

[Let **game** be the **ThatsLife** instance variable] -> **ThatsLife game;**

Game Part Being Tested	Test Description	Sample Values	Expected Result	Actual Result	P/F
<i>Choosing of Starting Path</i>	The player is at the starting space junction of the board.	game.getCurrentPlayer ().getSpaceNum () = <b>0</b>  game.getBoard ().isAtStartJunction (game.getCurrentPlayer ().getSpaceNum ()) = <b>true</b>	The game will prompt the user to choose between two paths, <i>Career Path</i> or <i>College Path</i>	The game will prompt the user to choose between two paths, <i>Career Path</i> or <i>College Path</i>	P
<i>Starting Space - Junction</i>	The player opts to choose the <i>Career Path</i> at the start of the game.	game.playerChoice = <b>2</b> (player chose <i>Career Path</i> )  game.getBoard ().chosenPath (game.getCurrentPlayer ().getSpaceNum ()) = <b>16</b> ( <b>starting space number for the career path to be assigned to the space number of the player</b> )	The game will prompt the user to draw a career card until the user draws a career card that does not require a degree.	The game will prompt the user to draw a career card until the user draws a career card that does not require a degree.	P
	The player opts to choose the <i>Career Path</i> at the start of the game. The player draws a career card that requires a degree.	game.getCurrentPlayer ().getStatus ().hasDegree () = <b>false</b> (player does not have a college degree)  game.getCareerCardDrawn ().degreeRequired () = <b>true</b>	The game will prompt the player to draw a card again.	The game will prompt the player to draw a card again.	P

	<p>The player opts to choose the <i>Career Path</i> at the start of the game. The player draws a career card that does not require a degree.</p>	<p>game.getCurrentPlayer ().getStatus ().hasDegree () = <b>false</b> (player does not have a college degree)</p> <p>game.getCareerCardDrawn ().degreeRequired () = <b>false</b></p>	<p>The game will now ask the user to draw a salary card. The game will then assign both cards to the player.</p>	<p>The game will now ask the user to draw a salary card. The game will then assign both cards to the player.</p>	P
	<p>The player chooses the <i>College Path</i> at the start of the game.</p>	<p>game.playerChoice = <b>1</b> (player chose <i>College Path</i>)</p> <p>game.getBoard ().chosenPath (game.getCurrentPlayer ().getSpaceNum ()) = <b>16</b> <b>(starting space number for the college path to be assigned to the space number of the player)</b></p>	<p>The game will prompt the user to spin the wheel. The player will then move accordingly to the resulting spin number.</p>	<p>The game will prompt the user to spin the wheel. The player will then move accordingly to the resulting spin number.</p>	P
	<p>The player chooses the <i>College Path</i> at the start of the game.</p>	<p>game.playerChoice = <b>1</b> (player chose <i>College Path</i>)</p> <p>game.getCurrentPlayer ().getCash () = <b>0</b></p> <p>game.getCurrentPlayer ().getLoanBalance () = <b>0</b></p>	<p>\$40,000 will be added to the player's cash</p> <p>game.getCurrentPlayer ().getCash () = <b>40000</b></p> <p>\$50,000 will be added to the player's loan balance</p> <p>game.getCurrentPlayer ().getLoanBalance () = <b>50000</b></p> <p>(Player borrowed money from the bank 2 times)</p>	<p>\$40,000 will be added to the player's cash</p> <p>game.getCurrentPlayer ().getCash () = <b>40000</b></p> <p>\$50,000 will be added to the player's loan balance</p> <p>game.getCurrentPlayer ().getLoanBalance () = <b>50000</b></p>	P

				(Player borrowed money from the bank 2 times)	
<i>Turn Rotation</i>	The player finishes their current turn.		The game will prompt the current player that it is already the next player's turn.	The game will prompt the current player that it is already the next player's turn.	P
<i>Orange Spaces</i>	The player lands on an <i>Orange Space</i> .	game.getBoard ().isAtOrangeSpace (game.getCurrentSpace ().getType ()) = <b>true</b>	The game will prompt the user to draw an action card.	The game will prompt the user to draw an action card.	P
	The player lands on an <i>Orange Tile</i> . The player draws a <i>Collect from the Bank</i> action card.	game.getCardDrawn ().getName () = <b>"Tax Refund!"</b>  ((ValueCard) game.getCardDrawn ().getAmount () = <b>50000</b>	The player will collect \$50,000 from the bank.	The player will collect \$50,000 from the bank.	P
	The player lands on an <i>Orange Space</i> . The player draws a <i>Pay the Bank</i> action card.	game.getCardDrawn ().getName () = <b>"Traffic Violation!"</b>  ((ValueCard) game.getCardDrawn ().getAmount () = <b>100000</b>	The player will pay \$100,000 to the bank.  The player's cash will decrease by \$100,000	The player will pay \$100,000 to the bank.  The player's cash will decrease by \$100,000	P
	The player lands on an <i>Orange Space</i> . The player draws a <i>Pay the Player</i> action card. There are 2 players in the game.	game.getCardDrawn ().getName () = <b>"Lawsuit!"</b>  ((ValueCard) game.getCardDrawn ().getAmount () = <b>50000</b>  game.getNumOfActivePlayers () = <b>2</b>	The game will automatically transfer \$50,000 from Player 1 to Player 2. Player 1 will not be prompted to choose since there are only two players in the game.	The game will automatically transfer \$50,000 from Player 1 to Player 2. Player 1 will not be prompted to choose since there are only two players in the game.	P

	<p>The player lands on an <i>Orange Space</i>. The player draws a <i>Pay the Player</i> action card. There are 3 players in the game.</p>	<pre>game.getCardDrawn ().getName () = "Lawsuit!"  ((ValueCard) game.getCardDrawn ()).getAmount () = 50000  game.getNumOfActivePlayers () = 3</pre>	<p>The game will prompt the player to select between players 2 and 3. Player 1 selects player 3. The game will automatically transfer \$50,000 from Player 1 to Player 3.</p>	<p>The game will prompt the player to select between players 2 and 3. Player 1 selects player 3. The game will automatically transfer \$50,000 from Player 1 to Player 3.</p>	P
	<p>The player lands on an <i>Orange Space</i>. The player draws a <i>Pay all Players</i> action card. There are 2 players in the game.</p>	<pre>game.getCardDrawn ().getName () = "Christmas Bonus!"  ((ValueCard) game.getCardDrawn ()).getAmount () = 50000  game.getNumOfActivePlayers () = 2</pre>	<p>The game will automatically transfer \$50,000 from Player 1 to Player 2.</p>	<p>The game will automatically transfer \$50,000 from Player 1 to Player 2.</p>	P
	<p>The player lands on an <i>Orange Space</i>. The player draws a <i>Pay all Players</i> action card. There are 3 players in the game.</p>	<pre>game.getCardDrawn ().getName () = "Christmas Bonus!"  ((ValueCard) game.getCardDrawn ()).getAmount () = 50000  game.getNumOfActivePlayers () = 3</pre>	<p>The game will automatically deduct \$100,000 from Player 1 and then split it between players 2 and 3. Players 2 and 3 will receive \$50,000 each.</p>	<p>The game will automatically deduct \$100,000 from Player 1 and then split it between players 2 and 3. Players 2 and 3 will receive \$50,000 each.</p>	P
	<p>The player lands on an <i>Orange Space</i>. The player draws a <i>Collect from a Player</i> action card. There are 2 players in the game.</p>	<pre>game.getCardDrawn ().getName () = "File a Lawsuit!"  ((ValueCard) game.getCardDrawn ()).getAmount () = 50000  game.getNumOfActivePlayers () = 2</pre>	<p>The game will automatically transfer \$50,000 from Player 2 to Player 1. Player 1 will not be prompted to choose since there are only two players in the game.</p>	<p>The game will automatically transfer \$50,000 from Player 2 to Player 1. Player 1 will not be prompted to choose since there are only two players in the game.</p>	P
	<p>The player lands on an <i>Orange Space</i>. The player</p>	<pre>game.getCardDrawn ().getName () = "File a Lawsuit!"</pre>	<p>The game will prompt the player to select</p>	<p>The game will prompt the player to select</p>	P

