Customizing by Doing for Responsive Video Game Characters

Andrea Kleinsmith (alk@cise.ufl.edu)
Department of Computing and Information Science and Engineering
University of Florida
Gainesville, Florida 32611
Marco Gillies (m.gillies@gold.ac.uk, corresponding author)
Department of Computing
Goldsmiths, University of London
London SE14 6NW UK

ABSTRACT

This paper presents a game in which players can customize the behavior of their characters using their own movements while playing the game. Players' movements are recorded with a motion capture system. The player then labels the movements and uses them as input to a machine learning algorithm that generates a responsive behavior model. This interface supports a more embodied approach to character design that we call "Customizing by Doing". We present a user study that shows that using their own movements made the users feel more engaged with the game and the design process, due in large part to a feeling of personal ownership of the movement.

KEYWORDS

Interactive machine learning, embodied design, body expressions, video game characters

1. Introduction

Customization of video game characters by players is now an important part of the experience, particularly in multiplayer online games, where customization can establish a distinctive identity for a player's avatar. A customized avatar serves both to increase a player's identification with the avatar and as a means to communicate a player's identity to other players. However, the aspects of a character that can be customized are often fairly limited. The most common form of customization is visual appearance, most often via selecting a combination or blend of a set of standard visual elements (hair, eyes, mouth etc). Customization of a character's movements is much rarer, and mostly consists of replacing individual animated actions from a standard set. For example, Second Life allowed players to buy new animations, but the creation of these animations was a difficult process, requiring either expertise with 3D animation tools or access to an expensive motion capture setup. However, customization can also go deeper than appearance and movement. Players could customize how their characters respond to events, how they select actions in response to what is happening in the game. This means customizing the set of algorithms that perform action selection; what in game development is called the Artificial Intelligence (AI) of the character. This is an important aspect as the nature of a character is not simply determined by how they perform actions (animation) but by which actions they choose in response to specific events. Altering the character's set of animations can change the responses, but only to a fixed set of triggers. To truly customize responsive behavior, a player should be able to customize the triggers to which a character responds as well as the responses produced by those triggers. Another approach to customization is to parameterize the algorithms, presenting players with a set of parameters; the values of which control the character's decision making processes. However, if these parameters are to be set by end users they must be easy to understand and the set of parameters must be kept small, so that players are able to customize them effectively. This innately limits the complexity of customization that is possible[1]. The question we ask in this paper is whether it is possible to create accessible user interfaces that allow game players to customize the responses of game characters; meaning customizing both the actions that are performed and the environmental triggers in response to which the actions are performed.

The advent of cheap motion capture controllers like the Microsoft Kinect[tm] enables new approaches to customization. Customizing a character's movement no longer means relying on pre-set animations designed by experts; players can use their own movements directly as the basis of their characters, i.e., as the character's animations. This paper investigates whether it could also be a means of allowing players to customize the responses of a character, as defined above. Players can customize how characters respond to different contexts by directly providing examples of actions in those contexts, through recording their own movements within the game. They can customize the game by playing the game.

---

[1] This is somewhat a function of genre, certain "hardcore" gamers are positively attracted to the possibility of tweaking large numbers of complex parameters but this would be daunting to the vast majority of the market.

This approach builds on the growing research area of Interactive Machine Learning [3] in which machine learning algorithms are used to train models based on examples provided interactively with users. This enables a form of design that Fiebrink [4] calls "embodied". Interaction is designed through physically doing the action rather than tweaking parameters. This links into the ideas of HCI thinkers such as Dourish [2] or Klemmer et al. [10] who highlight the importance of embodiment in interaction. Both stress the importance of action for thought. One of Klemmer et al's key themes is "thinking through doing" while Dourish states that "action both produces and draws upon meaning; meaning both gives rise to and arises from action" [2]. As Hummels et al. phrase it "[in order to design movement interfaces] one has to be or become an expert in movement, not just theoretically, by imagination or on paper, but by doing and experiencing while designing" [9]. These ideas support our aim of having players customize their character with their own movements. In this paper we attempt to do this by presenting a system that allows people to customize the responses of video game characters by directly playing the game using their own body movements in motion capture.

