# A Machine Learning Approach for Data Protection in Virtual Reality Therapy Applications

Mircea Maria-Mădălina

WeADL 2021 Workshop

The workshop is organized under the umbrella of WeaMyL, project funded by the EEA and Norway Grants under the RO-NO-2019-0133.
Contract: No 26/2020

# Introduction

- Internet
  → rapid development
  → social media
  → growing interest to spend time online
  → personal information – sellable good
- Trading personal data – uses:
  - Targeted ads
  - Increasing a company's revenue
  - …
  - Changing political views
- Health information → protected asset → can leak to third parties
- Health applications sometimes fail to keep the data private
- 2018 → GDPR

# Introduction

- User authentication = determine a user's identity
  - Knowledge-based → most utilized (PINs, passwords, etc.)
  - Token-based
  - Biometric-based
- Virtual Reality (VR)
  → great potential in therapy (e.g. physical and emotional trauma, disorders)
  → usually used for gaming → lower need of data security
  → secure identification required: medical applications, virtual presence (e.g. conferences), access to private resources, etc.

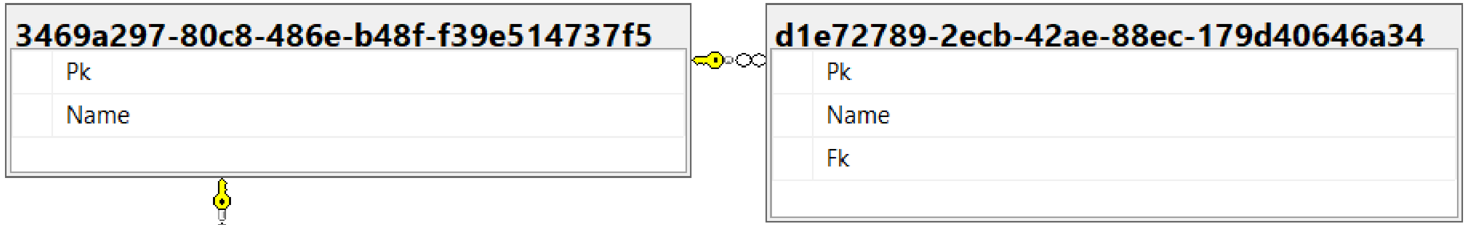
# Introduction

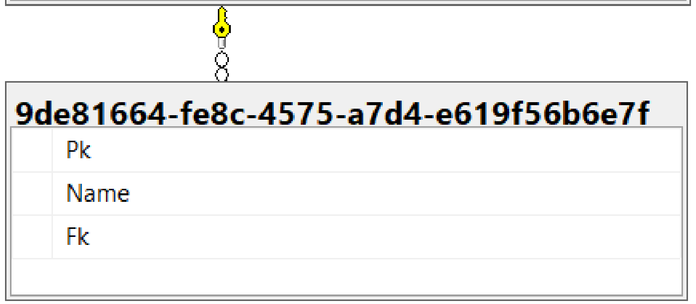
- Utilizing user data → ethical if the user gives consent → issues occur when the user is unaware of the data that is being used
- Artificial Emotional Intelligence (e.g. Amazon's Alexa)
  - Infer emotion from voice, behaviours, etc.
  - Humane purposes → improve mood (e.g. with jokes, music)
  - Negative purposes → control purchasing habits, political views, etc.
- Can we use health information for research?
  - Yes, if the data is anonymised (≠ pseudonymisation !)
  - Yes, if the user gives consent (on-going process, not one time event !)
  - Provide means to remove personal data from storage !

