READINESS

THE LARP

Game Master's Report
Projections on the Upcoming Civil War

OCT-NOV 2020

Game Master's Report
Projections on the Upcoming Civil War

Between the 16th and the 18th of October 2020, Readiness the LARP, a chamber live action roleplaying game took stage in Athens, Greece. The chamber was located in a secreted, highly migrated area of central Athens, near the Pakistani and the Chinese community. Readiness the Larp was designed to narrate and foreplay the pending scenarios of an upcoming American Civil War. It sparked the friction between a group roleplaying "the Preppers" (this movement of doomsday enthusiasts who marshal for the "end of the world as we know it"), an additional bunch of "Boogaloo Bois" (Civil War accelerationists flexing tactical gear and Hawaian shirts), and a group playing the "Golden Horde" (the allegedly criminal horde of the unprepared citizens molded in preppers' jargon and literature). The game also featured a number of neutral figures inspired by popular digital culture: The Oracle, Pudge, Sensei, the Operator and Jeso (a non binary variation of the messiah). The game consisted of 3 acts meant to unravel in 3 consequent days.

Readiness is a serialized simulation of the unfolding political antagonism. It adapts real, often scary and cringey collective identities to a world permeated by fantasy aesthetics and infantile hopepunk sentimentalism.

The Preppers were played as herbalists by two participants, catering their tented plantation of blood-producing crops. Their main task was to protect the garden and the Egg (home of Jeso) against the attacks of the Golden Horde, provide ingredients for the Oracle's potions, and develop a fungus that could exterminate the Horde. On the second day, a prepper conspirated with the Boogaloo Bois to snatch his fellow prepper and carry out sick experiments on him.

The Golden Horde started in a dismal state. Batu, the legendary leader of the Horde, woke up captured and imprisoned by Boogaloo Bois. The ExCop (member of the Horde) was also chained and detained. The Looter (Horde) struggled through a hidden tunnel to free the ExCop, while the Activist (Horde) provided guidance from the "Art of War". Their main goal was to steal everything that was not nailed down (weapons, lockets, chains, props etc.) and use it to conquer the Egg (home of Jeso). By the 3rd day of the Game, the Horde prevailed, bashing and humiliating the filthy Preppers and Boogaloo Bois. They then published the "Declaration of the Golden Horde", and disseminated memes for the global empowerment of the Horde.

The Boogaloo Bois were monitoring and torturing Batu, the captured leader of the Horde. Their task was to capture most members of the horde, cage them, and experiment with bootleg neuralink Brain Implants (-1 Intelligence, +1 Perception) on the heads of their caged specimens. The neuralink tryouts were ordered and commissioned by their boss, whose tech company is about to launch a new project to develop neural connections between electronic devices and the brain, aiming to monitor brain activity. Meanwhile, those space boys enjoyed to mock their fellow preppers for their obsolete attachment to the cultivation of the soil.

The Oracle casted spells against the Horde, sanctified the Preppers' weapons and fed the preppers with a magical potion (+2 Constitution), using a sacred spoon. She sat on her throne, communicated with Jeso through a sacred lore and delivered ritual litanies to protect the Egg. On the 3rd day, she turned sick and spread a disease to all.

Pudge raided the camp on the second day causing total mayhem. It cried out a duress to the camera but it was ignominiously put down by the Sensei.

The Sensei supervised the game's armory taking trades to provide players with weapons. A true warlady, Sensei taught defence techniques (+2 to Unarmed Combat) to participants and flexed breakdance moves. A system of barter economy governed the relations between clans and persons. Anyone was allowed to barter.

Jeso dwelled the Egg, the sacred capsule that the Horde sought to conquer. Operating through its secluded space, Jeso sang anthems and blessings but it was mostly preoccupied with narcissistic self-curation and texting on social media and Youtube's live chat. On the 3rd day, Jeso blessed the Horde and sided with their victorious insurgency against preppers and Boogaloo Bois.

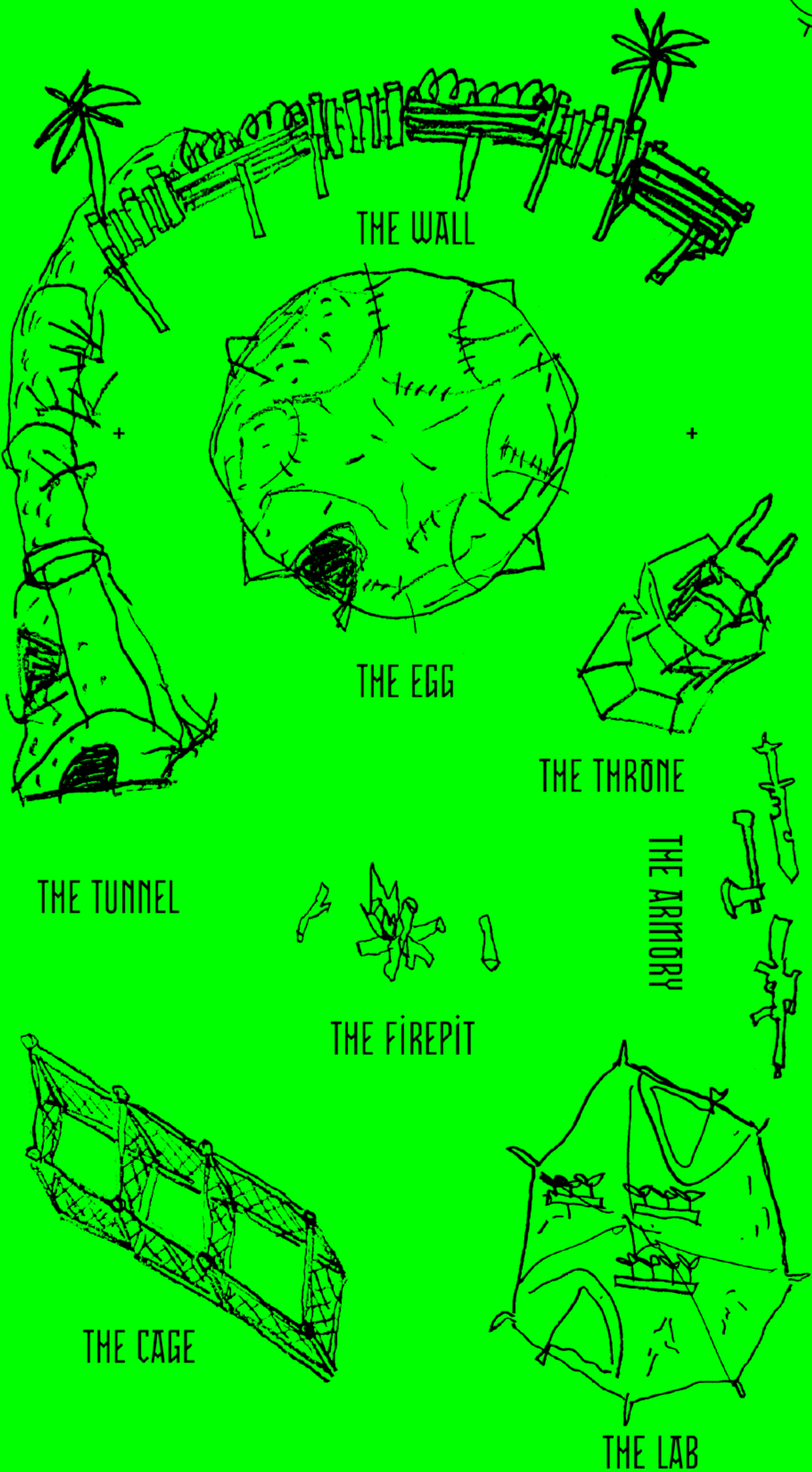
The Operator had installed a CCTV monitoring system and a series of sensors distributed to the subjects. The system was simultaneously broadcasting the experiment to a group of observers. Data was extensively collected from the head mounted cameras, neuralink waveforms, crowd tracking system, player vectors and the postgame forensics scan. This data will now be analysed and used as the primary Data Set fed into an Artificial Neural Network trained to predict the outcome of the upcoming Civil War.

