Einleitung

Unsere ursprüngliche Projektidee trug den Namen „Hackerrätsel“ und soll in Form von kleinen Rätseln auf bekannte Hacking-Methoden und Sicherheitslücken aufmerksam machen. Da in der Gruppe dann unterschiedliche Interessen und Kompetenzen zusammenkamen, haben wir uns darauf geeinigt eine Rätselplattform aufzubauen, die für jedes Thema genutzt werden kann.

Repository auf Github: https://github.com/keyseven123/labprepare_sw_hw_riddles

Team Frontend

Im Teamfrontend wurde ein Dashboard erstellt, um eine Umgebung für die Bearbeitung der Rätsel bereitszustellen. Über HTTP Request werden dabei rätselrelevante Informationen von Server gefetched und im Frontend dargestellt.

Im Dashboardbereich ist die Rätselwahl zu finden. Wurde ein Rätsel ausgewählt, so stehen dort relevante Informationen wie Rätselname, Completion State und die Schwierigkeitsstufe.

Im Bereich Task ist eine Beschreibung des Rätsels zu finden. Hier wird das Rätsel genau beschrieben, um das Rätsel zu lösen. Falls die Rätsel zu schwer sind, gibt es im Bereich Hint nützliche Tipps, welche helfen können, um das Rätsel zu lösen. Bei Explanation ist eine zusätzliche Erklärung zum Thema zu finden, welches das Rätsel addressiert.

Für das Frontend wurde das Node.js Framework react.js genutzt. Als Texteditor wurde Visual Studio Code verwendet.

Team Backend

Ein Node-Server soll als Registrierungsstelle für alle Rätsel dienen und nachdem Rätsel ihre Lösung melden jeweilige Ports am Raspberry Pi schalten. Soweit ist dies auch exemplarisch möglich, bis auf die Auswahl bestimmter Ports. Ein Rätsel, dass in beliebiger Sprache programmiert sein kann, sendet über HTTP, GET- oder POST-Anfragen Informationen zum Node-Server, um sich zu registrieren oder ein Rätsel als gelöst zu setzen. Das Frontend kann sich dann, auch über eine Anfrage, alle registrierten Rätsel zur Verfügung stellen lassen, um sie für den Nutzer angenehm darzustellen (inklusive Tipps, Schwierigkeitsgrad, etc…).

Ein Beispielrätsel in Python um den Registrierungsablauf und verschiedene Funktionen des Servers zu zeigen ist hier zu finden:
https://github.com/keyseven123/labprepare_sw_hw_riddles/blob/main/Backend/NodeServer/exampleRiddle.py

Team Arduino

Im Team Arduino wurde die Umgebung für die Hardwarerätsel aufgebaut. Ursprünglich war geplant, dass das Arduino-Modul als Schnittstelle zwischen dem Raspberry Pi (kurz Raspi) und dem Gehäuse

dienen soll. Aufgrund zeitlicher und koordinativer Probleme haben wir beschlossen, alle Bestandteile modular und unabhängig zu bauen.

