DOXCS

d = episteme(VoI)

by Nikola Supuković

//////----------……...

VR Experiment, 2023 (Oculus Quest 2, Headphones, Custom Software, XBOX 360 Controller, PC)

Messy Source Code available on Github:

CONTENT:

WORK DESCRIPTION (ACADEMIC BIG WORDS BS VERSION)

This artistic research project explores the interconnected nature of data extraction/processing, big data practices, and gamification while aiming to elucidate the relationships between game/behavior design, flow states, and the seemingly voluntary actions that lead to data production.

//GAME
At the level of a computer game, the project presents an interactive framework for voluntary engagement. Players willingly enter a designed environment, which operates with a certain level of opacity. The design of the system, including its affordances and rules, guides players towards achieving flow states. In this context, a metaphorical representation of "banging one's head against a wall" is employed to induce a meditative and iterative condition—a state of objective delusion. This repetitive action allows players to attempt to transcend the current context by iteratively engaging with it. The system provides players with modulated sounds that can be controlled and adjusted according to their preferences, enabling them to immerse themselves in the provided environment. After a certain period of time, the virtual room's impressions become distorted and ominous. At this point, players are prompted to categorize what they observe based on machine-generated instructions ("Press the button when you see a X"). In a comfortable state, their categorizations are recorded and stored on a computer for future use in training machine learning algorithms.

//CONTROL
The project assumes that control in modern societies is predominantly acquired through voluntary actions. These actions can be seen as products of specific structural preconditions, system affordances, and the influence of collective, political, individual, and economic nodes and actors. Computer games, as a medium, embody neoliberal principles: they depict the complexities of globalized systems and contribute to the construction of an image of the individual as an empowered agent. Within this worldview, players willingly enter various virtual worlds, gaining a sense of empowerment through their ability to exercise control over the virtual environment. However, leaving these worlds is often not a voluntary choice. By participating in this game and engaging with

it, players consent to having their in-game decisions recorded. While the initial impulse may have been voluntary, the subsequent emphasis on individual responsibility and autonomy disregards any manipulative elements that may have influenced the player's choice.

//CATEGORIZATION
Machine learning contexts embody a particular notion of intelligence that can be associated with instrumental reason. Within this framework, intelligence is closely linked to generating useful outcomes, solving problems to achieve technical goals, and historically, has been employed to categorize individuals in military and racial contexts. In other words, the prevailing understanding of intelligence, as represented, categorized, and measured in established tests, already assumes and incorporates a simplified, functional, and formalized form of intelligence, while suppressing alternative conceptualizations. Intelligence, in this sense, becomes a tool to be utilized, and machine learning systems are expected to contribute to this instrumental worldview. During a flow state, players effectively become machine learners themselves, faced with the task of reducing a complex visual and auditory environment into categorized data.

//DATA
In contrast to the notion that data simply exists in its natural state, this project creates an environment that exemplifies how data is acquired. The design of the test incorporates certain assumptions about intelligence that are closely tied to categorization and subsequent applicability. The ideology of dataism posits that data is inherently available for extraction and utilization, disregarding the selection, assumption, acquisition, and classification processes that form the foundation of data sets. By inducing a "state of objective delusion" or a "veil of ignorance" (drawing on John Rawls' concept, which suggests that individuals can hypothetically detach themselves from their social and epistemic experiences to enter a purely rational realm where ideal societal contracts can be formulated), the project simplifies the complexities of perception.

WORK DESCRIPTION (ACCESSIBLE VERSION)

This research project explores how data extraction, big data practices, and gamification are interconnected. Its goal is to understand the relationships between game design, flow states, and the actions that generate data.

//GAME
At the core of the project is a computer game that allows players to voluntarily engage with an interactive environment. The game's design, with its unique features and rules, guides players toward achieving flow states. Players are encouraged to immerse themselves in the game by adjusting and controlling various elements, like sounds. Eventually, the game prompts them to categorize what they see based on instructions. These categorizations are recorded and used to train machine learning systems.