Report Signed by : Prepper Herbalist & The Operator

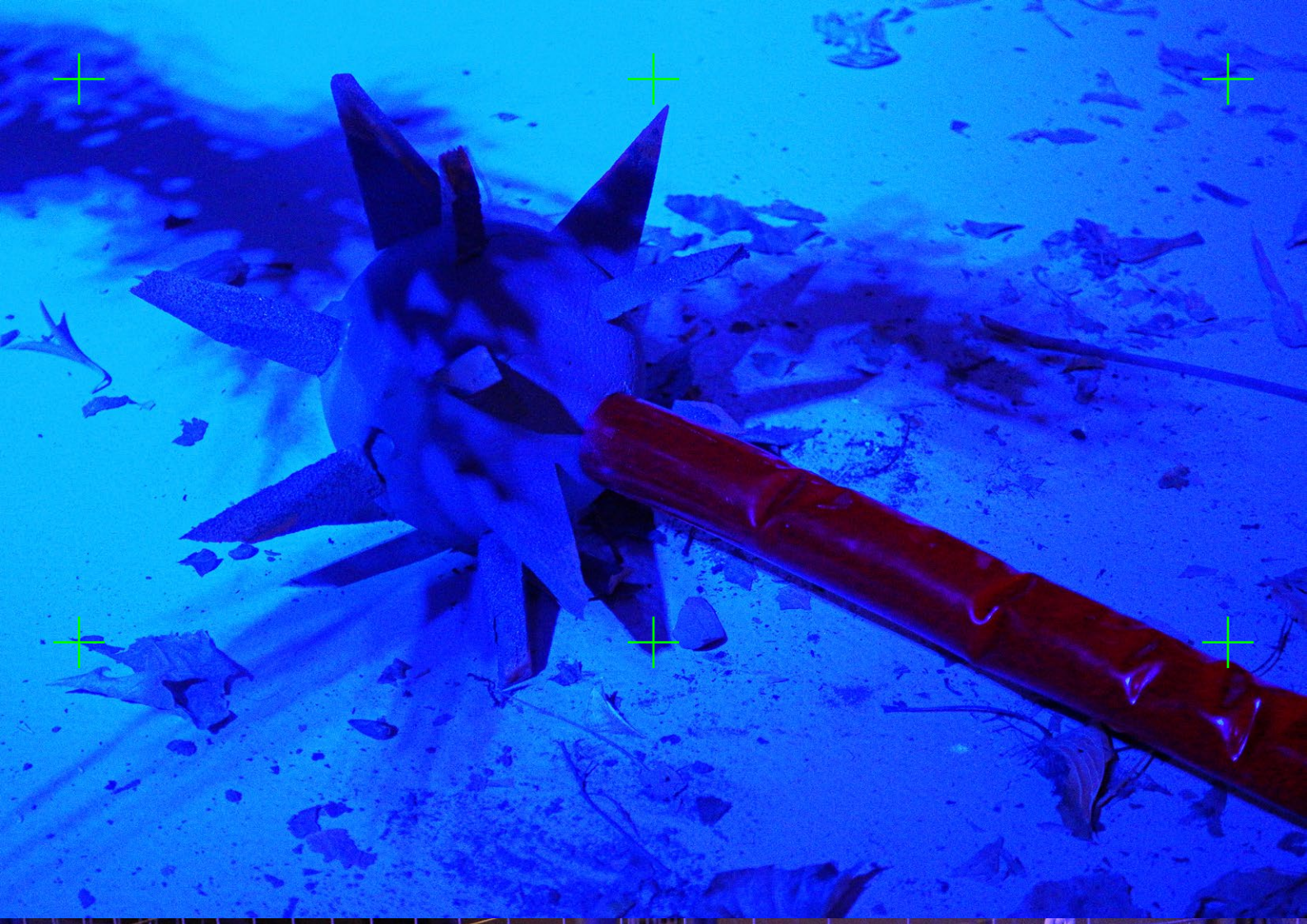
November 2020


BRIDGESTONE

Readiness v1.0
Zooom v45.134f
Streamlabis v4.56



OFF GAME AREA




MACK

PLAYER GUIDE:

READINESS

THE LARP

Readiness is a Chamber LARP for 8-16 Players and a Game Master that takes place inside the “Encampment”, a fictional campsite constructed by the Preppers to protect themselves against an upcoming apocalypse of mysterious origins. Readiness is informed by current global turmoil and the attitudes of survival, alienation and vigilance built around it. It combines elements of gaming culture, performance, lyric theater and post-media art.

The LARP will play out over 4 days, including an introduction and character building workshop and 3 Acts played in real time over 3 days. Players of Readiness will be assigned to one of 3 Clans (Preppers, Horde, Boogaloos, Neutral) and given a Character Outline, who they can further design and refine. The LARP will be broadcasted as a Live Stream and recorded for further editing.

The world of Readiness is a bleak apocalyptic dystopia where madness, misery and toxic personalities have won over reason. It takes place within the Encampment. We are unsure what lies beyond. The Encampment consists of a series of Key Locations. These are The Egg, the Tunnels, the Firepit, the Lab, the Throne, the Cage, the Armory and the Wall.

Readiness is not a combat based LARP. But when needed the following rules can be used:
First hit gets you wounded. Second gets you unconscious and bleeding. Third gets you dead. Light armor gives you 1 point of resistance, heavy armor 2 points. Entire body is 1 armor zone, so wherever you get hit your protection is weakened. You only get protection where you are covered by armor. 2-handed weapons will ignore light armor. Shields are unusable after 10 strikes by a 2-handed weapon. Don’t hit the head, neck or groin areas. Getting hit, wounds and unconsciousness need to be roleplayed. When you’re reduced to unconscious, you bleed for 15 minutes – if no one starts healing you by then, your character dies and can no longer be played. Safety is the first rule – call a hold if there’s any danger. LARP is a physical activity and you play at your own risk and responsibility. Roleplay good, play nice and have fun.

Bring a starting attire that matches your character. Additional accessories, props and weapons will be provided. Wear sports shoes. Additional Costumes, Props and Weapons will be provided. Items can be Traded or Looted during the game. Upgrades and Special items can be acquired at the Armory and the Lab. Players can also Craft custom items during the game.

Meta Techniques: The Game Master will be intervening minimally, to help guide the main plot using a few meta techniques that will be explained during the intro.

Play to Lose - Play to Lift : A technique to create better drama by not trying to win, but letting your character lose. It is used in a collaborative play style rather than a competitive play style. Play to Lift means that the responsibility for your drama and your character also rests on all your co-players. You have to lift each other.

The Clans

Boogaloos
Far-right, pro-gun, anti-government, and extremist militia group born out of 4chan memes. Inspired by right wing accelerationist ideas.
Goal: Ignite a Civil War
Special: Meme Magick. Force Neutral Characters to take a side for 5 minutes.

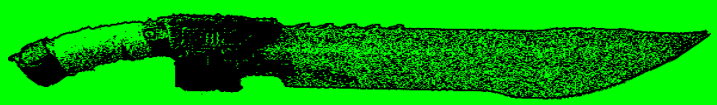
Preppers
Organized and obsessed with self-reliance, stockpiling supplies, and survival knowledge.
Goal: Protect their Property.
Special: Survival Skills. Can Craft new objects and Escape tough situations.

