Ren Ramlan
20.405 Principles of Synthetic Biology

1-Bit Pixels Encoded in E. Coli for the Display of Interactive Digital Media

Motivation

Doom is a first-person shooter game released 1993 in which a player assumes the role of a space marine, unofficially referred to as Doomguy, to fight off hordes of undead humans and demons trying to take over the Solar System[1]. What has brought Doom out of the shadows of the 90s and into the modern light is not its riveting gameplay or questionable protagonists, but rather its appealing computational design. Built on the id Tech 1 engine[2], the game was designed to require only the most modest of setups to be played. As da Vinci once said, "Simplicity is the ultimate sophistication."

As time advanced following Doom's release, as did humanity's technology. In 2007, a revolution began as Apple released its first iPhone - the iPod Touch. Screens soon became an integral part of 21st century existence, integrated into more and more products and daily tools. Many unexpected modern appliances such as the fridge, toaster, cash registers, cars, vapes, and even pregnancy tests were soon fitted with screens of their own. Along with this came an opportunity.

To run Doom, all one needs is a screen and willpower. As such, the concept of "Doom running on everything" (henceforth referred to as DROE) was born, giving rise to a culture in which Doom has been played on ATMs, calculators, lego bricks, potatoes, and even the aforementioned pregnancy tests[34] (Fig. 1). Ultimately, this begs the question of how biological systems might be engineered to host this classic millennial FPS.

Fig. 1. Doom being run on a 18x32 pixel pregnancy test screen, controlled by a QWERTY keyboard created by user @Foone on Twitter circa 2020

[1] https://en.wikipedia.org/wiki/Doom_(1993_video_game)
[2] https://en.wikipedia.org/wiki/Id_Tech
[3] https://www.reddit.com/r/itrunsdoom/
[4] https://www.gamesradar.com/12-things-that-prove-that-doom-will-run-on-literally-anything/

Background

Genuine efforts to integrate Doom with biology are well underway. Neuronal computing has given us neurons capable of controlling flight simulators[5], piloting robots[6], playing Pong[7], and recently, playing Doom itself. However, I argue that programming cells to *play* Doom is not the same as running the game themselves.

Implementation

To run Doom solely on cells would be a behemoth feat that I cannot even imagine approaching. However, within the culture of DROE, simply getting Doom displayed on the screen of a novel device has been sufficient to consider the game played on that technology. As such, the challenge is engineering a way to display Doom on cells, a realistic possibility inspired by creations such as the E. coli digital display from Shin et al. 2020[8].

Fundamentally, in order to display Doom on cells, they must be able to function as pixels. In a simple 1-bit black-and-white display, a pixel is either on or off, represented as 1 and 0 respectively. I propose a system whereby cells are contained within a 32x48 well plate connected to a display controller that processes the graphics output of Doom and translates the binary arrays into the addition or omission of a repressor controlling the fluorescence of the cells in that well. This simple system requires only a single genetic part: A repressor-operator pair controlling the expression of a fluorescent protein such as GFP. The output of this model would help determine the frame rate at which Doom could theoretically be displayed on cells.

The genetic system that I will be modeling is based on an inducer from the Registry of Standard Biological Parts, BBa_K3893028. This plasmid, designed for E. coli, contains two interacting genetic circuits. The first contains a promoter J23106 and RBS B0030 that constitutively express the luxR repression factor. The second contains the promoter Plux, which is repressed through the binding of LuxR to the promoter, and RBS B0032 to produce GFP. In the absence of LuxR, this system has a basal expression for GFP of ~0.75×10^6 MEFL. Below is the dose response curve for this system, which saturates at a LuxR concentration of 100 nM and has a maximum expression of ~9×10^6 MEFL (Fig. 2). Parameters for the system have been experimentally determined as follows. An effective translation rate of 8.11e8 molecules/min, a dissociation constant of inducer of 15.34 nM, a hill coefficient of 0.958, a basal expression coefficient of 0.1049, and a dilution rate of AHL of 0.0115 /min. I optimized the model by converting these to compatible nM scale values and adjusting parameters to get reasonable behaviors (Table 1).