	draws a <i>Collect from a Player</i> action card. There are 3 players in the game.	((ValueCard) game.getCardDrawn ().getAmount () = <b>50000</b>  game.getNumOfActivePlayers () = <b>3</b>	between players 2 and 3. Player 1 selects player 3. The game will automatically transfer \$50,000 from Player 3 to Player 1.	between players 2 and 3. Player 1 selects player 3. The game will automatically transfer \$50,000 from Player 3 to Player 1.	
	The player lands on an <i>Orange Space</i> . The player draws a <i>Collect from all Players</i> action card. There are 2 players in the game.	game.getCardDrawn ().getName () = <b>"Happy Birthday!"</b>  ((ValueCard) game.getCardDrawn ().getAmount () = <b>75000</b>  game.getNumOfActivePlayers () = <b>2</b>	The game will automatically transfer \$75,000 from Player 2 to Player 1. Player 1 will not be prompted to choose since there are only two players in the game.	The game will automatically transfer \$75,000 from Player 2 to Player 1. Player 1 will not be prompted to choose since there are only two players in the game.	P
	The player lands on an <i>Orange Space</i> . The player draws a <i>Collect from all Players</i> action card. There are 3 players in the game but one player is already retired.	game.getCardDrawn ().getName () = <b>"Happy Birthday!"</b>  ((ValueCard) game.getCardDrawn ().getAmount () = <b>75000</b>  game.getNumOfActivePlayers () = <b>2 (one player has retired, therefore 2 active players left)</b>	The game will automatically transfer \$75,000 from Player 2 to Player 1. Player 1 will not be prompted to choose since one of the players is now retired and there are now only two active players left in the game.	The game will automatically transfer \$75,000 from Player 2 to Player 1. Player 1 will not be prompted to choose since one of the players is now retired and there are now only two active players left in the game.	P
<i>Green Spaces</i>	The player lands on a <i>Green Space</i> .	game.getBoard ().isAtGreenSpace (game.getCurrentSpace ().getType ()) = <b>true</b>	The game will prompt the user to execute the action of the green space.	The game will prompt the user to execute the action of the green space.	P
	The player lands on a <i>Green Space</i> . A <i>Pay Day</i>	game.getCurrentPlayer ().getStatus ().getCurSalary () =	The game will automatically add cash	The game will automatically add	P

	event is triggered.	<b>60000</b>  game.getCurrentPlayer ().getCash () = <b>100000</b>	to the player, based on their current salary. Player 1's cash will now be \$160,000	cash to the player, based on their current salary. Player 1's cash will now be \$160,000	
	The player lands on a <i>Green Space</i> . A <i>Pay Raise</i> event is triggered. Player had <b>not yet reached</b> the max of the number of times a pay raise can be done as stated in the owned career card.	game.getCurrentPlayer ().getStatus ().getCurSalary () = <b>60000</b>  game.getCurrentPlayer ().getStatus ().getPayRaises () = <b>3 (number of times the player did a pay raise)</b>  game.getCurrentPlayer ().getStatus ().getCareerCard ().getMaxPayRaise () = <b>5</b> <b>(max number of times a pay raise can be done)</b>  game.getCurrentPlayer ().getStatus ().getSalaryCard ().getPayRaise () = <b>10000</b> <b>(pay raise value to be added to the current salary of the player)</b>	The game will automatically increase the current salary of the player. Player 1's new salary is now \$70,000	The game will automatically increase the current salary of the player. Player 1's new salary is now \$70,000	P
	The player lands on a <i>Green Space</i> . A <i>Pay Raise</i> event is triggered. Player had <b>reached</b> the max of the number of times a pay raise can be done as stated in the owned career card.	game.getCurrentPlayer ().getStatus ().getCurSalary () = <b>80000</b>  game.getCurrentPlayer ().getStatus ().getPayRaises () = <b>5 (number of times the player did a pay raise)</b>  game.getCurrentPlayer	The game will prompt the user that pay raise cannot be done anymore as the max number of times a pay raise can be done as stated in the owned career card has been reached.	The game will prompt the user that pay raise cannot be done anymore as the max number of times a pay raise can be done as stated in the owned career card has been reached.	P

		().getStatus ().getCareerCard ().getMaxPayRaise () = <b>5</b> <b>(max number of times a pay raise can be done)</b>			
<i>Blue Spaces</i>	The player lands on a <i>Blue Space</i> .	game.getBoard ().isAtBlueSpace (game.getCurrentSpace ().getType ()) = <b>true</b>	The game will prompt the user to draw a blue card.	The game will prompt the user to draw a blue card.	P
<i>Blue Space - Lawsuit</i>	The player lands on a <i>Blue Space</i> . The player draws a <i>Lawsuit</i> blue card. The player is a Lawyer.	game.getCardDrawn ().getName () = <b>"Lawsuit"</b> <b>(Action of the Blue Card)</b>  ((BlueCard) game.getCardDrawn ()).getCareer () = <b>"Lawyer"</b> <b>(Career assigned to the Blue Card is "Lawyer")</b>  ((ValueCard) game.getCardDrawn ()).getAmount () = <b>75000</b>  game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>"Lawyer"</b> <b>(Career of player is "Lawyer")</b>  game.isMatchingCareer () = <b>true</b> (player matches the career)	The player receives \$15,000.	The player receives \$15,000.	P
	The player lands on a <i>Blue Space</i> . The player draws a <i>Lawsuit</i> blue card. There is another player who is a Lawyer	game.getCardDrawn ().getName () = <b>"Lawsuit"</b> <b>(Action of the Blue Card)</b>  ((BlueCard) game.getCardDrawn ()).getCareer () = <b>"Lawyer"</b> <b>(Career assigned to the Blue Card is "Lawyer")</b>  ((ValueCard) game.getCardDrawn	Player 1 pays Player 2 \$50,000.	Player 1 pays Player 2 \$50,000.	P