The latter part of this paper presents a study that investigates how customizing by doing can support the customization of characters' responses. We address two major research questions:

1)What, if any, is the benefit of using players' own movement for customization in general and particularly customization of responsive behavior?

2)To what degree can interactive machine learning approaches support customization by doing of responsive behavior?

Previous research that has informed this work is presented in the next section. The application is described in Section 3. Sections 4 and 5 describe experimental user study which investigates how players' experience of designing a character's behavior using their own body movements compares to designing with a pre-existing set of movements and the methods they use to link the results of embodied behavior to the logic of the game. Conclusions are presented in Section 6.

2. Background

2.1. Expressive Virtual Characters

One of the key research challenges for interactive characters in games and other virtual environments is to emulate the non-verbal expressions that we use in daily life. There has been a wide range of research in this area as surveyed by Vinayagamoorthy et al [23]; examples include models of emotional expression [7] and social interaction [22]. There are a number of ways of customizing this type of expressive behavior. One approach is to alter the style in which a movement is performed; for example, the work of Taylor and Hinton [20] or Neff and Kim [15]. The manner in which an action is performed is transformed without changing the action itself. This can provide considerable scope for variation, but much of the variation between individuals is in what they do as well as how they do it. A number of approaches to this have been suggested. For example, Noot and Ruttkay [16] tag particular utterances with probabilities of performing particular non-verbal gestures, while Prendinger et al [18] allow users to define simple scripted actions in response to perceptual input. Both of these systems use an XML definition language that is well suited to professional game designers but is unlikely to be sufficiently accessible to end users. In this paper we present an interface for customizing behavior that is based on Interactive Machine Learning.

2.2. Interactive Machine Learning

Interactive machine learning (IML), proposed by Fails and Olsen [3], centers on human interaction to improve supervised learning systems by allowing the user to implement her expert knowledge to guide the learning process. An issue with traditional machine learning techniques is that they typically require considerable technical understanding, which comes from experience [3], [9]. A significant advantage to IML is that it is meant to be easily used by non machine learning experts. This is becoming increasingly important as machine learning techniques are being applied in a variety of research disciplines, such as HCI. Indeed, the results of a survey of over 100 HCI researchers showed that at least a third of them have used machine learning in their studies [14].

A main aim for many IML systems is to support an iterative approach to the improvement of the model created. Doing so allows the user to more easily explore different model variations than in traditional machine learning. For instance, Kapoor et al [9] have implemented ManiMatrix which allows users to interactively control a confusion matrix. By using their knowledge of the data, the users specify their preferences for classification and use an iterative process to optimize the model. For instance, it may be more important to the users to obtain higher accuracy rates for some classes over others. Talbot et al [21] propose EnsembleMatrix which allows users to combine classifiers and interactively explore the relationships between them by directly manipulating the visualizations.

In addition to the importance of iterative refinement, the creators of other IML systems operate under the premise that the users should be able to provide and edit their own training data. For instance, Fails and Olsen [3] present the Crayons system which enables non-expert users to create image processing classifiers in an iterative process by drawing on images that they

provide as training data. Fiebrink et al [4], [5] considered composers and students as end-users in building new, gesturally-controlled musical instruments in which the users interactively created and labelled their own training data and iteratively evaluated and improved the resulting model. In this system, the end-user becomes even more involved in the process of creating the system through an embodied interaction; designing gestures to control by performing gestures to music. They found that using embodied interaction to create an IML system creates a more engaging experience and connects the user to the process in order to better understand how best to achieve a usable system.

