InteractML: Node Based Tool to Empower Artists and Dancers in using Interactive Machine Learning for Designing Movement Interaction

CLARICE M HILTON

Goldsmiths, University of London, UK

PHOENIX PERRY

University of the Arts, UK

NICOLA PLANT

Goldsmiths, University of London, UK

CARLOS GONZALEZ DIAZ

University of York, UK

REBECCA FIEBRINK

University of the Arts, UK

BRUNO MARTELLI

Studio Gibson/Martelli, London, UK

RUTH GIBSON

Coventry University, UK

MICHAEL ZBYSZYŃSKI

Goldsmiths, University of London, UK

MARCO GILLIES

Goldsmiths, University of London, UK

Artists, and particularly dancers, have a deep knowledge of movement that can inform movement interaction for immersive media. There are few tools that enable movement interaction in immersive technology to be easily designed by nonprogrammers. We explain the design rationale behind InteractML a node-based interactive machine learning tool for designing movement-based interaction. We also describe a design principle:that the design should serve a dual purpose of increasing understanding of machine learning whilst giving access to customise the machine learning system.

CCS CONCEPTS •Human-centered computing~Human computer interaction (HCI)~Interactive systems and tools~User interface toolkits •Human-centered computing~Human computer interaction (HCI)~Interactive systems and tools~User interface programming •Applied computing~Arts and humanities~Performing arts

Additional Keywords and Phrases: Interactive Machine Learning, HCI, Dance

ACM Reference Format:

First Author's Name, Initials, and Last Name, Second Author's Name, Initials, and Last Name, and Third Author's Name, Initials, and Last Name. 2018. The Title of the Paper: ACM Conference Proceedings Manuscript Submission Template: This is the subtitle of the paper, this document both explains and embodies the submission format for authors using Word. In Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY. ACM, New York, NY, USA, 10 pages. NOTE: This block will be automatically generated when manuscripts are processed after acceptance.

1 INTRODUCTION

HCI discourse now embraces the cultural world as a resource for designing interfaces, respecting the importance of emotion, experience and aesthetics[1]. Theories are increasingly inspired by dance and movement practice based in somatic design and embodiment. This is particularly true of immersive technologies such as VR or AR, where the use of body movement in interaction is able to increase the sense of presence[2].

Interactive Machine Learning (IML) has the potential of enabling non-programmers to create, control and customise machine learning (ML) applications. We define IML as a workflow of rapid cycles of training, evaluation of performance and editing the model to improve performance[3]. Systems that enable the user to provide and edit training data deciding how this data is classified.

In this paper we explain the rationale for the design of InteractML a node based IML tool for designing movement based interaction for virtual and augmented reality for artists[4], dancer and indie games developers. It will detail the justification for our design process as well as the challenges that have come up in our user based research. At the heart of our approach is the aim to make IML more democratised and accessible to a wider range of people. Designing an interface that is not only easy to use but improves understanding of what ML is and it works so that users are empowered to use it in innovative and interesting ways.

2 BACKGROUND AND RELATED WORK: INTERACTIVE MACHINE LEARNING

IML is increasingly being implemented to design and build new gestural controls to allow users to create and define examples improvisationally, and to evaluate models through experimenting with controllers in real time. Several toolkits have been designed for the use of programmers such as GRT[6], RapidMix, RapidLib and ml.Lib. Tools have been designed for less expert users such as Delft AI Toolkit and Wekinator offering a graphical user interface. Wekinator[7] was built for the creation of new musical instruments, it allows for custom mappings between gesture and computer responses. IML is also being used by programmers working with dancers to implement their artistic work[8]. These are used in collaboration with the artists controlled by the programmer rather than creating tools which put the power in the hands of the dancer.

To make immersive technology it is necessary to use a game engine such as Unity3D. Linking IML libraries into interactive 3D has proven challenging. Schedel and Perry[9] document the process needed to connect Wekinator to Unity 3D and Max/MSP simultaneously via Open Sound Control (OSC). The complexity of this tool chain applies limitations on creators when it comes to creating, sharing and distributing creative work.

For artists and dancers to create movement based interaction artworks they must rely on a developer to programme a way for the system to understand a particular movement. There are many issues with this design process; it is non-iterative, uses linear problem solving and means technical limitations stifle creative ideas[8]. InteractML intends to give this artistic power back to the artist giving them the ability to control the ML system and the movement which they want to express. In the ethos of the design is not only for them to be able to use the system themselves, but for them to gain a higher understanding of how the system works and what it is capable of so they are able to create more original uses of the system and enrich their artwork.