		<p><code>().getAmount () = <b>75000</b></code></p> <p><code>game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>"Athlete"</b></code>  <b>(Career of player is "Athlete")</b></p> <p><code>game.isMatchingCareer () = <b>false</b></code>  (player does not match the career)</p> <p>By <code>findPlayerWithCareer ()</code>,  <code>game.playerChoice = <b>2</b></code>  <b>(player 2 matches the career)</b></p>			
	<p>The player lands on a <i>Blue Space</i>. The player draws a <i>Lawsuit</i> blue card. There are no Lawyers in the game.</p>	<p><code>game.getCardDrawn ().getName () = <b>"Lawsuit"</b></code>  <b>(Action of the Blue Card)</b></p> <p><code>((BlueCard) game.getCardDrawn ().getCareer () = <b>"Lawyer"</b></code>  <b>(Career assigned to the Blue Card is "Lawyer")</b></p> <p><code>((ValueCard) game.getCardDrawn ().getAmount () = <b>50000</b></code></p> <p><code>game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>"Athlete"</b></code>  <b>(Career of player is "Athlete")</b></p> <p><code>game.isMatchingCareer () = <b>false</b></code>  (player does not match the career)</p> <p>By <code>findPlayerWithCareer ()</code>,  <code>game.playerChoice = <b>-1</b></code>  <b>(no player matched the career)</b></p>	<p>The player pays the bank \$50,000 instead</p>	<p>The player pays the bank \$50,000.</p>	P
<i>Blue Space - Salary Tax Due</i>	<p>The player lands on a <i>Blue Space</i>. The player draws a</p>	<p><code>game.getCardDrawn ().getName () = <b>"Salary Tax Due"</b></code></p>	<p>The player receives \$15,000.</p>	<p>The player receives \$15,000.</p>	P



	<p><i>Salary Tax Due</i> blue card. The player is an Accountant.</p>	<p><b>(Action of the Blue Card)</b></p> <p>((BlueCard) game.getCardDrawn ().getCareer () = <b>“Accountant”</b> <b>(Career assigned to the Blue Card is “Accountant”)</b></p> <p>game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>“Accountant”</b> <b>(Career of player is “Accountant”)</b></p> <p>game.isMatchingCareer () = <b>true</b> <b>(player matches the career)</b></p>			
	<p>The player lands on a <i>Blue Space</i>. The player draws a <i>Salary Tax Due</i> blue card. There is another player who is an Accountant.</p>	<p>game.getCardDrawn ().getName () = <b>“Salary Tax Due”</b> <b>(Action of the Blue Card)</b></p> <p>((BlueCard) game.getCardDrawn ().getCareer () = <b>“Accountant”</b> <b>(Career assigned to the Blue Card is “Accountant”)</b></p> <p>game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>“Athlete”</b> <b>(Career of player is “Athlete”)</b></p> <p>game.isMatchingCareer () = <b>false</b> <b>(player does not match the career)</b></p> <p>By findPlayerWithCareer (), game.playerChoice = <b>2</b> <b>(player 2 matches the career)</b></p>	<p>Player 1 pays Player 2 with the tax due.</p>	<p>Player 1 pays Player 2 with the tax due.</p>	<p>P</p>
	<p>The player lands on a <i>Blue Space</i>. The player draws a <i>Salary Tax Due</i> blue card. There are no Accountants in the game.</p>	<p>game.getCardDrawn ().getName () = <b>“Salary Tax Due”</b> <b>(Action of the Blue Card)</b></p> <p>((BlueCard) game.getCardDrawn ().getCareer () = <b>“Accountant”</b></p>	<p>The player pays the bank \$10,000 instead.</p>	<p>The player pays the bank \$10,000 instead</p>	<p>P</p>

		<p><b>(Career assigned to the Blue Card is "Accountant")</b></p> <p>game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>"Athlete"</b></p> <p><b>(Career of player is "Athlete")</b> game.isMatchingCareer () = <b>false</b> (player does not match the career)</p> <p>By findPlayerWithCareer (), game.playerChoice = <b>-1</b> <b>(no player matched the career)</b></p>			
Blue Space - Tip the Server	<p>The player lands on a <i>Blue Space</i>. The player draws a <i>Tip the Server</i> blue card. The player spins the wheel to multiply with the amount (1000). The player is a Server.</p>	<p>game.getCardDrawn ().getName () = <b>"Tip The Server"</b> <b>(Action of the Blue Card)</b></p> <p>((BlueCard) game.getCardDrawn ().getCareer () = <b>"Server"</b> <b>(Career assigned to the Blue Card is "Server")</b></p> <p>game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>"Server"</b> <b>(Career of player is "Server")</b></p> <p>((ValueCard) game.getCardDrawn ().getAmount () = <b>1000</b></p> <p>game.isMatchingCareer () = <b>true</b> (player matches the career)</p> <p>game.getWheel ().getNumber () = <b>9</b> <b>(number spinned from the wheel)</b></p>	The player receives \$15,000.	The player receives \$15,000.	P
	<p>The player lands on a <i>Blue Space</i>. The player draws a <i>Tip the Server</i> blue card. The player spins the wheel to multiply with the amount</p>	<p>game.getCardDrawn ().getName () = <b>"Tip The Server"</b> <b>(Action of the Blue Card)</b></p> <p>((BlueCard) game.getCardDrawn</p>	Player 1 pays Player 2 \$9,000 (1000 * 9).	Player 1 pays Player 2 \$9,000 (1000 * 9).	P

	<p>(1000).There is another player who is a Server.</p>	<p><code>()</code>.getCareer () = <b>“Server”</b>  <b>(Career assigned to the Blue Card is “Server”)</b></p> <p><code>game.getCurrentPlayer ()</code>.getStatus  <code>()</code>.getCareerCard ().getName () =  <b>“Athlete”</b>  <b>(Career of player is “Athlete”)</b></p> <p><code>game.isMatchingCareer ()</code> = <b>false</b>  (player does not match the career)</p> <p><code>((ValueCard) game.getCardDrawn  ()).getAmount ()</code> = <b>1000</b></p> <p>By <code>findPlayerWithCareer ()</code>,  <code>game.playerChoice</code> = <b>2</b>  <b>(player 2 matches the career)</b></p> <p><code>game.getWheel ()</code>.getNumber () = <b>9</b>  <b>(number spun from the wheel)</b></p>			
	<p>The player lands on a <i>Blue Space</i>. The player draws a <i>Tip the Server</i> blue card. The player spins the wheel to multiply with the amount (1000).There are no Servers in the game.</p>	<p><code>game.getCardDrawn ()</code>.getName () =  <b>“Tip The Server”</b>  <b>(Action of the Blue Card)</b></p> <p><code>((BlueCard) game.getCardDrawn  ()).getCareer ()</code> = <b>“Server”</b>  <b>(Career assigned to the Blue Card is “Server”)</b></p> <p><code>game.getCurrentPlayer ()</code>.getStatus  <code>()</code>.getCareerCard ().getName () =  <b>“Athlete”</b>  <b>(Career of player is “Athlete”)</b></p> <p><code>game.isMatchingCareer ()</code> = <b>false</b>  (player does not match the career)</p> <p><code>((ValueCard) game.getCardDrawn  ()).getAmount ()</code> = <b>1000</b></p>	<p>The player pays the bank \$9,000 (1000 * 9) instead.</p>	<p>The player pays the bank \$9,000 (1000 * 9) instead.</p>	<p>P</p>