5
https://philosophy.as.uky.edu/sites/default/files/Adaptive%20Flight%20Control%20With%20Living%20Neuronal%20Networks%20on%20Microelectrode%20Trays%20-%20De%20Marse%2C%20Dockendorf.pdf
[6] https://dl.acm.org/doi/10.1162/NECO_a_00632
[7] https://www.nature.com/articles/d41586-022-03229-y
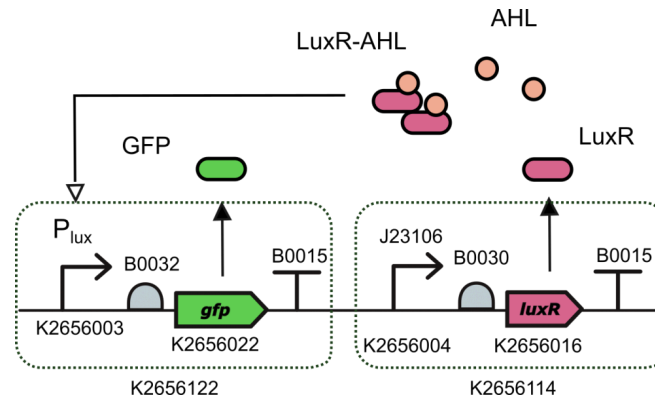[8] https://www.embopress.org/doi/full/10.15252/msb.20199401

Fig. 2. BBa_K3893028 - Diagram representing the AHL induced expression of GFP (including luxR constitutive expression) from the iGEM Parts Catalog.
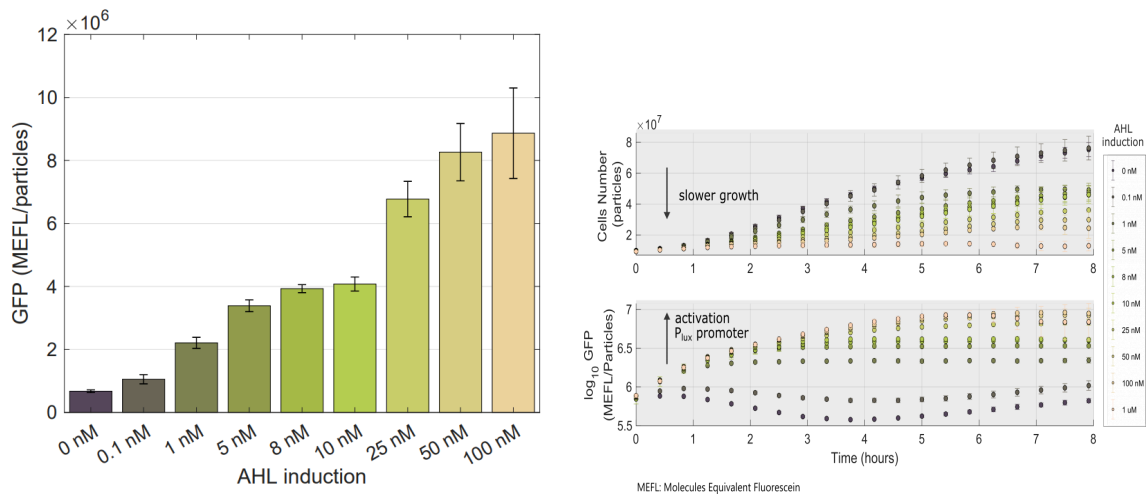

Fig. 3. GFP expression as a function of AHL induction from the iGEM Parts Registry.

k_trl	KdLux	n	k_pdeg	k_const	K_dA	K_onA	GFP0	Lux0	AHL0
100	0.1	0.958	0.0115	1	0.0001	0.01	2nM	10nM	100 or 0 nM

Table 1. Model contents and their values. K_trl: transcription rate constant, KdLux: dissociation constant of LuxR from pLux, n: hill coefficient, k_pdeg: protein degradation constant, k_const: constitutive promoter constant, K_dA: dissociation constant of AHL from LuxR, K_onA: forward binding constant of AHL to LuxR, GFP0/Lux0/AHL0: starting concentration of each species.