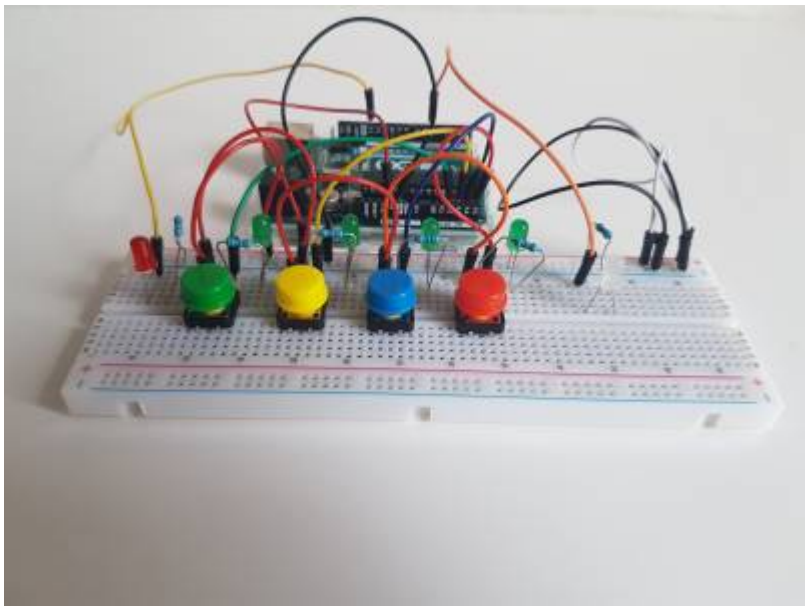
Programmierung

Zum Schreiben des Arduino-Codes wurde Visual Studio Code (https://code.visualstudio.com) und PlatformIO (https://platformio.org) verwendet. Letzteres erstellt eine Entwicklerumgebung, um große umfassende Arduino-Projekte verwalten zu können. Im Allgemeinen besteht der Arduino-Code aus einem Setup und einer Loop. Im Setup wurde festgelegt, welche Pins wir brauchen und ob diese Input oder Outputs sind. In der Loop wird zunächst überprüft, ob ein Knopf gedrückt wurde. Falls ja, dann läuft je nach dem Farbe des Knopfes ein eigenes Programm zur Überprüfung des Rätsels.

Den kompletten Code ist auf unserem Github-Repository.

Button-Panel

Mit dem Button-Panel kann man die gelösten Rätsel überprüfen lassen. Möchte man z.B. Rätsel 3 checken lassen, muss der blaue Button gedrückt werden. Der Zustand eines Button wird anhand von einem Analog-Pin gelesen. Nicht gedrückt liest der Pin die Spannung des GND also 0 V, gedrückt werden 5 V gelesen. Ein Widerstand ist nötig, um einen Kurzschluss zu meiden. Die grünen LEDs verhindern in der Funktion als Dioden das sogenannte Ghosting, wo fälschlicherweise ein gedrückter Button erkannt wird.



Wenn der Arduino gerade nichts tut, kann ein Button gedrückt werden. Dies wird durch die rote LED angezeigt. Bei einem richtig gelösten Rätsel leuchtet nach Überprüfung die weiße LED für 5 Sekunden. Die weiße LED ist eigentlich nur ein Ersatz für den Elektromotor des Automaten.