.3 Machine Learning for Virtual Characters

Machine learning has been applied to virtual characters in a number of ways. It has been used most often in contexts where there is a clear score that the system has to optimize to produce the best possible character at playing a game (e.g. Lee and Lee's 2004 work [12] on reinforcement learning for a boxing character). Others have used learning algorithms to approximate examples of real behavior. For example, Kwon et al. [11] learn dynamic bayesian networks that reproduce the interaction between two fighting characters. Similarly, Gillies [6] has created characters that respond expressively to game events (in this case, spectators at a football match). This was learned from motion capture of a person responding to example events. Levine et al [13] use a similar approach for learning to control gestures to accompany spoken conversation. These examples form the basis of our current work. We extend this approach to use interactive machine learning, in which the human is not simply a source of data but is actively involved in guiding the learning process.

3. Embodied pong

We have created a software application that enables users to design, using their own movement, the responsive behavior that controls a video game character. They design the character's behavior by recording their own movements while playing the game and then training a decision tree model based on these movements.

Another aim for the study was to investigate how participants link their own actions to the logic of the game. In order to do so we wanted our application to support two methods. The first is an interactive machine learning approach in which participants label the motion data but the responsive behavior is created automatically. However, we also wanted to give participants the ability to manually edit the system themselves. We therefore wanted to create a system that would give scope for expressive variations of body movement in the responses while at the same time being conceptually simple enough that participants were capable of both manual and automatic creation of the character's responses. For this reason we chose a relatively simple game but decided to design the characters' emotional responses to game events rather than the game play itself. This gives users considerable scope for creativity in their expressions, unlike game play AI which would focus on optimizing the score. The game mechanic gives a relatively simple and clear context for the expressive behavior, making the design of the character's responsive behavior easier.

We have chosen to implement a 3D version of the classic video game Pong (Figure 1). This is a simple and abstract representation of a game of tennis in which players control paddles represented as lines. They can use these paddles to hit a ball back and forth in an imitation of a tennis game. In our version players are able to control the paddle with their own body movements as captured with a motion capture system (there is also an option for more traditional control with a mouse, which is used for testing as described in the motion labeling section). The player is represented as a stick figure avatar as shown in Figure 1.

When the game is controlled with a mouse, at the end of each point (when either player misses a ball) the avatar performs an animation which depends on the current state of the game. The aim of the design process is to create a system which appropriately determines which animation to play given the current state of the game. The state of the game is represented by a number of numerical contextual variables:

- Last Miss: 1 or -1 depending on which player missed the last ball
- Score: the player's current score
- Relative score: the players current score minus the other player's score
- Ball direction: the left/right velocity of the ball
- Distance: the distance of the ball from the player when it was missed
- Volley length: how many times the ball passed between the players before the point ended

These variables were chosen to intentionally give a wide range of different types of information and to have some that were more obviously useful than others, in order to test participants' understanding of the game and variables. When the game is played with motion capture, the player's movements between the end of one point and the start of the next are recorded and saved as a motion clip together with values for the game state variables.

