Proof Of Humanity

1/ Plan

-2 march → 9 march (1 week) : Research and Setup

-9 march → 21 march (1.5 week) : Practice and preps for prototype build (Research on tech tools Python libraries, Cairo, useful firmwares…).

***** Upload PrepDoc*****

-21 april → 4 april (2 weeks) : Cahier des charges (And bibliographie)

***** Upload Cahier des charges *****

-4 april → 11 april (1 week) : Conception document

***** Upload Doc de Conc *****

-11 april → 2 may (3,4 weeks) : Local Prototype blocks

*****Upload V1*****

-9 may → 16 may (2 weeks) : Tests, and Updates

*****Upload Final V*****

####### HALF TIME ####### (2 weeks lifeboat)

-30 may → OnChain migration (2 months)
- 1 august → RAPPORT (1 month)

2/ Ideas

- Machine Learning algorithm that detects bots from real humans.
- Neural network that distinguishes human faces from “other” pictures.
- Human mouse trajectory recognition.
- Completely decentralized algorithm (Open source) *not rely, or as less as possible on using Security by obscurity.*
- Not worrying about the robustness of the algorithm*, the goal is to discourage a certain threshold of bots. Simply adding an extra task already removes (personal estimation) 50% of bots. Machine learning algorithms repulce up to 80%, 90% of most advanced bots. There must*

be a real motivation and dedication to make a powerful intelligent bot, but then, the danger of employing real human farms exists also. Then the effort needed to develop a 100% bot repellent algorithm, seems unnecessary. Complementary document verification is necessary for highly critical systems.

- Preserving privacy as much as possible.
- Integration de starkware en live.
- Proof of livenes: Is 100% robustness even necessary? After a difficulty level, easier to hire real humans.

3/ Project Summary

4/ Meetings

Nicolas Perrin-Gilbert : perrin@isir.upmc.fr ISIR Researcher (Friday 04/03/22 at 17h):

- Proof of livenes by Machine Learning, (Without document verification), that is completely robust, **Doesn't exist** to date. Big companies protect themselves by using abnormally huge amounts of data to train their Machine Learning algorithms and that way ensuring that very few teams in the world have that much data to train on. For example Facebook is able to do training on user mouse trajectory on Millions of subjects per day. That way having access to the top quality of data which cant be attained in normal conditions.
- **Warning**: Keep in mind that for successful training, a very large data set is required.
- Proof of livenes is generally not based on machine learning, because it's not completely robust. A more **dynamic** solution is required: Changing the nature of the test before there is a breach. The Goal is to make the task of breaching the test by a bot as hard as possible before it has the time to train himself to resolve the test. Nevertheless We have to keep in mind that the test is never 100% impenetrable.
- **Particular problem** of supervised learning: Algorithms must classify a face and "something else".Training by giving to the "else" category random pictures is dangerous. ML algorithms have pretty unpredictable responses to unseen data. To Check: GAN (Adverserial NN). Generator and discriminant methods.
- **Warring**: Security by obscurity, big companies hide in top secrecy their trained neural network.
- **Decentralization**, and open sourcenes gives the project an interesting aspect as we must avoid using hiding anything.

5/ Bibliographie

[1] - Nicolas Perrin-Gilbert, course on supervised learning

[2] - Gunjan, V. K., Senatore, S., Kumar, A., Gao, X. Z., & Merugu, S. (Éds.). (2020). Advances in Cybernetics, Cognition, and Machine Learning for Communication Technologies. Lecture Notes in Electrical Engineering. https://doi.org/10.1007/978-981-15-3125-5

Chapter 1:
Chapter 8:

[3] - Meng Joo Er, Shiqian Wu, Juwei Lu, & Hock Lye Toh. (2002). Face recognition with radial basis function (RBF) neural networks. *IEEE Transactions on Neural Networks*, *13*(3), 697-710. https://doi.org/10.1109/tnn.2002.1000134

[4] - Lior Goldberg Shahar Papini Michael Riabzev. (2021) Cairo – a Turing-complete STARK-friendly CPU architecture.

[5] - Alejandro Acien , Aythami Morales , Julian Fierrez, Ruben Vera-Rodriguez. (2021). BeCAPTCHA-Mouse: Synthetic Mouse Trajectories and Improved Bot Detection. https://arxiv.org/pdf/2005.00890.pdf