Message Locker: and encryption decryption tool

Kaci Bourgua, Katerina Kovaleva, Gemma Milman



Contents

- Project plan
- Description of game
- Challenges
- Process
- Highlights
- → Demo (ONE MINUTE)

Project plan

| 01 | Phase 1 - morning | Define plan in jira Discuss & agree on game workflow and features Design flowchart |
|----|---------------------|--|
| 02 | Phase 2 - afternoon | Pseudocode and repartitionCodingAssimilate code |
| 03 | Phase 3 - evening | Create\ finalise presentationTidying up documentation |

Description of Message locker

This tool allows you to encode a message wrote in English, and then exports it as an encrypted message in a text file.

The user can then come back at a later date to decrypt the message.



Challenges faced

Agreeing design

Many different possibilities:

Coding : Decrypting

Lots of time trying to decrypt using dictionary values to return key

Time..

Maybe too ambitious for amount of time assigned we let go of some features e.g. password

And code not as annotated as desired

Start Protocol/Welcome message Creating the encryption/decrypt choice of encyrption or decryption procedure Do you want to -Decryptionencrypt or decrypt Message to encrypt Message to decrypt decryption procedure Encryption procedure Encrypted message Displaye decrypted generated message Export encrypted message in a txt file End

Features dropped:

Security feature with a user name and password

Process

Key Features: Encryption key

- Encryption key
 - 3 characters per letter e.g. A= @f&
 - Generated using random.choice
 - No duplication of key codes
 - Only generated once
 - Dictionary for character encryption key created



Key Features: Encryption procedure

- Encryption procedure
 - Encrypts each character from input string
 - String -> list of individual characters [H, E, L, L, O]
 - Iterate through each character, matching with encryption key in dictionary
- Outputs decrypted message as text file

Key Features: Decryption procedure

- Uses regex to slice encrypted message every 3 characters
- Loops through encrypted message,
 - For every 3 letters in message, key found in dictionary
 - character appended to new message
- Message converted to string

Encryption List : ['"K_', 'i#H',]

Dictionary: {'h': '"K_', 'i': 'i#H',}

List : ['h', 'i',]

String: "hi"

Highlights

Getting to grips (or not) with tools like git and jira

Working as a team

For coding this was quite cool, except when we got stuck and then it sucked up man hours

Having a working tool...

And now for the demo...