Project Title: The Tower of Cipher Breaking

Description: This is a decoding game.

Huffman tree - data decompression (binary number -> encoded text)

Background:

The player wants to rescue the person trapped on the top of the tower. They need to decode the message with a limited amount of money to rescue the person. They can choose the difficult level by themselves. In this process, they also learn how the huffman tree works.

Game Mechanics:

The player according to the huffman tree on the right to decode the message. Then they can guess the message. They can input the message they guessed. If verified, they can start to collect chars.

Then they choose the difficult level to collect the characters of the words with a limited amount of money. The simple level is unweighted, every path from char to char costs the same amount of money. The complex level is weighted, every path from char to char costs different amounts of money. They can use different hints - (BFS for unweighted, dijkstra… but if they choose dijkstra they will spend more money to buy).

The post-MVP might include negative weights.

Addition game elements:

Progress bar; Hints and time pressure;

Win the Game:

Once the player does the games three times, the people will be rescued. .

Similar Projects:

Cryptool:

Features:

Interactive Visualizations: CrypTool offers dynamic visual representations of cryptographic algorithms, aiding in comprehension.

Step-by-Step Tutorials: It provides guided walkthroughs for various cryptographic methods, enhancing user learning.

Hands-On Exercises: Users can engage in practical exercises to apply cryptographic techniques.

Huffman Encoder by NERDfirst Resources:

It allows users to input text and visualize the Huffman encoding process, including the frequency table and bit string output.

Features:

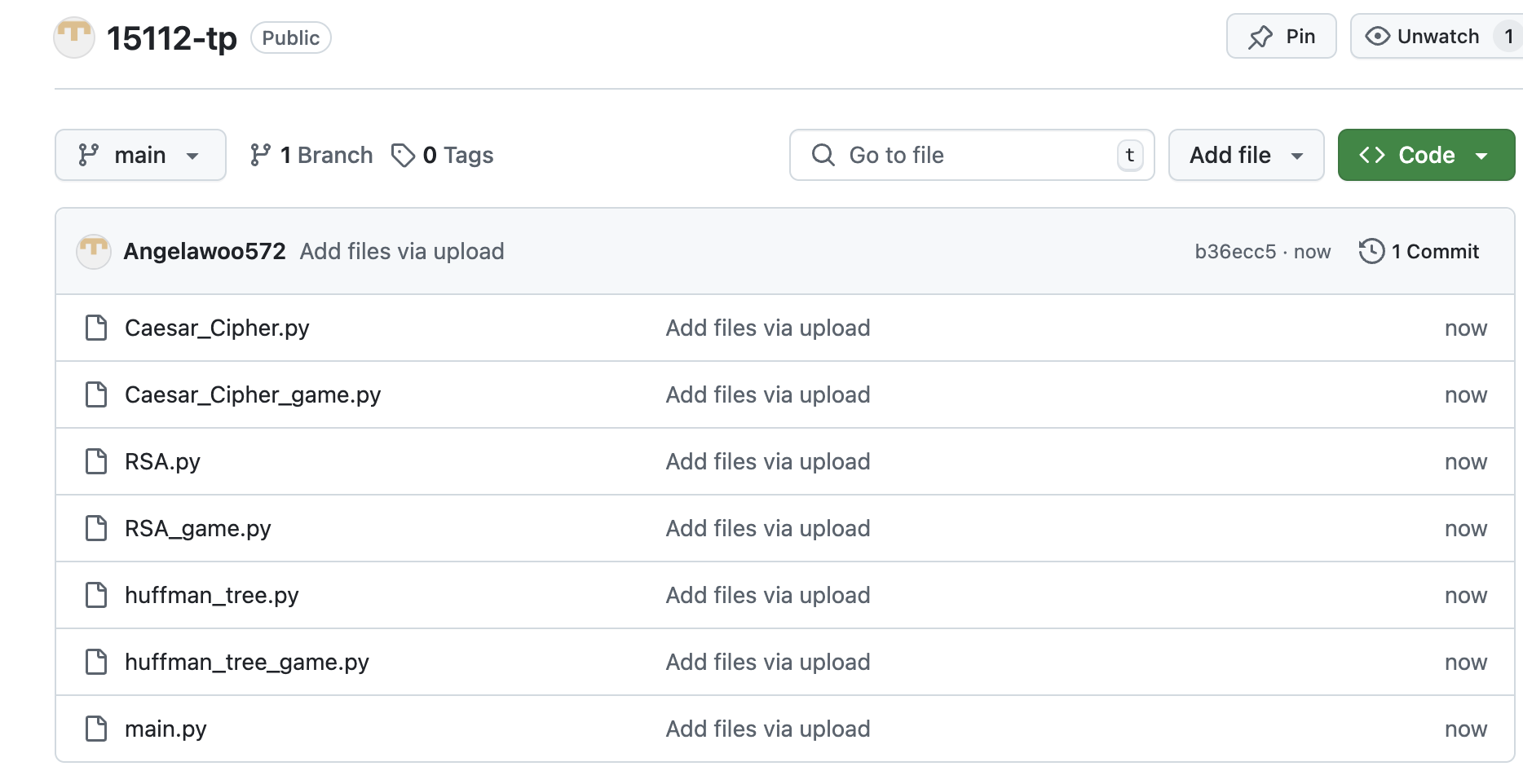
User Input Integration: Users can enter custom text to see how Huffman encoding applies to their data.

Comprehensive Output: The tool displays the frequency table, Huffman tree, and encoded bit string, providing a holistic view of the process.

Version Control / Backup Plan:

I am using Git version control to manage and back up my code regularly. My repository is hosted on GitHub, which allows me to keep my code safe in the cloud. Every day, I commit my latest changes and push them to the remote GitHub repository to ensure that my work is backed up and accessible from any device, in case something happens to my local machine.

Additionally, I periodically create zip archives of my project and upload them to Google Drive for an extra layer of safety. This ensures that I have multiple copies of my project stored in different places, reducing the risk of losing my progress.



Tech List:

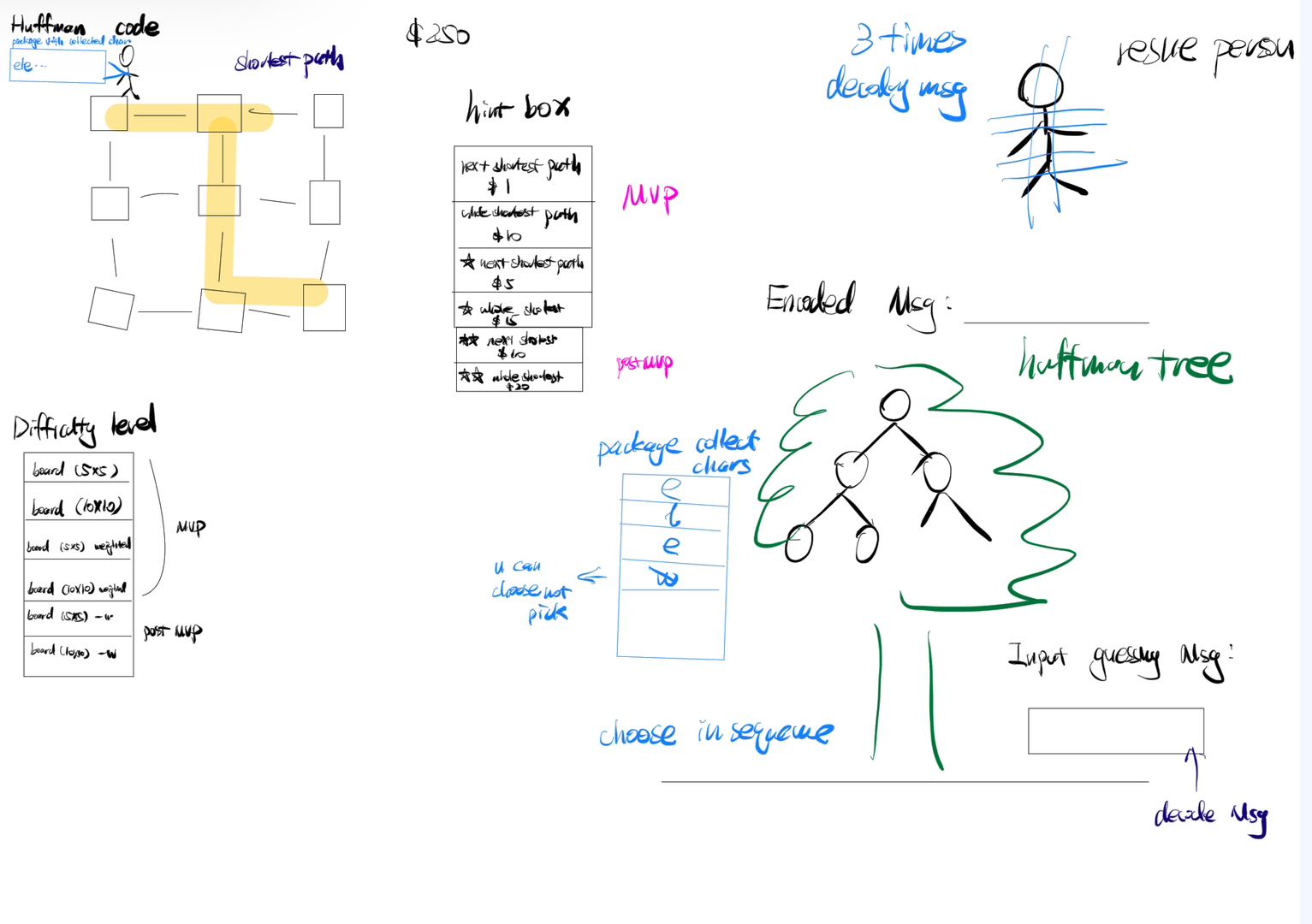
I think there are no other external modules.