# Background. Related work

- 2016, Yu et al
  - Compared 3 VR authentication methods (PIN, password, 3D password)
  - 2 experiments, 15 participants
    1. Select password → Insert password 5 times → record error rate → PIN is easiest
    2. Shoulder surfing → film users authenticating → show to other users → record success rate → 3D password is safest
- 2017, Lee et al
  - Lip reading for authentication
  - LSTM architecture → analyse a sequence of images of the user's lips

# Methodology




- Database security
  - Pseudonyms for tables and table fields
  - Information encrypted
  - Encryption → Fernet algorithm, password computed at runtime using the user’s information
  - GDPR → each user has their own ID and only has access to their own information
  - Minimal data → records of the user’s dance moves
  - Data is stored in encrypted files on the server
  - Path to data is computed at runtime

# Methodology

- Secure authentication – 3 versions – advantages and disadvantages
  1. Username – voice recording; Password – dynamic movement
     - Failed – voice analysis did not work
  2. Identify only by dynamic movement
     - Accuracy > 99%, did not perform well in practice
  3. Username – text; Password – dynamic movement
     - Accuracy > 99%, performs inconsistently in practice

# Methodology

- Dynamic movement records:
  1. Positions and rotations
  2. Only positions


```
{
  "X": 0.4135810434818268,
  "Y": 1.6037466526031494,
  "Z": 0.16181637346744537
},
{
  "X": 0.4072014391422272,
  "Y": 1.6022557020187378,
  "Z": 0.1623678207397461
},
{
  "X": 0.40542149543762207,
  "Y": 1.601191520690918,
  "Z": 0.17611461877822876
}
],
"headsetRotations": [
{
  "X": -0.019815094769001007,
  "Y": 0.14819036424160004,
  "Z": -0.03293139860033989,
  "W": -0.9882118701934814
},
{
  "X": -0.02947123534977436,
  "Y": 0.15355893969535828,
  "Z": -0.026276519522070885,
  "W": -0.9873502850532532
},
```



```
{
  "X": 23.111183166503906,
  "Y": 2.154688835144043,
  "Z": 21.23886291333008
},
{
  "X": 23.11316294384766,
  "Y": 2.1573691368103027,
  "Z": 21.235750198364258
}
]
},
"leftControllerPositions": {
  "records": [
    {
      "X": -0.1711234301328659,
      "Y": 1.2772496938705444,
      "Z": -0.006181010976433754
    },
    {
      "X": -0.20724187791347504,
      "Y": 0.7210021615028381,
      "Z": 0.12436452507972717
    },
    {
      "X": -0.15876558423042297,
      "Y": 0.5392007827758789,
      "Z": 0.0904918685555458
    },
```

