

Weapon of Math Destruction

ChessMirror WMD – Individuele gedragsprofieling en subtile beïnvloeding in een schaakomgeving

1. Inleiding

Voor deze opdracht ontwikkelde ik ChessMirror WMD, een volledig lokaal draaiende webapplicatie die gebruikersgedrag op individueel niveau verzamelt, analyseert en inzet om de gebruiker op subtile wijze te beïnvloeden. Het project fungeert als een concrete demonstratie van een *Weapon of Math Destruction (WMD)*: een systeem dat grootschalige dataverzameling, abstracte modellering en geautomatiseerde beslissingen combineert om invloed uit te oefenen op individuen, zonder volledige transparantie.

Hoewel de applicatie zich aan de oppervlakte presenteert als een schaakspel, is het eigenlijke doel niet het verbeteren van schaakvaardigheden. Het systeem is ontworpen om aan te tonen hoe ogenschijnlijk onschuldige digitale interacties kunnen worden omgezet in gedetailleerde gedragsprofielen en hoe deze profielen gebruikt kunnen worden om het gedrag van gebruikers te sturen.

2. Overzicht van het systeem

Het project bestaat uit vier duidelijk afgebakende maar samenwerkende onderdelen:

1. User-facing applicatie
Een interactieve schaakinterface waarin gebruikers spelen tegen een adaptieve bot, hints ontvangen en feedback krijgen tijdens het spel.
2. Backend & databank
Een Express API gekoppeld aan een PostgreSQL-database die alle gebruikersinteracties valideert, verwerkt en persistent opslaat.
3. Admin dashboard
Een beheerdersinterface waarin verzamelde data wordt gevisualiseerd, geanalyseerd en actief kan worden ingezet om gebruikersgedrag te beïnvloeden.
4. Reflectief en analytisch luik
Een schriftelijke analyse waarin de impact, beperkingen en ethische risico's van het systeem worden besproken.

Het volledige systeem draait volledig lokaal binnen een Docker-omgeving en kan worden opgestart via één .env-bestand en het commando docker compose up --build.

3. Dataverzameling op individueel niveau

Elke gebruiker krijgt bij de start van een sessie een unieke UID, die fungeert als primaire sleutel voor alle dataverzameling. Alle interacties worden strikt per UID gelogd en geanalyseerd.

De verzamelde data omvat onder andere:

- Elke schaakzet (SAN- en UCI-notatie)
- Denktijd per zet (in milliseconden)
- Aantal hovers per zet
- Hoverpatronen over tijd
- Klikgedrag (links- en rechtsklik)
- Navigatie tussen verschillende views
- Gebruik van hints
- Focus- en blur-events
- Zetkwaliteit (bijvoorbeeld *blunder, good*)
- Botinteracties
- Basis device metadata (timezone, taal, schermgrootte en user-agent, opgeschoond)

Elke interactie, hoe klein ook, wordt beschouwd als waardevolle data en wordt gevalideerd voordat ze in de databank wordt opgeslagen.

4. Backend & dataverwerking

De backend vervult een centrale rol binnen het systeem en is verantwoordelijk voor:

- Validatie van inkomende events
- Opschoning en normalisatie van data
- Persistente opslag in PostgreSQL
- Aggregatie van ruwe events tot betekenisvolle statistieken en signalen

Voorbeelden van berekende signalen zijn:

- Gemiddelde denktijd per zet
- Blunderpercentage

- Aantal hovers per zet
- Consistentie over meerdere zetten
- Evolutie van gedrag binnen één sessie

Deze afgeleide statistieken vormen de basis voor verdere gedragsprofiling.

5. Gedragsprofiling

Op basis van de geaggregeerde signalen wordt elke gebruiker dynamisch ingedeeld in een gedragssegment. Deze segmenten zijn niet statisch, maar kunnen veranderen naarmate er meer data beschikbaar komt.

Voorbeelden van gebruikte segmenten:

- Impulsive – zeer korte denktijd gecombineerd met een hoog foutpercentage
- Hesitant – lange denktijd en veel exploratie via hovers
- Explorer – uitzonderlijk veel hovers per zet
- Reflective – lange denktijd met een lage foutgraad
- Balanced – geen uitgesproken extreme patronen
- Warming up – onvoldoende data om betrouwbare conclusies te trekken

Deze labels reduceren complex menselijk gedrag tot begrijpbare categorieën, maar verbergen tegelijk de nuance en context van de onderliggende data.