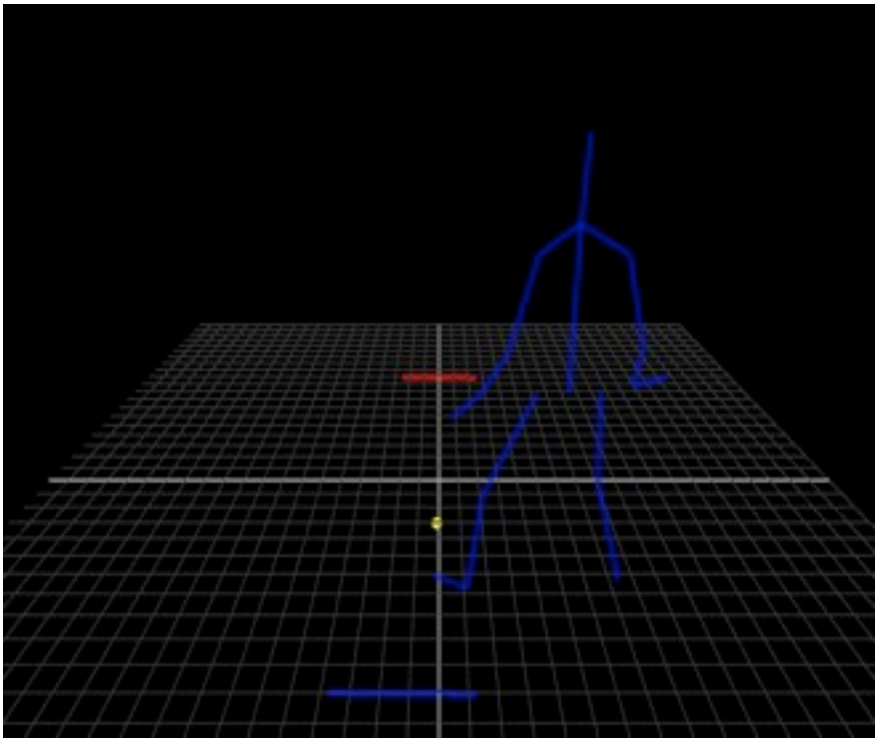

Figure 1. The embodied pong game. The player is represented as a stick figure

3.1. Decision Tree Based Responsive Behavior

The responsive behavior is based on Decision Trees [19]; a method of classifying multivariate data into different classes. A classification rule is represented as a tree. Each internal node of the tree represents a decision based on a single variable. The value of the variable is compared to a threshold and depending on the outcome, the data is passed down to one of the node's two children. For example, the current score could be passed to the left child if it is less than 10 or the right if it is greater than 10. The child node makes a decision based on its variable and value (for example deciding on distance). Leaf nodes are labeled with classes, so when a leaf node is reached the data item can be assigned to a class. This class is then used to select a motion clip to play.

A typical example in this game would be to play a celebration when the player wins a point or a disappointed animation when the player misses the ball. These animations would also be more extreme when the player is near winning or near losing the entire game. We could therefore have 4 classes: Win, Good Win, Lose, Bad Lose. A decision tree for this situation is shown in Figure 2. The top node makes a decision based on the last miss variable. The next nodes down then assign a class based on relative score.

Decision Trees can be learned automatically from a set of data labeled with classes. This is done by first selecting a variable and threshold value that best split the data into two sets. A best split is defined in such as way that there is minimal overlap of classes between the two sets, i.e, the sets are as close to containing distinct classes as possible. There are a number of possible measures for this, we use one of the most popular, the Gini index = $1 - \sum_c (f_c/N)^2$, where f_c is the number of items labeled with class c in the set, N is the total number of items in the set and the summation is over the classes c. Choosing a variable and threshold that minimizes the Gini index over both sets will tend to make them as different as possible. Once a split has been chosen, a node is created and the procedure is repeated recursively with each of the resulting sets in order to create the children of the node. If a set is found that only contains one class, a leaf node is created.

Figure 2. An example of a decision tree based on the variables last miss and relative score that were used for the pong game