//CONTROL
The project assumes that control in modern societies is primarily achieved through voluntary actions. Computer games, as a medium, reflect this idea by presenting players with opportunities to exercise control within virtual worlds. However, the project acknowledges that leaving these virtual worlds is often not a voluntary choice. By participating in the game, players consent to having their decisions recorded, emphasizing individual responsibility while disregarding any manipulative elements that might have influenced their choices.

//CATEGORIZATION
Machine learning contexts define intelligence in terms of practical problem-solving and categorization. This definition, often based on established tests, simplifies intelligence and overlooks alternative perspectives. During flow states in the game, players take on the role of machine learners themselves, working to categorize a complex visual and auditory environment into data.

//DATA
Contrary to the notion that data exists naturally, this project demonstrates how data is acquired through a designed environment. It challenges the idea that data is simply available for extraction and utilization, highlighting the complex processes involved in selecting, assuming, acquiring, and classifying data sets. By creating a state of "objective delusion" or a "veil of

ignorance," the project simplifies perception, allowing players to focus on specific elements for observation and categorization.

WORK DESCRIPTION (SIMPLE NO BS VERSION)

This research project explores how we collect and use data in computer games. The goal is to understand how game design, the state of being fully engaged (called "flow"), and the actions players take all contribute to producing data.

//GAME
The project involves a computer game where players can choose to participate. The game has its own rules and features that help players get into a focused and immersed state. For example, players can control sounds in the game to make it more enjoyable. After a while, the game asks players to identify certain things they see. These answers are then stored and used to teach computer systems how to recognize things.

//CONTROL
In our society, we often gain a sense of control through the choices we make. Computer games let players feel in control by giving them the power to make decisions within the game world. However, once players start the game, they can't easily leave. By playing, they agree to have their choices recorded, which puts more emphasis on their own responsibility, even if the game might have influenced their decisions.

//CATEGORIZATION
In this project, we look at how we define intelligence in computer systems. Intelligence is often seen as the ability to solve problems and put things into categories. But this view can limit other ways of thinking about intelligence. When players are fully engaged in the game, they become like the computer systems, categorizing things they see in the game.

//DATA
We usually think of data as something that's just there, ready for us to use. But in reality, data is collected through specific processes. This project creates an environment that shows how we collect data. We carefully choose what we want to see and ignore other things to

make it easier to understand. It's like focusing on certain details while ignoring the rest.

WORK DESCRIPTION (GERMAN SIMPLE NO BS VERSION)

Dieses Forschungsprojekt untersucht, wie Datenextraktion, Big Data-Praktiken und Gamification miteinander verbunden sind und zielt darauf ab, die Beziehungen zwischen Spiel-/Verhaltensdesign, Flow-Zuständen und den scheinbar freiwilligen Handlungen, die zur Datenproduktion führen, zu verdeutlichen.

//SPIEL
m Kern des Projekts steht ein Computerspiel, das den Spieler:innen ermöglicht, mit einer virtuellen Umgebung zu interagieren. Das Design des Spiels mit seinen spezifischen Eigenschaften und Regeln führt die Spieler:innen zu Flow-Zuständen. Eine Vertiefung ins Spiel soll durch die Modulation verschiedener Klänge und des Anschlagtempos ermöglicht werden. Schließlich werden sie aufgefordert, basierend auf Anweisungen Gesehenes zu kategorisieren. Diese Kategorisierungen werden aufgezeichnet und als Trainingsdaten für Machine Learners verwendet.

