Selling Price 96000

HOUSE CARD

Name: Java Farmhouse

Buying Price: 140000

Selling Price 112000

HOUSE CARD

Name: Bali Farmhouse

Buying Price: 89000

Selling Price 71200

HOUSE CARD

Name: Havsis Villa

Buying Price: 135000

Selling Price 108000

HOUSE CARD

Name: Jacobian Mansion

Buying Price: 128000

Selling Price 102400

HOUSE CARD

Name: Swing Penthouse

Buying Price: 79000

Selling Price 63200

HOUSE CARD

Name: Edwardian Mansion
Buying Price: 148000
Selling Price 118400

HOUSE CARD
Name: Purogh Townhome
Buying Price: 113000
Selling Price 90400

B-5-03

Drawn card:
HOUSE CARD
Name: Edwardian Mansion
Buying Price: 148000
Selling Price 118400

Remaining cards:
House Card Deck
Num. of Cards (Original): 10
Num. of Cards (Current): 8

HOUSE CARD
Name: Hai Lin Villa
Buying Price: 90000
Selling Price 72000

HOUSE CARD
Name: Florence Penthouse
Buying Price: 120000
Selling Price 96000

HOUSE CARD

Name: Java Farmhouse

Buying Price: 140000

Selling Price 112000

HOUSE CARD

Name: Bali Farmhouse

Buying Price: 89000

Selling Price 71200

HOUSE CARD

Name: Havsis Villa

Buying Price: 135000

Selling Price 108000

HOUSE CARD

Name: Jacobian Mansion

Buying Price: 128000

Selling Price 102400

HOUSE CARD

Name: Swing Penthouse

Buying Price: 79000

Selling Price 63200

HOUSE CARD

Name: Purogh Townhome

Buying Price: 113000

Selling Price 90400

B-5-04

Drawn card:

HOUSE CARD

Name: Jacobian Mansion

Buying Price: 128000

Selling Price 102400

Remaining cards:

House Card Deck

Num. of Cards (Original): 10

Num. of Cards (Current): 7

HOUSE CARD

Name: Hai Lin Villa

Buying Price: 90000

Selling Price 72000

HOUSE CARD

Name: Florence Penthouse

Buying Price: 120000

Selling Price 96000

HOUSE CARD

Name: Java Farmhouse

Buying Price: 140000

Selling Price 112000

HOUSE CARD

Name: Bali Farmhouse

Buying Price: 89000

Selling Price 71200

HOUSE CARD

Name: Havsis Villa

Buying Price: 135000

Selling Price 108000

HOUSE CARD

Name: Swing Penthouse

Buying Price: 79000

Selling Price 63200

HOUSE CARD

Name: Purogh Townhome

Buying Price: 113000

Selling Price 90400

C-2-01

Action Card Discard Pile

Num. of Cards (Current): 5

ACTION CARD

Name: Bonus Pay Day

Amount Collected from Bank: Player salary

ACTION CARD

Name: Christmas Bonus

Amount Paid to Other Player: 12000 (to every player)

ACTION CARD

Name: Buy an Item
Amount Paid to Bank: 2000

ACTION CARD

Name: File a Lawsuit
Amount Collected from Other Player: 90000 (from chosen player)

ACTION CARD

Name: Bonus Pay Day
Amount Collected from Bank: Player salary

C-4-01

CAREER CARD
Is Degree Required: true
Max Number of Pay Raises Range: [3, 7]
Max Number of Pay Raises: 7
Career: Computer Consultant

Career Card Deck
Num. of Cards (Original): 7
Num. of Cards (Current): 6

CAREER CARD
Is Degree Required: true
Max Number of Pay Raises Range: [5, 8]
Max Number of Pay Raises: 7
Career: Lawyer

CAREER CARD
Is Degree Required: false

Max Number of Pay Raises Range: [2, 8]

Max Number of Pay Raises: 3

Career: Racecar Driver

CAREER CARD

Is Degree Required: false

Max Number of Pay Raises Range: [1, 4]

Max Number of Pay Raises: 4

Career: Server

CAREER CARD

Is Degree Required: false

Max Number of Pay Raises Range: [1, 6]

Max Number of Pay Raises: 3

Career: Athlete

CAREER CARD

Is Degree Required: true

Max Number of Pay Raises Range: [4, 7]

Max Number of Pay Raises: 4

Career: Accountant

CAREER CARD

Is Degree Required: true

Max Number of Pay Raises Range: [5, 8]

Max Number of Pay Raises: 7

Career: Doctor

C-4-02

CAREER CARD

Is Degree Required: true

Max Number of Pay Raises Range: [5, 8]