		By findPlayerWithCareer (), game.playerChoice = <b>-1</b> <b>(no player matched the career)</b>  game.getWheel ().getNumber () = <b>9</b> <b>(number spinned from the wheel)</b>			
<i>Blue Space - Ski Accident</i>	The player lands on a <i>Blue Space</i> . The player draws a <i>Ski Accident</i> blue card. The player is a Doctor.	game.getCardDrawn ().getName () = <b>"Ski Accident"</b> <b>(Action of the Blue Card)</b>  ((BlueCard) game.getCardDrawn ().getCareer () = <b>"Doctor"</b> <b>(Career assigned to the Blue Card is "Doctor")</b>  game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>"Doctor"</b> <b>(Career of player is "Doctor")</b>  game.isMatchingCareer () = <b>true</b> (player matches the career)  ((ValueCard) game.getCardDrawn ().getAmount () = <b>10000</b>	The player receives \$15,000.	The player receives \$15,000.	P
	The player lands on a <i>Blue Space</i> . The player draws a <i>Ski Accident</i> blue card. There is another player who is a Doctor.	game.getCardDrawn ().getName () = <b>"Ski Accident"</b> <b>(Action of the Blue Card)</b>  ((BlueCard) game.getCardDrawn ().getCareer () = <b>"Doctor"</b> <b>(Career assigned to the Blue Card is "Doctor")</b>  game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>"Athlete"</b> <b>(Career of player is "Athlete")</b>  game.isMatchingCareer () = <b>false</b>	Player 1 pays Player 2 \$10,000.	Player 1 pays Player 2 \$10,000.	P

		(player does not match the career)  ((ValueCard) game.getCardDrawn ()).getAmount () = <b>10000</b>  By findPlayerWithCareer (), game.playerChoice = <b>2</b> <b>(player 2 matches the career)</b>			
	The player lands on a <i>Blue Space</i> . The player draws a <i>Ski Accident</i> blue card. There are no Doctors in the game.	game.getCardDrawn ().getName () = <b>"Ski Accident"</b> <b>(Action of the Blue Card)</b>  ((BlueCard) game.getCardDrawn ()).getCareer () = <b>"Doctor"</b> <b>(Career assigned to the Blue Card is "Doctor")</b>  game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>"Athlete"</b> <b>(Career of player is "Athlete")</b>  game.isMatchingCareer () = <b>false</b> (player does not match the career)  ((ValueCard) game.getCardDrawn ()).getAmount () = <b>10000</b>  By findPlayerWithCareer (), game.playerChoice = <b>-1</b> <b>(no player matched the career)</b>	The player pays the bank \$10,000 instead.	The player pays the bank \$10,000 instead.	P
<i>Blue Space - Computer Repair</i>	The player lands on a <i>Blue Space</i> . The player draws a <i>Computer Repair</i> blue card. The player is prompted to spin the wheel to determine odd or even number results. The player is a Computer Consultant.	game.getCardDrawn ().getName () = <b>"Computer Repair"</b> <b>(Action of the Blue Card)</b>  ((BlueCard) game.getCardDrawn ()).getCareer () = <b>"Computer Consultant"</b> <b>(Career assigned to the Blue Card is "Computer Consultant")</b>	The player receives \$15,000.	The player receives \$15,000.	P

		<p>game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>“Computer Consultant”</b> (Career of player is “Computer Consultant”)</p> <p>game.isMatchingCareer () = <b>true</b> (player matches the career)</p>			
	<p>The player lands on a <i>Blue Space</i>. The player draws a <i>Computer Repair</i> blue card. There is another player who is a Computer Consultant. The player spins for an odd number.</p>	<p>game.getCardDrawn ().getName () = <b>“Computer Repair”</b> (Action of the Blue Card)</p> <p>((BlueCard) game.getCardDrawn ().getCareer () = <b>“Computer Consultant”</b> (Career assigned to the Blue Card is “Computer Consultant”)</p> <p>game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>“Athlete”</b> (Career of player is “Athlete”)</p> <p>game.isMatchingCareer () = <b>false</b> (player does not match the career)</p> <p>By findPlayerWithCareer (), game.playerChoice = <b>2</b> (player 2 matches the career)</p> <p>game.getWheel ().getNumber () = <b>1</b> (odd number, pay 10000)</p>	<p>Player 1 pays Player 2 \$10,000 (odd number spinned).</p>	<p>Player 1 pays Player 2 \$10,000 (odd number spinned).</p>	P
	<p>The player lands on a <i>Blue Space</i>. The player draws a <i>Computer Repair</i> blue card. There is another player who is a Computer Consultant. The player spins for an even number.</p>	<p>game.getCardDrawn ().getName () = <b>“Computer Repair”</b> (Action of the Blue Card)</p> <p>((BlueCard) game.getCardDrawn ().getCareer () = <b>“Computer Consultant”</b></p>	<p>Player 1 pays Player 2 \$5,000 (even number spinned).</p>	<p>Player 1 pays Player 2 \$5,000 (even number spinned).</p>	P

		<p><b>(Career assigned to the Blue Card is “Computer Consultant”)</b></p> <p>game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>“Athlete”</b></p> <p><b>(Career of player is “Athlete”)</b></p> <p>game.isMatchingCareer () = <b>false</b> (player does not match the career)</p> <p>By findPlayerWithCareer (), game.playerChoice = <b>2</b> <b>(player 2 matches the career)</b></p> <p>game.getWheel ().getNumber () = <b>2</b> <b>(even number, pay 5000)</b></p>			
	<p>The player lands on a <i>Blue Space</i>. The player draws a <i>Computer Repair</i> blue card. There are no Computer Consultant in the game. The player spins for an odd number.</p>	<p>game.getCardDrawn ().getName () = <b>“Computer Repair”</b> <b>(Action of the Blue Card)</b></p> <p>((BlueCard) game.getCardDrawn ().getCareer () = <b>“Computer Consultant”</b> <b>(Career assigned to the Blue Card is “Computer Consultant”)</b></p> <p>game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>“Athlete”</b> <b>(Career of player is “Athlete”)</b></p> <p>game.isMatchingCareer () = <b>false</b> (player does not match the career)</p> <p>By findPlayerWithCareer (), game.playerChoice = <b>-1</b> <b>(no player matched the career)</b></p> <p>game.getWheel ().getNumber () = <b>1</b> <b>(odd number, pay 10000)</b></p>	<p>The player pays the bank \$10,000 (odd number spun) instead.</p>	<p>The player pays the bank \$10,000 (odd number spun) instead.</p>	P

	<p>The player lands on a <i>Blue Space</i>. The player draws a <i>Computer Repair</i> blue card. There are no Computer Consultant in the game. The player spins for an even number.</p>	<p>game.getCardDrawn ().getName () = <b>“Computer Repair”</b> (Action of the Blue Card)</p> <p>((BlueCard) game.getCardDrawn ().getCareer () = <b>“Computer Consultant”</b> (Career assigned to the Blue Card is “Computer Consultant”)</p> <p>game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>“Athlete”</b> (Career of player is “Athlete”) game.isMatchingCareer () = <b>false</b> (player does not match the career)</p> <p>By findPlayerWithCareer (), game.playerChoice = -1 (no player matched the career)</p> <p>game.getWheel ().getNumber () = <b>2</b> (even number, pay 5000)</p>	<p>The player pays the bank \$5,000 (even number spun) instead.</p>	<p>The player pays the bank \$5,000 (even number spun) instead.</p>	P
<i>Blue Space - World Cup</i>	<p>The player lands on a <i>Blue Space</i>. The player draws a <i>World Cup</i> blue card. The player is an Athlete.</p>	<p>game.getCardDrawn ().getName () = <b>“World Cup”</b> (Action of the Blue Card)</p> <p>((BlueCard) game.getCardDrawn ().getCareer () = <b>“Athlete”</b> (Career assigned to the Blue Card is “Athlete”)</p> <p>game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>“Athlete”</b> (Career of player is “Athlete”)</p> <p>game.isMatchingCareer () = <b>true</b> (player matches the career)</p>	<p>The player receives \$15,000.</p>	<p>The player receives \$15,000.</p>	P
	<p>The player lands on a <i>Blue</i></p>	<p>game.getCardDrawn ().getName () =</p>	<p>Player 1 pays Player 2</p>	<p>Player 1 pays Player</p>	P