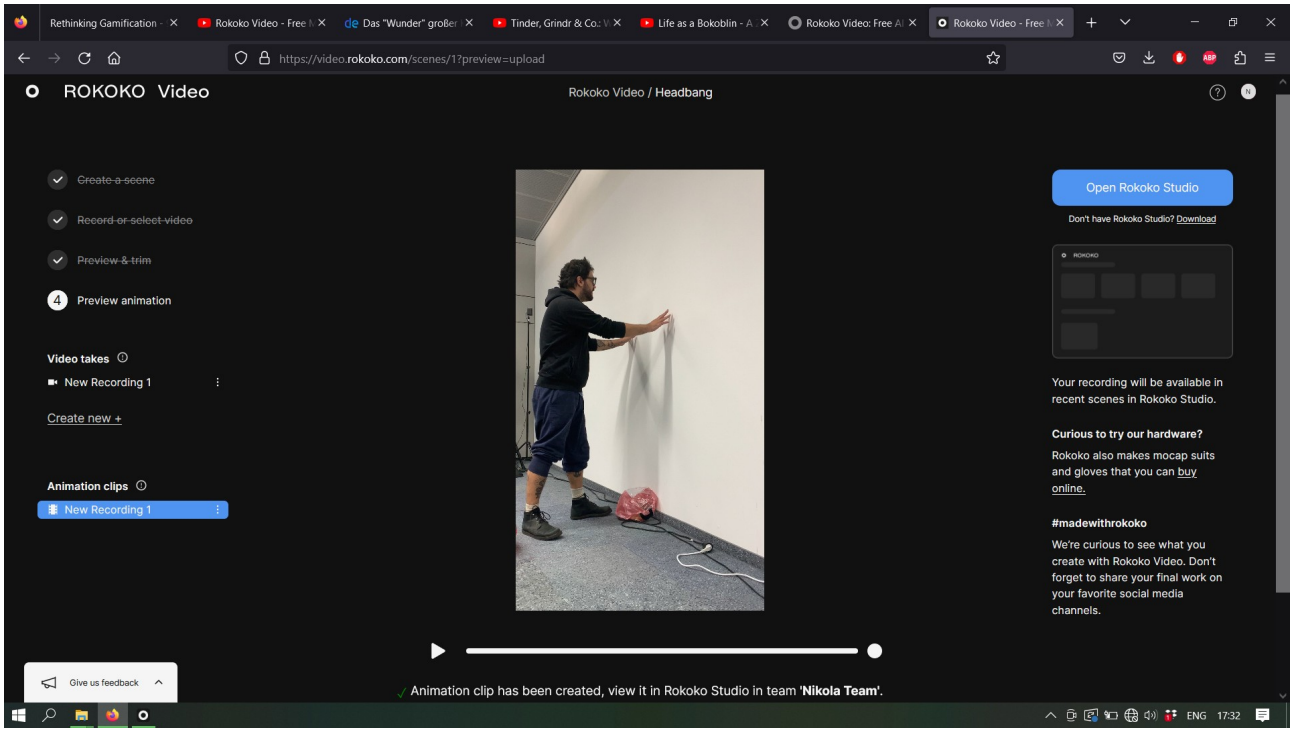
//KONTROLLE
Das Projekt geht davon aus, dass Kontrolle in modernen Gesellschaften hauptsächlich durch freiwillige Handlungen ermöglicht wird. Computerspiele repräsentieren und reproduzieren neoliberale Ideologeme, indem sie den Spieler:innen Möglichkeiten bieten, innerhalb virtueller Welten Kontrolle auszuüben. Allerdings ist das Verlassen dieser virtuellen Welten oft keine freiwillige Entscheidung. Durch die Teilnahme am Spiel und die Auseinandersetzung damit willigen die Spieler:innen ein, dass ihre Entscheidungen aufgezeichnet werden. Dabei wird der Fokus auf individuelle Verantwortung gelegt, während mögliche manipulative Elemente, die die Entscheidungen beeinflusst haben könnten, außer Acht gelassen werden.