I want to find a dictionary of English words modules (it might be realized after MVP).

StoryBoard:

Tower of Cipher Breaking

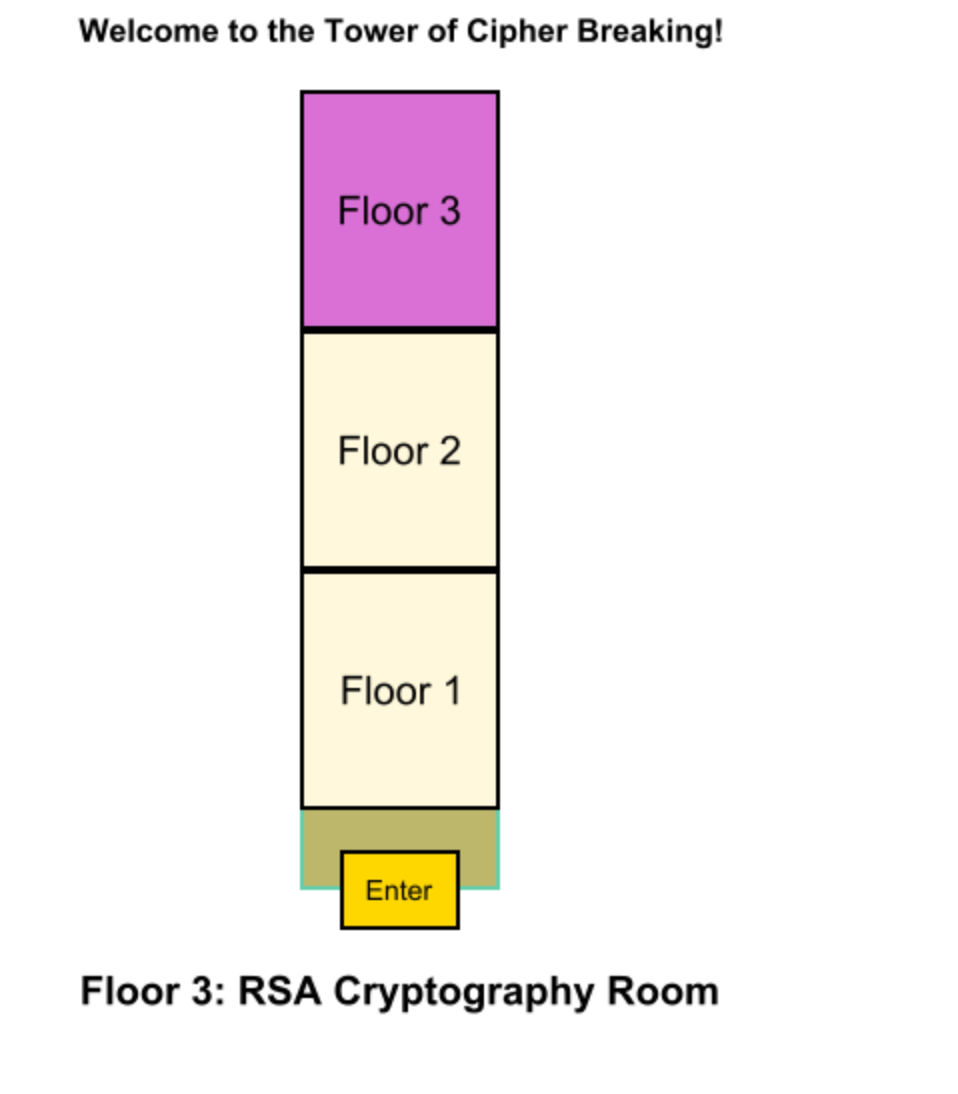
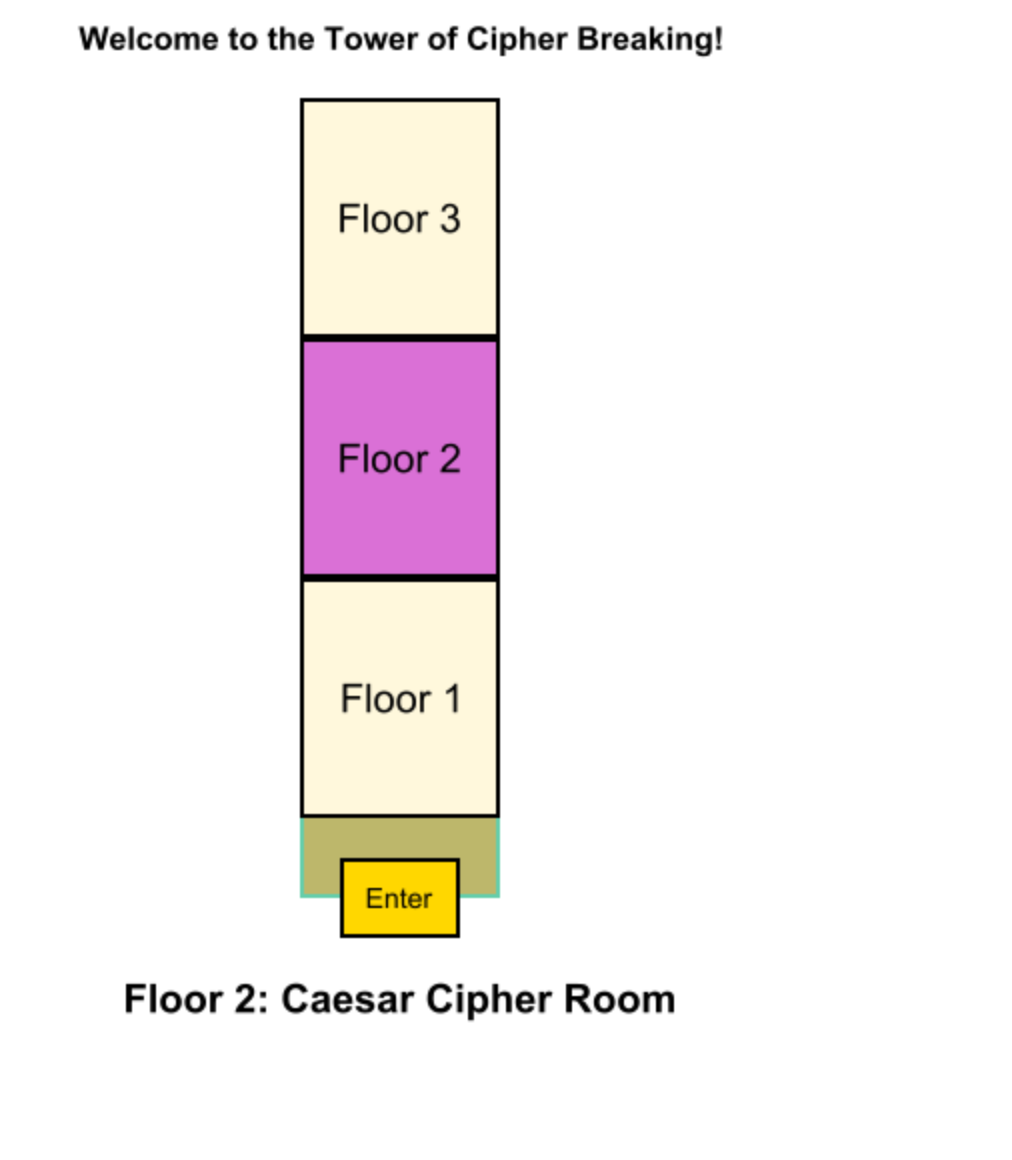
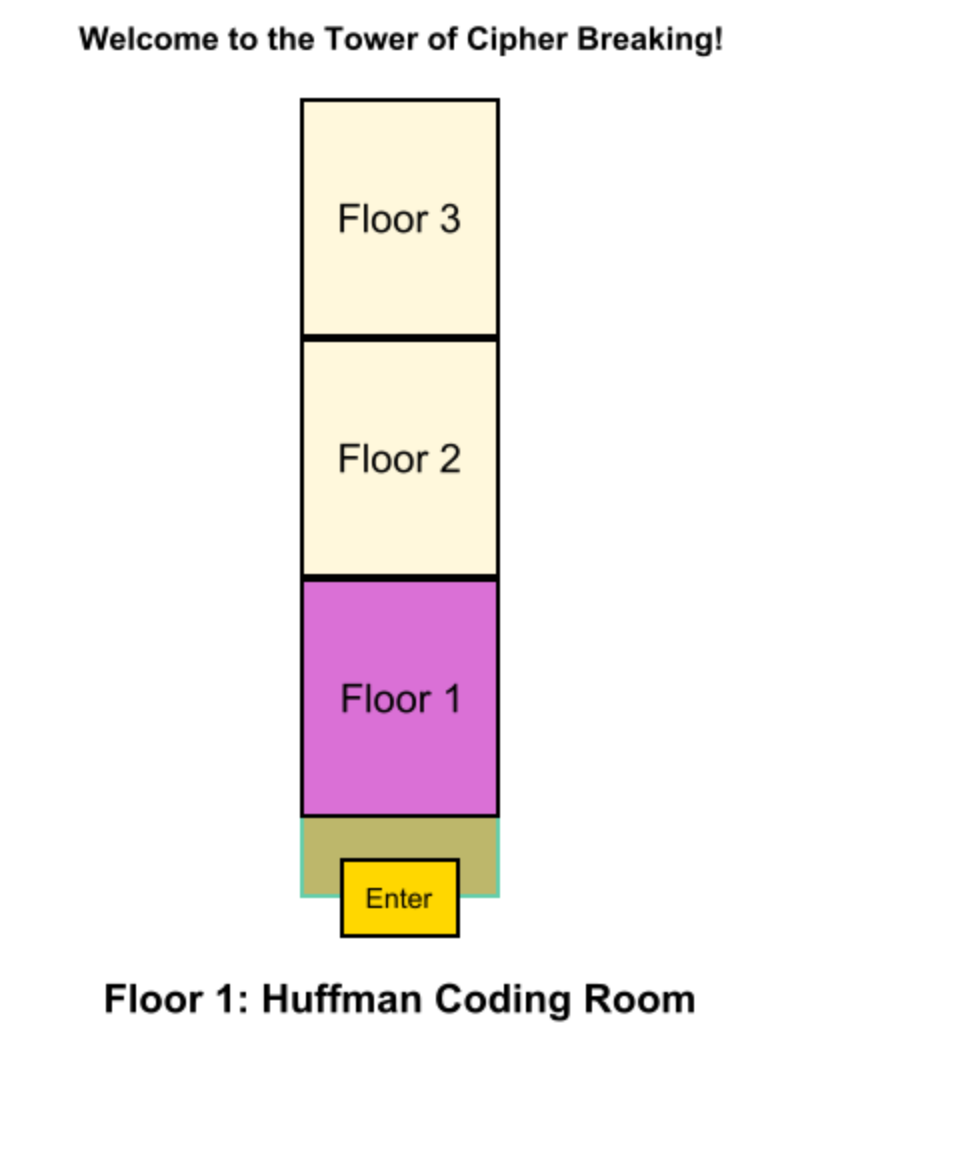
Goal: the player needs to ascend the tower and rescue the person.