	<p><i>Space</i>. The player draws a <i>World Cup</i> blue card. There is another player who is an Athlete. There are 2 players in the game.</p>	<p><b>“World Cup” (Action of the Blue Card)</b></p> <p>((BlueCard) game.getCardDrawn ().getCareer () = <b>“Athlete”</b> <b>(Career assigned to the Blue Card is “Athlete”)</b></p> <p>game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>“Lawyer”</b> <b>(Career of player is “Lawyer”)</b></p> <p>game.isMatchingCareer () = <b>false</b> (player does not match the career)</p> <p>By findPlayerWithCareer (), game.playerChoice = <b>2</b> <b>(player 2 matches the career)</b></p> <p>game.getNumOfActivePlayers () = <b>2</b></p>	\$10,000 (5000 * 2).	2 \$10,000 (5000 * 2).	
	<p>The player lands on a <i>Blue Space</i>. The player draws a <i>World Cup</i> blue card. There is another player who is an Athlete. There are 3 players in the game.</p>	<p>game.getCardDrawn ().getName () = <b>“World Cup”</b> <b>(Action of the Blue Card)</b></p> <p>((BlueCard) game.getCardDrawn ().getCareer () = <b>“Athlete”</b> <b>(Career assigned to the Blue Card is “Athlete”)</b></p> <p>game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>“Lawyer”</b> <b>(Career of player is “Lawyer”)</b></p> <p>game.isMatchingCareer () = <b>false</b> (player does not match the career)</p> <p>By findPlayerWithCareer (), game.playerChoice = <b>2</b> <b>(player 2 matches the career)</b></p>	Player 1 pays Player 2 \$15,000 (5000 * 3).	Player 1 pays Player 2 \$15,000 (5000 * 3).	P

		game.getNumOfActivePlayers () = 3			
	<p>The player lands on a <i>Blue Space</i>. The player draws a <i>World Cup</i> blue card. There are no Athletes in the game. There are 2 players in the game.</p>	<p>game.getCardDrawn ().getName () = <b>“World Cup”</b>  <b>(Action of the Blue Card)</b></p> <p>((BlueCard) game.getCardDrawn ().getCareer () = <b>“Athlete”</b>  <b>(Career assigned to the Blue Card is “Athlete”)</b></p> <p>game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>“Lawyer”</b>  <b>(Career of player is “Lawyer”)</b></p> <p>game.isMatchingCareer () = <b>false</b>  (player does not match the career)</p> <p>By findPlayerWithCareer (),  game.playerChoice = -1  <b>(no player matched the career)</b></p> <p>game.getNumOfActivePlayers () = 2</p>	<p>The player pays the bank \$10,000 (5000 * 2).</p>	<p>The player pays the bank \$10,000 (5000 * 2).</p>	P
	<p>The player lands on a <i>Blue Space</i>. The player draws a <i>World Cup</i> blue card. There are no Athletes in the game. There are 3 players in the game.</p>	<p>game.getCardDrawn ().getName () = <b>“World Cup”</b>  <b>(Action of the Blue Card)</b></p> <p>((BlueCard) game.getCardDrawn ().getCareer () = <b>“Athlete”</b>  <b>(Career assigned to the Blue Card is “Athlete”)</b></p> <p>game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>“Lawyer”</b>  <b>(Career of player is “Lawyer”)</b></p> <p>game.isMatchingCareer () = <b>false</b>  (player does not match the career)</p>	<p>The player pays the bank \$15,000 (5000 * 3).</p>	<p>The player pays the bank \$15,000 (5000 * 3).</p>	P

		<p>By findPlayerWithCareer (), game.playerChoice = -1 <b>(no player matched the career)</b></p> <p>game.getNumOfActivePlayers () = 3</p>			
Blue Space - F1 Race	<p>The player lands on a <i>Blue Space</i>. The player draws a <i>F1 Race</i> blue card. The player is a Racecar Driver.</p>	<p>game.getCardDrawn ().getName () = <b>"F1 Race"</b> <b>(Action of the Blue Card)</b></p> <p>((BlueCard) game.getCardDrawn ().getCareer () = <b>"Racecar Driver"</b> <b>(Career assigned to the Blue Card is "Racecar Driver")</b></p> <p>game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>"Racecar Driver"</b> <b>(Career of player is "Racecar Driver")</b></p> <p>game.isMatchingCareer () = <b>true</b> (player matches the career)</p>	The player receives \$15,000.	The player receives \$15,000.	P
	<p>The player lands on a <i>Blue Space</i>. The player draws a <i>F1 Race</i> blue card. There is another player who is a Racecar Driver.</p>	<p>game.getCardDrawn ().getName () = <b>"F1 Race"</b> <b>(Action of the Blue Card)</b></p> <p>((BlueCard) game.getCardDrawn ().getCareer () = <b>"Racecar Driver"</b> <b>(Career assigned to the Blue Card is "Racecar Driver")</b></p> <p>game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>"Lawyer"</b> <b>(Career of player is "Lawyer")</b></p> <p>game.isMatchingCareer () = <b>true</b> (player matches the career)</p>	Player 1 pays Player 2 \$10,000 (100000 / 10).	Player 1 pays Player 2 \$10,000 (100000 / 10).	P

		<p>By findPlayerWithCareer (), game.playerChoice = <b>2</b> (player 2 matched the career)</p> <p>game.getCurrentPlayer ().getStatus ( ).getCurSalary () = <b>100000</b></p>			
	<p>The player lands on a <i>Blue Space</i>. The player draws a <i>F1 Race</i> blue card. There are no Racecar Drivers in the game.</p>	<p>game.getCardDrawn ().getName () = <b>"F1 Race"</b> <b>(Action of the Blue Card)</b></p> <p>((BlueCard) game.getCardDrawn ( )).getCareer () = <b>"Racecar Driver"</b> <b>(Career assigned to the Blue Card is "Racecar Driver")</b></p> <p>game.getCurrentPlayer ().getStatus ( ).getCareerCard ().getName () = <b>"Lawyer"</b> <b>(Career of player is "Lawyer")</b></p> <p>game.isMatchingCareer () = <b>true</b> (player matches the career)</p> <p>By findPlayerWithCareer (), game.playerChoice = <b>-1</b> (no player matched the career)</p> <p>game.getCurrentPlayer ().getStatus ( ).getCurSalary () = <b>100000</b></p>	<p>The player pays the bank \$10,000 (100000 / 10).</p>	<p>The player pays the bank \$10,000 (100000 / 10).</p>	P
<i>Magenta Space</i>	<p>The player lands on a <i>Magenta Space</i> and the number of movements needed as spinned by the wheel is not yet achieved.</p>	<p>game.getBoard ( ).isAtMagentaSpace (game.getCurrentSpace ( ).getType ()) = <b>true</b></p> <p>game.getWheel ().getNumber () = 5</p> <p>number of movements achieved by the player (counter) = 3</p>	<p>Player will STOP! Immediately proceed to doing the specified magenta space action.</p>	<p>Player will STOP! Immediately proceed to doing the specified magenta space action.</p>	P