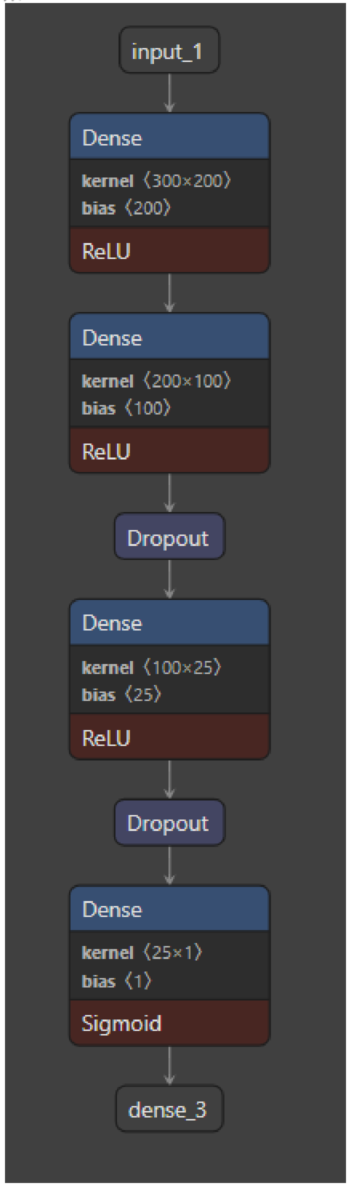
# Methodology

- Dynamic movement records
  - User records the same move multiple times (4 times)
  - Create augmented data → move the positions on X and Z
  - Normalize data → between -100 and 100 on X and Z, between 0 and 3 on Y
  - Flatten array
  - Pad to the right to obtain an array of length 300


```
{
  "records": [
    0.6150910015106201,
    0.7134119669596354,
    0.5030754550075531,
    0.6150895900726319,
    0.71261994043986,
    0.5030754479503632,
    0.615071174621582,
    0.715578556060791,
    0.5030753558731079,
    0.6150764770507813,
    0.7147644360860189,
    0.5030753823852538,
    0.615080644607544,
    0.7128190994262695,
    0.5030754032230377,
    0.6150563545227051,
    0.7139293352762858,
    0.5030752817726135,
    0.6150056667327881,
    0.7122228940327963,
    0.5030750283336639,
    0.6150550575256348,
    0.711285670598348,
```

# Methodology

- Artificial Neural Network (ANN)
  - Regression model → one neuron on the output layer (1 = true, 0 = false)
  - Input → 200-dim Dense → 100-dim Dense → Dropout 0.5 → 25-dim Dense → Dropout 0.25 → Output
  - ReLU activation on hidden layers
  - Sigmoid activation on output (for [0, 1] interval)
  - Compiled with binary_crossentropy, Adam optimizer, learning rate 0.00001
  - 80% of data for training, 20% for testing
  - Trained for 150 epochs, with 25% of data used for validation

