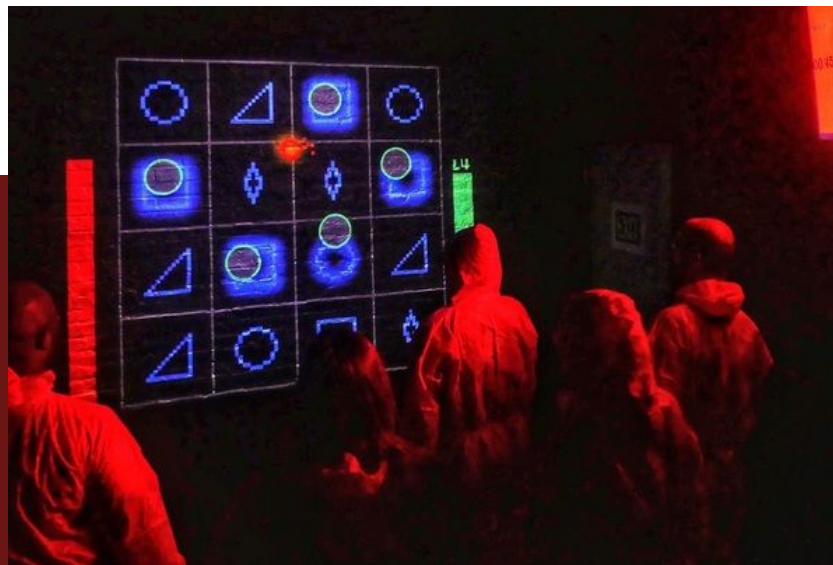


Escape room automation

Dan Egnor - Stanford CS 64



Dan Egnor, escape enthusiast

Way too many rooms played (983 total so far)

Former Guinness World Record holder for
escape rooms played in 24 hours (22)
(since overtaken; record is now 32)

Member of the design team at
Palace Games in San Francisco

ex-Google, ex-Microsoft

Kind of a big dork



But wait! What is an "escape room"??

People argue about this a lot, like "what is a video game" or "what is sci-fi"

BUT most escape rooms fit most of these genre conventions

- Adventure challenge games played as a team (~4-10)
- Played in a purpose built physical space
- Most of the game requires thinking/searching/insight, not physical skill
- You're not told what to do, you have to figure it out
- The game lasts an hour or so, if you finish in time you all "win"
- Often there's a storyline or theme (*sometimes* scary/horror, often not)

ESCAPE!



What did we see there?

Big puzzle blocks reveal data (symbols) when assembled correctly

Wheels on the wall are an input for the data

When the puzzle was assembled, there was some kind of noise to startle players (and mark completion!) - probably triggered by game operator

When the wheels were positioned correctly, a door opened (such excitement!)

This was a simple puzzle, but representative; more complex puzzles would have less-obvious relationships between things, indirection, more complicated actions to take, hidden sensors & motors, etc

Also there's flickery lights and other non-puzzle decor elements

Escape room technology "generations"

Labeled by [Shawn Fischstein \(Escape Games Canada\)](#), not everyone agrees!

Gen 1: Combination locks, padlocks, hidden keys, written clues

Gen 2: Actuators, energy, basic self contained input/output control

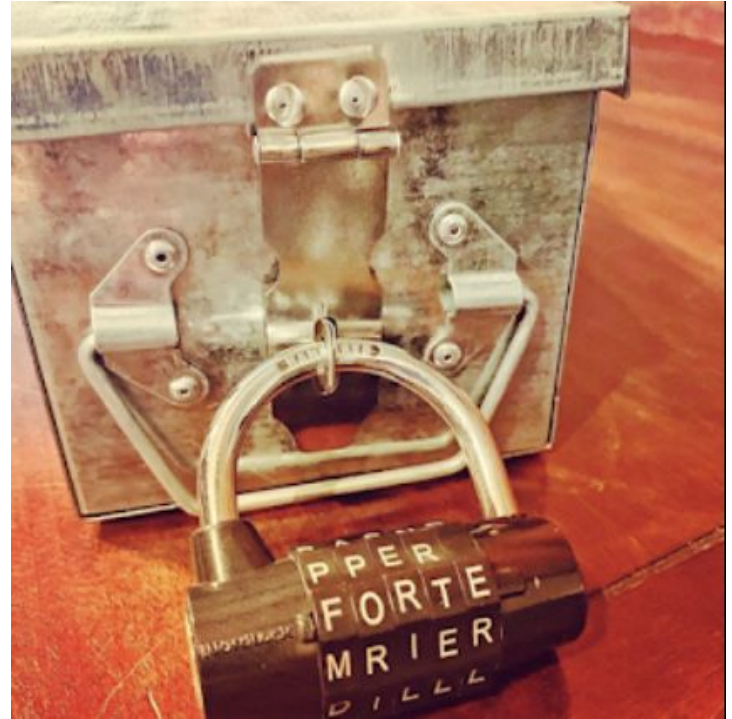
Gen 3: "Smart" room, integration across systems, mostly computer controlled

Gen 4: "Adaptive" room, managed player experience, computer "showrunner"