<p><i>Magenta Space - Graduation</i></p>	<p>The player lands on the <i>Graduation Space</i>.</p>	<pre>game.getBoard () .isAtMagentaSpace (game.getCurrentSpace ().getType ()) = <b>true</b>  game.getCurrentPlayer ().getStatus ().hasDegree () = <b>false</b></pre>	<p>The player will now gain a degree.</p> <pre>game.getCurrentPlayer ().getStatus ().graduate ()</pre>	<p>The player will now gain a degree.</p> <pre>game.getCurrentPlay er ().getStatus ().graduate ()</pre>	<p>P</p>
<p><i>Magenta Space - College Career Choice</i></p>	<p>The player lands on the <i>College Career Choice Space</i>. The game presents the player with two sets of cards: Lawyer with \$80,000 salary and Athlete with \$90,000 salary. The player selects the set with the Lawyer card.</p>	<pre>game.getBoard ().isCareerChoiceSpace (game.getCurrentSpace.getMagenta ()) = <b>true</b>  game.drawCareerCard () (2x) game.drawSalaryCard () (2x)  game.getCareerCardDrawn ().getName () = "<b>Lawyer</b>"  game.getSalaryCardDrawn ().getSalary () = <b>80000</b>  game.getSecondCareerCardDrawn ().getName () = "<b>Athlete</b>"  game.getSecondSalaryCardDrawn ().getSalary () = <b>90000</b></pre>	<p>The game assigns the Lawyer career card along with the \$80,000 salary card to the player. These cards are now owned, therefore it cannot be drawn by the other players.</p>	<p>The game assigns the Lawyer career card along with the \$80,000 salary card to the player. These cards are now owned, therefore it cannot be drawn by the other players.</p>	<p>P</p>
	<p>The player lands on the <i>College Career Choice Space</i>. The game presents the player with two sets of cards: Lawyer with \$80,000 salary and Doctor with \$60,000 salary. The player selects the set with the Doctor card.</p>	<pre>game.getBoard ().isCareerChoiceSpace (game.getCurrentSpace.getMagenta ()) = <b>true</b>  game.drawCareerCard () (2x) game.drawSalaryCard () (2x)  game.getCareerCardDrawn ().getName () = "<b>Lawyer</b>"  game.getSalaryCardDrawn ().getSalary () = <b>80000</b></pre>	<p>The game assigns the Doctor career card along with the \$60,000 salary card to the player. These cards are now owned, therefore it cannot be drawn by the other players.</p>	<p>The game assigns the Doctor career card along with the \$60,000 salary card to the player. These cards are now owned, therefore it cannot be drawn by the other players.</p>	<p>P</p>

		<p>game.getSecondCareerCardDrawn ().getName () = <b>"Doctor"</b></p> <p>game.getSecondSalaryCardDrawn ().getSalary () = <b>60000</b></p>			
Magenta Space - Job Search	<p>The player lands on the <i>Job Search Space</i>. The game presents the player with a new career and salary card: Computer Consultant with a salary of \$100,000. The player's current career is a Server with a \$40,000 salary. The player chose to change their career.</p>	<p>game.getBoard ().isJobSearchSpace (game.getCurrentSpace.getMagenta ()) = <b>true</b></p> <p>game.drawCareerCard () game.drawSalaryCard ()</p> <p>game.getCareerCardDrawn ().getName () = <b>"Computer Consultant"</b></p> <p>game.getSalaryCardDrawn ().getSalary () = <b>100000</b></p> <p>game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>"Server"</b></p> <p>game.getCurrentPlayer ().getStatus ().getSalaryCard ().getSalary () = <b>40000</b></p>	<p>The game assigns the Computer Consultant career card along with the \$100,000 salary card to the player. The player's old cards are now disowned and could be drawn by other players already.</p>	<p>The game assigns the Computer Consultant career card along with the \$100,000 salary card to the player. The player's old cards are now disowned and could be drawn by other players already.</p>	P
	<p>The player lands on the <i>Job Search Space</i>. The game presents the player with a new career and salary card: Computer Consultant with a salary of \$100,000. The player's current career is a Server with a \$40,000 salary. The player chose to keep their current career.</p>	<p>game.getBoard ().isJobSearchSpace (game.getCurrentSpace.getMagenta ()) = <b>true</b></p> <p>game.drawCareerCard () game.drawSalaryCard ()</p> <p>game.getCareerCardDrawn ().getName () = <b>"Computer Consultant"</b></p> <p>game.getSalaryCardDrawn ().getSalary () = <b>100000</b></p>	<p>The game does not assign the Computer Consultant career card along with the \$100,000 salary card to the player. The player's old cards are still in their possession. Other players can draw the Computer Consultant career card along with</p>	<p>The game does not assign the Computer Consultant career card along with the \$100,000 salary card to the player. The player's old cards are still in their possession. Other players can draw the Computer Consultant</p>	P

		<p>game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>"Server"</b></p> <p>game.getCurrentPlayer ().getStatus ().getSalaryCard ().getSalary () = <b>40000</b></p>	the \$100,000 salary card.	career card along with the \$100,000 salary card.	
	<p>The player lands on the <i>Job Search Space</i>. <b>The player does not have a degree yet.</b> The game presents the plate with a new career (with no degree) and salary card: Athlete with a salary of \$70000. The player's current career is a Server with a \$40,000 salary. The player chose to keep their current career.</p>	<p>game.getBoard ().isJobSearchSpace (game.getCurrentSpace.getMagenta ()) = <b>true</b></p> <p><b>game.drawNoDegreeCareer ()</b> game.drawSalaryCard ()</p> <p>game.getCareerCardDrawn ().getName () = <b>"Athlete"</b></p> <p>game.getSalaryCardDrawn ().getSalary () = <b>70000</b></p> <p>game.getCurrentPlayer ().getStatus ().getCareerCard ().getName () = <b>"Server"</b></p> <p>game.getCurrentPlayer ().getStatus ().getSalaryCard ().getSalary () = <b>40000</b></p>	The game does not assign the Athlete career card along with the \$70,000 salary card to the player. The player's old cards are still in their possession. Other players can draw the Athlete career card along with the \$70,000 salary card.	The game does not assign the Athlete career card along with the \$70,000 salary card to the player. The player's old cards are still in their possession. Other players can draw the Athlete career card along with the \$70,000 salary card.	P
<i>Magenta Space - Buy a House</i>	<p>The player lands on the <i>Buy a House Space</i>. The game presents the player with three house choices: (1) The Bungalow, (2) The Mountain, and (3) The Mansion. The player selects The Bungalow. The player has enough cash to buy The Bungalow.</p>	<p>game.getBoard ().isBuyHouseSpace (game.getCurrentSpace ().getMagenta ()) = <b>true</b></p> <p>game.getCurrentPlayer ().getCash () = <b>700000</b></p> <p>game.getCardDrawn ().getName () = <b>"The Bungalow"</b></p> <p>((HouseCard) game.getCardDrawn ().getPrice () = <b>300000</b></p>	The game assigns The Bungalow house card to the player. \$300,000 is deducted from the player's cash. He is left with \$400,000. The Bungalow house card can no longer be chosen by other players.	The game assigns The Bungalow house card to the player. \$300,000 is deducted from the player's cash. He is left with \$400,000. The Bungalow house card can no longer be chosen by other players	P