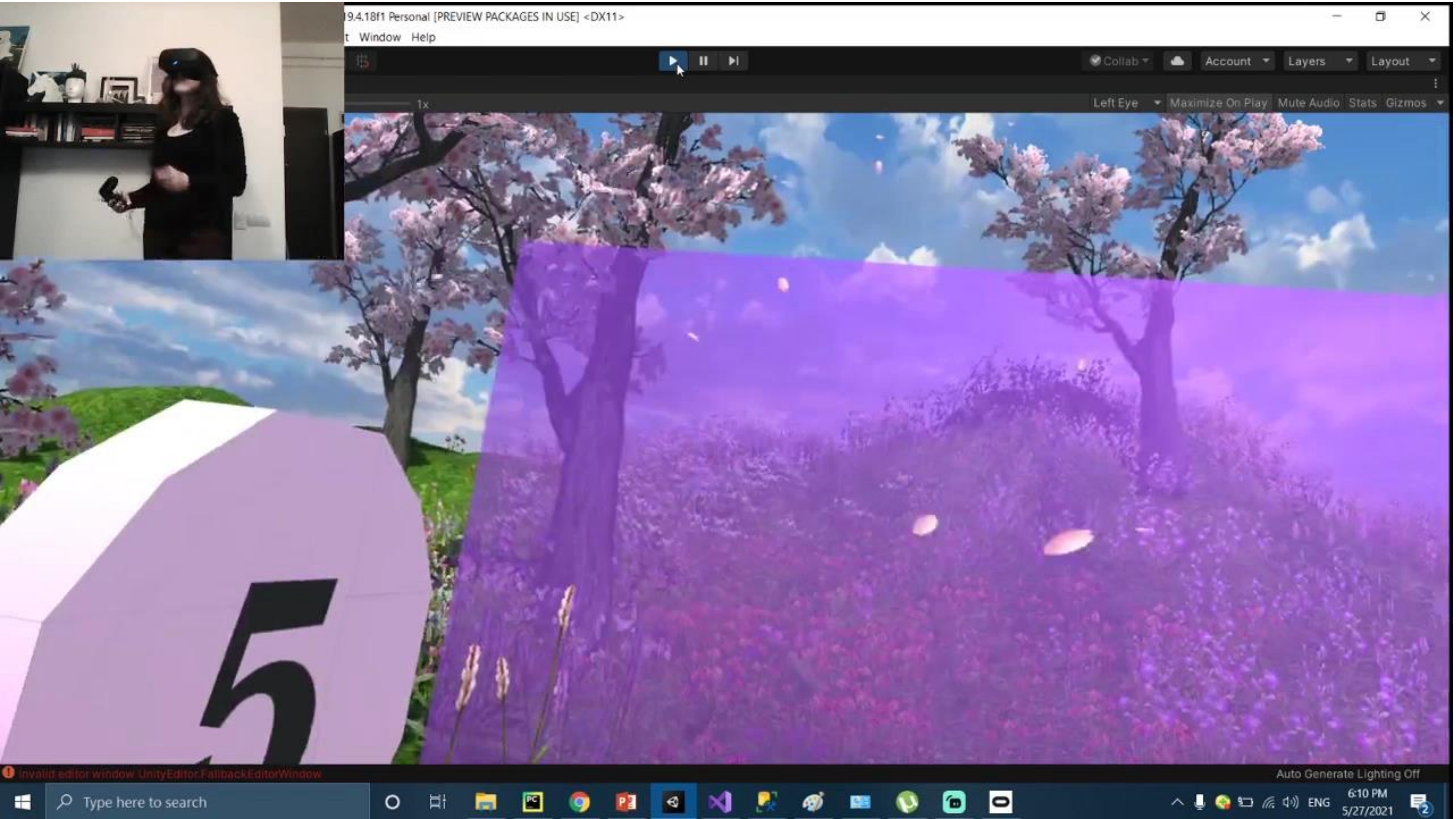


# Methodology

- Full flow:
  - Server receives registration data
  - Check to see if the username exists, return error if it does, create a new user in the database if it does not
  - Save the registration data into files
  - Create augmented files; create flattened files
  - Parse all of the data on the server → mark the new user's data with 1 and all of the other users' data with 0
  - Shuffle the labeled data
  - Select a number of 0-labeled records that is equal to the number of 1-labeled records (to obtain a balanced dataset)
  - Split data into training and testing
  - Train and test the new user's model
  - Retrain all of the other users' models using the new data