Welcome Screen

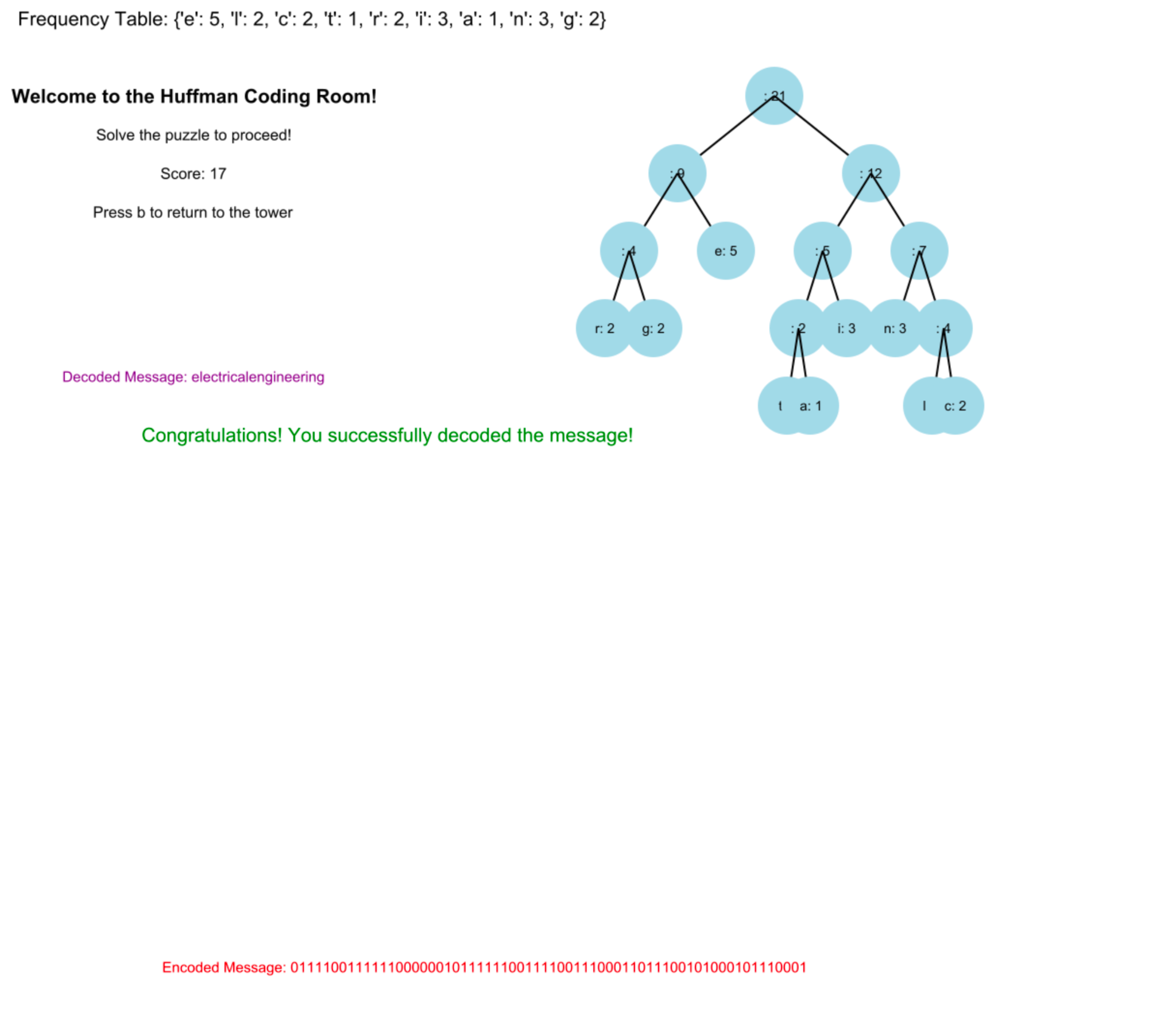
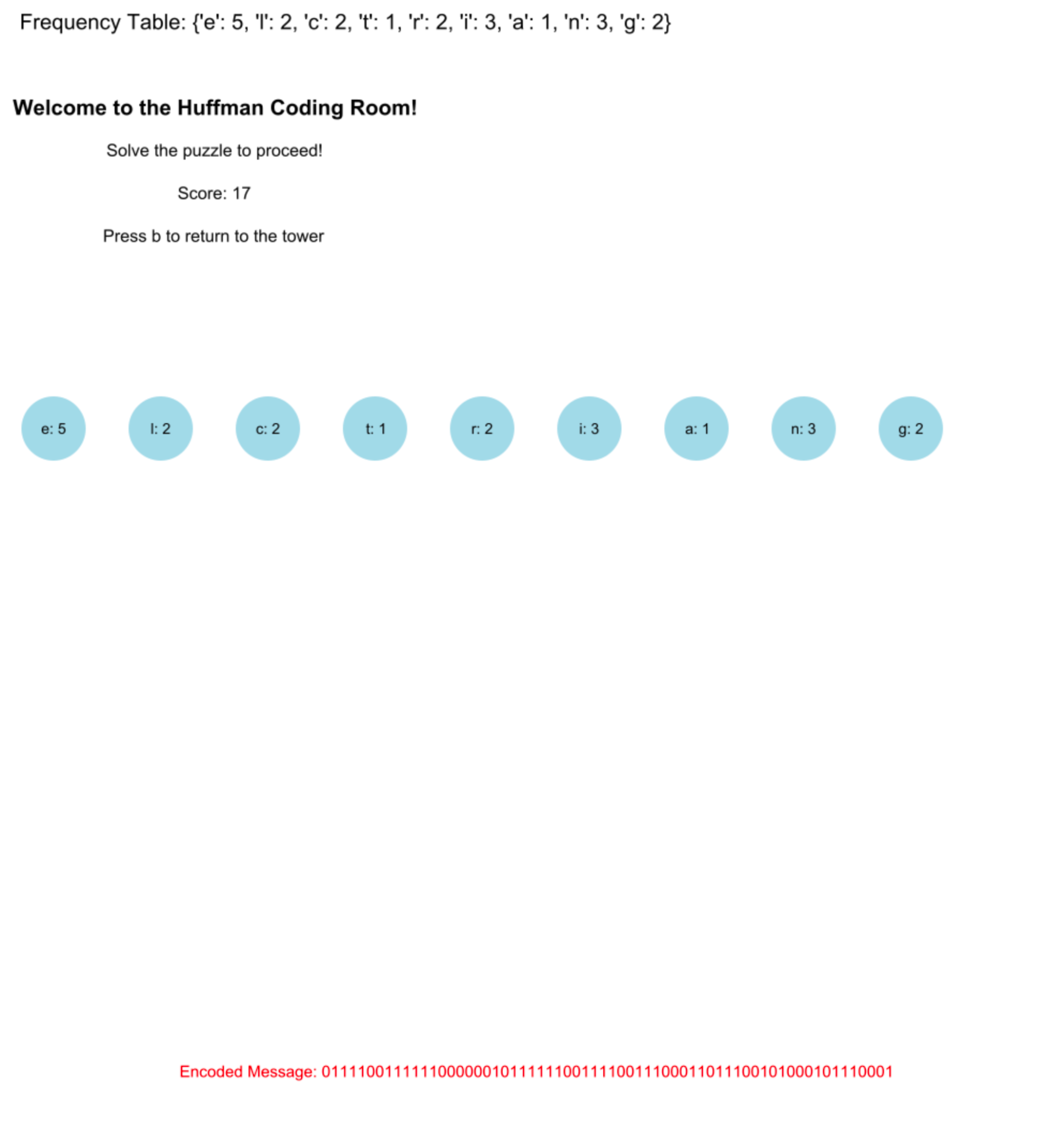
The player clicks space to start. There should be some characters that the player can pick.