Max Number of Pay Raises: 7

Career: Lawyer

Career Card Deck

Num. of Cards (Original): 7

Num. of Cards (Current): 5

CAREER CARD

Is Degree Required: false

Max Number of Pay Raises Range: [2, 8]

Max Number of Pay Raises: 3

Career: Racecar Driver

CAREER CARD

Is Degree Required: false

Max Number of Pay Raises Range: [1, 4]

Max Number of Pay Raises: 4

Career: Server

CAREER CARD

Is Degree Required: false

Max Number of Pay Raises Range: [1, 6]

Max Number of Pay Raises: 3

Career: Athlete

CAREER CARD

Is Degree Required: true

Max Number of Pay Raises Range: [4, 7]

Max Number of Pay Raises: 4

Career: Accountant

CAREER CARD

Is Degree Required: true

Max Number of Pay Raises Range: [5, 8]

Max Number of Pay Raises: 7

Career: Doctor

C-4-03

CAREER CARD

Is Degree Required: false

Max Number of Pay Raises Range: [2, 8]

Max Number of Pay Raises: 3

Career: Racecar Driver

Career Card Deck

Num. of Cards (Original): 7

Num. of Cards (Current): 4

CAREER CARD

Is Degree Required: false

Max Number of Pay Raises Range: [1, 4]

Max Number of Pay Raises: 4

Career: Server

CAREER CARD

Is Degree Required: false

Max Number of Pay Raises Range: [1, 6]

Max Number of Pay Raises: 3

Career: Athlete

CAREER CARD

Is Degree Required: true

Max Number of Pay Raises Range: [4, 7]

Max Number of Pay Raises: 4

Career: Accountant

CAREER CARD

Is Degree Required: true

Max Number of Pay Raises Range: [5, 8]

Max Number of Pay Raises: 7

Career: Doctor

C-4-04

SALARY CARD

Salary: 40000

Tax Due: 4000

Pay Raise: 8000

Salary Card Deck

Num. of Cards (Original): 10

Num. of Cards (Current): 9

SALARY CARD

Salary: 100000

Tax Due: 10000

Pay Raise: 20000

SALARY CARD

Salary: 90000

Tax Due: 9000
Pay Raise: 18000

SALARY CARD
Salary: 80000
Tax Due: 8000
Pay Raise: 16000

SALARY CARD
Salary: 60000
Tax Due: 6000
Pay Raise: 12000

SALARY CARD
Salary: 70000
Tax Due: 7000
Pay Raise: 14000

SALARY CARD
Salary: 50000
Tax Due: 5000
Pay Raise: 10000

SALARY CARD
Salary: 20000
Tax Due: 2000
Pay Raise: 4000

SALARY CARD
Salary: 10000
Tax Due: 1000

Pay Raise: 2000

SALARY CARD

Salary: 30000

Tax Due: 3000

Pay Raise: 6000

C-4-05

SALARY CARD

Salary: 100000

Tax Due: 10000

Pay Raise: 20000

Salary Card Deck

Num. of Cards (Original): 10

Num. of Cards (Current): 8

SALARY CARD

Salary: 90000

Tax Due: 9000

Pay Raise: 18000

SALARY CARD

Salary: 80000

Tax Due: 8000

Pay Raise: 16000

SALARY CARD

Salary: 60000

Tax Due: 6000

Pay Raise: 12000

SALARY CARD

Salary: 70000

Tax Due: 7000

Pay Raise: 14000

SALARY CARD

Salary: 50000

Tax Due: 5000

Pay Raise: 10000

SALARY CARD

Salary: 20000

Tax Due: 2000

Pay Raise: 4000

SALARY CARD

Salary: 10000

Tax Due: 1000

Pay Raise: 2000

SALARY CARD

Salary: 30000

Tax Due: 3000

Pay Raise: 6000

C-4-06

SALARY CARD

Salary: 90000

Tax Due: 9000

Pay Raise: 18000

Salary Card Deck

Num. of Cards (Original): 10

Num. of Cards (Current): 7

SALARY CARD

Salary: 80000

Tax Due: 8000

Pay Raise: 16000

SALARY CARD

Salary: 60000

Tax Due: 6000

Pay Raise: 12000

SALARY CARD

Salary: 70000

Tax Due: 7000

Pay Raise: 14000

SALARY CARD

Salary: 50000

Tax Due: 5000

Pay Raise: 10000

SALARY CARD

Salary: 20000

Tax Due: 2000

Pay Raise: 4000

SALARY CARD

Salary: 10000

Tax Due: 1000

Pay Raise: 2000

SALARY CARD

Salary: 30000

Tax Due: 3000

Pay Raise: 6000

C-4-07

ACTION CARD

Name: Win a Competition

Amount Paid to the Bank: 3000

Action Card Deck

Num. of Cards (Original): 50

Num. of Cards (Current): 49

ACTION CARD

Name: Buy an Item

Amount Paid to the Bank: 4000

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 50000 (to chosen player)