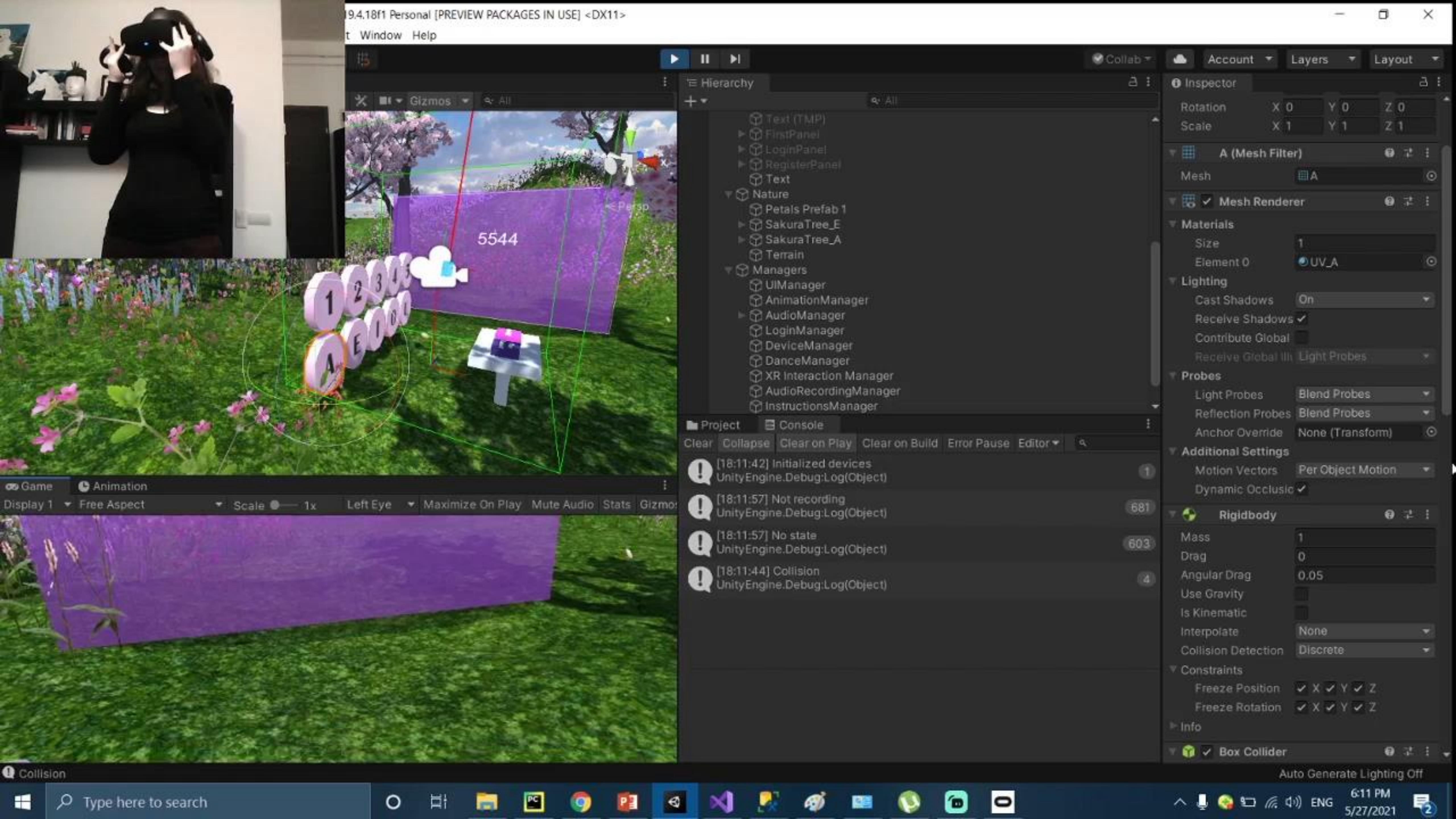
# Experimental Evaluation

- Server → Python (Flask)
  - Login()
    - Receives username, dance record JSON
    - Returns
      {"user_id": user_id, "code": 200, "error": ""} on SUCCESS
      {"user_id": "", "code": 404, "error": "User not recognized."} on FAILURE
  - Register()
    - Receives username, array of JSON dance records
    - Returns
      {"user_id": str(user_id), "code": 200, "error": username} on SUCCESS
      {"user_id": "", "code": 404, "error": "Could not create user!"} on FAILURE


9.4.18f1 Personal [PREVIEW PACKAGES IN USE] <DX11>
t Window Help
Collab
Account
Layers
Layout
1x
Left Eye
Maximize On Play
Mute Audio
Stats
Gizmos
5
Invalid editor window UnityEditor.FallbackEditorWindow
Auto Generate Lighting Off
Type here to search
ENG
6:10 PM
5/27/2021