	<p>The player lands on the <i>Buy a House Space</i>. The game presents the player with three house choices: (1) The Bungalow, (2) The Mountain, and (3) The Mansion. The player selects The Mountain. The player has enough cash to buy The Mountain.</p>	<pre>game.getBoard ().isBuyHouseSpace (game.getCurrentSpace ().getMagenta ()) = <b>true</b>  game.getCurrentPlayer ().getCash () = <b>700000</b>  game.getCardDrawn ().getName () = <b>"The Mountain"</b>  ((HouseCard) game.getCardDrawn ()).getPrice () = <b>400000</b></pre>	<p>The game assigns The Mountain house card to the player. \$400,000 is deducted from the player's cash. He is left with \$300,000. The Mountain house card can no longer be chosen by other players.</p>	<p>The game assigns The Mountain house card to the player. \$400,000 is deducted from the player's cash. He is left with \$300,000. The Mountain house card can no longer be chosen by other players.</p>	P
	<p>The player lands on the <i>Buy a House Space</i>. The game presents the player with three house choices: (1) The Bungalow, (2) The Mountain, and (3) The Mansion. The player selects The Mansion. The player has enough cash to buy The Mansion.</p>	<pre>game.getBoard ().isBuyHouseSpace (game.getCurrentSpace ().getMagenta ()) = <b>true</b>  game.getCurrentPlayer ().getCash () = <b>700000</b>  game.getCardDrawn ().getName () = <b>"The Mansion"</b>  ((HouseCard) game.getCardDrawn ()).getPrice () = <b>600000</b></pre>	<p>The game assigns The Mansion house card to the player. \$600,000 is deducted from the player's cash. He is left with \$100,000. The Mansion house card can no longer be chosen by other players.</p>	<p>The game assigns The Mansion house card to the player. \$600,000 is deducted from the player's cash. He is left with \$100,000. The Mansion house card can no longer be chosen by other players.</p>	P
	<p>The player lands on the <i>Buy a House Space</i>. The game presents the player with three house choices: (1) The Bungalow, (2) The Mountain, and (3) The Mansion. The player selects The Mansion. The player does not have enough cash to buy The Mansion.</p>	<pre>game.getBoard ().isBuyHouseSpace (game.getCurrentSpace ().getMagenta ()) = <b>true</b>  game.getCurrentPlayer ().getCash () = <b>0</b>  game.getCurrentPlayer ().getLoanBalance () = <b>0</b>  game.getCardDrawn ().getName () = <b>"The Mansion"</b>  ((HouseCard) game.getCardDrawn ()).getPrice () = <b>600000</b></pre>	<p>The game assigns The Mansion house card to the player. The player then loans \$600,000 from the bank. The player's loan balance is now \$750,000.</p> <pre>game.getCurrentPlayer ().getLoanBalance () = 750000</pre>	<p>The game assigns The Mansion house card to the player. The player then loans \$600,000 from the bank. The player's loan balance is now \$750,000.</p> <pre>game.getCurrentPlay er ().getLoanBalance () = 750000</pre>	P



Magenta Space - Get Married	The player lands on a <i>Get Married Space</i> . The player is not married yet. The player spins for an odd number. There are 2 players in the game.	<p>game.isActionRequiredSpin () = <b>true</b> (<b>magenta action requires spin</b>)</p> <p>game.getCurrentPlayer ().getStatus ().isMarried () = <b>false</b> (<b>not yet married</b>)</p> <p>game.getWheel ().getNumber () = <b>1</b> (<b>odd number spinned</b>)</p> <p>game.getNumOfActivePlayers () = <b>2</b></p>	The player gains a total of \$5,000. This is taken from the other player in the game.	The player gains a total of \$5,000. This is taken from the other player in the game.	P
	The player lands on a <i>Get Married Space</i> . The player is not married yet. The player spins for an odd number. There are 3 players in the game.	<p>game.isActionRequiredSpin () = <b>true</b> (<b>magenta action requires spin</b>)</p> <p>game.getCurrentPlayer ().getStatus ().isMarried () = <b>false</b> (<b>not yet married</b>)</p> <p>game.getWheel ().getNumber () = <b>1</b> (<b>odd number spinned</b>)</p> <p>game.getNumOfActivePlayers () = <b>3</b></p>	The player gains a total of \$10,000. \$5,000 is taken from each player in the game.	The player gains a total of \$10,000. \$5,000 is taken from each player in the game.	P
	The player lands on a <i>Get Married Space</i> . The player is not married yet. The player spins for an even number. There are 2 players in the game.	<p>game.isActionRequiredSpin () = <b>true</b> (<b>magenta action requires spin</b>)</p> <p>game.getCurrentPlayer ().getStatus ().isMarried () = <b>false</b> (<b>not yet married</b>)</p> <p>game.getWheel ().getNumber () = <b>2</b> (<b>even number spinned</b>)</p> <p>game.getNumOfActivePlayers () = <b>2</b></p>	The player gains a total of \$10,000. This is taken from the other player in the game.	The player gains a total of \$10,000. This is taken from the other player in the game.	P
	The player lands on a <i>Get Married Space</i> . The player is not married yet. The player spins for an even number. There are 3	<p>game.isActionRequiredSpin () = <b>true</b> (<b>magenta action requires spin</b>)</p> <p>game.getCurrentPlayer ().getStatus ().isMarried () = <b>false</b></p>	The player gains a total of \$20,000. \$10,000 is taken from each player in the game.	The player gains a total of \$20,000. \$10,000 is taken from each player in the game.	P

	players in the game.	<p><b>(not yet married)</b></p> <p>game.getWheel ().getNumber () = <b>2</b> <b>(even number spun)</b></p> <p>game.getNumOfActivePlayers () = <b>3</b></p>			
	The player lands on a <i>Get Married Space</i> . The player is married already.	<p>game.isActionRequiredSpin () = <b>true</b> <b>(magenta action requires spin)</b></p> <p>game.getCurrentPlayer ().getStatus ().isMarried () = <b>true</b> <b>(already married)</b></p>	Nothing will happen on the <i>Get Married Space</i> .	Nothing will happen on the <i>Get Married Space</i> .	P
<i>Magenta Space - Have Children</i>	The player lands on a <i>Have Children Space</i> . The player is not married yet.	<p>game.getBoard ().isHaveChildrenSpace (game.getCurrentSpace ().getMagenta ()) = <b>true</b></p> <p>game.getCurrentPlayer ().getStatus ().isMarried () = <b>false</b></p>	Nothing will happen on the <i>Have Children Space</i> .	Nothing will happen on the <i>Have Children Space</i> .	P
	The player lands on a <i>Have Children Space</i> . The player is married. The player does not have any children yet. The space gives the player a baby (one child, by random chance).	<p>game.getBoard ().isHaveChildrenSpace (game.getCurrentSpace ().getMagenta ()) = <b>true</b></p> <p>game.getCurrentPlayer ().getStatus ().isMarried () = <b>true</b></p> <p>game.getCurrentPlayer ().getStatus ().getFamily () = <b>0</b></p>	<p>The player's number of children will increase by 1.</p> <p>player.childrenAmt = 1</p>	<p>The player's number of children will increase by 1.</p> <p>player.childrenAmt = 1</p>	P
	The player lands on a <i>Have Children Space</i> . The player is married. The player does	<p>game.getBoard ().isHaveChildrenSpace (game.getCurrentSpace ().getMagenta ()) = <b>true</b></p>	The player's number of children will increase by 2.	The player's number of children will increase by 2.	P