Horde
The mass of the unprepared and unorganized turned animalistic in their agony for survival.
Goal: Loot.
Special: Strength of the Night. The Horde is 50% Invisible during nighttime.

Neutral
Neutral Characters are in this either for personal gain or for the greater good. It's hard to say.

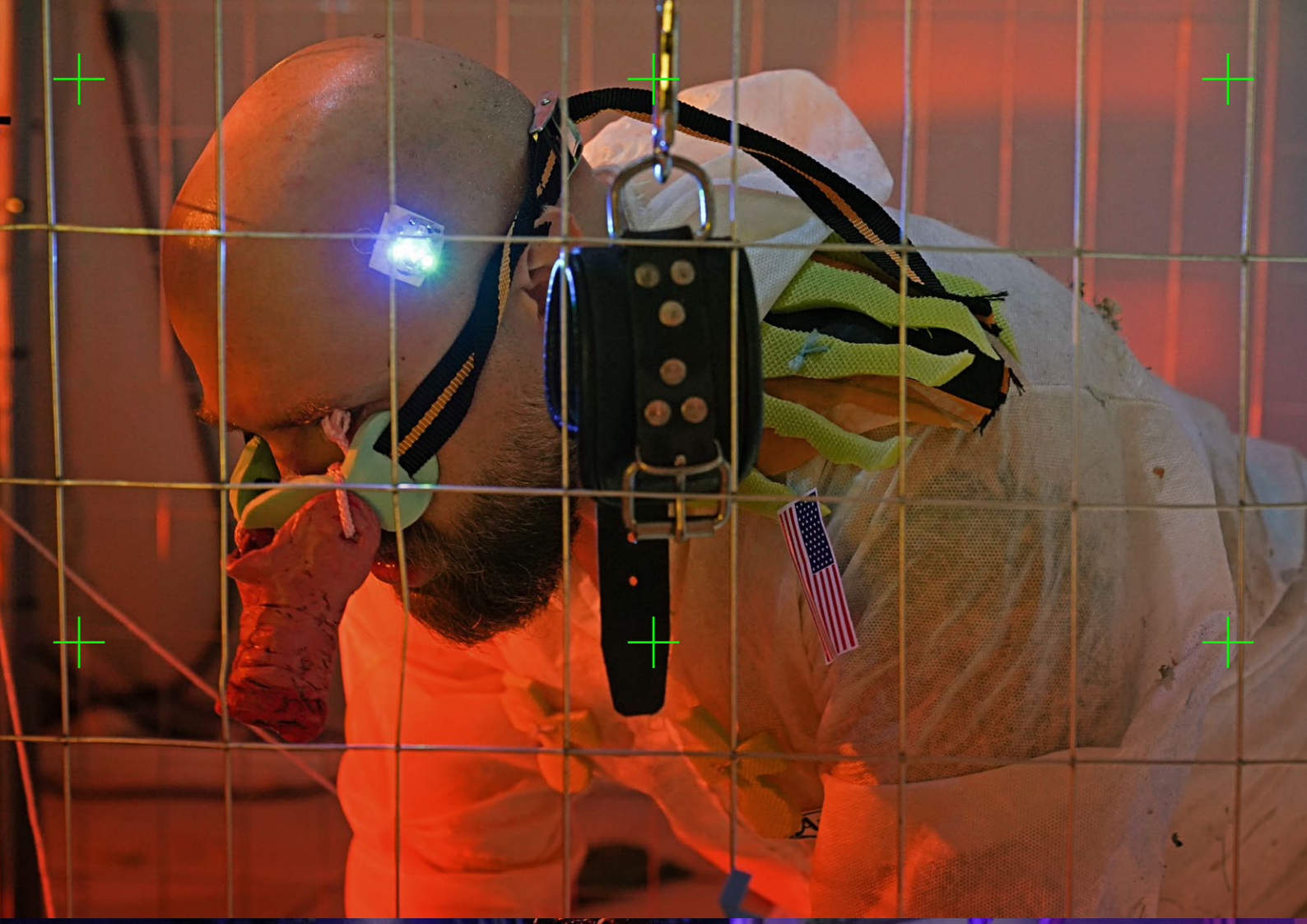
The Characters

Oracle (Neutral) - Psychic that has insights on the catastrophe. Skill: Psy Spells & Potions
Messiah (Neutral) - Elevated Spiritual Being. Skill: Release Civil Tension
Bard (Neutral) - A true artist of the community, respected by all Clans. Skill: Messenger
Operator (Prepper) - Controls the Security System at the Encampment. Skill: Assign Cam
Sensei (Prepper) - Martial Arts Expert. Skill: Teach Melee Combat
Bunker Bitch (Prepper) - Previously Privileged Alpha Female. Skill: Persuasion
Herbalist (Prepper) - Bonsai Hydroponics gardener. Skill: Heal / Revive
Boi 1 (Boogaloo) - Ideologically Confused Beta Haiwaian Tactical. Skill: Games
Boi 2 (Boogaloo) - Ideologically Confused Beta Haiwaian Tactical. Skill: Firearms
Batu (Horde) - Retired Warrior Leader of the Golden Horde. Skill: Art of War
ExCop (Horde) - Former SWAT cop with PTSD. Conspiracy Theorist. Skill: Combat
Activist (Horde) - Dancer Medic Protester. Skill: Heal / Revive
Looter (Horde) - Here to loot for fashion brands and resell on ebay. Skill: Traps/Lockpick




AVILA
C. TSALERAS S.A.


PLUG
& PLAY




MOTORE
NO FEA


novation


BLACK


panda

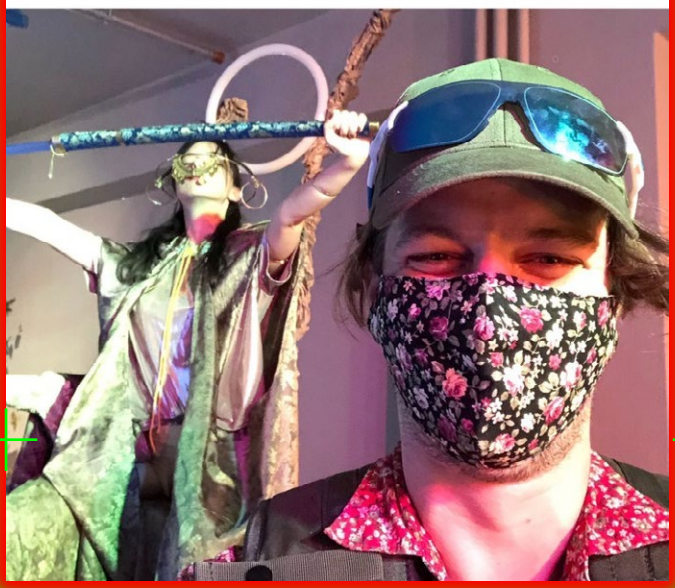
THE DECLARATION OF THE GOLDEN HORDE

(AS IT WAS TRUMPETED BY THE HORDE ITSELF AT THE ENDING PART OF THE GAME)