... in reality it's not as neat and tidy, but things are getting more integrated

Gen 1: Locks and boxes

- + Simple and easy!
- + Actually pretty satisfying to open
- + Some great rooms built this way
- + No tech skills needed
- Limited interaction types
- Need to be manually re-locked
- Some people get snobby about this
- Hard to explain why an ancient tomb has Master Lock® padlocks all over

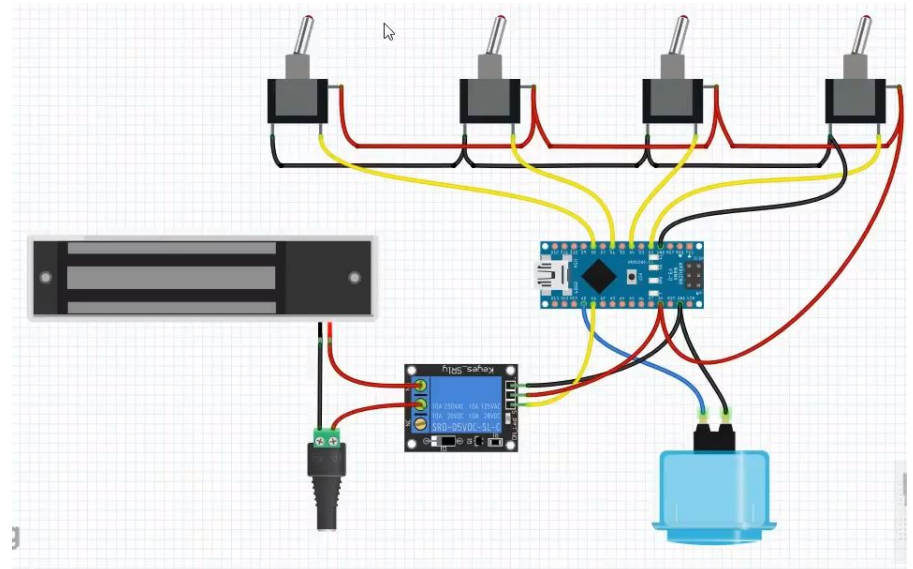


Gen 2: Sensors and triggers

Each input/output directly linked

Simple Arduino-type controllers (or direct wiring with relays and such)

- + Fine for the puzzle in the video!
- + Each puzzle is independent
- + Off the shelf systems available
- Integration can be difficult
- Lots of separately programmed widgets to install and manage

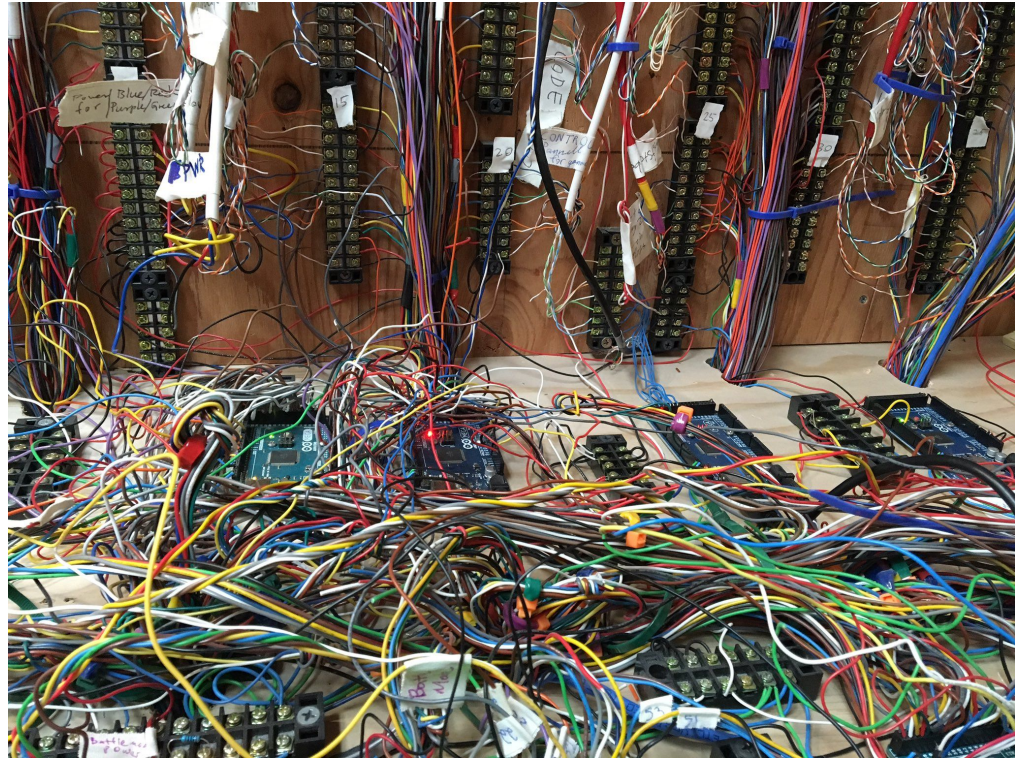


Gen 2 gone wild

Houdini Room at Palace Games
(ca. 2016; since cleaned up with
later generation technology)

Basic puzzle-at-a-time Gen 2
architecture expanded with
interactions (A/V coordination,
puzzles activate other puzzles,
conditional timing, etc) gets messy

Imagine debugging a loose wire!