1. Tower



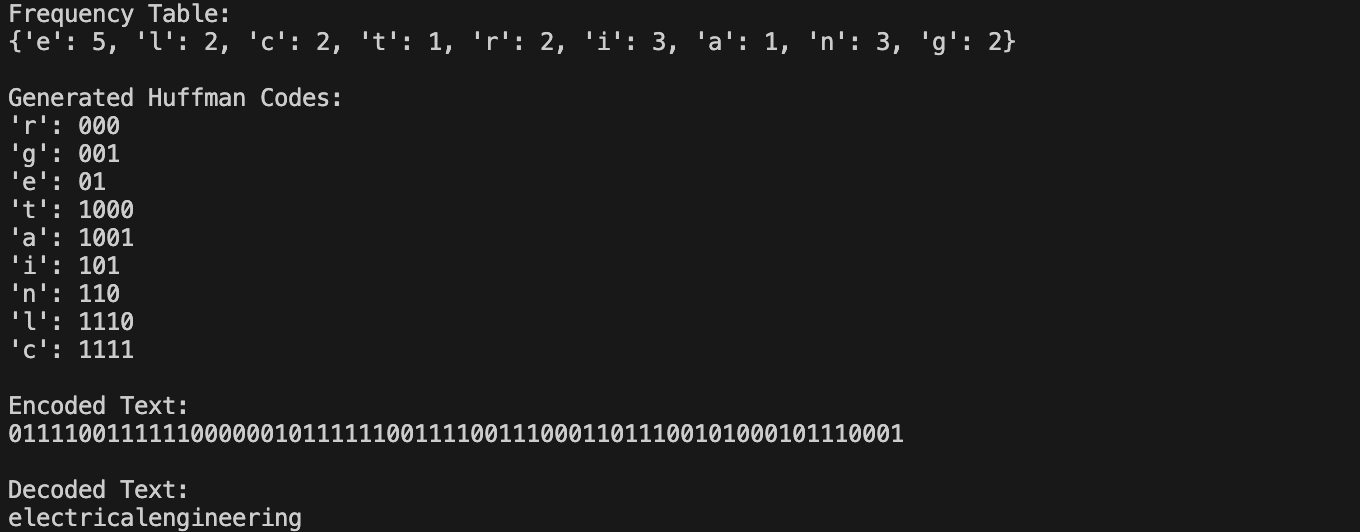
The tower I need to insert in the picture is good looking. Each time, when a player complete one floor challenge, it will go back to this page.

1. Huffman Coding Room



The player needs to be familiar with huffman principle. The player merges nodes to build a huffman tree - binary. They should merge from two nodes having the smallest frequencies.