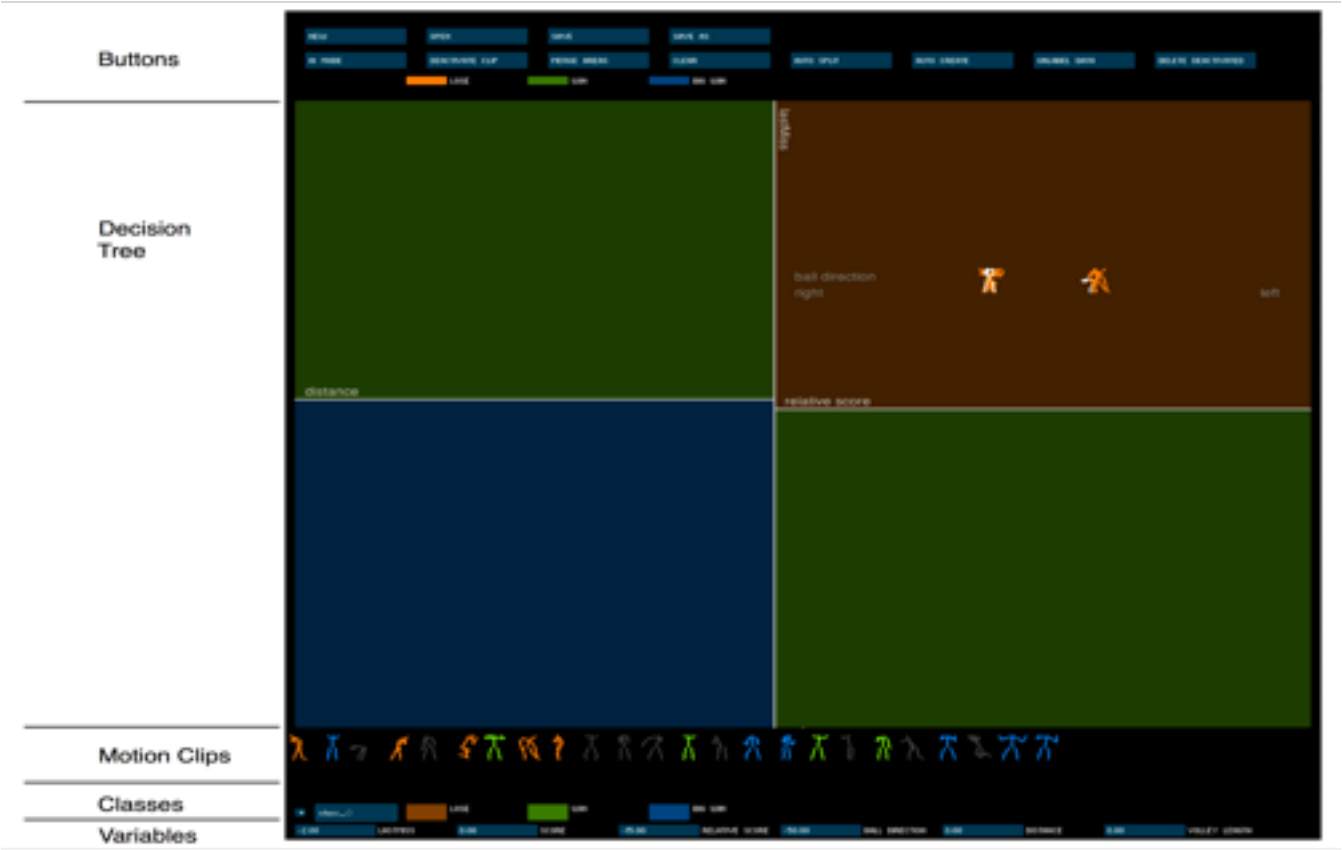

Figure 3. The interface used to design the character's responses. As shown from bottom to top of the Figure: variables are the variables representing each point in the game. Classes are the categories defined by the player. Motion clips correspond to the body movement recorded at each point in the game. Decision tree is the interactive visualization that players use to manually edit the decision tree model. The colors in the decision tree correspond to the classes and the lines correspond to the variables.

Decision trees were chosen partly for the reasons cited by Fails and Olsen [3]: that they are fast to learn and produce an efficient feature selection. However, the primary reason is that they can be both learned automatically from data and, unlike most machine learning algorithms, are conceptually simple enough to be created and edited manually. Therefore, they provide a fair test of automatic learning versus manual editing as a design process.

3.2. The Design Interface

Figure 3 shows the interface used to design the character's responses. After a game has been played and the motion capture data has been gathered, users can use this interface to create the decision trees that will control the character. There are two basic ways in which users can interact with the interface. They can label clips with classes and then tell the system to automatically learn a decision tree or they can design the tree manually using a visualization labeled as "Decision Tree" shown in Figure 3 and described below.