LIFE IN THIS SOCIETY BEING, AT BEST, AN UTTER BORE AND NO ASPECT OF SOCIETY BEING AT ALL RELEVANT TO THE GOLDEN HORDE, THERE REMAINS TO CIVIC-MINDED, RESPONSIBLE, THRILL-SEEKING HORDERS ONLY TO OVERTHROW THE GOVERNMENT, ELIMINATE THE MONEY SYSTEM, INSTITUTE COMPLETE AUTOMATION AND DESTROY THE PREPPERS. IT IS NOW TECHNICALLY FEASIBLE TO REPRODUCE WITHOUT THE AID OF PREPPERS OR, FOR THAT MATTER, HORDERS AND TO PRODUCE ONLY HORDERS. WE MUST BEGIN IMMEDIATELY TO DO SO. RETAINING THE PREPPER HAS NOT EVEN THE DUBIOUS PURPOSE OF REPRODUCTION. THE PREPPER IS A BIOLOGICAL ACCIDENT: THE Y PREPPER GENE IS AN INCOMPLETE X HORDE GENE, THAT IS, IT HAS AN INCOMPLETE SET OF CHROMOSOMES. IN OTHER WORDS, THE PREPPER IS AN INCOMPLETE HORDER, A WALKING ABORTION, ABORTED AT THE GENE STAGE. TO BE PREPPER IS TO BE DEFICIENT, EMOTIONALLY LIMITED; PREPAREDNESS IS A DEFICIENCY DISEASE AND PREPPERS ARE EMOTIONAL CRIPPLES. THE PREPPER IS COMPLETELY EGOCENTRIC, TRAPPED INSIDE HIMSELF, INCAPABLE OF EMPATHIZING OR IDENTIFYING WITH OTHERS, OR LOVE, FRIENDSHIP, AFFECTION OR TENDERNESS. HE IS A COMPLETELY ISOLATED UNIT, INCAPABLE OF RAPPORT WITH ANYONE. HIS RESPONSES ARE ENTIRELY VISCERAL, NOT CEREBRAL; HIS INTELLIGENCE IS A MERE TOOL IN THE SERVICES OF HIS DRIVES AND NEEDS; HE IS INCAPABLE OF MENTAL PASSION, MENTAL INTERACTION; HE CANOT RELATE TO ANYTHING OTHER THAN HIS OWN PHYSICAL SENSATIONS. HE IS A HALF-DEAD, UNRESPONSIVE LUMP, INCAPABLE OF GIVING OR RECEIVING PLEASURE OR HAPPINESS; CONSEQUENTLY, HE IS AT BEST AN UTTER BORE, AN INOFFENSIVE BLOB, SINCE ONLY THOSE CAPABLE OF ABSORPTION IN OTHERS CAN BE CHARMING. HE IS TRAPPED IN A TWILIGHT ZONE HALFWAY BETWEEN HUMANS AND APES, AND IS FAR WORSE OFF THAN THE APES BECAUSE, UNLIKE THE APES, HE IS CAPABLE OF A LARGE ARRAY OF NEGATIVE FEELINGS - HATE, JEALOUSY, CONTEMPT, DISGUST, GUILT, SHAME, DOUBT - AND MOREOVER, HE IS AWARE OF WHAT HE IS AND WHAT HE ISNOT.

THE DECLARATION WAS AN UNAPOLOGETIC APPROPRIATION OF A FAMOUS PART OF SCUM MANIFESTO BY VALERIE SOLANAS.

-“Dad, im gonna study fine arts”
Fine arts:



“The golden horde is near”
- The golden horde:

No one:
Preppers:

You VS the guy she told you not to worry about

OperatorLog.txt - Data Written on 18 Oct 2020

18:23:25.697: CPU Name: Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz
18:23:25.697: CPU Speed: 2808MHz
18:23:25.697: Physical Cores: 4, Logical Cores: 8
18:23:25.697: Physical Memory: 16340MB Total, 10729MB Free
18:23:25.697: Windows Version: 10.0 Build 18362 (revision: 1082; 64-bit)
18:23:25.697: Running as administrator: false
18:23:25.697: Aero is Enabled (Aero is always on for windows 8 and above)
18:23:25.697: Windows 10 Gaming Features:
18:23:25.697: Game DVR: On
18:23:25.699: Sec. Software Status:
18:23:25.701: Windows Defender Antivirus: enabled (AV)
18:23:25.702: Windows Firewall: enabled (FW)
18:23:25.702: Current Date/Time: 2020-10-18, 18:23:25
18:23:25.702: Browser Hardware Acceleration: true
18:23:25.702: Portable mode: false
18:23:26.190: OBS 24.0.3 (64-bit, windows)
18:23:26.190: ---------------------------------
18:23:26.214: ---------------------------------
18:23:26.214: audio settings reset:
18:23:26.214: samples per sec: 44100
18:23:26.214: speakers: 2
18:23:26.215: ---------------------------------
18:23:26.215: Initializing D3D11...
18:23:26.215: Available Video Adapters:
18:23:26.217: Adapter 0: NVIDIA GeForce GTX 1060
18:23:26.217: Dedicated VRAM: 3132096512
18:23:26.217: Shared VRAM: 4272400384
18:23:26.217: output 0: pos={0, 0}, size={1920, 1080}, attached=true
18:23:26.220: Loading up D3D11 on adapter NVIDIA GeForce GTX 1060 (0)
18:23:26.274: D3D11 loaded successfully, feature level used: b000
18:23:26.274: D3D11 GPU priority setup failed (not admin?)
18:23:26.765: ---------------------------------
18:23:26.765: video settings reset:
18:23:26.765: base resolution: 1920x1080
18:23:26.765: output resolution: 1920x1080
18:23:26.765: downscale filter: Bicubic
18:23:26.765: fps: 30/1
18:23:26.765: format: NV12
18:23:26.765: YUV mode: 601/Partial
18:23:26.765: NV12 texture support enabled
18:23:26.767: Audio monitoring device:
18:23:26.767: name: Default
18:23:26.767: id: default
18:23:26.767: ---------------------------------
18:23:26.770: [CoreAudio encoder]: CoreAudio AAC encoder not installed on the system or couldn't be loaded
18:23:26.771: Failed to load 'en-US' text for module: 'decklink-ouput-ui.dll'
18:23:26.807: [AMF] AMF Test failed due to one or more errors.
18:23:26.807: Failed to initialize module 'enc-amf.dll'
18:23:26.827: [obs-browser]: Version 2.7.15
18:23:26.831: NVENC supported
18:23:26.897: Couldn't find VLC installation, VLC video source disabled
18:23:26.904: No blackmagic support
18:23:26.911: ---------------------------------
18:23:26.911: Loaded Modules:
18:23:26.911: win-wasapi.dll
18:23:26.911: win-mf.dll
18:23:26.911: win-dshow.dll
18:23:26.911: win-decklink.dll
18:23:26.911: win-capture.dll
18:23:26.911: vlc-video.dll
18:23:26.911: text-freetype2.dll
18:23:26.911: rtmp-services.dll
18:23:26.911: obs-x264.dll
18:23:26.911: obs-vst.dll
18:23:26.911: obs-transitions.dll
18:23:26.911: obs-text.dll
18:23:26.911: obs-qsv11.dll
18:23:26.911: obs-outputs.dll
18:23:26.911: obs-filters.dll
18:23:26.911: obs-ffmpeg.dll
18:23:26.911: obs-browser.dll
18:23:26.911: image-source.dll
18:23:26.911: frontend-tools.dll
18:23:26.911: enc-amf.dll
18:23:26.911: droidcam-obs.dll
18:23:26.911: decklink-ouput-ui.dll
18:23:26.911: coreaudio-encoder.dll
18:23:26.911: ---------------------------------
18:23:26.911: ==== Startup complete ===
18:23:26.918: All scene data cleared
18:23:26.919: --
18:23:27.202: WASAPI: Device 'Speakers (Realtek High Definition Audio)' initialized
18:23:27.280: WASAPI: Device 'Speakers (Realtek High Definition Audio)' initialized
18:23:27.350: WASAPI: Device 'Microphone (Realtek High Definition Audio)' initialized
18:23:27.384: adding 46 milliseconds of audio buffering, total audio buffering is now 46 milliseconds (source: Mic/Aux)
18:23:27.384:
18:23:27.431: WASAPI: Device 'Microphone (HD Pro Webcam C920)' initialized
18:23:27.456: WASAPI: Device 'Microphone (HD Webcam C310)' initialized
18:23:27.467: WASAPI: Device 'Microphone (Elgato Sound Capture)' initialized
18:23:27.468: [Media Source 'Pigs_Prezi']: settings:
18:23:27.468: input: D:/Users/Raptor/Documents/Projects/Readiness/TheLarp/Pigs_Prezi.mp4
18:23:27.468: input_format: (null)
18:23:27.468: speed: 100
18:23:27.468: is_looping: yes
18:23:27.468: is_hw_decoding: no
18:23:27.468: is_clear_on_media_end: yes
18:23:27.468: restart_on_activate: yes
18:23:27.468: close_when_inactive: no
18:23:27.567: [Media Source 'CreditsClip']: settings:
18:23:27.567: input: D:/Users/Raptor/Documents/Projects/Readiness/TheLarp/EndCredits_FX.mp4
18:23:27.567: input_format: (null)
18:23:27.567: speed: 100
18:23:27.567: is_looping: yes
18:23:27.567: is_hw_decoding: no
18:23:27.567: is_clear_on_media_end: no
18:23:27.567: restart_on_activate: yes
18:23:27.567: close_when_inactive: no
18:23:27.568: [Media Source 'IntroClip']: settings:
18:23:27.568: input: D:/Users/Raptor/Documents/Projects/Readiness/TheLarp/IntroScreen_FX.mp4
18:23:27.568: input_format: (null)
18:23:27.568: speed: 100
18:23:27.568: is_looping: yes
18:23:27.568: is_hw_decoding: no
18:23:27.568: is_clear_on_media_end: no
18:23:27.568: restart_on_activate: yes
18:23:27.568: close_when_inactive: no
18:23:28.110: [DroidCamOBS] create(source=00000256177B0AA0) v1.0
18:23:30.189: [DroidCamOBS] activated=1, deactivateWNS=0, is_showing=0, enable_audio=0
18:23:30.189: [DroidCamOBS] video_format=avc video_resolution=1280x720
18:23:30.189: [DroidCamOBS] device_info.id=dev_id_wifi device_info.ip=192.168.43.201 device_info.port=4747 device_info.type=1
18:23:30.190: [DroidCamOBS] video_thread start
18:23:30.190: [DroidCamOBS] video_decode_thread start
18:23:30.190: [DroidCamOBS] audio_thread start
18:23:30.192: Switched to scene '4xCAMS'
18:23:30.193: --
18:23:30.193: Loaded scenes:
18:23:30.193: - scene 'Scene':
18:23:30.193: - source: 'Display Capture' (monitor_capture)
18:23:30.193: - source: 'Game Capture' (game_capture)
18:23:30.193: - scene 'Intro':
18:23:30.193: - source: 'IntroClip' (ffmpeg_source)
18:23:30.193: - scene '4xCAMS':
18:23:30.193: - source: 'LaptopCam' (dshow_input)
18:23:30.193: - source: '1_Logitec720' (dshow_input)
18:23:30.193: - filter: 'Color Correction' (color_filter)
18:23:30.193: - source: '2_Logitec720' (dshow_input)
18:23:30.193: - filter: 'Color Correction' (color_filter)
18:23:30.193: - source: '3_Logitec1080' (dshow_input)
18:23:30.193: - source: 'Space_Elgato Cam' (dshow_input)
18:23:30.193: - filter: 'Color Correction' (color_filter)
18:23:30.193: - filter: 'Noise Suppression' (noise_suppress_filter)
18:23:30.193: - source: 'DroidCam OBS' (droidcam_obs)
18:23:30.193: - source: 'OV_4CAM' (image_source)
18:23:30.193: - scene '3_1xCAMs':
18:23:30.193: - source: 'Space_Elgato Cam' (dshow_input)
18:23:30.193: - filter: 'Color Correction' (color_filter)
18:23:30.193: - filter: 'Noise Suppression' (noise_suppress_filter)
18:23:30.193: - source: '1_Logitec720' (dshow_input)
18:23:30.193: - filter: 'Color Correction' (color_filter)
18:23:30.193: - source: '2_Logitec720' (dshow_input)
18:23:30.193: - filter: 'Color Correction' (color_filter)
18:23:30.193: - source: '3_Logitec1080' (dshow_input)
18:23:30.193: - source: 'LaptopCam' (dshow_input)
18:23:30.193: - source: 'OV_3_1' (image_source)
18:23:30.193: - scene 'S_SpaceCam':
18:23:30.193: - source: 'LaptopCam' (dshow_input)
18:23:30.193: - source: '1_Logitec720' (dshow_input)
18:23:30.193: - filter: 'Color Correction' (color_filter)
18:23:30.193: - source: '2_Logitec720' (dshow_input)
18:23:30.193: - filter: 'Color Correction' (color_filter)
18:23:30.193: - source: '3_Logitec1080' (dshow_input)
18:23:30.193: - source: 'Space_Elgato Cam' (dshow_input)
18:23:30.193: - filter: 'Color Correction' (color_filter)
18:23:30.193: - filter: 'Noise Suppression' (noise_suppress_filter)
18:23:30.193: - source: 'OV_Solo_Misc' (image_source)
18:23:30.193: - scene 'S_Cam3':
18:23:30.193: - source: 'LaptopCam' (dshow_input)
18:23:30.193: - source: '1_Logitec720' (dshow_input)
18:23:30.193: - filter: 'Color Correction' (color_filter)
18:23:30.193: - source: '2_Logitec720' (dshow_input)
18:23:30.193: - filter: 'Color Correction' (color_filter)
18:23:30.193: - source: '3_Logitec1080' (dshow_input)
18:23:30.193: - source: 'Space_Elgato Cam' (dshow_input)
18:23:30.193: - filter: 'Color Correction' (color_filter)
18:23:30.193: - filter: 'Noise Suppression' (noise_suppress_filter)
18:23:30.193: - source: 'OV_Solo_Misc' (image_source)
18:23:30.193: - scene 'S_Cam1':
18:23:30.193: - source: 'LaptopCam' (dshow_input)
18:23:30.193: - source: '1_Logitec720' (dshow_input)
18:23:30.193: - filter: 'Color Correction' (color_filter)
18:23:30.193: - source: '2_Logitec720' (dshow_input)
18:23:30.193: - filter: 'Color Correction' (color_filter)
18:23:30.193: - source: '3_Logitec1080' (dshow_input)
18:23:30.193: - source: 'Space_Elgato Cam' (dshow_input)
18:23:30.193: - filter: 'Color Correction' (color_filter)
18:23:30.193: - filter: 'Noise Suppression' (noise_suppress_filter)
18:23:30.193: - source: 'OV_Solo_Misc' (image_source)
18:23:30.193: - scene 'S_Cam2':
18:23:30.193: - source: 'LaptopCam' (dshow_input)
18:23:30.193: - source: '1_Logitec720' (dshow_input)
18:23:30.193: - filter: 'Color Correction' (color_filter)
18:23:30.193: - source: '2_Logitec720' (dshow_input)
18:23:30.193: - filter: 'Color Correction' (color_filter)
18:23:30.193: - source: '3_Logitec1080' (dshow_input)
18:23:30.193: - source: 'Space_Elgato Cam' (dshow_input)
18:23:30.193: - filter: 'Color Correction' (color_filter)
18:23:30.193: - filter: 'Noise Suppression' (noise_suppress_filter)
18:23:30.193: - source: 'OV_Solo_Misc' (image_source)
18:23:30.193: - scene 'S_CamOp':
18:23:30.193: - source: 'LaptopCam' (dshow_input)
18:23:30.193: - source: '1_Logitec720' (dshow_input)
18:23:30.193: - filter: 'Color Correction' (color_filter)
18:23:30.193: - source: '2_Logitec720' (dshow_input)
18:23:30.193: - filter: 'Color Correction' (color_filter)
18:23:30.193: - source: '3_Logitec1080' (dshow_input)
18:23:30.193: - source: 'Space_Elgato Cam' (dshow_input)
18:23:30.193: - filter: 'Color Correction' (color_filter)
18:23:30.193: - filter: 'Noise Suppression' (noise_suppress_filter)