3 DESIGN GOALS AND JUSTIFICATIONS

3.1 Node Based Systems

Node based programming allows for the manipulation of pre-created modules of code. Nodes are connected inside a visual graph. The logic written into nodes, which the end user does not engage with on programmatic level, composes the functionality of the system. The output of one node becomes the input for the next, allowing for flexible routing of data to accommodate a range of use cases. This form of programming provides a way for individuals to use a system's functionality without necessarily learning to program that functionality themselves.

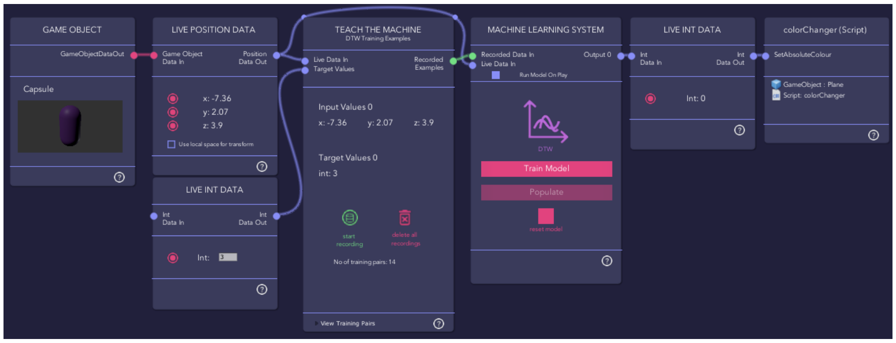

Figure 1: InteractML Node Based System

3.2 Usability for non-programmers

This form of programming is suited to individuals unfamiliar with programming as a way to encourage developing competencies in ML[10]. Node based systems have shown to support embodied interaction design in creative practice design students engaging with these concepts at the level of higher education. Within systems such as this, the aim is to empower creative thinking around designing embodied interfaces for non-programmers.

The systems creative practitioners are familiar with informed the choice to create a node-based environment. Popular tools for embodied creative practitioners include Max/MSP, Isadora, Pure Data and Touch Designer. These software packages use flow-based programming to allow development of bespoke interactive systems. Using a familiar visual design language established provides a mental model for users of these paradigms to build from. A mental model is the understanding a user has of how a system works [11]. Game engines such as Unreal, Twine, and Unity are supporting node-based development within their engines. However, a challenge remains that, while 3D game engines support VR and the creation of embodied systems, dancers are not as familiar with ML and its use to support movement based creative applications mentioned above.

To help address this situation, InteractML implements a flow-based system allowing for entry level or non-programmers to build IML applications by focusing on the embodied interaction design within Unity3D. It allows for creators to engage with the concepts needed to understand IML without the need to learn to implement the required algorithms. The systems are capable of empowering a high level of personalization and expressivity.

4 INTERACTML

InteractML is a Unity 3D plugin that enables creators to configure, train and run an IML system within the game editor. It is based on a similar model to Fiebrink's Wekinator [7] whereby the user provides examples of their movement, classifies this data with a target value and can use this to trigger outcomes in the system. It supports classification and dynamic time warping which are KNN and regression MLP algorithm using RapibLib library.

InteractML enables user to extract data from game controllers, headset, MOCAP suite or any sensor that can be integrated into Unity. Any movement data can be extracted from this including position and rotation, there are additional nodes that give access to further information such as velocity or distance between two values. The user records training examples in the Teach The Machine node using the data extracted from these objects along with a target value as the classifier, as configured by the user. By specifying which Machine Learning System node is connected they choose the algorithm. In this node they can train the model and run the system to start recognising a movement or gesture. When a movement is classified the target value is produced and can be extracted from the system to trigger events. In the design of ML tools one of the biggest challenges has been users not understanding how systems behave and what they are capable of[12]. Central to the design of InteractML is designing an interface which increases the understanding of ML as well as being easy to use.

In the design of InteractML we have used a user centered design process. We have run workshops using the tool and a tool hackathon with project collaborators Gibson and Martelli to learn about how the tool is used and to inform the design. Amershi[13] emphasises the importance of studying the use of a machine learning tool with the lack of proven design principles in this area.

5 CHALLENGE - INPUT/OUTPUT TERMINOLOGY AND FLOW

A challenging design problem when creating interfaces for ML is the terminology around describing inputs and outputs of the system. A typical flow would see that, for instance, users working with movement sensors would understand the live positional data as 'input', then some sort of computational processing would occur, out of which an 'output' would be produced[14]. With a IML system in contrast, data flow is slightly different.

When using the InteractML system, the process of working with inputs and outputs requires an additional stage: generating training data. For training, the user will connect sensor input, then configure the classifier, which the system output when the gesture is recognized, sometimes referred to as the target value [9]. The result of this recording, is used as input to train the ML model. When the user wishes to run the model, they must input live movement data, based on which an output value or values are outputted. Finally, the output will then need to be mapped to something outside of InteractML in the game engine, via code.