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 110000 (to chosen player)

ACTION CARD

Name: Win a Competition

Amount Paid to the Bank: 1000

ACTION CARD

Name: Setup School

Amount Collected from the Bank: 110000

ACTION CARD

Name: Tax Refund

Amount Collected from the Bank: Player's tax

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 15000 (from every player)

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: Write a Book

Amount Collected from the Bank: 13000

ACTION CARD

Name: Traffic Violation

Amount Paid to the Bank: 1000

ACTION CARD

Name: Buy an Item

Amount Paid to the Bank: 3000

ACTION CARD

Name: Hiking

Amount Paid to the Bank: 60000

ACTION CARD

Name: Watch a Show

Amount Paid to the Bank: 2000

ACTION CARD

Name: Setup School

Amount Collected from the Bank: 50000

ACTION CARD

Name: Setup School

Amount Collected from the Bank: 50000

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 7000 (from every player)

ACTION CARD

Name: Setup School

Amount Collected from the Bank: 100000

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: Traffic Violation

Amount Paid to the Bank: 5000

ACTION CARD

Name: Christmas Bonus

Amount Paid to Other Player: 8000 (to every player)

ACTION CARD

Name: Hiking

Amount Paid to the Bank: 20000

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: File a Lawsuit

Amount Collected from Other Player: 50000 (from chosen player)

ACTION CARD

Name: Buy an Item

Amount Paid to the Bank: 6000

ACTION CARD

Name: Setup School

Amount Collected from the Bank: 60000

ACTION CARD

Name: File a Lawsuit

Amount Collected from Other Player: 140000 (from chosen player)

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: Hiking

Amount Paid to the Bank: 60000

ACTION CARD

Name: Sell an Item

Amount Collected from the Bank: 4000

ACTION CARD

Name: Christmas Bonus

Amount Paid to Other Player: 7000 (to every player)

ACTION CARD

Name: Traffic Violation

Amount Paid to the Bank: 4000

ACTION CARD

Name: Win a Competition

Amount Paid to the Bank: 4000

ACTION CARD

Name: Visit a Place

Amount Paid to the Bank: 20000

ACTION CARD

Name: Hiking

Amount Paid to the Bank: 60000

ACTION CARD

Name: Write a Book

Amount Collected from the Bank: 18000

ACTION CARD

Name: Write a Book

Amount Collected from the Bank: 15000

ACTION CARD

Name: Win a Competition

Amount Paid to the Bank: 3000

ACTION CARD

Name: Sell an Item

Amount Collected from the Bank: 5000

ACTION CARD

Name: Watch a Show

Amount Paid to the Bank: 5000

ACTION CARD

Name: Watch a Show

Amount Paid to the Bank: 1000

ACTION CARD

Name: Write a Book

Amount Collected from the Bank: 8000

ACTION CARD

Name: Win a Competition

Amount Paid to the Bank: 5000

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 6000 (from every player)

ACTION CARD

Name: Christmas Bonus

Amount Paid to Other Player: 15000 (to every player)

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: Traffic Violation

Amount Paid to the Bank: 4000

Action Card Discard Pile
Num. of Cards (Current): 1

ACTION CARD
Name: Win a Competition
Amount Paid to the Bank: 3000

C-4-08

ACTION CARD
Name: Buy an Item
Amount Paid to the Bank: 4000

Action Card Deck
Num. of Cards (Original): 50
Num. of Cards (Current): 48

ACTION CARD
Name: Lawsuit
Amount Paid to Other Player: 50000 (to chosen player)

ACTION CARD
Name: Lawsuit
Amount Paid to Other Player: 110000 (to chosen player)

ACTION CARD
Name: Win a Competition
Amount Paid to the Bank: 1000

ACTION CARD

Name: Setup School

Amount Collected from the Bank: 110000

ACTION CARD

Name: Tax Refund

Amount Collected from the Bank: Player's tax

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 15000 (from every player)