READINESS :: THE LARP
SOLO CAM_


LIVE 2020
READINESS :: THE LARP
SOLO CAM_


LIVE 2020


FIRST PERSON CAM_


SECOND PERSON CAM_


LIVE 2020
GAME SPACE CAM


FIRST PERSON CAM

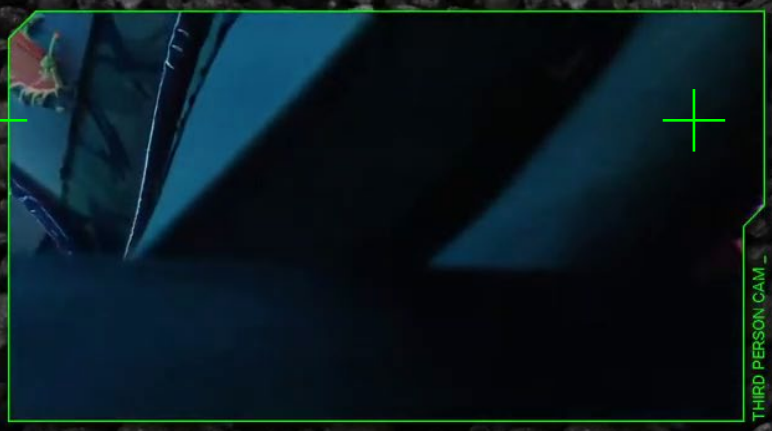

SECOND PERSON CAM
READINESS :: THE LARP


THIRD PERSON CAM


LIVE 2020
GAME SPACE CAM


FIRST PERSON CAM


SECOND PERSON CAM
READINESS :: THE LARP


THIRD PERSON CAM


LIVE 2020
GAME SPACE CAM


FIRST PERSON CAM

READINESS

THE LARP

Dark Web Leak: Norwegian Preppers testing Bootleg Neuralink on Human Pigs

Participants have been held captives for nearly 2 months and subjected to DIY skull surgeries. Medical specialists worried, while Elon Musk refused to make any comments.

JC By Jelisa Castrodale

October 14, 2020, 9:52pm Share Tweet Snap

Yesterday, the LinkedIn page for Neuralink shared its first update in more than four months, posting a screenshot of a recent tweet from Elon Musk and reiterating that it's hiring. "As Elon mentions, we're looking for engineers who've solved hard problems with phones and wearables," it wrote. "The world has done so much to optimize telecommunications signals. Now it's time to do the same for brain signals."

The San Francisco-based company describes itself as a developer of "ultra-high bandwidth brain-machine interfaces to connect humans and computers," which is a multisyllabic way of saying that it wants to implant a tiny chip in people's brains. Neuralink was co-founded by Musk in 2016, but it hasn't shared anything new about its mission or its progress since last July, when it posted an ODESZA-soundtracked introductory video and Musk briefly discussed the concept of having ultra-thin "threads" implanted in your brain to "achieve a sort of symbiosis with artificial intelligence."

Earlier this month, Musk took a sec from posting Dogecoin memes to write that an update about Neuralink's progress would be coming on August 28. He also responded to someone who asked him whether, after implementing neuralink, it would be possible to "listen to music directly from our chips?" Musk simply wrote "Yes," but that has caused all new speculation about whether or not that's a possibility, with Neuralink or anything else.

The idea of transmitting some sounds directly into our heads without headphones or speakers isn't new; Beethoven apparently connected one end of a rod to the sounding board inside his piano and bit down on the other end so he could "hear" the vibrations. Some 'bone-conduction' headphones,

MORE LIKE THIS

Tech

'Scam' Spyware Vendor Gets Caught, Once Again

LORENZO FRANCESCHI-BICCHIERAI

05.19.20

Tech

How One Company Collected Browsing Data Via Android Apps

JOSEPH COX

07.31.20

Entertainment

The Truth Behind Those Videos of Monkeys Riding Tiny Motorcycles

JELISA CASTRODALE

05.07.20

Food

Some Companies Are Finally Offering 'Period Leave'