This must be carefully managed in the interface, so the user can clearly differentiate the input and output at each point of the workflow and conceptualise how they relate to each other within the system. The challenge is using terminology that is consistent across the system but also allows for this differentiation, and designing the flow of data is visualised through the system to clearly demonstrate the relation of input/output at each stage.

The language used in the system is vital so that it can be understandable to non-expert users and newcomers to ML. We wanted the language to increase the understanding of how ML works, describing what the process is. In the system we avoid the use of input opting to describe what the system is doing, live data in is used to describe the data coming from the sensors. Throughout the system non-technical language that increases understanding is used for example the system is the system this is also done for example Teach The Machine for the node where generation of training data is controlled to explain that stage of the system.

6 CONCLUSION

We have explained the design rationale behind InteractML using a node based interface, that the design should have a dual role of deepening understanding of ML as well as giving access to controlling the system. A design that does both these things overcomes some of the most prevalent challenges in designing ML systems. Our research also throws up interesting questions: What can movement practice bring to the design of IML? When a system is built inspired by and to support an art where improvisation is key what can this add to how we design? What are the benefits and the difficulties of node based systems? How can the design of interfaces help people understand what ML is and how it works? At the core of our research is the aim to democratise technology and enable more people to use machine learning enriching what is possible to be built by IML.

REFERENCES

[1] S. Bødker, ‘When second wave HCI meets third wave challenges’, in *Proceedings of the 4th Nordic conference on Human-computer interaction changing roles - NordiCHI ’06*, Oslo, Norway, 2006, pp. 1–8, doi: 10.1145/1182475.1182476.

[2] M. Slater, ‘Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments’, *Phil. Trans. R. Soc. B*, vol. 364, no. 1535, pp. 3549–3557, Dec. 2009, doi: 10.1098/rstb.2009.0138.

[3] J. A. Fails and D. R. Olsen, ‘Interactive Machine Learning’, p. 7.

[4] C. Gonzalez Diaz, P. Perry, and R. Fiebrink, ‘Interactive Machine Learning for More Expressive Game Interactions’, in *2019 IEEE Conference on Games (CoG)*, London, United Kingdom, Aug. 2019, pp. 1–2, doi: 10.1109/CIG.2019.8848007.

[5] M. Ware, E. Frank, G. Holmes, M. Hall, and I. H. Witten, ‘Interactive machine learning: letting users build classifiers’, *International Journal of Human-Computer Studies*, vol. 55, no. 3, pp. 281–292, Sep. 2001, doi: 10.1006/ijhc.2001.0499.

[6] N. Gillian and J. A. Paradiso, ‘The Gesture Recognition Toolkit’, in *Gesture Recognition*, S. Escalera, I. Guyon, and V. Athitsos, Eds. Cham: Springer International Publishing, 2017, pp. 497–502.

[7] R. Fiebrink and D. Trueman, ‘End-User Machine Learning in Music Composition and Performance’, p. 4.

[8] S. Fdili Alaoui, ‘Making an Interactive Dance Piece: Tensions in Integrating Technology in Art’, in *Proceedings of the 2019 on Designing Interactive Systems Conference*, San Diego CA USA, Jun. 2019, pp. 1195–1208, doi: 10.1145/3322276.3322289.

[9] P. Perry, M. Schedel, and R. Fiebrink, ‘Wekinating 000000Swan: Using Machine Learning to Create and Control Complex Artistic Systems’, *Proceedings of the International Conference on New Interfaces for Musical Expression - 30 May - 1 June 2011*, p. 4, 2011.

[10] S. A. Brown, M. Gratton, and B. Williams, ‘A Visual Programming Tool for Creative Practice Pedagogy in Embodied Interaction and Media Arts’, in *Proceedings of the 31st Australian Conference on Human-Computer-Interaction*, Fremantle WA Australia, Dec. 2019, pp. 364–368, doi: 10.1145/3369457.3369495.

[11] D. A. Norman, *The design of everyday things*, Revised and Expanded edition. New York, New York: Basic Books, 2013.

[12] Q. Yang, A. Steinfeld, C. Rosé, and J. Zimmerman, ‘Re-examining Whether, Why, and How Human-AI Interaction Is Uniquely Difficult to Design’, in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, Honolulu HI USA, Apr. 2020, pp. 1–13, doi: 10.1145/3313831.3376301.

[13] S. Amershi, B. Lee, A. Kapoor, R. Mahajan, and B. Christian, ‘CueT: human-guided fast and accurate network alarm triage’, in *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI ’11*, Vancouver, BC, Canada, 2011, p. 157, doi: 10.1145/1978942.1978966.

[14] F. Bernardo, M. Zbyszy´ nski, R. Fiebrink, and M. Grierson, ‘Interactive Machine Learning for End-user Innovation’, *Association for the Advancement of Artificial Intelligence 2017*.