In order for this design to be compatible with a customizable display, we want to be able to induce GFP expression by introducing an external signaling molecule to the "on" pixel wells. Induction of this system can be achieved by repressing the repressor LuxR. AHL is a small and easily-synthesized molecule that can bind to LuxR and prevent Lux repression of the GFP promoter. The dose response curve for AHL-mediated activation is shown below.
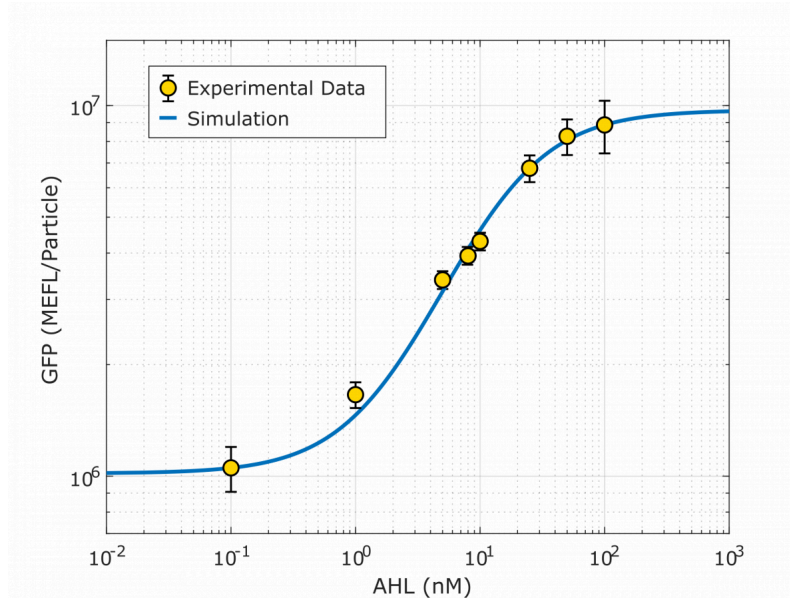
Fig. 4. BBa_K3893028 induction curve from the iGEM Parts Registry.

After planning the system and mining parameter values, I then had to code the system. I started by describing the changes in each species over time using ODEs (fig. 5), assuming hill-like binding of the repressor and steady state mRNA dynamics, as well as universal protein degradation rates. I also assume GFP expression is proportional to promoter repression/induction, and that cell growth is negligible.

$$\frac{d\,GFP}{dt} = K_{trl} \cdot \frac{K_{dL}^{n}}{K_{dL}^{n} + LacR^{n}} - K_{pdeg}[GFP]$$

$$\frac{d\,Lac}{dt} = K_{const.} - K_{onA}[AHL][Lac] + K_{dA}[AHL \cdot LAC] - K_{pdeg}[Lac]$$

$$\frac{d\,AHL}{dt} = K_{dA}[AHL \cdot Lac] - K_{onA}[AHL][Lac] - K_{pdeg}[AHL]$$

$$\frac{d\,AHLLAC}{dt} = K_{onA}[AHL][Lac] - K_{dA}[AHLLAC]$$

Fig. 5. System of ODEs describing the model for a single bit in the display system. This model assumes that GFP expression is proportional to repressor binding at the promoter, that mRNA dynamics are at steady state, and that protein degradation is constant between the species.

Coding

I then coded this system into Python using Colab. After solving the ODE and fine tuning parameter values, I was able to model a system in which, when AHL0 is set to 100nM, produces a pulse of GFP expression (fig. 6), and when AHL0 is set to 0, it does not (fig. 7).
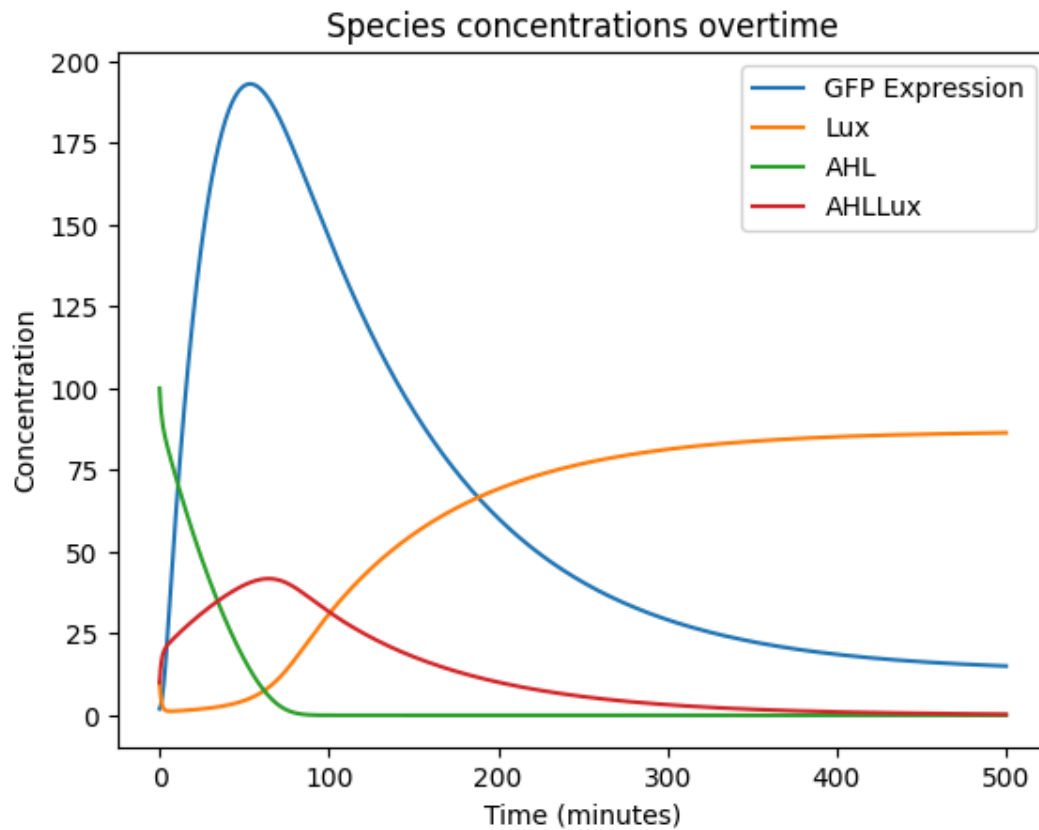