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: Write a Book

Amount Collected from the Bank: 13000

ACTION CARD

Name: Traffic Violation

Amount Paid to the Bank: 1000

ACTION CARD

Name: Buy an Item

Amount Paid to the Bank: 3000

ACTION CARD

Name: Hiking

Amount Paid to the Bank: 60000

ACTION CARD

Name: Watch a Show

Amount Paid to the Bank: 2000

ACTION CARD

Name: Setup School

Amount Collected from the Bank: 50000

ACTION CARD

Name: Setup School

Amount Collected from the Bank: 50000

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 7000 (from every player)

ACTION CARD

Name: Setup School

Amount Collected from the Bank: 100000

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: Traffic Violation

Amount Paid to the Bank: 5000

ACTION CARD

Name: Christmas Bonus

Amount Paid to Other Player: 8000 (to every player)

ACTION CARD

Name: Hiking

Amount Paid to the Bank: 20000

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: File a Lawsuit

Amount Collected from Other Player: 50000 (from chosen player)

ACTION CARD

Name: Buy an Item

Amount Paid to the Bank: 6000

ACTION CARD

Name: Setup School

Amount Collected from the Bank: 60000

ACTION CARD

Name: File a Lawsuit

Amount Collected from Other Player: 140000 (from chosen player)

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: Hiking

Amount Paid to the Bank: 60000

ACTION CARD

Name: Sell an Item

Amount Collected from the Bank: 4000

ACTION CARD

Name: Christmas Bonus

Amount Paid to Other Player: 7000 (to every player)

ACTION CARD

Name: Traffic Violation

Amount Paid to the Bank: 4000

ACTION CARD

Name: Win a Competition

Amount Paid to the Bank: 4000

ACTION CARD

Name: Visit a Place

Amount Paid to the Bank: 20000

ACTION CARD

Name: Hiking

Amount Paid to the Bank: 60000

ACTION CARD

Name: Write a Book

Amount Collected from the Bank: 18000

ACTION CARD

Name: Write a Book

Amount Collected from the Bank: 15000

ACTION CARD

Name: Win a Competition

Amount Paid to the Bank: 3000

ACTION CARD

Name: Sell an Item

Amount Collected from the Bank: 5000

ACTION CARD

Name: Watch a Show

Amount Paid to the Bank: 5000

ACTION CARD

Name: Watch a Show

Amount Paid to the Bank: 1000

ACTION CARD

Name: Write a Book

Amount Collected from the Bank: 8000

ACTION CARD

Name: Win a Competition

Amount Paid to the Bank: 5000

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 6000 (from every player)

ACTION CARD

Name: Christmas Bonus

Amount Paid to Other Player: 15000 (to every player)

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: Traffic Violation

Amount Paid to the Bank: 4000

Action Card Discard Pile

Num. of Cards (Current): 2

ACTION CARD

Name: Buy an Item

Amount Paid to the Bank: 4000

ACTION CARD

Name: Win a Competition

Amount Paid to the Bank: 3000

C-4-09

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 50000 (to chosen player)

Action Card Deck

Num. of Cards (Original): 50

Num. of Cards (Current): 47

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 110000 (to chosen player)

ACTION CARD

Name: Win a Competition

Amount Paid to the Bank: 1000

ACTION CARD

Name: Setup School

Amount Collected from the Bank: 110000

ACTION CARD

Name: Tax Refund

Amount Collected from the Bank: Player's tax

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 15000 (from every player)

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: Write a Book

Amount Collected from the Bank: 13000

ACTION CARD

Name: Traffic Violation

Amount Paid to the Bank: 1000

ACTION CARD

Name: Buy an Item

Amount Paid to the Bank: 3000

ACTION CARD

Name: Hiking

Amount Paid to the Bank: 60000

ACTION CARD

Name: Watch a Show

Amount Paid to the Bank: 2000

ACTION CARD

Name: Setup School

Amount Collected from the Bank: 50000

ACTION CARD

Name: Setup School

Amount Collected from the Bank: 50000

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 7000 (from every player)

ACTION CARD

Name: Setup School

Amount Collected from the Bank: 100000

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: Traffic Violation

Amount Paid to the Bank: 5000

ACTION CARD

Name: Christmas Bonus

Amount Paid to Other Player: 8000 (to every player)

ACTION CARD

Name: Hiking

Amount Paid to the Bank: 20000

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: File a Lawsuit

Amount Collected from Other Player: 50000 (from chosen player)

ACTION CARD

Name: Buy an Item

Amount Paid to the Bank: 6000

ACTION CARD

Name: Setup School

Amount Collected from the Bank: 60000

ACTION CARD

Name: File a Lawsuit

Amount Collected from Other Player: 140000 (from chosen player)