JELISA CASTRODALE

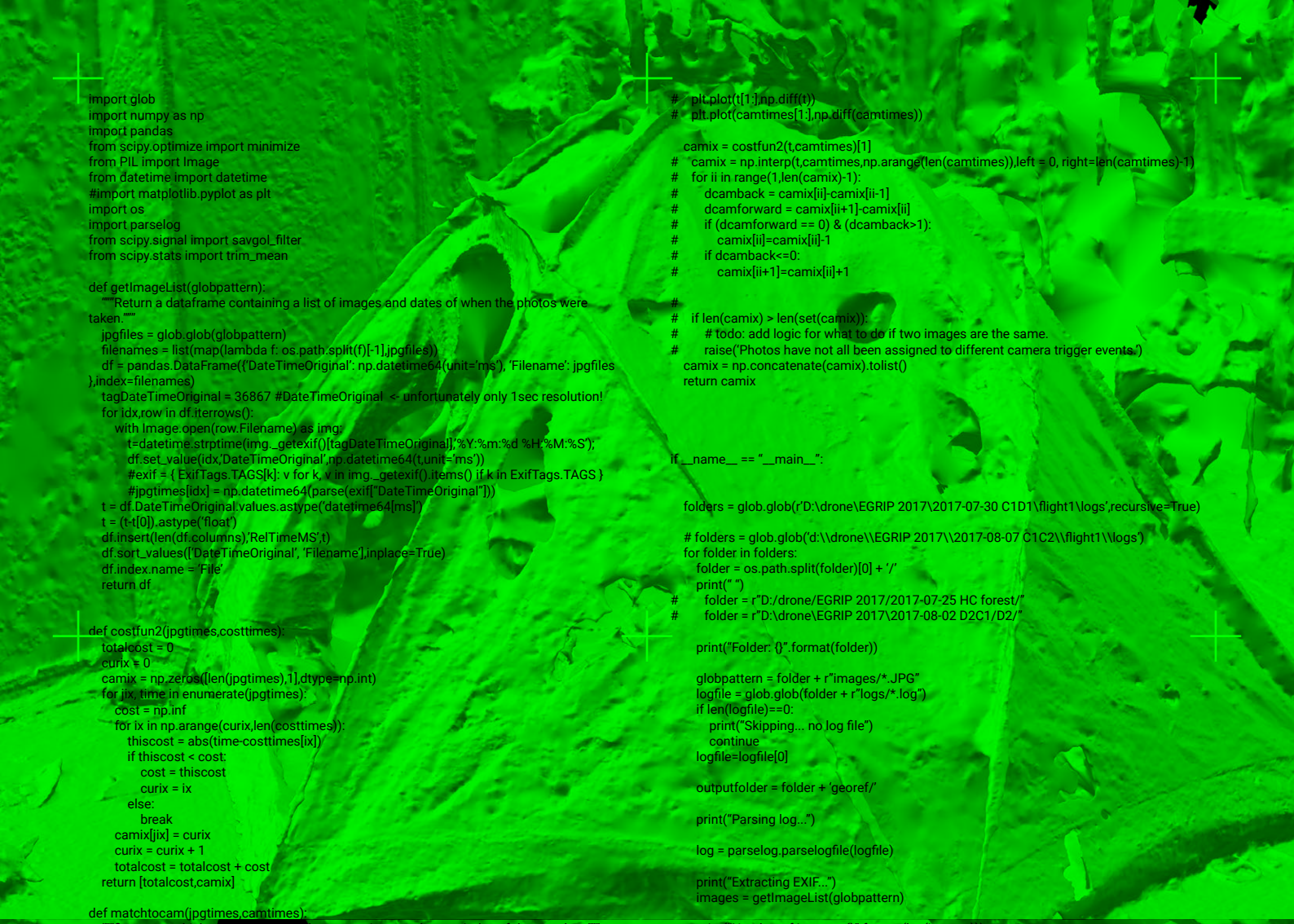
08.12.20

Sex

A Security Flaw in a Chastity Cage for Men Could Permanently Lock in Their Penis

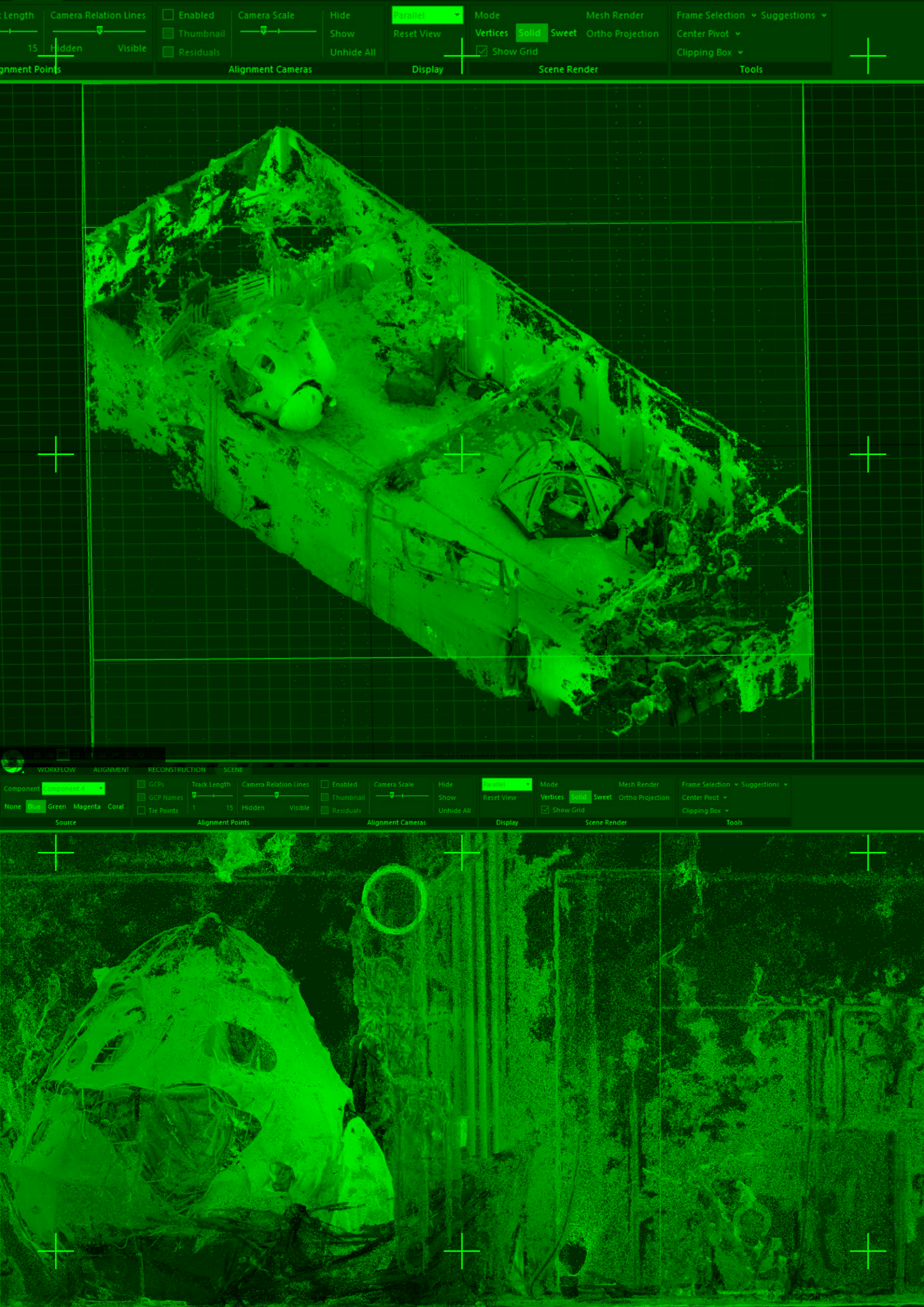
VARSHA RANI

10.07.20



```
import glob
import numpy as np
import pandas
from scipy.optimize import minimize
from PIL import Image
from datetime import datetime
#import matplotlib.pyplot as plt
import os
import parselog
from scipy.signal import savgol_filter
from scipy.stats import trim_mean

def getImageList(globpattern):
    """Return a dataframe containing a list of images and dates of when the photos were taken."""
    jpgfiles = glob.glob(globpattern)
    filenames = list(map(lambda f: os.path.split(f)[-1],jpgfiles))
    df = pandas.DataFrame({'DateTimeOriginal': np.datetime64(unit='ms'), 'Filename': jpgfiles},index=filenames)
    tagDateTimeOriginal = 36867 #DateTimeOriginal  <- unfortunately only 1sec resolution!
    for idx,row in df.iterrows():
        with Image.open(row.Filename) as img:
            t=datetime.strptime(img._getexif()[tagDateTimeOriginal],'%Y:%m:%d %H:%M:%S');
            df.set_value(idx,'DateTimeOriginal',np.datetime64(t,unit='ms'))
            #exif = { ExifTags.TAGS[k]: v for k, v in img._getexif().items() if k in ExifTags.TAGS }
            #jpgtimes[idx] = np.datetime64(parse(exif["DateTimeOriginal"]))
    t = df.DateTimeOriginal.values.astype('datetime64[ms]')
    t = (t-t[0]).astype('float')
    df.insert(len(df.columns),'RelTimeMS',t)
    df.sort_values(['DateTimeOriginal', 'Filename'],inplace=True)
    df.index.name = 'File'
    return df


def costfun2(jpgtimes,costtimes):
    totalcost = 0
    curix = 0
    camix = np.zeros([len(jpgtimes),1],dtype=np.int)
    for jix, time in enumerate(jpgtimes):
        cost = np.inf
        for ix in np.arange(curix,len(costtimes)):
            thiscost = abs(time-costtimes[ix])
            if thiscost < cost:
                cost = thiscost
                curix = ix
            else:
                break
        camix[jix] = curix
        curix = curix + 1
        totalcost = totalcost + cost
    return [totalcost,camix]

def matchtocam(jpgtimes,camtimes):
    """figure out which r[illegible]h camera times and return index of the matches."""

#    # deal with limited [illegible]
#    dt = trim_mean(np.d[illegible]roportiontocut=.1)
#    window = 2000./dt #ap[illegible]cond smoothing filter
#    window = np.round([illegible]l #ensure an odd filter
#    if window>=3:
#        tt = [illegible]filter(jpg[illegible]ow_length=int(window),polyorder=1,mode='nearest')
#        ix =
#        jpgti[illegible]

    [illegible]onsto[illegible]minimizer works better if we are closer to zero.
    camtin[illegible]rset
    [illegible]f con[illegible]list we can interpolate in to figure out distance to closest
    [illegible]imes
    [illegible]ost
    [illegible]t =
    [illegible]ost[illegible]e(-0.5,len(costtimes),.5),np.arange(len(costtimes)),cost-
    [illegible]s,left[illegible]ght=costtimes[-1]+np.max(dt))
#    cost = costtimes * 0
#    cost[2:-1:2] = dt/2
#    cost[0] = np.max(dt) #special treatment of edges.
#    cost[-1] = np.max(dt)
#
#    # this function calculates the mean(square(temporaldistance)) to nearest photo.
#    costfun = lambda offset: np.mean((np.interp(jpgtimes+offset,costtimes,cost,left=cost[0], right=cost[-1]))**2.0)

    constoffset = camtimes[0] #the minimizer works better if we are closer to zero.
    camtimes = camtimes - constoffset


    # this function calculates the mean(square(temporaldistance)) to nearest photo.
    costfun = lambda offset: costfun2(jpgtimes+offset,camtimes)[0]

    # Search for which cam corresponds to jpgtime ...
    bestcost = np.Inf
    bestoffset = 0
    for camtime in camtimes:
        for dt in np.arange(-1000,1001,250):
            curoffset = camtime - jpgtimes[0] +dt #todo: pick a less random one
            curcost = costfun(curoffset)
            if curcost < bestcost:
                bestoffset = curoffset
                bestcost = curcost
    # then optimize
    offset = minimize(costfun, bestoffset) #the cost fun has many local minima, so we need to be close to the optimal time

    t = jpgtimes + offset.x[0]

#    plt.plot(t[1:],np.diff(t))
#    plt.plot(camtimes[1:],np.diff(camtimes))

    camix = costfun2(t,camtimes)[1]
#    camix = np.interp(t,camtimes,np.arange(len(camtimes)),left = 0, right=len(camtimes)-1)
#    for ii in range(1,len(camix)-1):
#        dcamback = camix[ii]-camix[ii-1]
#        dcamforward = camix[ii+1]-camix[ii]
#        if (dcamforward == 0) & (dcamback>1):
#            camix[ii]=camix[ii]-1
#        if dcamback<=0:
#            camix[ii+1]=camix[ii]+1

#
#    if len(camix) > len(set(camix)):
#        # todo: add logic for what to do if two images are the same.
#        raise('Photos have not all been assigned to different camera trigger events.')
    camix = np.concatenate(camix).tolist()