Figure 6. Species concentrations as determined by ODE solving in python. The legend shows which species correspond to which colors. When AHL is present, initially, the AHL-Lux complex AHLLux is formed, depleting Lux and resulting in GFP expression. As AHL is consumed and degraded, Lux concentration increases overtime, decreasing GFP and resulting in the pulse behavior of the system.
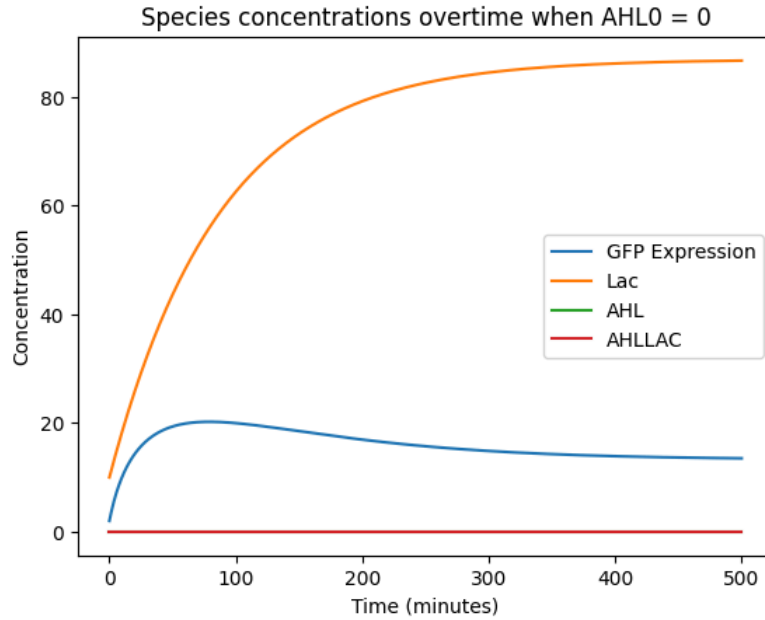
Figure 7. Species concentrations over time in the absence of AHL inducer. When AHL is not added to the system, Lux rapidly saturates and prevents significant GFP expression). (Lac here represents the presence of LuxR, while AHLLAC represents the complex of AHL and LuxR)

