Documentation Template

|  |
| --- |
| ***You MUST provide evidence showing how the problem has been decomposed, how the components have been developed and trialled, and of how they have been assembled and tested to create a final, working outcome.*** |

Getting Lucky

**Game Summary**

Choose number of rounds 10-30, objective is to reach the specified money goal based off the rounds within the number of rounds.

The money goal = rounds \* 10

Each round user gets to choose 4 options, Loot, Steal, Job, Shop. Choosing any options except shop will pass a round. E.G. Round 1: User chooses Loot, game now moves on to round 2.

**Loot:** gives you a chance to get a medium amount of money but also a chance to not get any money that round.

(Generate random, 80% chance to gain $8 and 20% chance to get nothing that round.)

**Steal:** gives you a chance to get a money but also a chance to lose money that round.

(50% chance to lose 30% of current balance or 50% chance to gain 30% of current balance)

**Job:** Nothing to lose but have to answer a question like 2+2 to get a small amount of money.

($6 if user gets question right and if wrong user gains nothing during that round.)

**Shop:** User can access a shop which displays a menu of upgrades with costs.

(Shop sells upgrades:

Increase reward from loot by $3 for $10

Increase percentage to get money from steal by 10% for $15

Increase reward to $10 from jobs for $20)

When game ends user gets asked if they want to play again.

### Outline / Decomposition

*Please write down your task decomposition here (a numbered list is a good idea)*

1. Base Component
2. Welcome User
3. Yes and No checker
4. Instructions
5. Ask user how many rounds to play (A whole number more than or equal to 10 and less than or equal to 20)
6. Game checks for loot, steal, job, shop and XXX each round.
7. Generate random for loot
8. Workout percentage of balance for steal and 50/50 chance to lose or gain money
9. Generate random question for job
10. Work out balance after each round
11. When user inputs shop display menu of upgrades with cost
12. When user buys from shop minus from balance and apply the upgrade
13. Looping the game / ask user to play again

### 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.*

### Assembled Outcome Testing

*Please show testing for your assembled outcome below. This should include a test plan followed by screenshot proof*

### Usability Testing

*Write a list of things improvements which need to be made based on your usability testing. Then write down what you changed.*

### Post Usability Test…

*Show that your post usability testing program works correctly*

### Social and End User Considerations…

**How did you ensure that your task was suitable for your chosen audience?**

*Answer here*

**How have you honoured copyright?**

*Answer here*

**How did you make your quiz easy to use?**

*Answer here*