# Scenario – Bug Reports

## Scenario Description

* In this document, all the reported bugs are tested using user acceptance testing methodology
* Each bug have its own test script, with pre condition, post condition, require data and processes, expected result and test result with necessary teardown steps

## Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| 1.0 | 09/10/2017 | Binaya Subedi | Initial Draft |
| 1.1 | 10/10/2017 | Binaya Subedi | Initial Version |

## Test Scripts

The following script will cover the scenario

* 1. Game doesn’t pay out current level
  2. Player cannot reach betting limit
  3. Odds in the game do not appear to be correct
  4. Combination of three symbols never change after first roll
  5. Spade is never selected, and only other five symbols are randomly selected

## Use Case

* **Playing the Game**

## Test Components/Requirements

This test scenario covers the following high-level test requirements (see scripts below for specific requirements covered by each test script):

* **Player to play the game**

## User Group

* **Players playing game**

## Script 1: Bug – Game does not pay out at correct level

### Script Description

Player will launch the game, and if he wins the match, player balance should increase adding bet value to the player current balance.

### Testing Requirements

In this test, first player will choose the symbol to bet on, if the player wins the match that means rolled dice symbols matches the player selected symbol, thus the player balance should be increased ,with below specified rules.

|  |  |
| --- | --- |
| if the symbol appears on the uppermost face of 3 dice | 3 to 1 |
| if the symbol appears on the uppermost face of 2 dice 2 to | 2 to 1 |
| if the symbol appears on the uppermost face of 1 dice 1 to 1 | 1 to 1 |

### Setup

* A user should be created with balance greater than allowed limit which is great than 0.
* 3 dice should be initialized and ready to roll.

### Tear Down

* At the end of the game, there are different results, according to the dice symbols.
* If player gets the selected symbol, in 1 dice out of 3, then user balance should increase with bet amount, which in this case 5
* If player gets the selected symbol, in 2 dice out of 3, then user balance should increase with bet amount multiply by 2, which in this case is 10
* If player gets the selected symbol, in 3 dice out of 3, then user balance should increase with bet amount multiply by 3, which in this case is 15

### Script Steps

| **Step #** | **Test Action** | **Expected Results** | **Pass/ Fail** |
| --- | --- | --- | --- |
| 1 | Create the player | Player is create with name ‘Fred’, with initial balance 100 | Pass |
| 2 | Assign the pick, one symbol to player | Random symbol should be assigned to player. | Pass |
| 3 | Three dice should be initialized, and rolled | Three symbols which mean names should be returned. | Pass |
| 4 | Check the pick against the three dice value. | Give winning to the player according to the value matched:-  If pick is matched to 1 dice, add 5 to balance  If pick is matched to 2 dices, add 10 to balance  If pick is matched to 3 dices, add 15 to balance | Pass |
| 5 | Check player balance | Check if the balance increases according to the pick and matched dice values | Fail |

### Test Execution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date/Time | Tester | Test ID | Test Phase | Status |
| 10/10/2017 08:30pm | Binaya | PlayerBalanceTest1 | System Cycle 1 | Fails |

## Script 2: Bug – Player cannot reach betting limit

### Script Description

Player will play until, the balance is less than the betting amount.

### Testing Requirements

In this test, player should be able to play the game, up to the point where his balance is equal to the bet he has placed

### Setup

* Player should be created with name and initial balance
* Bet should be placed

### Tear Down

* At the end of the game the player balance should be zero, or less than bet amount.

### Script Steps

| **Step #** | **Test Action** | **Expected Results** | **Pass/ Fail** |
| --- | --- | --- | --- |
| 1 | Create the player | Player is create with name ‘Fred’, with initial balance 100 | Pass |
| 2 | Pick betting amount | The default betting amount is 5 | Pass |
| 3 | Check player balance limit | User should be able to play until the balance is less than bet amount | Fail |

### Test Execution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date/Time | Tester | Test ID | Test Phase | Status |
| 12/10/2017 10:00 pm | Binaya | PlayerBalanceLimitTest1 | System Cycle 1 | Fails |

## Script 3: Bug – Win/Lose ratio doesn’t equals 0.42

### Script Description

At the end of the game, the win+lose ratio should be o.42

### Testing Requirements

In the game, the game is 8% biased towards the house, thus the ration of winning a match should equals 42 percent.

### Setup

* Game should be started
* Game should end whether the balance is 0 or the balance exceeds maximum limit, which is 200.

### Tear Down

* At the end of the game, the win/lose ratio should be 42%.

### Script Steps

| **Step #** | **Test Action** | **Expected Results** | **Pass/ Fail** |
| --- | --- | --- | --- |
| 1 | Start the game | Initial setup must be done to start the game | Pass |
| 2 | End of the game | Player balance is 0 or 200 | Pass |
| 3 | Win+lose ratio is 42 percentage | The user winning the game, should be 42 percentage of total turn. | Fail |

### Test Execution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date/Time | Tester | Test ID | Test Phase | Status |
| 13/10/2017 10:00 pm | Binaya | WinLoseratiotest | System Cycle 1 | Fails |