The next step was to take this system and replicate it into a 32x48 array to simulate each well of cells. After doing this, I needed a way to translate the frames of Doom into binary inputs for the addition or nonaddition of AHL to each well. I took the first several frames of Doom and compressed them into 32x48 grayscale arrays of average neighboring pixel values to make the frames the same size as the cell array (fig. 8b). I then performed thresholding on the images to determine which pixels should be considered "on" and have AHL added, and which pixels are off (fig. 8c). I then pass this thresholded binary array into my simulation code as the determinant of which wells have the inducer added and generate a video of the GFP concentration in each pixel "well" over time, as shown in the final video.
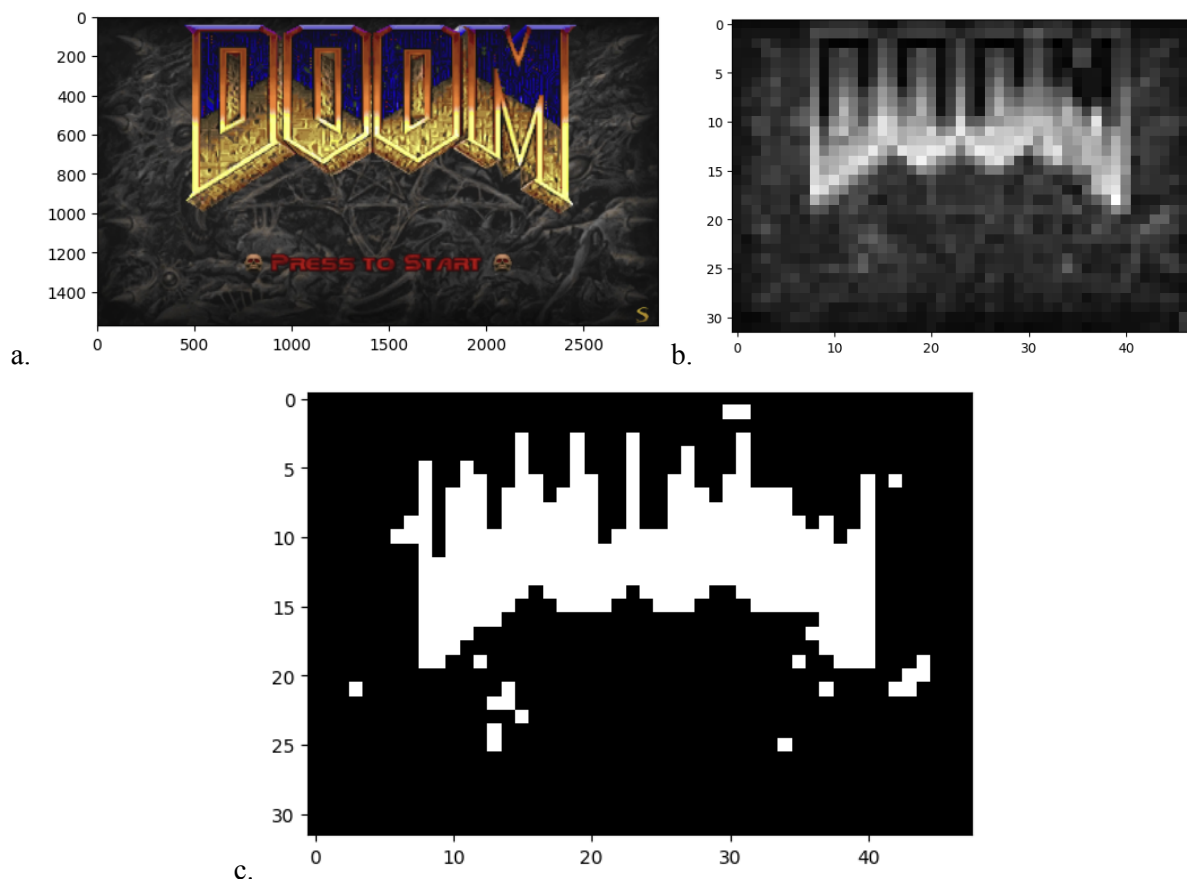
Figure 8. Image processing of Doom frames. Pictured is the first frame of Doom, as an example. a) The original frame 1 of Doom. b) The 32x48 grayscale compressed first frame of Doom. c) The thresholded 32x48 array that is used to determine which pixels are "on" and which are "off". Threshold was set to pixel values from the grayscale image greater than 70.

In conclusion, it takes approximately 70 minutes to reach the peak display output of GFP in an on well, and a total of 8 hours and 20 minutes for the cell to return approximately to the starting state. The framerate for the original doom is capped at 35 frames/second[9], and the average playthrough time is ~5 hours[10]. Doing the math, this means that it would take 599 years to run Doom on cells, according to this simulation.

To improve this time and model, there are several future directions for this work. Firstly, more experimentally derived values for the parameters would make the system more indicative of a real-life implementation of the design. Second, the incorporation of a quenching mechanism to speed up the rate of turning back off would increase the framerate. Furthermore, the incorporation of a memory and prediction system would mean that pixels that are on multiple frames in a row could remain on, likely decreasing the time required to change between the frames. More work is needed to simulate the maximum fluorescence the system must decrease to after the pulse before the next frame can be loaded in.