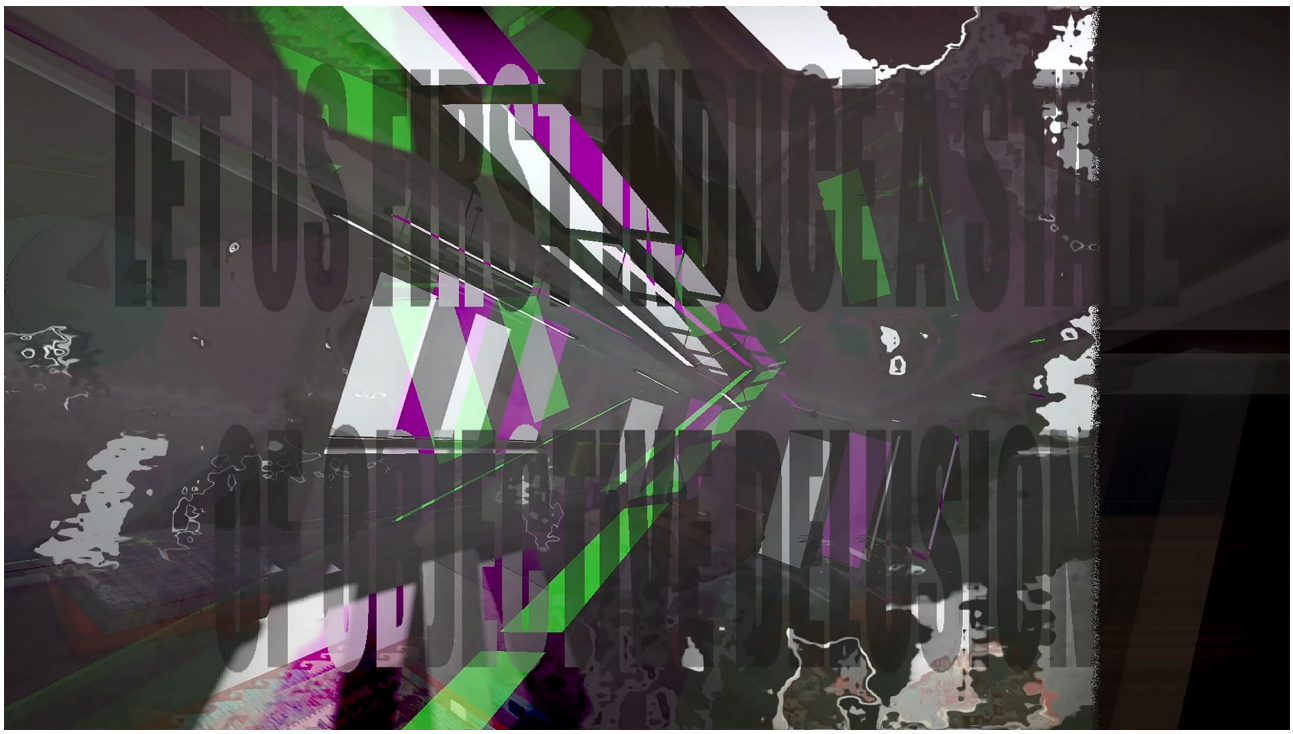
//KATEGORISIERUNG
Maschinelles Lernen definiert Intelligenz oft als die Fähigkeit, Probleme praktisch zu lösen und Dinge in Kategorien einzuteilen. Diese Definition kann jedoch alternative Sichtweisen auf Intelligenz einschränken. Während des Flow-Zustands im Spiel übernehmen die Spieler