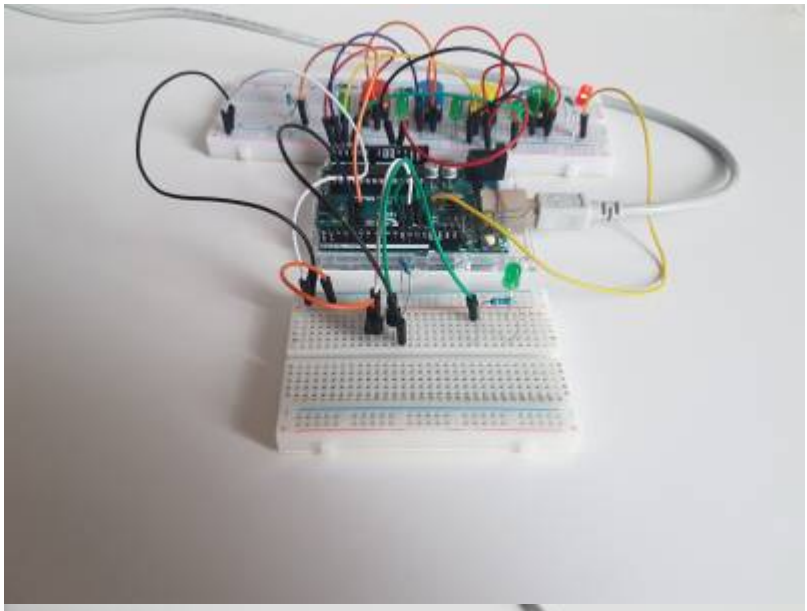
Hardwarerätsel

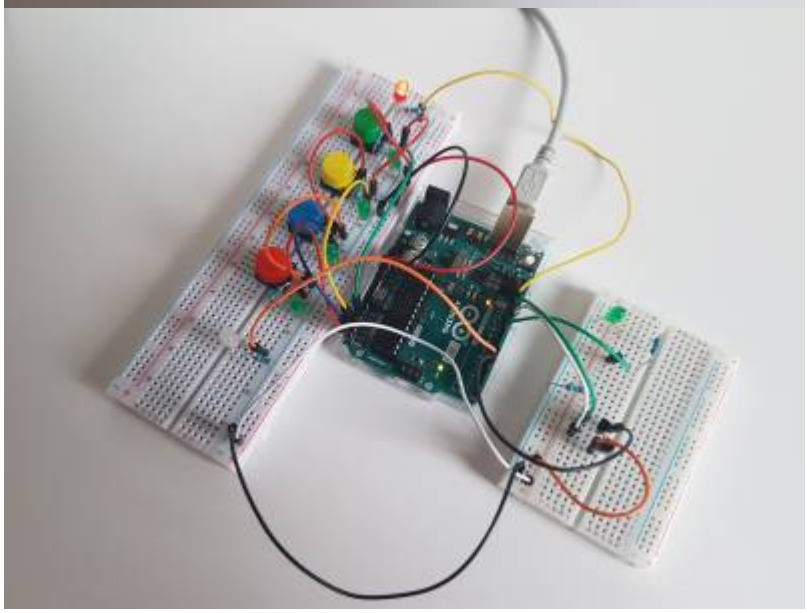
Auch wenn der Begriff „Hardwarerätsel“ erstmal kompliziert klingt, sind die folgenden Rätsel einfach und sollen eine Einführung in elektronischen und digitalen Schaltungen sein. Am Ende konnten vier

Rätsel zu folgenden drei Themen erstellt werden: LED, Logikgatter mit NMOS-Transistoren und Spannungsteiler. Im Rätseltext steht immer welche Bauteile benötigt sind und welche Pins wie verbunden werden müssen.

Alle Rätsel sind hier auf Drive.

Hier gibt es ein Video vom Ablauf eines Checks.





Cipher-rätsel (ciphero)

Mathematisches Rätsel, das das Kennenlernen einiger wichtigen Verschlüsselungsmethoden in der Geschichte ermöglicht. Teilnehmer sollen am Ende des Spieles ein Basiswissen der Verschlüsselungsanalysis erlangen.

Epochen & Levels

Das Spiel wird in 3 Epochen unterteilt, die einige wichtigen Entwicklungen der Kryptography in der Geschichte aufzeigen. In jeder Epoche sind verschiedene Levels(Stuffen) in Form von Aufgaben, die der Teilnehmer zu lösen hat. Die Aufgaben sind zwar miteinander verknüpft, um ein gutes Narativ zu konstruieren, bauen aber nicht aufeinander auf. Nichtsdestotrotz wird empfohlen die Hints und Explanations zu nutzen und möglichst alle Aufgaben nacheinander zu lösen. So könnte die geschichtliche Abhängigket der Methoden besser nachvollzogen werden.

Epoche I
Caeser-cipher
Reverse cipher
Book code
Epoche II
One-Time-Pad
Enigma
Epoche III
Steganography
DES
RSA

Es wird ebenso ein Superlevel für besonders schnelle Teilnehmer geben.
Die Rätsel wurden sowohl in python, als auch in javascript geschrieben.

Die Spieler erhalten am Ende des Spieles außer der Belohnung, eine Reihe von Links, um sich in den jeweiligen Epochen einzulesen und eventuell ihr Wissen zu erweitern.

Eine erweiterte Version des Spieles mit integrierten Punktenvergabe wird in der Zukunft auf der Webseite ciphero aufrufbar sein.