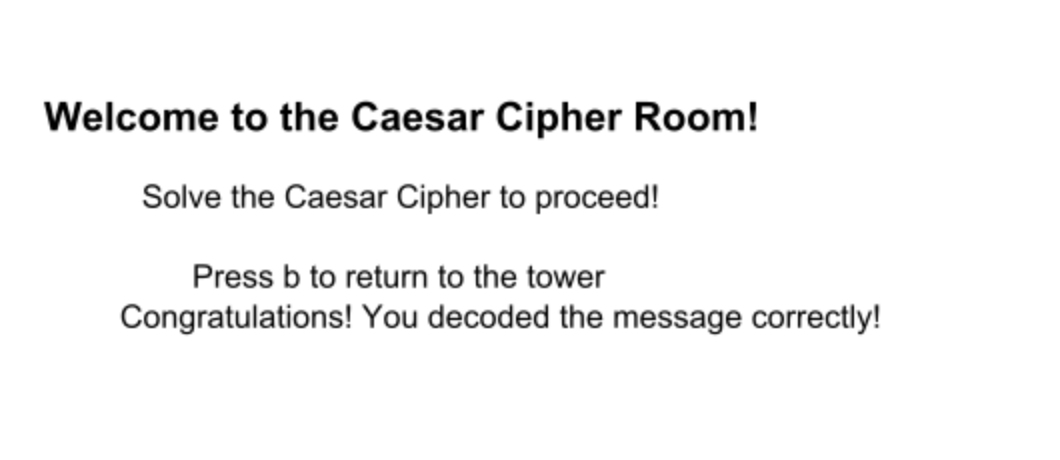
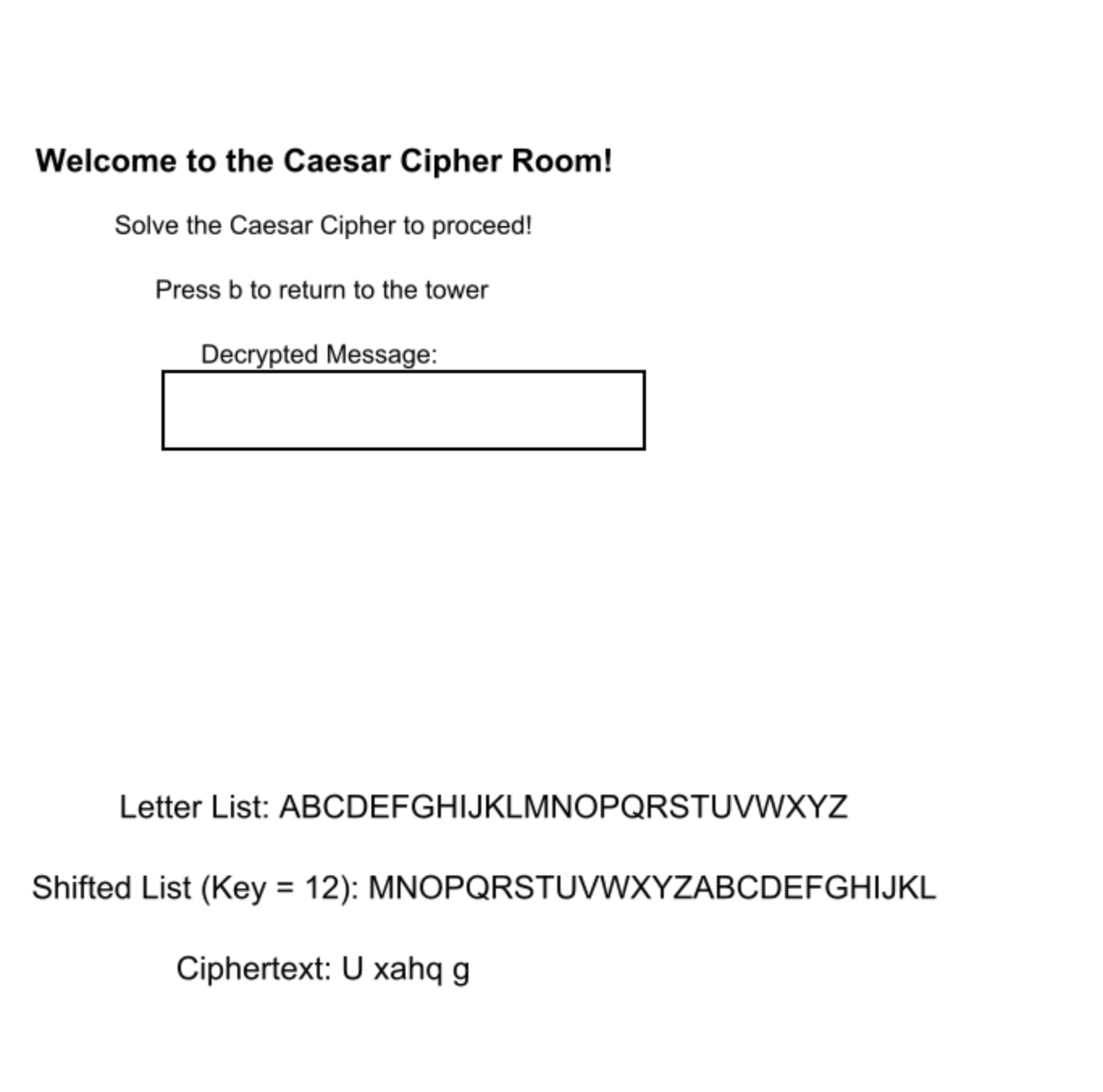
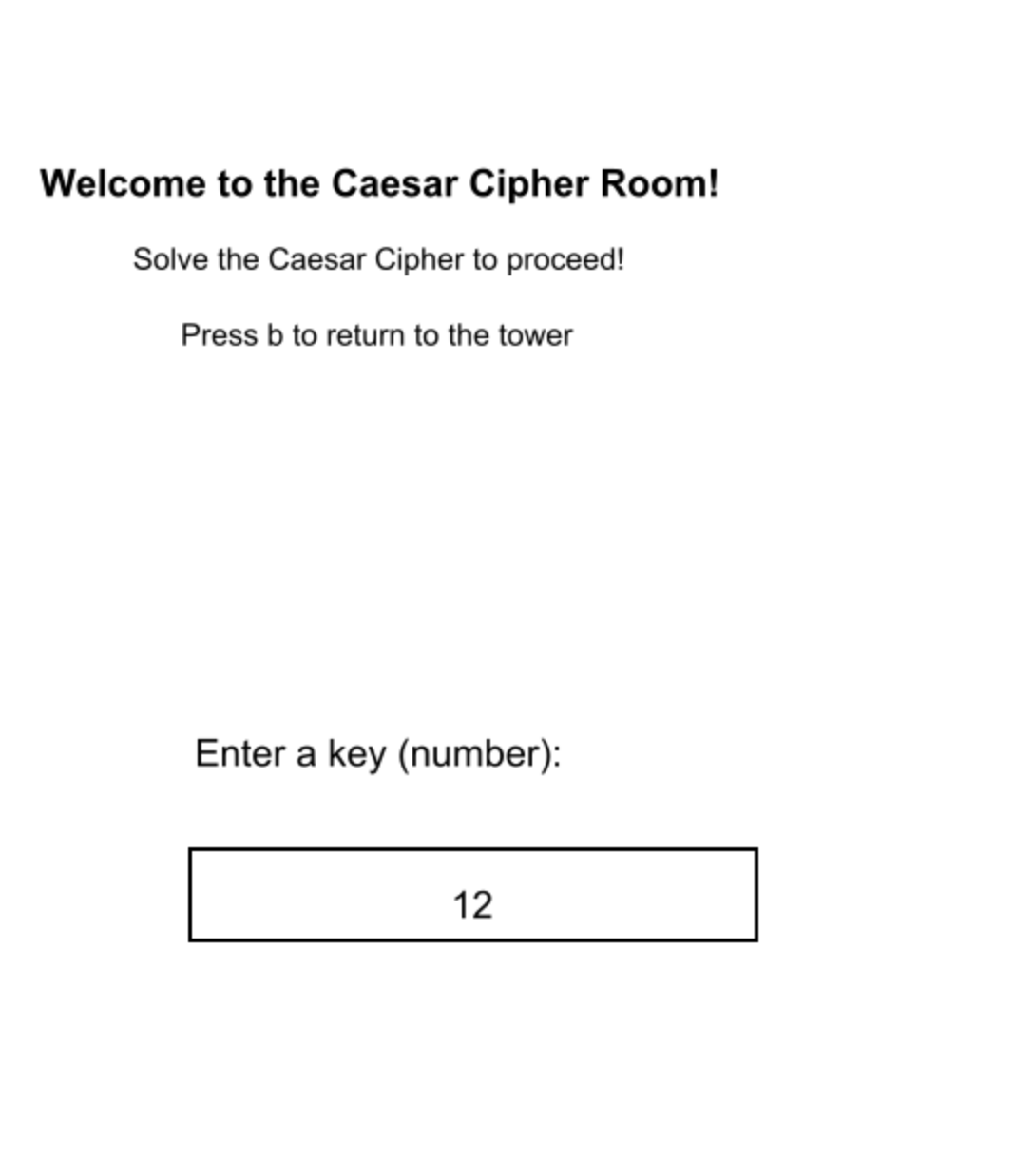
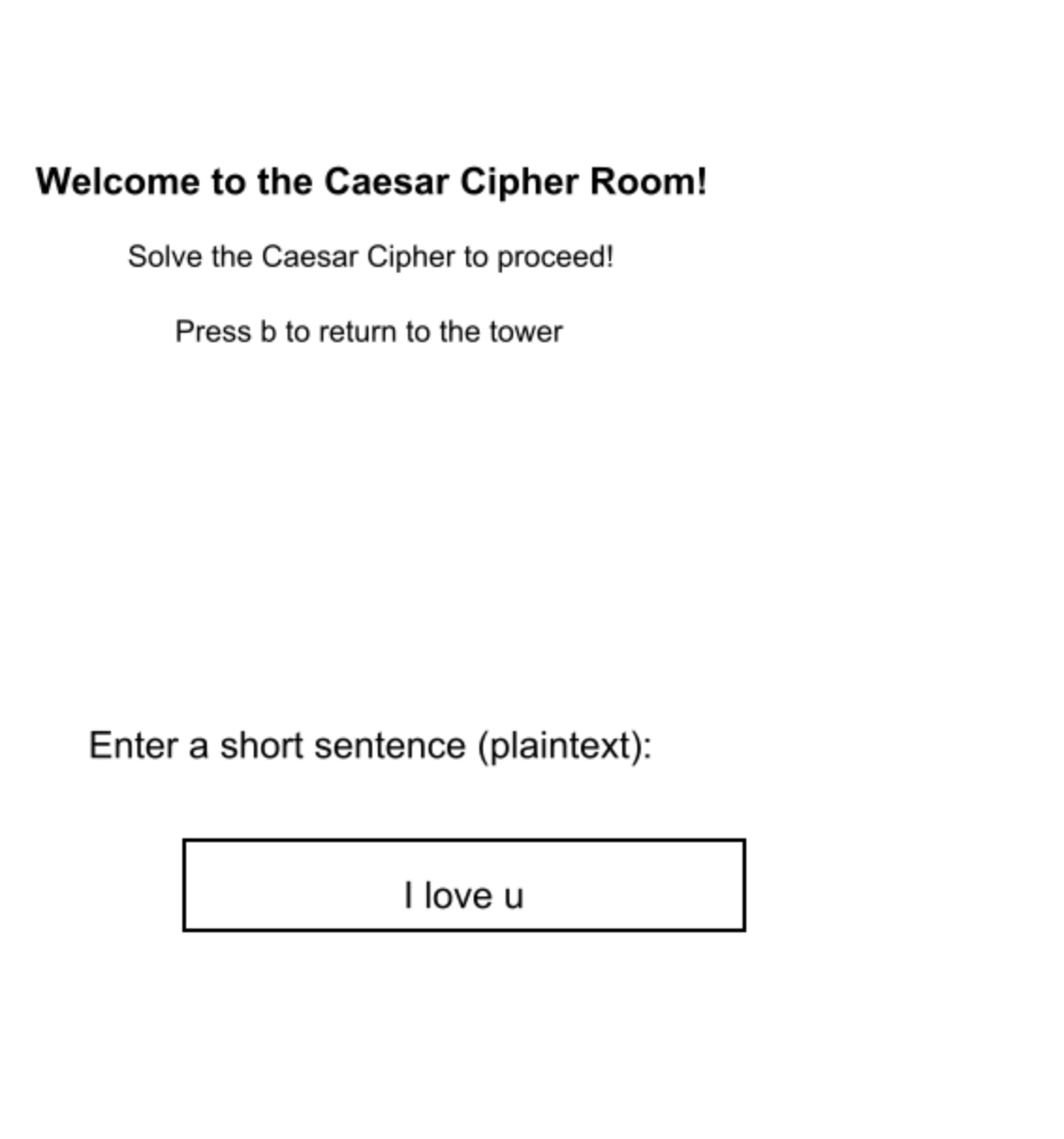
There should be some hints or instructions to tell the player. The player has three times to figure out the correct way to decompress data without losing points. At 4th time, they will lose points or fail to restart the game.



During the 3 times of trying, if the player did it correctly the first time, the added score should be the biggest. The second time, the “Generated Huffman Codes” will appear to hint the player to match with an encoded message. The third time, the player will be hinted that it’s a readable word.