	not have any children yet. The space gives the player twins (two children, by random chance.)	game.getCurrentPlayer ().getStatus ().isMarried () = <b>true</b>  game.getCurrentPlayer ().getStatus ().getFamily () = <b>0</b>	player.childrenAmt = 2	player.childrenAmt = 2	
	The player lands on a <i>Have Children Space</i> . The player is married. The player has 2 children already. The space gives the player a baby.	game.getBoard ().isHaveChildrenSpace (game.getCurrentSpace ().getMagenta ()) = <b>true</b>  game.getCurrentPlayer ().getStatus ().isMarried () = <b>true</b>  game.getCurrentPlayer ().getStatus ().getFamily () = <b>2</b>	The player's number of children will increase by 1.  player.childrenAmt = 3	The player's number of children will increase by 1.  player.childrenAmt = 3	P
	The player lands on a <i>Have Children Space</i> . The player is married. The player has 2 children already. The space gives the player twins.	game.getBoard ().isHaveChildrenSpace (game.getCurrentSpace ().getMagenta ()) = <b>true</b>  game.getCurrentPlayer ().getStatus ().isMarried () = <b>true</b> game.getCurrentPlayer ().getStatus ().getFamily () = <b>2</b>	The player's number of children will not increase since it will reach the max family (4).  player.childrenAmt = 2	The player's number of children will not increase since it will reach the max family (4). player.childrenAmt = 2	P
Magenta Space Junction - Change Career Path	The player lands on the <i>Career Junction</i> . The player opts to go to the <i>Change Career Path</i> .	game.getBoard ().isAtCareerJunction (game.getCurrentPlayer ().getSpaceNum ()) = <b>true</b>  game.playerChoice = <b>1</b> <b>(player chose the Change Career Path)</b>  game.getBoard ().chosenPath (game.getCurrentPlayer ().getSpaceNum ()) = <b>60</b> <b>(starting space of the Change Career Path)</b>	The player will move to the starting space (60) of the <i>Change Career Path</i>	The player will move to the starting space (60) of the <i>Change Career Path</i>	P
	The player lands on the <i>Career Junction</i> . The player opts to go to the <i>Keep Career Path</i> .	game.getBoard ().isAtCareerJunction (game.getCurrentPlayer ().getSpaceNum ()) = <b>true</b>	The player will move to the starting space (73) of the <i>Keep Career Path</i> .	The player will move to the starting space (73) of the <i>Keep Career Path</i> .	P

		<p>game.playerChoice = 2 <b>(player chose the Keep Career Path)</b></p> <p>game.getBoard ().chosenPath (game.getCurrentPlayer ().getSpaceNum ()) = 73 <b>(starting space of the Change Career Path)</b></p>			
Magenta Space Junction - Family Path	The player lands on the <i>Family Junction</i> . The player opts to start a family.	<p>game.getBoard ().isAtFamilyJunction (game.getCurrentPlayer ().getSpaceNum ()) = true</p> <p>game.playerChoice = 1 <b>(player chose the Start A Family Path)</b></p> <p>game.getBoard ().chosenPath (game.getCurrentPlayer ().getSpaceNum ()) = 108 <b>(starting space of the Start A Family Path)</b></p>	The player will move to the starting space (108) of the <i>Start A Family Path</i> .	The player will move to the starting space (108) of the <i>Start A Family Path</i> .	P
	The player lands on the <i>Family Junction</i> . The player opts to not start a family.	<p>game.getBoard ().isAtFamilyJunction (game.getCurrentPlayer ().getSpaceNum ()) = true</p> <p>game.playerChoice = 2 <b>(player chose the Don't Start A Family Path)</b></p> <p>game.getBoard ().chosenPath (game.getCurrentPlayer ().getSpaceNum ()) = 124 <b>(starting space of the Don't Start A Family Path)</b></p>	The player will move to the starting space (124) of the <i>Don't Start A Family Path</i> .	The player will move to the starting space (124) of the <i>Don't Start A Family Path</i> .	P
Retirement Space	The player lands on the <i>Retirement Space</i> and the number of movements needed as spinned by the	<p>game.getBoard ().isAtRetirementSpace (game.getCurrentSpace ().getType ()) = true</p>	Player will STOP! Immediately proceed to retirement.	Player will STOP! Immediately proceed to retirement.	P

	wheel is not yet achieved.	game.getWheel ().getNumber () = 5  number of movements achieved by the player (counter) = 3			
	The player proceeds to the retirement process with the final player stats	game.getCurrentPlayer ().getCash = <b>900000</b>  game.getCurrentPlayer ().getLoanBalance () = <b>100000</b>  game.getCurrentPlayer ().getStatus ().getHouseCard ().getPrice = <b>300000</b>  game.getCurrentPlayer ().getStatus ().getFamily () = <b>4</b>  game.retirePlace = <b>1</b> <b>(player is the first to retire, get 100000)</b>	Player will have a final cash of \$1240000 as a result from retirement and with no outstanding loan balance.	Player will have a final cash of \$1240000 as a result from retirement and with no outstanding loan balance.	P
	The player proceeds to the retirement process with the final player stats, there are 3 players in the game.	game.getCurrentPlayer ().getCash = <b>0</b>  game.getCurrentPlayer ().getLoanBalance () = <b>100000</b>  game.getCurrentPlayer ().getStatus ().getHouseCard () = <b>null</b> <b>(no house owned)</b>  game.getCurrentPlayer ().getFamily () = <b>0</b> <b>(no children)</b>	Player will have a final cash of \$20000 as a result from retirement and with an outstanding loan balance of \$100000 as player cannot pay the loan balance given the final cash.	Player will have a final cash of \$20000 as a result from retirement and with an outstanding loan balance of \$100000 as player cannot pay the loan balance given the final cash.	P

		game.retirePlace = 3 <b>(player is the last to retire, get 20000)</b>			
	The player proceeds to the retirement process with the final player stats, there are 2 players in the game.	game.getCurrentPlayer ().getCash = <b>500000</b>  game.getCurrentPlayer ().getLoanBalance () = <b>500000</b>  game.getCurrentPlayer ().getStatus ().getHouseCard ().getPrice = <b>900000</b>  game.getCurrentPlayer ().getStatus ().getFamily () = <b>2</b> game.retirePlace = 2 <b>(player is the last to retire, get 50000)</b>	Player will have a final cash of \$970000 as a result from retirement and with no outstanding loan balance.	Player will have a final cash of \$970000 as a result from retirement and with no outstanding loan balance.	P