Team CAD

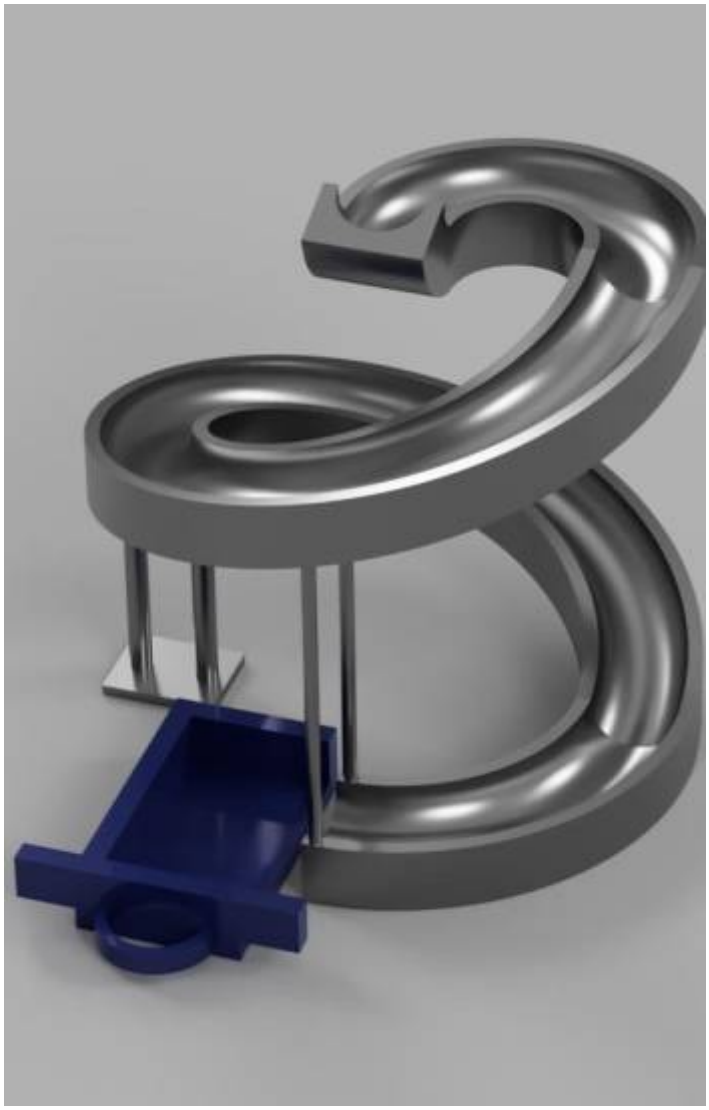
Das Team CAD hat sich um die Ausgabe einer Belohnung gekümmert. Die Idee war es sofern ein Rätsel korrekt beantwortet wurde eine Belohnung, z.B. Maoams oder Kaugummis, herauszugeben. Hierfür sendet der Arduino ein Signal an einen Elektromotor. Die Aufgabe des Elektromotors ist es dafür zu sorgen, dass die Belohnung eines Spielers bzw. einer Spielerin aufgenommen werden kann.

Design

Für das Design wurden Fusion 360 (https://www.autodesk.de/products/fusion-360/overview) sowie MakerCase (https://de.makercase.com/#/) benutzt. Das gesamte Design wurde erst einmal in Fusion 360 gebaut. Die Gesamtgröße wurde gemeinsam mit der gesamten Gruppe beraten und beträgt `300mm x 300mm x 595mm` für das gesamt Design.

Ein nebensächliches Ziel war es das Design so modular wie möglich zu gestalten. Dementsprechend gibt es folgende Module:

1. Box: Dient als Auffangbecken für die Belohnung.
2. Rutsche: So dass die Belohnung nicht einfach nur herunter fällt, gibt es eine Rutsche für die Belohnung
3. Vorratsbehälter: um nicht nach jeder Belohnung eine neue einzufüllen gibt es einen Behälter, welcher eine gesamte Packung von typischen Belohungen halten sollte
4. Schaufeln: Dient dazu, dass nur genau eine Belohnung herunter fällt.



Fertigung

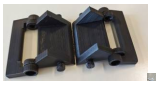
Als überwiegende Fertigungsmethode wurde der 3D Druck gewählt mit einem `Creality CR-10` als Drucker. Mit dem 3D-Druckverfahren wurden alle Teile gedruckt bis auf die äußeren Wände. Hierfür wurde ein Lasercutter benutzt um aus 8mm dicken Acrylplatten ein Gehäuse zusammenzuschneiden.

Gehäuse

Das Gehäuse wurde mit MakerCase erstellt und als SVG heruntergeladen. Anschließend wurde das Design in Inkscape geöffnet und die fehlenden Öffnungen reinskizziert. Nachdem die SVG Datei fertig gestellt wurde, wurden die Teile mit dem Lasercutter aus Acrylplattern herausgeschnitten.

Rest

Alle anderen Teile wurden mit dem 3D Drucker gedruckt. Dafür wurde jedes einzelne Modul in Fusion 360 unterteilt in druckbare Submodule und diese Submodule als STL Version exportiert. Die aktuellen Resultate sind hier als Bilder dargestellt



Ausstehend

Leider war es zeitlich nicht möglich alles fertigzustellen, da es nicht möglich war alle Module fertig zu drucken. Es fehlen noch die andere Hälfte des Vorratbehälters sowie der Hauptbestandteil der Rutsche. Ebenfalls müssen noch alle Teile miteinander verbaut werden sowie ein genereller Test von allen Komponenten muss erfolgen.

From:
http://www.labprepare.tu-berlin.de/wiki/ - **Project Sci.Com Wiki**

Permanent link:
http://www.labprepare.tu-berlin.de/wiki/doku.php?id=ss21:riddle

Last update: **2021/10/26 22:22**