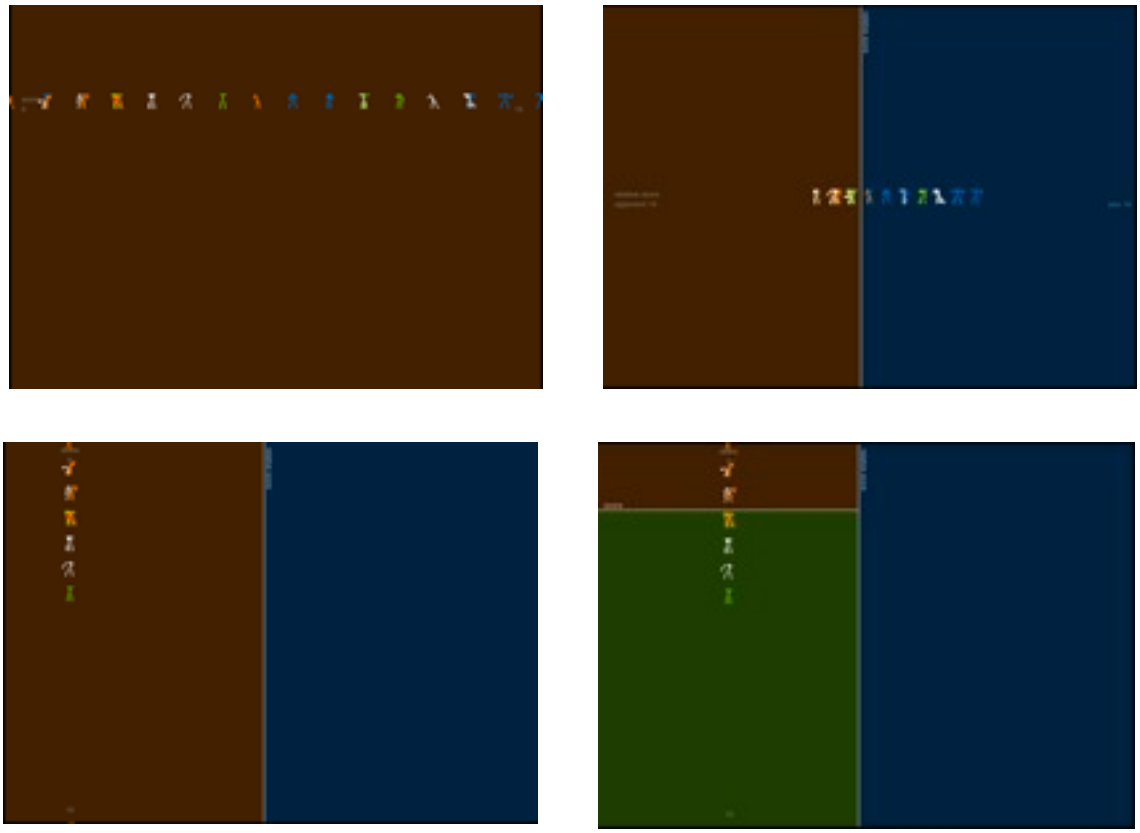

Figure 4. The decision tree visualization. Top left: the tree is empty. Hovering the mouse in the space shows the motion clip thumbnails ordered according to variable. Top right shows an internal node in the decision tree according to the selected variable. The line (i.e., threshold) can be modified by dragging it. Bottom left shows that clicking in an area selects that area and displays just those clips that are classified as belonging to that leaf node. Bottom right: the decisions are alternately split horizontally and vertically at alternate levels of the tree; so at the next level down the motion clips are displayed vertically and shift clicking results in a vertical split

3.3. Motion Labeling

While the game is played in motion capture mode the movements of the player at the end of a point are saved together with the state of the game. Thumbnails of these motion clips are shown in the section of the interface labeled “Motion Clips”. Users can click on the thumbnail to view the motion in the original game window (Figure 1). They can also see the values of game state variables in the section labeled “Variables”.

The interface can be used to label these clips with classes. Users can define and select classes using the section of the interface labeled “Classes”. Each class has a name chosen by the user and a color which is used consistently throughout the interface. To label a clip the user must select the clip by clicking on it and then clicking on the appropriate class. The thumbnail is then shown in the color of the class. Clips can also be deselected if the user does not want to use them (shown grayed out in the interface).