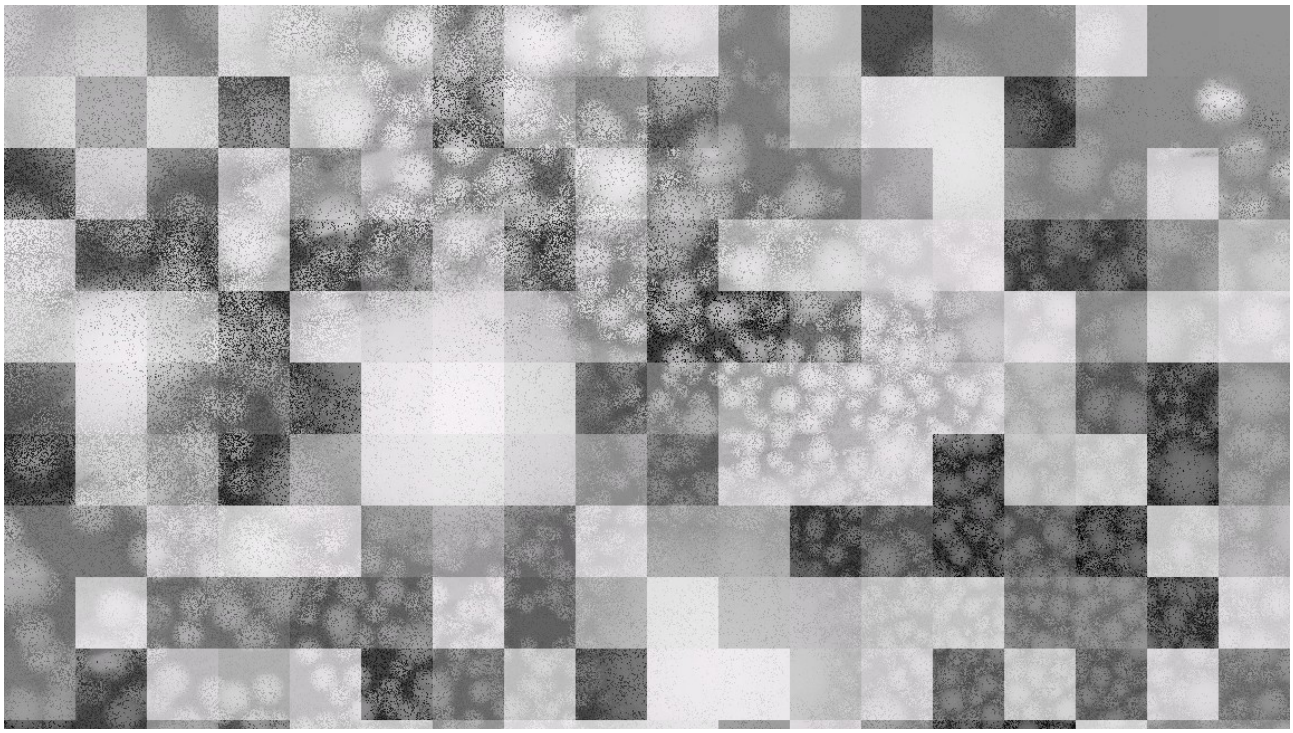
selbst die Rolle des maschinellen Lernens und kategorisieren Dinge, die sie im Spiel sehen.

//DATEN
Wir denken oft, dass Daten einfach vorhanden sind und darauf warten, von uns genutzt zu werden. In Wirklichkeit werden Daten jedoch durch spezifische Prozesse gesammelt. Dieses Projekt schafft eine Umgebung, die zeigt, wie Daten erhoben werden. Wir wählen sorgfältig aus, was wir sehen möchten, und ignorieren andere Dinge, um es einfacher zu machen, zu verstehen.