Gen 3 & 4: Integrated control

All systems coordinated: puzzle progress, lighting, A/V, motion

Advanced sensing (LIDAR, depth cameras) and control (motion stages, spatial sound, projection mapping)

- + Theme-park like cinematic experience and immersion
- + Central control and design
- Not easy to engineer!
- Prone to catastrophic failure

Gen 3 vs 4 difference is in software

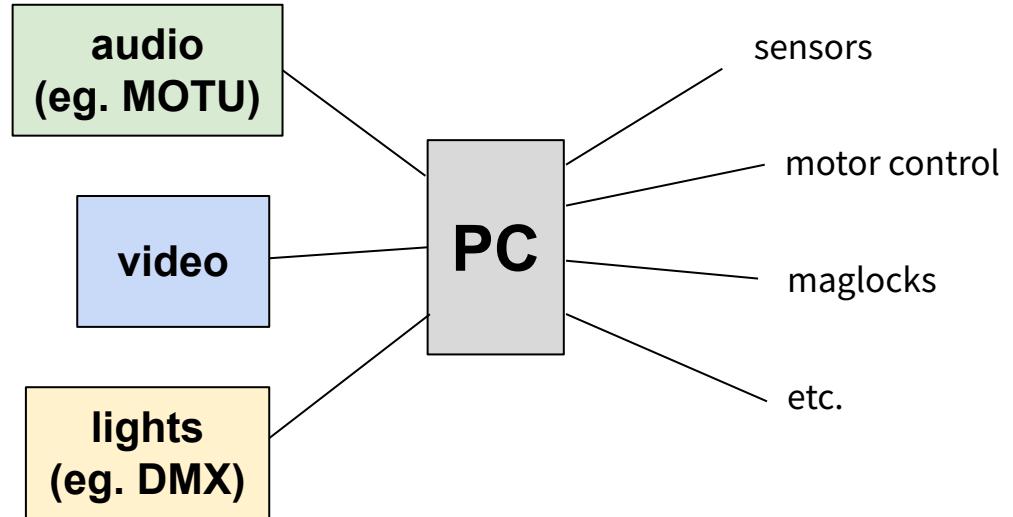


Gen 3 & 4 option: centralized architecture

In this design, one central computer does everything for the room

- + everything in one place, controlled by one program
- + write & debug ordinary software in your favorite language
- big wiring tentacle monster
- if that one program crashes...

This is how software engineers tend to build escape rooms!

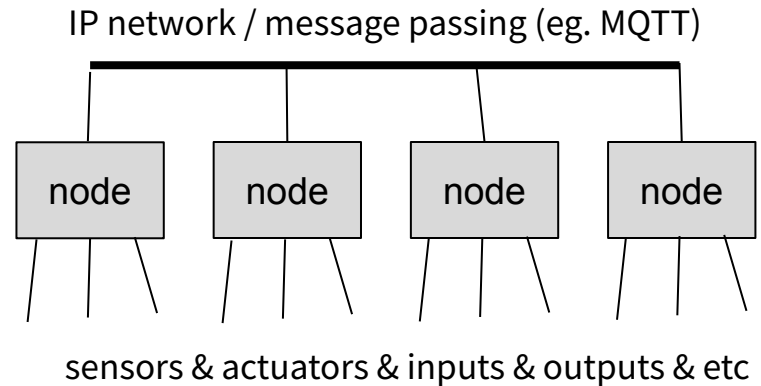


Gen 3 & 4 option: distributed architecture

In this design, distributed compute nodes (eg. Raspberry Pi) collaborate on room control

- + graceful failure degradation
- + less tentacle-monster wiring
- more computers to manage
- more complex coding of "microservice" architecture

This is how hardware engineers tend to build escape rooms!



But what about all that software?

Gen 3 & 4 systems are heavily software dependent, but there's no standard software architecture for building these systems

Architectures from robotics (message passing, subsumption architecture, hierarchical control) are common in this space

It's often a little difficult to test & debug, good automated testing and simulation mocks are not yet common in this space

On the bright side... you're usually only running one copy and you can monitor it yourself and include human overrides as needed

Gen 4: Adaptive game flow

With central control comes opportunities for adaptive/dynamic flow, some people call this "gen 4" design

For example, if a team is slow, puzzles can be made easier, or lighting/sound used as subtle cues to indicate next steps. If a team is fast, bonus challenges can be added. Ideally, players don't know this is happening

"Pipelining" becomes possible, bringing teams into earlier parts of the experience once the previous team has cleared, rather than limiting the whole build to one team at a time

In summary

There are lots of valid architectures; great experiences don't need high tech

These "generations" aren't "real" – there are no bright lines in the design space – they're just something some people use to talk about design directions

A lot of designers are experimenting with ways to build bolder experiences – "Puzzle Disneyland" (or "Zelda IRL") is the dream – but nobody knows the design+tech+business formula to combine the immersive hands-on joy of escape rooms with the scale of theme parks and the depth of video games

Q&A