Once the user has labeled clips, they can click “Auto Create” in the buttons section of the interface to automatically generate a decision tree. Once a tree has been created they can test the system by playing the game with the mouse interface and seeing the responses of the avatar.

3.4. Manual Editing

It is also possible to manually create and edit decision trees using the interactive visualization labeled “Decision Tree” in Figure 3. The visualization is a 2D plane divided into colored areas, each of which corresponds to a leaf node in the tree. The color of the area is a desaturated version of the color of the class associated with that leaf. The lines corresponding to the internal nodes are labeled with the variable name associated with that node.

The mapping of the data space into 2D is not straightforward, as the data is multidimensional. The visualization functions as a series of overlaid 1 dimensional representations. When the tree is empty (Figure 4, top left) hovering the mouse over the visualization shows the motion clip thumbnails laid out horizontally ordered by one of the game state variables. Hovering at different heights shows different variables.

Shift-clicking anywhere along the line of motion clips will split the space according to the variable being displayed (Figure 4, top right). This creates a new internal node in the decision tree that makes a decision according to the selected variable. The threshold is determined by the horizontal position at which the mouse is clicked. The user can then drag the dividing line to change the decision threshold. It also creates two leaf nodes assigned to the two new areas of the visualization. The class (and therefore color) of these nodes is determined by the majority class of the clips that are classified as being in that leaf. If there are no labeled clips then the node is not given a class.

The user can continue to split the resulting spaces to create new decision tree nodes. Clicking in an area selects that area and displays just those clips that are classified as belonging to that leaf node (Figure 4, bottom left). In order to maintain a reasonable visual layout of the space and avoid very thin areas the spaces are alternately split horizontally and vertically at alternate levels of the tree. So at the next level down the motion clips are displayed vertically and shift clicking results in a vertical split (Figure 4, bottom right). This does not affect the underlying decision tree.

4. User study

We conducted a user study to compare the design of a character's responsive behavior for the Pong game using the players' own movements versus existing movements. We provided both the interactive machine learning and manual editing approaches to the task. The study consisted of two tasks: one in which the players recorded their own movements while physically playing the game and one in which they were given a pre-existing set of movements recorded by one of the authors while playing the game, in the same manner as the participants. The goal of the experiment is two-fold: a) to examine how participants interact with the design interface and b) to evaluate whether differences exist between designing the character's responses using their own movements and existing movements. We hypothesize that participants will engage more with the interface when their own motions are used.

4.1. Participants

20 participants, ranging in age from 20-35 were recruited via posters and email. Half of the participants were computer science students, and the other half were from a variety of other, non-computer or programming based educational backgrounds in an effort to determine whether programming knowledge made it easier to use the system. Before beginning the experiment, participants were asked to read an information sheet after which they signed a consent form if they were happy to continue. The study lasted approximately two hours.

4.2. Task overview

An Optitrack optical motion capture system was used for the task involving the participant's own movements. Each participant was dressed in a motion capture suit to which 34 light-reflective markers are attached. The participants were asked to physically play pong and have their own body motions recorded. Eleven infrared cameras, mounted in a circular configuration above the motion capture space, track and reconstruct the markers in a 3D space. The motion capture data was streamed in real-time, directly into the pong game where the participant was represented as a stick figure as illustrated in Figure 1. The pong game was projected on a screen in the motion capture space and the participants moved in a horizontal direction to control their paddle. After recording their motions, the participants used them to create the character's responsive behavior. The participants did not physically play the game for the task using the existing motions.

4.3. Procedure

A within subjects study was carried out with two conditions, own movements versus existing movements. The task order was randomly determined and counter-balanced. Before starting either task, the participants underwent a training session in which they were shown a simple version of the pong game and asked to play a few points using the mouse and keyboard interface to see how the character responded for each point won or conceded. Next, a researcher gave a brief demonstration of the system and explained how to use it to create the rules of the game they had been shown. Participants were then given two short training tasks to familiarize them with how to use each part of the system (i.e., interactive machine learning and manual

editing) separately. One of the researchers observed in order to answer questions and make sure the participants understood the task. When the participants finished with the training tasks, the main part of the study began.