I would like to think about if I can use DFS to explain the concept of traversing to every leaf in order to build the Huffman encoding table

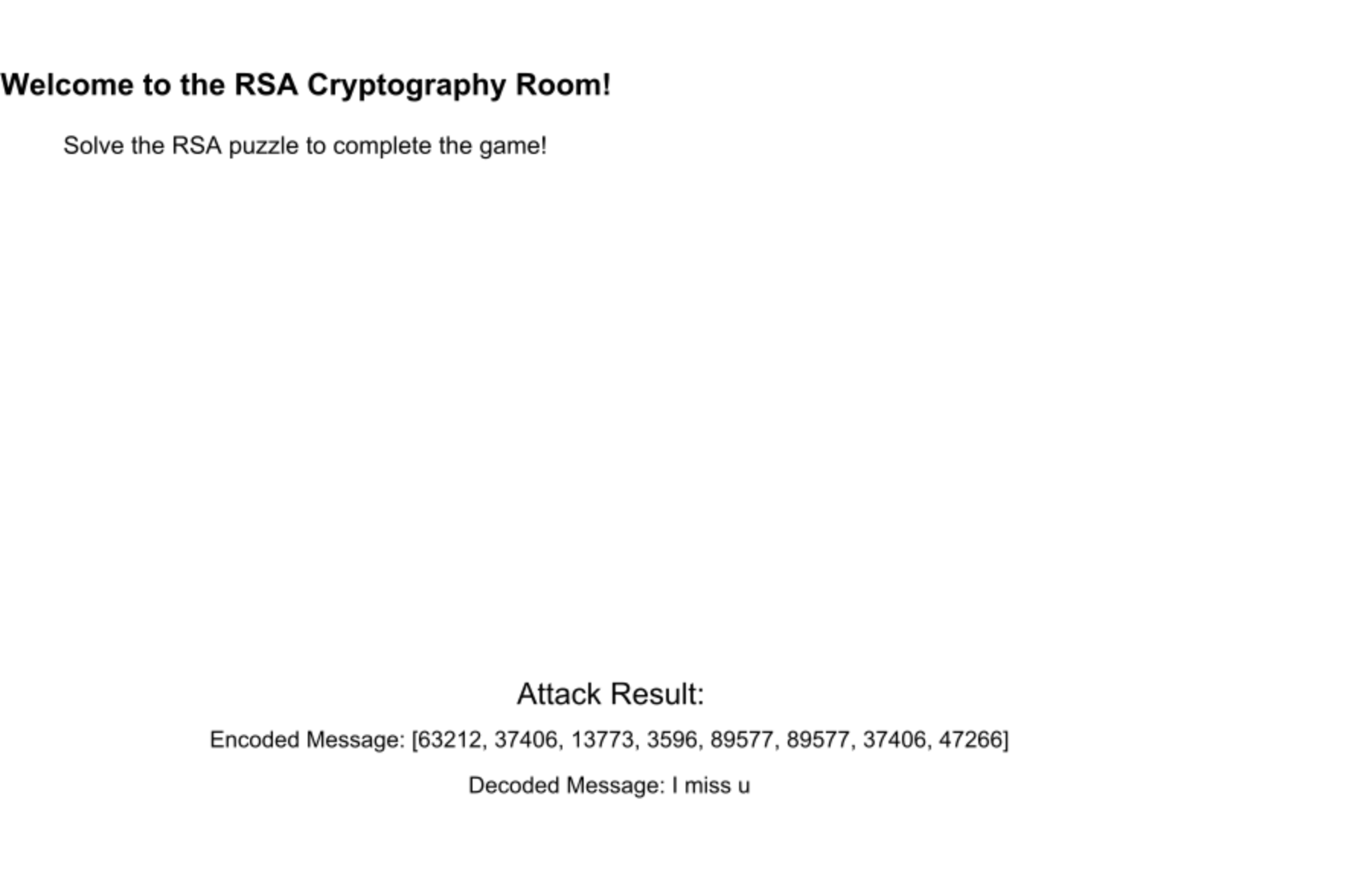
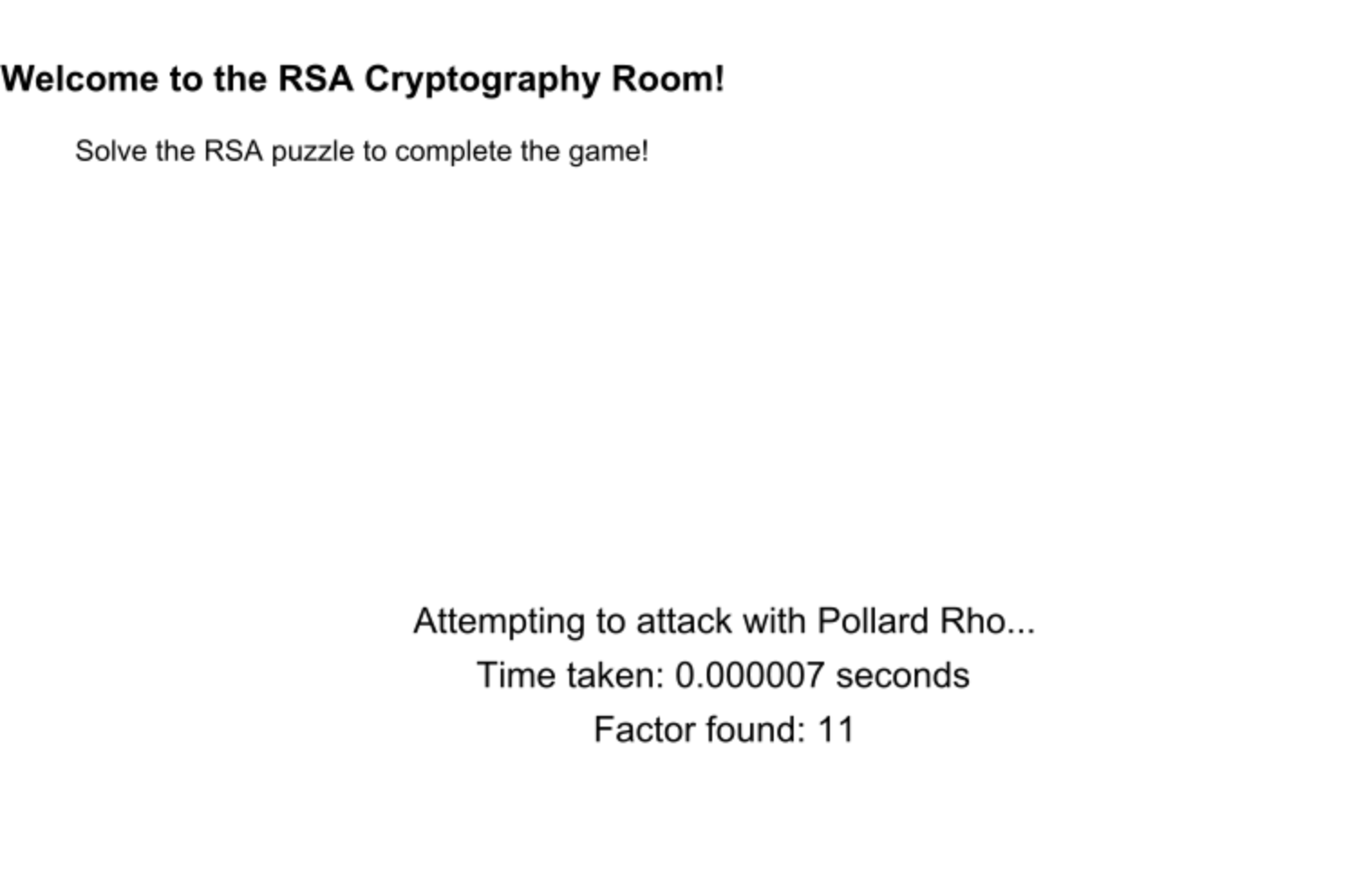
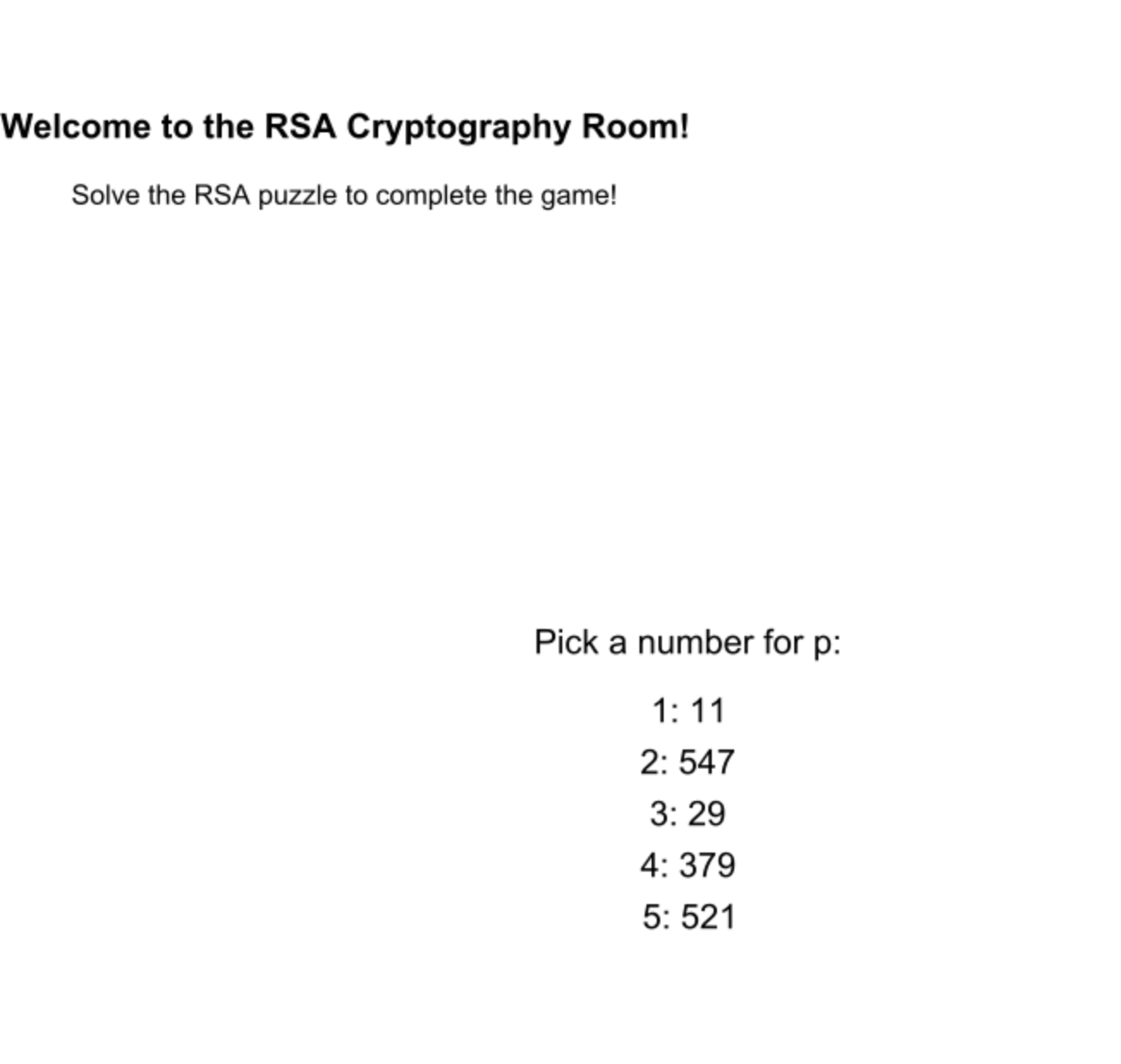
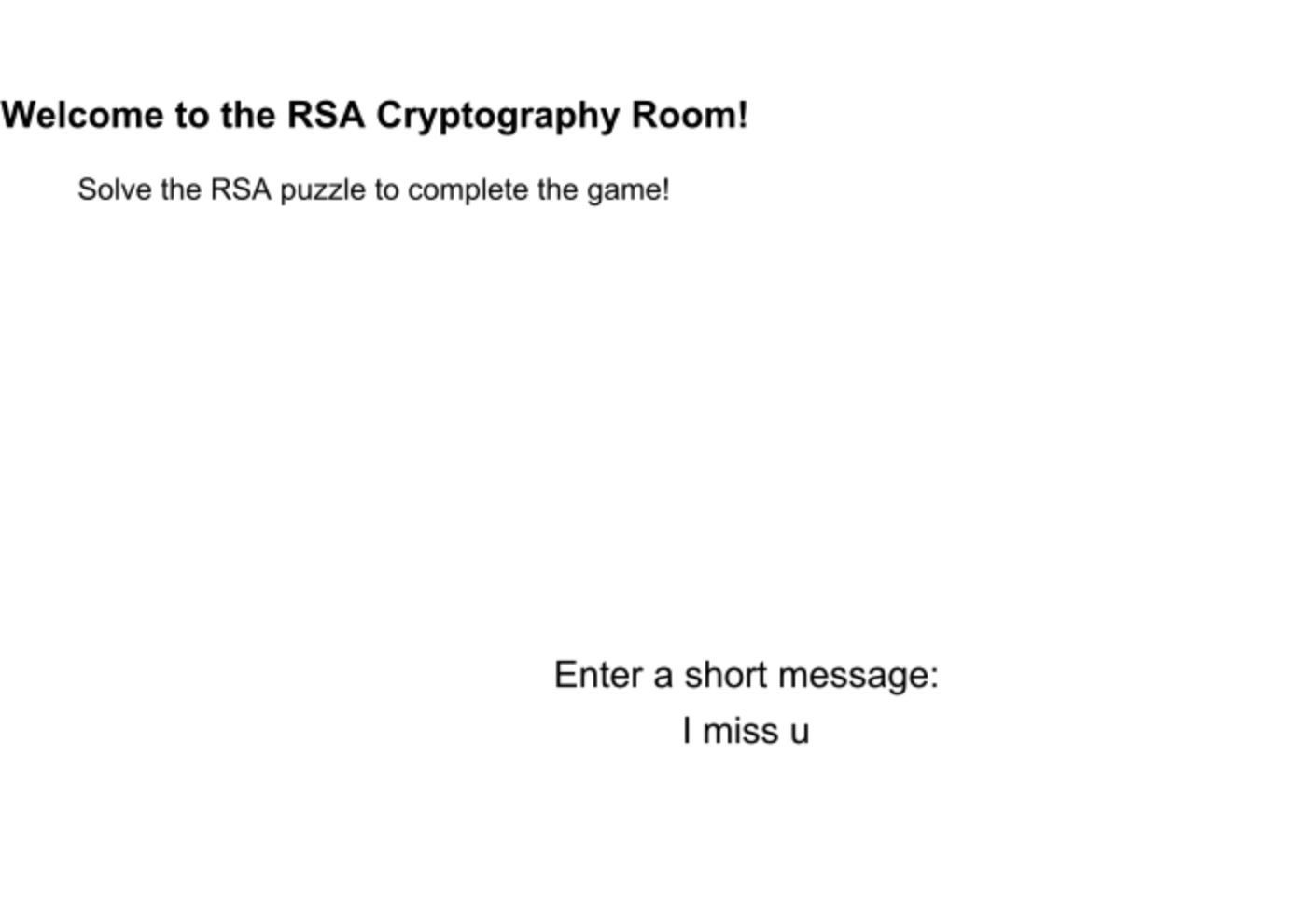
1. Caesar Cipher Room