ALL TEXTS EDITED BY A MACHINE


Rethinking Gamification
Rokoko Video - Free
Das "Wunder" großer
Tinder, Grindr & Co.:
Life as a Bokoblin - A
Rokoko Video: Free A
Rokoko Video - Free
https://video.rokoko.com/scenes/1?preview=upload
ROKOKO Video
Rokoko Video / Headbang
Create a scene
Record or select video
Preview & trim
4
Preview animation
Video takes
New Recording 1
Create new +
Animation clips
New Recording 1
Open Rokoko Studio
Don't have Rokoko Studio? Download
Your recording will be available in recent scenes in Rokoko Studio.
Curious to try our hardware?
Rokoko also makes mocap suits and gloves that you can buy online.
#madewithrokoko
We're curious to see what you create with Rokoko Video. Don't forget to share your final work on your favorite social media channels.
Give us feedback
Animation clip has been created, view it in Rokoko Studio in team 'Nikola Team'.
ENG
17:32





```
shader_type spatial;
render_mode unshaded, depth_test_disabled, cull_disabled;

uniform sampler2D displacement_noise;
uniform sampler2D screen_texture : hint_screen_texture;
uniform sampler2D instruction;
uniform float impactIntensity;
uniform float abberation_value;
uniform float stretch_multiplier;
uniform float blackout;
uniform float offset : hint_range(0.0, 1.0, 0.01);
uniform float invert : hint_range(0., 1.0, 1.0);
uniform float shake_intensity;
uniform float text_strength;
uniform float shader_trans;
uniform int pixel_size;
uniform float pixelation;
uniform float vignette_value : hint_range(0.0,1.0);
uniform sampler2D vignette_noise;

vec3 screen(vec3 base, vec3 blend){
	return 1.0 - (1.0 - base) * (1.0 - blend);
}


vec2 scale(vec2 uv, float x, float y)
{
	mat2 scale = mat2(vec2(x, 0.0), vec2(0.0, y));

	uv -= 0.5;
	uv = uv * scale;
	uv += 0.5;
	return uv;
}

float rand(vec2 co){
    return fract(sin(dot(co, vec2(12.9898, 78.233))) * 43758.5453);
}

void vertex(){
	// needs to be 1.0
	VERTEX *= 1.0;
	// needs to be 1.0
	POSITION = vec4(VERTEX, 1.0);
}
void fragment() {
	// Place fragment code here.
	vec2 texCoords = SCREEN_UV;
	float n = texture(displacement_noise, (texCoords*.5)-.75).r;
	texCoords.x += n * (impactIntensity * sin(20.*TIME));
	vec3 color = texture(screen_texture, texCoords).rgb;

	vec4 abberationColor = texture(screen_texture, SCREEN_UV);
	abberationColor.r = texture(screen_texture,SCREEN_UV+abberation_value).r;
	abberationColor.g =
texture(screen_texture,vec2(SCREEN_UV.x+abberation_value, UV.y-
abberation_value)).g;
	abberationColor.b = texture(screen_texture, SCREEN_UV-abberation_value).b;

	vec2 uv = SCREEN_UV;
	//random number, multily to make it very small.
```

```
	float r = rand(vec2(uv.x, uv.y)) * .008;
	//threshold: if UV.x is below offset, throw 1;
	//add some random offset to UV.x
	float threshold = step(offset, uv.x+r);
	//for threshold inversion
	threshold = abs(invert-threshold);
	vec2 sampleFrom = vec2(offset, uv.y);
	vec4 original = texture(screen_texture, SCREEN_UV);
	vec4 deformed = texture(screen_texture, sampleFrom);
	//use the threshold as mask again (transparent or not)
	vec4 colGlitch= deformed;
	colGlitch = mix(original, colGlitch, threshold);

	vec2 shakeUV = SCREEN_UV;
	shakeUV.x += sin(5.*TIME)*2.;
	vec3 shakeTex = texture(screen_texture,shakeUV).rgb;

	vec2 instrUV = SCREEN_UV;
	instrUV = scale(instrUV, 3., 3.);
	vec4 instrTex = texture(instruction, instrUV);

	vec3 newScreen = (abberationColor.rgb*abberation_value*.5) + color +
shakeTex*shake_intensity;
	vec3 withGlitch = mix(newScreen, colGlitch.rgb, threshold);
	vec3 fin = mix(newScreen, withGlitch, stretch_multiplier);

	//pixelation
	float x = float (int(FRAGCOORD.x) % pixel_size);
	float y = float (int(FRAGCOORD.y) % pixel_size);

	x = FRAGCOORD.x + floor(float(pixel_size) / 2.0) - x;
	y = FRAGCOORD.y + floor(float(pixel_size) / 2.0) - y;

	vec3 pixelated = texture(screen_texture, vec2(x,y) / VIEWPORT_SIZE).xyz;
	vec3 finfin = mix(original.rgb, fin, shader_trans);
	ALBEDO = mix(finfin, pixelated, pixelation);


	float vn = texture(vignette_noise, SCREEN_UV+TIME*.1).r;
	vec2 cUV = SCREEN_UV-.5;
	float d = length(cUV);
	d = smoothstep(0.0,vignette_value, d*vn);

	ALBEDO = mix(ALBEDO, vec3(0.), d-.3);
	ALPHA = 1.-instrTex.a*text_strength + d;
}
```