6. Visualisatie & admin dashboard

Het admin dashboard fungeert als het controlecentrum van de Weapon of Math Destruction. Vanuit deze interface kan een beheerder:

- Gebruikers (UID's) selecteren
- Gedetailleerde statistieken raadplegen
- Gedragstrends visualiseren (zoals denktijd over opeenvolgende zetten)
- Segmentlabels interpreteren
- Actieve beslissingen nemen over beïnvloedingsmechanismen

De beheerder beschikt hiermee over aanzienlijke macht over de gebruikerservaring, terwijl de gebruiker zelf geen volledige inzage heeft in de logica achter deze beslissingen.

7. Subtiele beïnvloeding (Influence)

De kern van het WMD-concept ligt in de manier waarop het systeem gebruikersgedrag stuurt zonder expliciete dwang of waarschuwingen.

Concrete beïnvloedingsmechanismen zijn onder andere:

- Willekeurig getoonde bevestigingsprompts (“Are you sure?”) bij snelle of impulsieve spelers
- Gedragsnudges zoals “*Take a second before committing*”
- Adaptief botgedrag dat inspeelt op fouten of tempo van de speler
- Variabele hints afhankelijk van het gebruikersprofiel

Deze interventies voelen aan als hulp of feedback, maar zijn strategisch getimed op basis van verzamelde gedragsdata.

8. Waarom dit systeem een Weapon of Math Destruction is

ChessMirror WMD voldoet aan de kernkenmerken van een Weapon of Math Destruction:

- Grootchalige dataverzameling via micro-interacties
- Abstracte modellering die gedrag herleidt tot labels en scores
- Geautomatiseerde beslissingen met directe impact op de gebruikerservaring
- Gebrek aan transparantie over hoe conclusies exact tot stand komen

De gebruiker ondergaat de gevolgen van het systeem, maar ziet niet hoe deze beslissingen worden genomen. Dit creëert een duidelijke machtsasymmetrie tussen systeem en gebruiker.

9. Beperkingen & tekortkomingen

Het project toont ook duidelijk de kwetsbaarheid van dergelijke systemen aan:

- Profielen zijn gebaseerd op beperkte context (enkel schaakgedrag)
- Korte sessies kunnen leiden tot foutieve conclusies
- Gedrag wordt soms verkeerd geïnterpreteerd
- Correlatie wordt niet altijd onderscheiden van causaliteit
- Externe factoren zoals vermoeidheid of stress worden niet gemeten

Deze beperkingen illustreren hoe snel een WMD onbetrouwbaar of zelfs schadelijk kan worden.

10. Reflectie & inzichten

Tijdens de ontwikkeling werd duidelijk hoe eenvoudig het is om:

- Mensen te reduceren tot statistische profielen
- Gedrag te sturen zonder expliciete toestemming
- Beslissingen te nemen die objectief lijken, maar dat niet zijn

Zelfs binnen een relatief klein en onschuldig systeem ontstaat een vorm van onzichtbare controle. Dit benadrukt het belang van ethische reflectie bij data-gedreven toepassingen.

11. Conclusie

ChessMirror WMD demonstreert op een technisch onderbouwde en inzichtelijke manier hoe data → profilering → invloed samen een Weapon of Math Destruction vormen. Het project voldoet aan alle technische, functionele en conceptuele vereisten van de opdracht en toont aan hoe dergelijke systemen in de praktijk problematisch kunnen worden wanneer transparantie en context ontbreken.

12. Technische context (samenvatting)

- Frontend: React
- Backend: Node.js & Express
- Database: PostgreSQL
- ORM: Prisma
- Environment: Docker & Docker Compose
- Scope: Volledig lokaal, geen externe API's

Bronnenlijst

Cursusmateriaal (Canvas – Erasmus Hogeschool Brussel)