In this part, the player inputs any sentence and any number they want, but they do not know the message they decode is the one they input. This part is easy for the player to find regulations, so the player only has 1 chance to get a score, otherwise fail/restart.

Maybe we should not use shifted lists - Implement BFS for systematically exploring all possible Caesar shifts. Let the user observe how each encoded message transforms until the correct plaintext is found.

1. RSA Cryptography Room



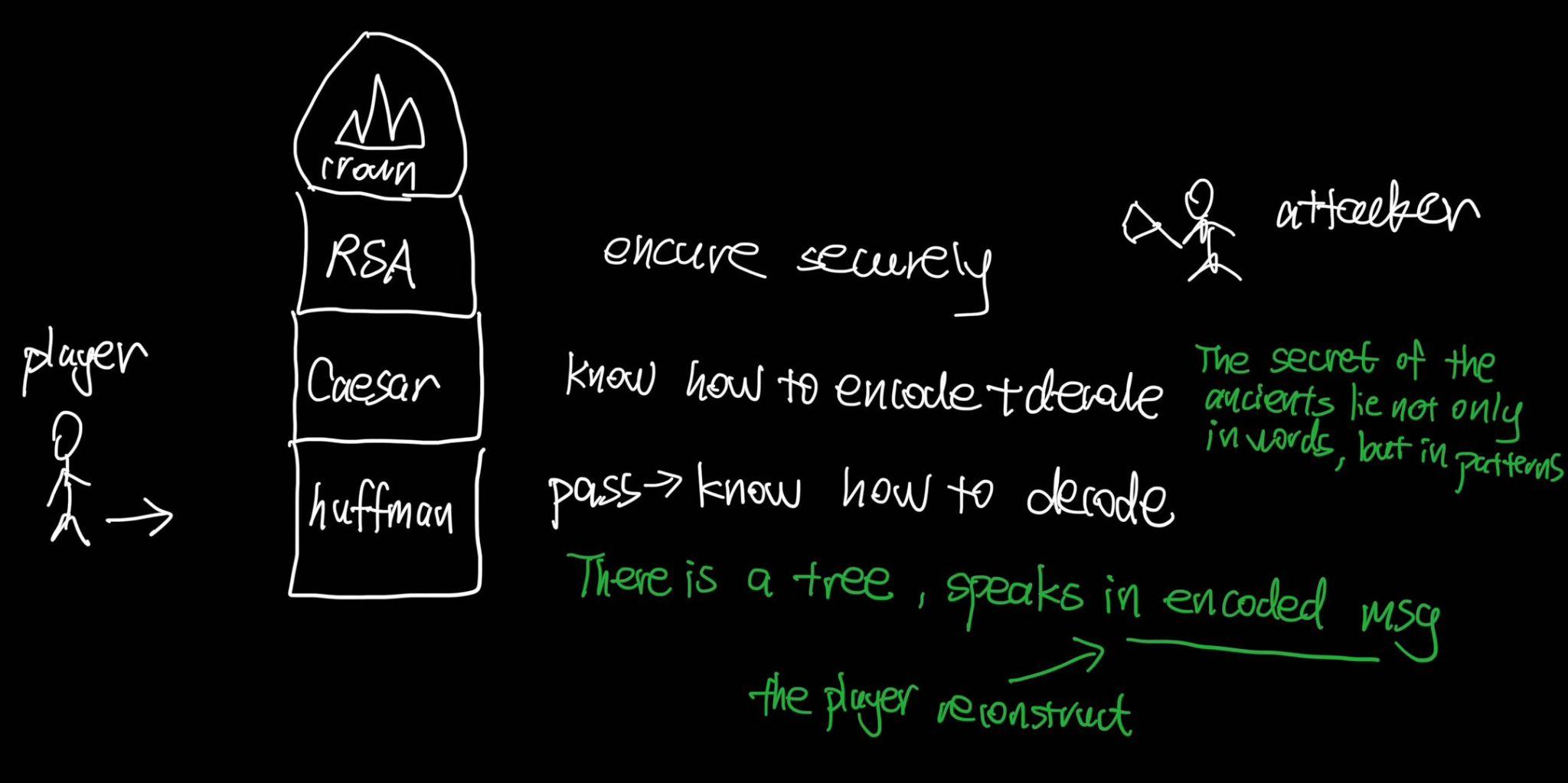
In this part, the player does not focus on decode the message. They should ensure the security of their message. Otherwise, another attacker will decode the message, and the player will fail.