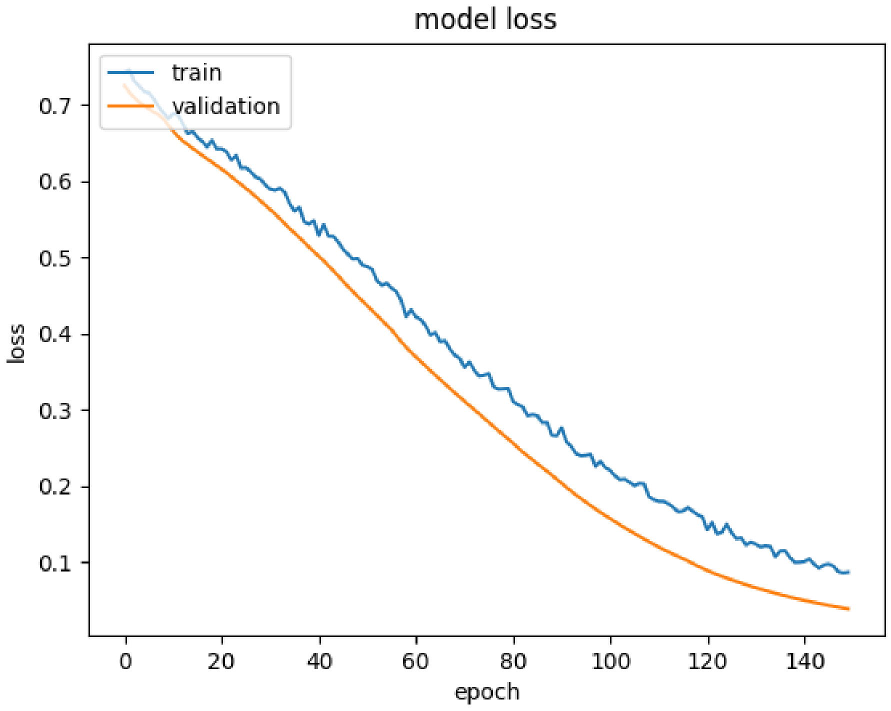
# Experimental Evaluation

- The flow is performed using States (e.g. RecordDanceState)
- To type the username, the user touches the keys on the virtual keyboard with the VR controllers
- To record the dance move, the user needs to hold the Grip button on the right controller
- When the user releases the grip button, the flow moves on to the next State in the StateSequence
- If the StateSequence is a RegisterStateSequence, then the user has to repeat the movement 4 times
- After repeating it the 4$^{th}$ time, the information is sent to the server
- If the StateSequence is a LoginStateSequence, the information is sent to the server after recording one dynamic movement


19.4.18f1 Personal [PREVIEW PACKAGES IN USE] <DX11>
t Window Help
Collab
Account
Layers
Layout
Gizmos
All
Hierarchy
Inspector
Text (TMP)
FirstPanel
LoginPanel
RegisterPanel
Text
Nature
Petals Prefab 1
SakuraTree_E
SakuraTree_A
Terrain
Managers
UIManager
AnimationManager
AudioManager
LoginManager
DeviceManager
DanceManager
XR Interaction Manager
AudioRecordingManager
InstructionsManager
5544
Persp
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
A (Mesh Filter)
Mesh A
Mesh Renderer
Materials
Size 1
Element 0 UV_A
Lighting
Cast Shadows On
Receive Shadows
Contribute Global
Receive Global Illi Light Probes
Probes
Light Probes Blend Probes
Reflection Probes Blend Probes
Anchor Override None (Transform)
Additional Settings
Motion Vectors Per Object Motion
Dynamic Occlusic
Rigidbody
Mass 1
Drag 0
Angular Drag 0.05
Use Gravity
Is Kinematic
Interpolate None
Collision Detection Discrete
Constraints
Freeze Position X Y Z
Freeze Rotation X Y Z
Info
Box Collider
Project
Console
Clear Collapse Clear on Play Clear on Build Error Pause Editor
[18:11:42] Initialized devices
UnityEngine.Debug:Log(Object)
1
[18:11:57] Not recording
UnityEngine.Debug:Log(Object)
681
[18:11:57] No state
UnityEngine.Debug:Log(Object)
603
[18:11:44] Collision
UnityEngine.Debug:Log(Object)
4
Game
Animation
Display 1
Free Aspect
Scale 1x
Left Eye
Maximize On Play
Mute Audio
Stats
Gizmos
Collision
Auto Generate Lighting Off
Type here to search
ENG
6:11 PM
5/27/2021

# Experimental Evaluation

- Validation loss ~ 0.02, Testing loss ~ 0.02
- Accuracy ~ 0.99
- The accuracy/loss did not decrease/increase when adding users (N=4)
- However, in practice, the login system fails to log any user in
  - Possible bug in code → will debug
  - Model overfitted → unlikely, but will try cross-validation and better augmented data
  - Data too specific → will try better augmented data



# Future work

- For login system
  - Debug; cross-validation; better data augmentation
  - More users
  - Instructions on screen
  - Full virtual keyboard & username implementation
- Use of login system
  - VR exposure therapy application for emetophobia

# Thank you!

Mircea Maria-Mădălina

WeADL 2021 Workshop

The workshop is organized under the umbrella of WeaMyL, project funded by the EEA and Norway Grants under the RO-NO-2019-0133.
Contract: No 26/2020