- Erasmus Hogeschool Brussel. *Weapon of Mass Destruction – cursusmateriaal.*
https://canvas.ehb.be/courses/44105/files/3603615?module_item_id=871885
Gebruikt als theoretisch en conceptueel kader voor het WMD-concept, machtsasymmetrie, profiling en invloed.
- Erasmus Hogeschool Brussel. *Data & profiling slides.*
<https://canvas.ehb.be/courses/44105/files?preview=3501657>
Gebruikt voor inzicht in dataverzameling, aggregatie, gedragsanalyse en interpretatie van signalen.
- Erasmus Hogeschool Brussel. *Docker & Docker Compose – lesmateriaal.*
<https://canvas.ehb.be/courses/44105/files?preview=3626131>
Gebruikt voor het opzetten van een lokaal draaiende multi-container omgeving met database, backend en frontend.
- Erasmus Hogeschool Brussel. *Git & versiebeheer – slides.*
<https://canvas.ehb.be/courses/44105/files?preview=3501652>
Gebruikt voor correct versiebeheer, gestructureerde commits en een overzichtelijke projectgeschiedenis.
- Erasmus Hogeschool Brussel. *Software & services – cursusmateriaal.*
<https://canvas.ehb.be/courses/44105/files?preview=3501662>
Gebruikt voor inzicht in architectuurkeuzes, services, en het scheiden van verantwoordelijkheden binnen een applicatie.

Technologie & officiële documentatie

- Docker Documentation.
<https://docs.docker.com>
Gebruikt voor het configureren en draaien van containers in een volledig lokale omgeving.
- Docker Compose Documentation.
<https://docs.docker.com/compose>
Gebruikt voor het definiëren en starten van meerdere services via één compose-bestand.
- PostgreSQL Documentation.
<https://www.postgresql.org/docs>
Gebruikt voor relationele databankopzet, schema's en persistente opslag van gebruikersdata.

- Prisma ORM Documentation.
<https://www.prisma.io/docs>
Gebruikt voor datamodellering, validatie en veilige interactie tussen backend en database.
- Express.js Documentation.
<https://expressjs.com>
Gebruikt voor het bouwen van een API die events valideert, verwerkt en opslaat.
- React Documentation.
<https://react.dev>
Gebruikt voor het bouwen van de user-facing interface en interactieve componenten.
- Chart.js Documentation.
<https://www.chartjs.org/docs>
Gebruikt voor het visualiseren van gebruikersdata en gedragsstatistieken in het admin dashboard.
- react-chessboard (npm)
<https://www.npmjs.com/package/react-chessboard>
Gebruikt voor de interactieve chessboard UI (drag & drop, onPieceDrop events) in de user-facing applicatie.
- nanoid (npm)
<https://www.npmjs.com/package/nanoid>
Gebruikt voor het genereren van unieke identifiers (UID) per gebruiker/sessie zonder login-systeem.

Web & interactie

- MDN Web Docs – Web APIs & Events.
<https://developer.mozilla.org>
Gebruikt voor correcte implementatie van browser-events zoals clicks, hovers, focus en blur.

Ethiek & reflectie

- ACM Code of Ethics and Professional Conduct.
<https://www.acm.org/code-of-ethics>
Gebruikt als ethisch referentiekader voor verantwoordelijk software- en datagebruik.

- GDPR – General Data Protection Regulation (EU).
<https://gdpr.eu>
Gebruikt om de risico's en beperkingen rond privacy en dataverzameling kritisch te reflecteren.

Chess.js Documentation

- <https://github.com/jhlywa/chess.js>
Gebruikt als hulpmiddel voor spelvalidatie, niet voor analyse of besluitvorming.

AI-ondersteuning

- Projectarchitectuur & Docker
<https://chatgpt.com/share/695a5543-b04c-800d-98bd-3a9ac97ed1fe>
Gebruikt voor het bepalen van de algemene projectarchitectuur en het lokaal opzetten van het systeem met Docker Compose.
- UID & individuele dataverzameling
<https://chatgpt.com/share/695a5639-7478-800d-b33a-660e8c190d53>
Gebruikt voor het ontwerpen van een UID-systeem en het loggen van gebruikersinteracties op individueel niveau.
- Backend & databaseverwerking
<https://chatgpt.com/share/695a566a-ba58-800d-84cc-e458e5e9aeb7>
Gebruikt voor de opbouw van de backend, validatie van events en persistente opslag in PostgreSQL.
- Gedragsprofieling & beïnvloeding
<https://chatgpt.com/share/695a5685-f13c-800d-b062-b81170499b7c>
Gebruikt voor het definiëren van gedragssegmenten en het toepassen van subtiele beïnvloeding in de gebruikersinterface.
- Admin dashboard & visualisatie
<https://chatgpt.com/share/695a56b4-6eb0-800d-9ce4-3d63df90de12>
Gebruikt voor het ontwerpen van het admin dashboard en het visualiseren van gebruikersdata.