Lucky Unicorn

### Graphical user interface, text, application Description automatically generatedOutline / Decomposition

### Graphical user interface, application Description automatically generated

### Flowchart

*Please show a developed flowchart of your program below (you may use draw.io to create your flowchart)*

### Version Log

*Your version log should go here. Annotated screenshots are a good idea at this point*

### Component Testing

*Show that you have tested each component here. You should have a test plan and then screenshot proof for each component. You should also include notes that justify the major decisions you made.*

**Instructions (and yes / no checker) Test Plan**

|  |  |
| --- | --- |
| **Test Case** | **Expected Values** |
| Have you played before? Maybe.  Have you played before? Yes | <error> Please choose y / n  Game Starts |
| Have you played before? No | Displays instructions and starts game |

**

**Token Generator Test Plan ( Check tokens randomly generated)**

|  |  |
| --- | --- |
| **Test Case** | **Expected Values** |
| Run the program | Program should output 20 tokens with at least one horse, donkey, zebra, and unicorn. Order of tokens should be random. |

**Round Test Plan**

For testing purposes, each round the user will lose $1.00 (ie: they get a donkey each time)

|  |  |
| --- | --- |
| **Test Case** | **Expected Values** |
| Starting amount: $5 Play until money runs out. | Game should go for five rounds. Each round should be correctly numbered. After 5 rounds, ‘Sorry you have run out of money’ should display |
| Start  ing amount: $5 Play 3 rounds | Game should go for 3 rounds. When user types ‘xxx’ (for round #4), game should end. Finale balance should be $2 |

**Rounds Testing**

**Text

Description automatically generated**

Ran out of money so game ends

Text

Description automatically generated

Exit code works correctly will add a ‘thanks for playing’ message.

Text

Description automatically generated

**Trial #1:** Program randomly chooses from a list of horse, unicorn, donkey and zebra.

**Token Generator Test Plan (Check house has an advantage)**

|  |  |
| --- | --- |
| **Test Case** | **Expected Values** |
| User starts with $100  Run progam | User should lose money (final balance should be less than $100) |

**Trial #2:** Results for 100 rounds, 10% chance of a unicorn, 30% change for each of donkey, zebra and horse.

****

**Trial #3:** Testing on shows only 10 rounds, 5% change of unicorn, 30# change for donkey, and 65% change of horse / zebra and when run more than 100 rounds house advantage becomes clear. Text

Description automatically generated

**Trial 3 when run for 100 rounds**

****

|  |
| --- |
| **Token Generator** |
| The payout schedule is…  Donkey: lose $1.00  Horse / Zebra: lose $0.50  Unicorn: win $4.00 (ie: if user starts with $1 and gets a unicorn, their balance will be $5.00) |

**Yes / No Checker Test Plan …**

*Table

Description automatically generated*

*Text

Description automatically generated*

**Table

Description automatically generatedNumber Checker**

*Text

Description automatically generated*

**Statement Decorator Test Plan**

|  |  |
| --- | --- |
| **Test Case** | **Expected** |
| Statement 1: Welcome to the Lucky Unicorn Game  Decoration: “\*”  Statement 2: Congratulations you got a Unicorn  Decoration “!” | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*Welcome to the Lucky Unicorn Game\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  !!! Congratulations you got a Unicorn !!!  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! |

### **Yes / No trialling**

Initial Trial: The code worked but uppercase didn’t work.

Trial 1 : Added both Upper and lower case worked and used or command to make code be able to accept yes and y and no and n statements.

Trial 2 : Put the code into a function which makes it easy to ask more than yes / no question in a program and added a question that asks the user if their having fun.

### Testing

### Text Description automatically generatedAssembled Outcome Testing

|  |  |
| --- | --- |
| **Test Case** | **Expected** |
| Run Program  Played before? **Maybe**  Instructions? Yes  Amount? Xlii, 0.5, 11, 1  Play until money runs out | Welcome message displays  Error – please type y / n  Error for xlii, 0.5, 0 and 11  Random Token generated and score updated ( most likely outcome is a single round game unless we get a unicorn)  Stats show starting amount of 1 and balance of either $0.50 or $0.00 |

### Assembled Outcome Testing part 2

|  |  |
| --- | --- |
| **Test Case** | **Expected** |
| Run Program  Instructions? N  Amount? 10  Play until balance is $3.00 /  $2.50, then press ‘xxx’ to quit | Welcome message displays  Instructions are displayed  Random Token generated each time user presses <enter> and balance is updated (lose $1 for donkeys, lose 50c for horse / zebra and gain $4.00 for unicorns). Balance should decrease over time  Stats show starting amount of $10 and final balance of either $2.50 or $3.00 |

*Text

Description automatically generated*

*Graphical user interface, application

Description automatically generated*

*Graphical user interface

Description automatically generated with medium confidence*

[*https://github.com/Fange-Wu/01-Lucky\_Unicorn*](https://github.com/Fange-Wu/01-Lucky_Unicorn)