    return camix



if __name__ == "__main__":


    folders = glob.glob(r'D:\drone\EGRIP 2017\2017-07-30 C1D1\flight1\logs',recursive=True)

    # folders = glob.glob('d:\\drone\\EGRIP 2017\\2017-08-07 C1C2\\flight1\\logs')
    for folder in folders:
        folder = os.path.split(folder)[0] + '/'
        print(" ")
#        folder = r"D:/drone/EGRIP 2017/2017-07-25 HC forest/"
#        folder = r"D:\drone\EGRIP 2017\2017-08-02 D2C1/D2/"

        print("Folder: {}".format(folder))

        globpattern = folder + r"images/*.JPG"
        logfile = glob.glob(folder + r"logs/*.log")
        if len(logfile)==0:
            print("Skipping... no log file")
            continue
        logfile=logfile[0]

        outputfolder = folder + 'georef/'

        print("Parsing log...")

        log = parselog.parselogfile(logfile)

        print("Extracting EXIF...")
        images = getImageList(globpattern)

        print("Number of images: {}".format(len(images)))
        print("Number of CAM messages in log: {}".format(len(log["CAM"])))

        print("Matching photo time to camera log")
        jpgtimes = images.RelTimeMS.values
        camtimes = log['CAM']['GPSTime'].values
        camix = matchtocam(jpgtimes,camtimes)
        print("Matched! index of first & last photo: {}-{}".format(camix[0],camix[-1]))

        jpgcams = log['CAM'].iloc[camix].copy()
        jpgcams.set_index(images.index.values,inplace =True)
        jpgcams.index.name = 'Filename'

        if not os.path.exists(outputfolder):
            os.makedirs(outputfolder)

        #jpgcams[['Lng','Lat','Alt','Yaw','Pitch','Roll']].to_csv(outputfolder + 'CamLocations_raw_CAM.txt')

        shutterdelayMS = 550 #Sony QX1
        print("Accounting for shutterlag of {} ms".format(shutterdelayMS))
        #TODO: get lat,long,alt etc from EKF1
        useEKF1 = True
        if useEKF1:
            datasource = 'EKF1' #ATT -OR- EKF1
            mappings = {'Lat': 'PN', 'Lng': 'PE', 'Alt': 'PD'}
            jt = jpgcams['GPSTime']+shutterdelayMS-log['gpstimeoffset']

            for jpgkey, ekfkey in mappings.items():
                y=log['GPS'][jpgkey]
                x=log['EKF1'][ekfkey]
                p=np.polyfit(x,y,1) #infer linear mapping...
                x=np.interp(jt,log['EKF1']['TimeMS'],log['EKF1'][ekfkey])
                jpgcams[jpgkey] = np.polyval(p,x)
        else:
            jt=jpgcams['GPSTime']+shutterdelayMS
            jpgcams['Lat'] = np.interp(jt,log['GPS']['TimeMS'],log['GPS']['Lat'])
            jpgcams['Lng'] = np.interp(jt,log['GPS']['TimeMS'],log['GPS']['Lng'])
            jpgcams['Alt'] = np.interp(jt,log['GPS']['TimeMS'],log['GPS']['Alt'])
            datasource = 'ATT' #ATT -OR- EKF1


        jt = jpgcams['GPSTime']+shutterdelayMS-log['gpstimeoffset']
        #todo: protect against circular overflows - quaternion interpolation... (Gimbal lock extremely unlikely though)
        jpgcams['Roll'] = np.interp(jt,log[datasource]['TimeMS'],log[datasource]['Roll'])
        jpgcams['Pitch'] = np.interp(jt,log[datasource]['TimeMS'],log[datasource]['Pitch'])
        log[datasource]['Yaw'] = np.unwrap(log[datasource]['Yaw']*np.pi/180)*180/np.pi
        jpgcams['Yaw'] = np.interp(jt,log[datasource]['TimeMS'],log[datasource]['Yaw']) % 360

        outputfilename = outputfolder + 'CamLocations_{}lag.txt'.format(shutterdelayMS)
        jpgcams[['Lng','Lat','Alt','Yaw','Pitch','Roll']].to_csv(outputfilename)
```


Enabled
Thumbnail
Residuals
Camera Scale
Hide
Show
Unhide All
Alignment Cameras
Perspective
Reset View
Display
Mode
Vertices
Solid
Sweet
Mesh Render
Ortho Projection
Show Grid
Scene Render
Frame Selection
Center Pivot
Clipping Box
Tools


Length
Camera Relation Lines
15
Hidden
Visible
Enabled
Thumbnail
Residuals
Camera Scale
Hide
Show
Unhide All
Parallel
Reset View
Mode
Vertices
Solid
Sweet
Mesh Render
Ortho Projection
Show Grid
Frame Selection
Suggestions
Center Pivot
Clipping Box
gnment Points
Alignment Cameras
Display
Scene Render
Tools
WORKFLOW
ALIGNMENT
RECONSTRUCTION
SCENE
Component
Component 4
None
Blue
Green
Magenta
Coral
GCPs
GCP Names
Tie Points
Track Length
1
15
Source
Alignment Points

READINESS
THE LARP

BY
KOSTIS STAFYLAKIS
THEO TRIANTAFYLLIDIS
ALEXIS FIDETZIS

WITH
VASILIS BAKALIS
STATHIS CHALKIAS
SOTIRIS FOKEAS
CHRISTOS FOUSEKIS
MARILIA KAISAR
KOSMAS KOSMOPOULOS
MARKELLA KSILOGIANNOPOULOU
ANNA SAMARA
LIA SMARAGDA
SAVVAS TSIMOURIS
VASSILIS VLASTARAS
POKA-YIO

MOVEMENT ADVISOR
MARIA GORGIA

MORE AT
SLIMETECH.ORG/READINESS