---

[9] "Uncapped Framerate of Doom." DoomWiki.Org, DoomWiki.org, 20 Feb. 2021, doomwiki.org/wiki/Uncapped_framerate.
[10] "How Long Is Doom?: Howlongtobeat." HowLongToBeat.Com, howlongtobeat.com/game/2701.

Overall, this was a very enjoyable project, and I hope you enjoyed reading about it. The future of DROE is bright, even if it is over half a millennium from now.

References

1. "Doom (1993 Video Game)." Wikipedia, Wikimedia Foundation, en.wikipedia.org/wiki/Doom_(1993_video_game).
2. "Id Tech." Wikipedia, Wikimedia Foundation, 27 July 2023, en.wikipedia.org/wiki/Id_Tech.
3. "Doom Running on Anything." Reddit, www.reddit.com/r/itrunsdoom/.
4. Hurley, Leon, et al. "15 Things That Prove That Doom Will Run on Literally Anything." Gamesradar, GamesRadar+, 15 Oct. 2021, www.gamesradar.com/12-things-that-prove-that-doom-will-run-on-literally-anything/.
5. DeMarse, Thomas B., and Karl P. Dockendorf. Adaptive Flight Control with Living Neuronal Networks on Microelectrode ..., philosophy.as.uky.edu/sites/default/files/Adaptive%20Flight%20Control%20With%20Living%20Neuronal%20Networks%20on%20Microelectrode%20Trays%20-%20De%20Marse%2C%20Dockendorf.pdf. Accessed 15 Nov. 2023.
6. Chapin, J.K., K.A. Moxon, R.S. Markowitz, and M.A.L. Nicolelis "Realtime control of a robot arm using simultaneously recorded neurons in the motor cortex," Nature Neuroscience, vol. 2(7), 664-670, 1999.
7. Ledford, Heidi. "Neurons in a Dish Learn to Play Pong - What's Next?" Nature News, Nature Publishing Group, 12 Oct. 2022, www.nature.com/articles/d41586-022-03229-y.
8. Programming Escherichia Coli to Function as a Digital Display - EMBO Press, www.embopress.org/doi/full/10.15252/msb.20199401.
9. Mitra, Ritwik. "The 15 Wildest Things That Doom Can Be Played On." Game Rant, 6 July 2022, gamerant.com/weirdest-consoles-you-can-play-doom-on/.
10. "Uncapped Framerate of Doom." DoomWiki.Org, DoomWiki.org, 20 Feb. 2021, doomwiki.org/wiki/Uncapped_framerate.
11. "BBa_K3893028." Part, parts.igem.org/Part:BBa_K3893028.
12. "How Long Is Doom?: Howlongtobeat." HowLongToBeat.Com, howlongtobeat.com/game/2701.

Videos and Media

1. https://www.youtube.com/watch?v=TIlOu1lxfEU&t=1s
2. https://www.youtube.com/watch?v=Q4GiCg_m8wA&t=15s
3. https://www.youtube.com/watch?v=kpnW68Q8ltc
4. https://www.youtube.com/watch?v=c6hnQ1RKhbo&t=625s
5. https://www.youtube.com/watch?v=GD0L46y3IqI
6. https://www.youtube.com/watch?v=nduMTX86Zl0&t=3s
7. https://www.youtube.com/watch?v=_-BH1XdgVPQ
8. https://www.youtube.com/watch?v=5dPK_OGU9FY&t=123s
9. https://www.youtube.com/watch?v=VOFI-f7rCWk&t=35s
10. https://www.youtube.com/watch?v=bEXefdbQDjw&t=419s
11. https://www.youtube.com/watch?v=PW5ELKTivbE&t=39s
12. https://www.youtube.com/watch?v=TJHSMpYi8bg
13. https://www.youtube.com/watch?v=0apFspOlK10

14. https://www.youtube.com/watch?v=JrEYSrUX1YM&t=272s
15. https://www.youtube.com/watch?v=MnqLJpgq7jc
16. https://www.youtube.com/watch?v=D5NTJSfUWDE
17. https://www.youtube.com/watch?v=RO90omga8D4
18. https://www.youtube.com/watch?v=5es64FPCO6k&t=10s
19. https://www.youtube.com/watch?v=KJwnIcj82a0&t=7s
20. https://www.youtube.com/watch?v=dbeCNiBuv3Q