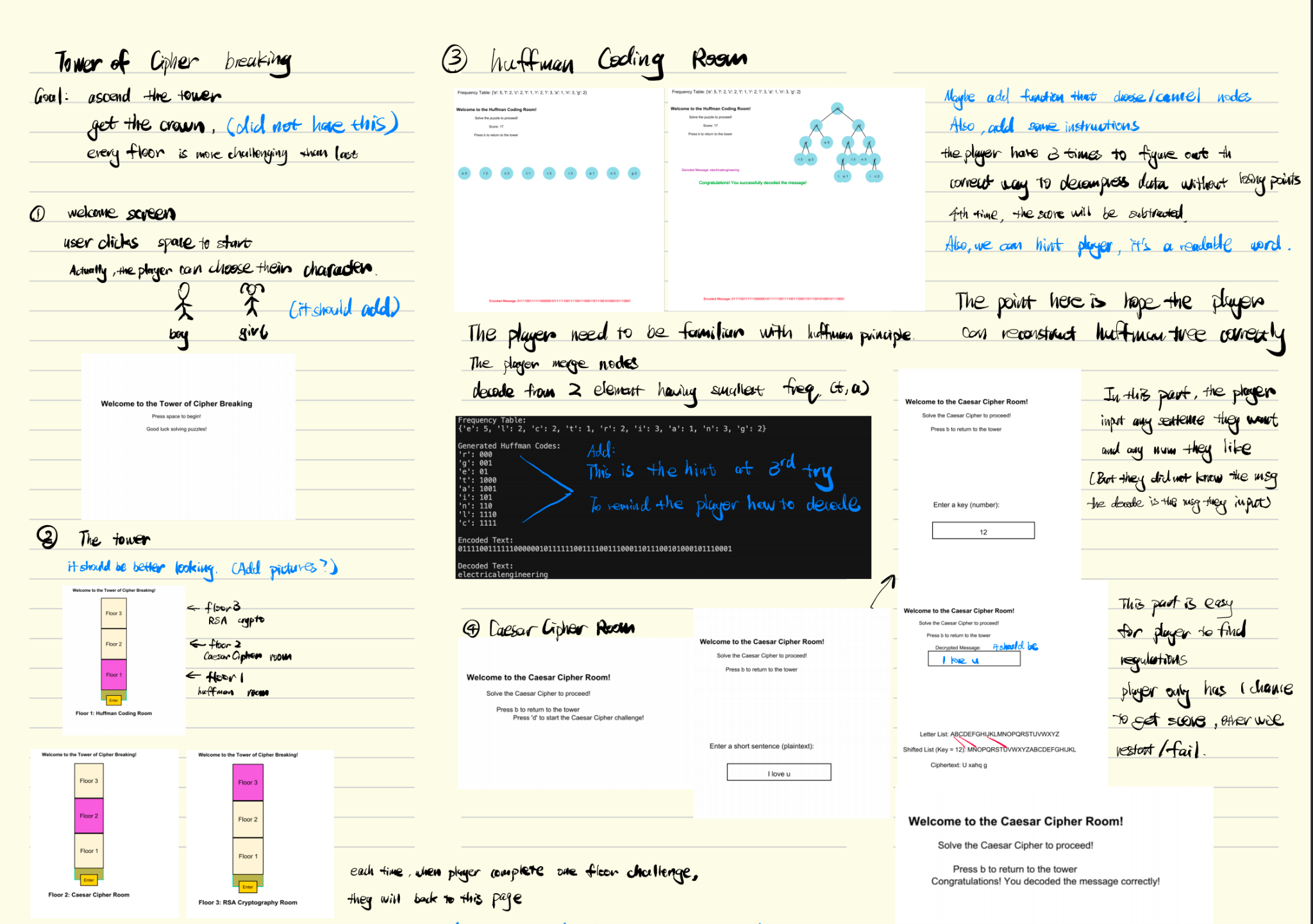
In this case, the player should pick the large prime number, which is not easy to factorize.

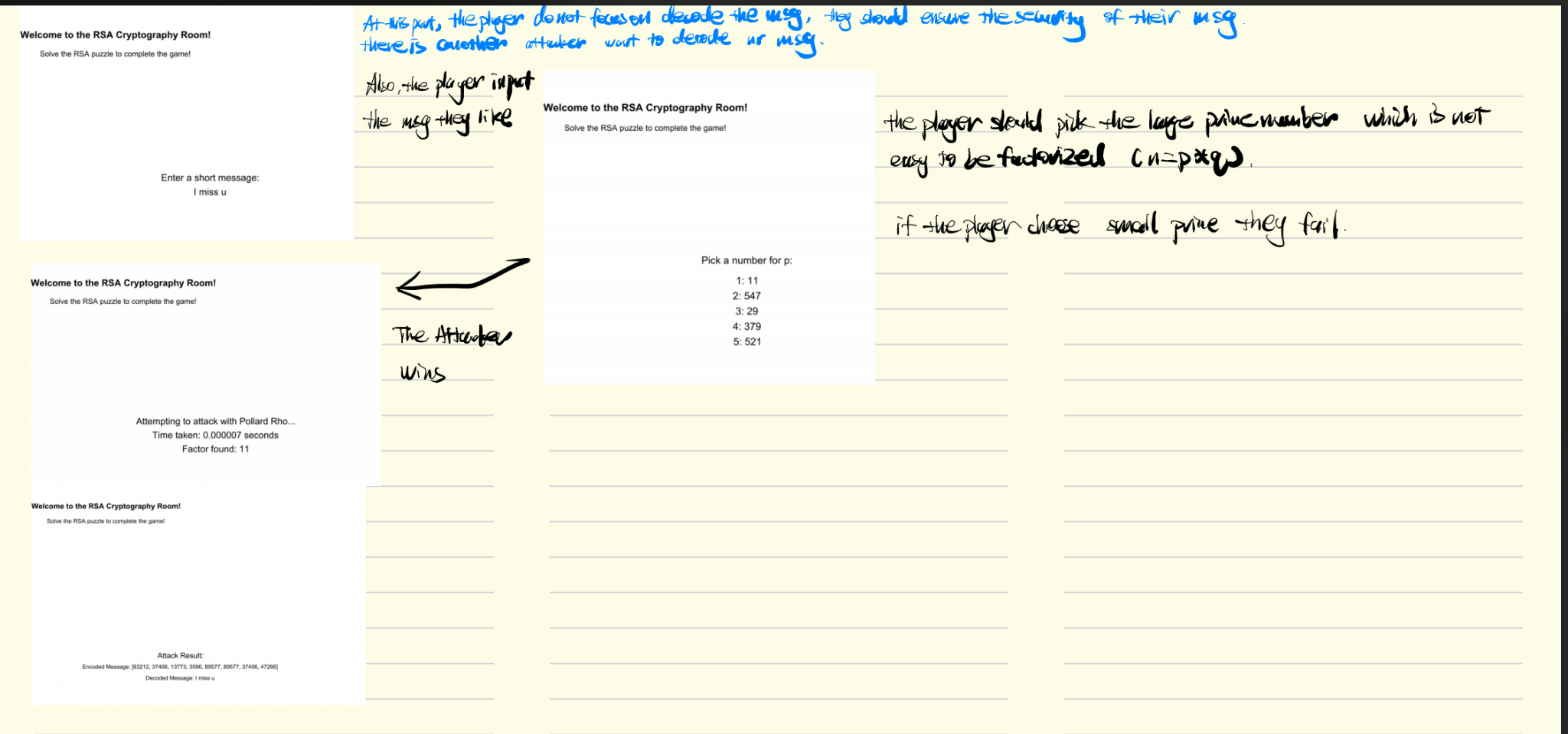
Also, my code here has a problem, if the value error is raised, it should not appear on the screen, and it should be randomized again.

1. Win the game

The player gets to the tower and wins the crown.







Graph (3 levels)

BFS (edge = 1):

Dijkstra: +weighted

Bellman-Ford: + negative weighted

2 hints:

find the node to next node shortest path

find the whole shortest path.