

Building an Interactive Escape Room as a Distributed System

Description Stage, COM3610

Simon Fish | Supervisor: Andrew Stratton

2019-10-03

Contents

Introduction	1
Analysis	1
Plan of Action	2

Introduction

The aim of this project is to build an autonomous escape room framework using an assortment of software and hardware tools. From a personal perspective, I intend to apply existing knowledge to the project and use it to learn further technologies, which are discussed in the next section.

Automation would provide a variety of benefits. A flow of events would lead to more accurate tracking metrics for participants during their escape, which could also be used by an escape room maintainer to identify the most difficult parts of their room. Improvements to the flow of data from registration to participation and beyond would make for a smoother user experience, as has already been demonstrated by The Gym Group Ltd.¹ - there, members are granted a unique passcode that works across branches, and their entry/exit times are recorded. Further data is available to them via their online profile. Automation of puzzles would allow maintainers more time to research and improve their room even while a group attends the room.

The topic lends itself well to iterative, agile development processes and testing, while having a lot of leeway for me to explore and strengthen my skills.

Analysis

Automating an escape room would not be without its flaws. Many escape rooms already use technology minimally for show, e.g. showing a video to explain a story, or using lasers in those set in futuristic settings. However, they lend themselves well to being manually supervised, as physical puzzles can be finicky to work and mental puzzles can be difficult to understand. I believe it is possible to create a fully autonomous escape room, despite the boundaries of human understanding that lead maintainers to manually interfere. I will need to surpass this norm.

¹<https://www.thegymgroup.com/>

For the escape room to stand on its own, we would have to assume all puzzles would be in a working state and would not break after participant interaction. However, this is an idealistic scenario. One problem I fully intend to tackle is to make sure that even if a puzzle malfunctions, the participants' ability to complete the escape room will not be hampered. Puzzles in escape rooms are usually dependent on each other, so this would be to ensure that there are fallbacks if a dependency malfunctions. An assumption I hold so far is that there will be one main server which acts as the source of truth and maintains the state of the escape room. It may poll each of the puzzles regularly to check that they are live.

While the fundamentals of the project would be the deliverables discussed in the introduction, I would like to use it as an opportunity to learn, explore and document my way around a variety of different tools. My aim is to use some technologies I am familiar with, take some of those to new depths, and explore some new ones during the process. The project is likely to use a majority of the following:

- Red Hat SSO/Keycloak
- Languages that compile to machine code (Rust/GO)
- Amazon Alexa Skills Kit
- React (potentially to power a web frontend to the API)
- Mobile app creation frameworks and tools (e.g. Flutter, React Native, Expo)
- Microcontrollers/small computers such as the Raspberry Pi, particularly with regards to GPIO input and output
- CI servers such as Jenkins or GitLab CI
- Docker Swarms/Kubernetes
- RFID tokens as an interface
- Quando

Quando, a tool for digital interactive exhibits, should be incorporated in prototyping and serve as something on which to base the puzzles. The AMQP protocol, via RabbitMQ, will allow me to quickly and easily distribute events through the escape room system and maintain participants' progress at one central server. There is potential for this to grow into a franchise of sorts - a hosted queue that receives events from many escape rooms, each with many participants whose details are held in a single sign-on system.

Plan of Action

I have identified the following major aims for defining the entire framework:

- **Initial escape room setup:** registering clients (puzzles) with a server
- **Escape room configuration:** defining an expected order in which puzzles should be completed, in order to cater for the three *puzzle paths* as described by Wiemker, Elumir, and Clare (2015).
- **Defining the puzzle framework:** sending a message from a client to a server, liveness and readiness endpoints
- **Setting up a single sign-on server:** enabling storage of users' data in a single sign-on system
- **Maintaining a user session:** securely signing in to the server using RFID (or a suitable stub) and a one-time passcode, perhaps as a QR code

The intention is that a user has something physical that can be scanned to identify them against their account, and are then sent a magic link/OTP/QR code, which they activate

as consent to sign in.

In the context of Agile, I would consider these epics as described by Rehkopf (n.d.). My approach to splitting up the work is influenced by my learnings from my year in industry - within each epic, I have defined and estimated stories that represent small, manageable territories of well-defined work. On this basis, I expect that the smallest epic will be configuring the single sign-on platform.

I have decided that I will aim to clear at least two of these this semester - **defining the puzzle framework** and **setting up a single sign-on server** - and research around the others, expecting to clear those with ample time to create sample puzzles for the end of the course mid-2020. I expect the project's scope to shift, so these estimates will need to be verified regularly.

Rehkopf, Max. n.d. "Epics, Stories, Themes, and Initiatives." *Atlassian*. Atlassian. <https://www.atlassian.com/agile/project-management/epics-stories-themes>.

Wiemker, Markus, Errol Elumir, and Adam Clare. 2015. "Escape Room Games." *Game Based Learning* 55.