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: Hiking

Amount Paid to the Bank: 60000

ACTION CARD

Name: Sell an Item

Amount Collected from the Bank: 4000

ACTION CARD

Name: Christmas Bonus

Amount Paid to Other Player: 7000 (to every player)

ACTION CARD

Name: Traffic Violation

Amount Paid to the Bank: 4000

ACTION CARD

Name: Win a Competition

Amount Paid to the Bank: 4000

ACTION CARD

Name: Visit a Place

Amount Paid to the Bank: 20000

ACTION CARD

Name: Hiking

Amount Paid to the Bank: 60000

ACTION CARD

Name: Write a Book

Amount Collected from the Bank: 18000

ACTION CARD

Name: Write a Book

Amount Collected from the Bank: 15000

ACTION CARD

Name: Win a Competition

Amount Paid to the Bank: 3000

ACTION CARD

Name: Sell an Item

Amount Collected from the Bank: 5000

ACTION CARD

Name: Watch a Show

Amount Paid to the Bank: 5000

ACTION CARD

Name: Watch a Show

Amount Paid to the Bank: 1000

ACTION CARD

Name: Write a Book

Amount Collected from the Bank: 8000

ACTION CARD

Name: Win a Competition

Amount Paid to the Bank: 5000

ACTION CARD

Name: It's Your Birthday

Amount Collected from Other Player: 6000 (from every player)

ACTION CARD

Name: Christmas Bonus

Amount Paid to Other Player: 15000 (to every player)

ACTION CARD

Name: Bonus Payday

Amount Collected from the Bank: Player's salary

ACTION CARD

Name: Traffic Violation

Amount Paid to the Bank: 4000

Action Card Discard Pile

Num. of Cards (Current): 3

ACTION CARD

Name: Lawsuit

Amount Paid to Other Player: 50000 (to chosen player)

ACTION CARD

Name: Buy an Item

Amount Paid to the Bank: 4000

ACTION CARD

Name: Win a Competition
Amount Paid to the Bank: 3000

C-4-10

HOUSE CARD
Name: Purogh Townhouse
Buying Price: 132000
Selling Price 105600

House Card Deck
Num. of Cards (Original): 10
Num. of Cards (Current): 9

HOUSE CARD
Name: Florence Penthouse
Buying Price: 121000
Selling Price 96800

HOUSE CARD
Name: Hai Lin Villa
Buying Price: 95000
Selling Price 76000

HOUSE CARD
Name: Jacobian Mansion
Buying Price: 119000
Selling Price 95200

HOUSE CARD
Name: Cece Townhouse

Buying Price: 102000

Selling Price 81600

HOUSE CARD

Name: Swing Penthouse

Buying Price: 98000

Selling Price 78400

HOUSE CARD

Name: Edwardian Mansion

Buying Price: 125000

Selling Price 100000

HOUSE CARD

Name: Java Farmhouse

Buying Price: 76000

Selling Price 60800

HOUSE CARD

Name: Havsis Villa

Buying Price: 138000

Selling Price 110400

HOUSE CARD

Name: Bali Farmhouse

Buying Price: 124000

Selling Price 99200

C-4-11

BLUE CARD

Career Match: Lawyer

Name: Lawsuit (Blue Card)

Received if Career Match: 15000

Paid to Other Player (or Bank): 140000

Blue Card Deck

Num. of Cards (Original): 7

Num. of Cards (Current): 7

BLUE CARD

Career Match: Athlete

Name: World Cup

Received if Career Match: 15000

Paid to Other Player (or Bank): Number of players * 5000

BLUE CARD

Career Match: Server

Name: Tip the Server

Received if Career Match: 15000

Paid to Other Player (or Bank): Randomly spun number * 1000

BLUE CARD

Career Match: Racecar Driver

Name: F1 Race

Received if Career Match: 15000

Paid to Other Player (or Bank): 10% of salary

BLUE CARD

Career Match: Computer Consultant

Name: Computer Repair

Received if Career Match: 15000

Paid to Other Player (or Bank): 10000 (if odd) 5000 (if even)

BLUE CARD

Career Match: Doctor

Name: Ski Accident

Received if Career Match: 15000

Paid to Other Player (or Bank): 10000

BLUE CARD

Career Match: Accountant

Name: Salary Tax Due

Received if Career Match: 15000

Paid to Other Player (or Bank): Tax Due

BLUE CARD

Career Match: Lawyer

Name: Lawsuit (Blue Card)

Received if Career Match: 15000

Paid to Other Player (or Bank): 140000