For both tasks, the participants were asked to design the character's responses in the game (with their movements or with pre-existing movements) and then test it by playing to see if the game reacted the way in which they expected. If it did not, they were instructed to continue to make modifications using the interface and check them with the game until they were happy with the outcome. The purpose was to explore if and how the participants used an iterative design process as this is considered an important aspect of IML. Each task was limited to 30 minutes.

After each task, participants answered several questions on a 7-point Likert scale about their use of the system: how much the manual and automatic parts of the system were used (1 = not at all, 7 = only), their level of happiness with the game they created (1 = not at all, 7 = extremely), how often the model they created reacted as they expected (1 = not at all, 7 = all the time) and the number of iterations they performed to create the model.After completing both tasks, participants were asked additional questions in a semi-structured interview format in order to obtain details about their strategies in how they used the system, and what difference, if any, their own movements made to their use of the system (e.g., was using their own movements an advantage or a disadvantage, did physically playing the game help and did that affect how they used the system).

5. Findings and discussion

Below we present results of the interview with participants. There were no statistical differences between participants based on their prior background in programming or gaming.

5.1. Use of participants' own motions

In the post-experiment interviews, the majority (17 out of 20) of the participants expressed a strong preference for using their own motions over using existing motions. The following quotes are good examples of the overall responses: "*it made it a bit more real, because you know, it was my reactions. It kinda felt like I was there"* – P12; *"when I was doing it, I knew what the meaning was" – P3; "[it was] an advantage because you know what you're trying to portray" – P7;* and *"[it was] definitely an advantage. I got to see my own expressions – the way I reacted" – P14.* Using their own motions was also considered to be *"more interesting" – P8* and *"added more fun to the game." – P4.*

Only one participant (out of 20) reported that using his own motions was a disadvantage, but only because of his expressions. *"In this case it was a disadvantage you could say. But if I had done some more extreme acting, of course it might have been an advantage." – P5.* Two participants out of the 20 initially felt it was neither an advantage nor a disadvantage. However, P10 qualified his statement by adding: *"I mean, I think it was an advantage because I could sort of, I knew like, I could put more emphasis where I knew... so when I watched back the clip, I remembered 'oh yeah, I really celebrated hard', so I knew that was what I was doing there."*

Though these results are pleasing, it is important to understand whether they would carry over to the real world situation of game customization. There are two elements to this question; the first is understanding why participants preferred customizing with their own movements, and whether the same factors would be at play in real, more complex game customization. The second element addresses the central question of this paper, does using their own movements support players in customizing not only animation but responsive behavior?

Engagement and personal ownership

Many comments suggest that the condition in which the participants recorded and used their own movements was more engaging and made the process more personal: *"It was incredible. Embodying the animation made it more personal" – P14.* Similarly, P15 commented *"it makes more sense and then I understand it more when I had my own animations. It becomes more personal"*. P11 said that the result *"felt more like it was my baby because those were my own moves."* These, and other, quotations suggest that designing with their own motions produces a strong sense of personal ownership and attachment to the results which are likely to provide strong motivations to end users customizing their video game avatars.

Significantly, many participants linked physically playing and designing closely to their emotional state: *"When it was my own game […] because I knew what I was doing at the time of the movements, it was easier to divide; whereas with the second one [non-embodied task], they weren't my own [movements], so I had to sort of think of what emotion that animation might be. So definitely, it was easier to physically play, because it helped me then make my own game because I knew what I was feeling as I was playing it" – P9.* P18 commented that it *"helped in that I knew my emotions. It connected me more with like, thinking about what the body might be doing when it's feeling different things. So it made me much more aware of my own physical reactions to things. […] It got me more emotionally connected to it. And the fact that you know it's yourself there, too, makes it more enjoyable. You think, it's cool, it's me, and it's something you can relate to*." Indeed, Berthouze et al

[1] postulate that greater involvement of the body in game play causes a stronger emotional experience for the player. Our results suggest that embodied design is particularly applicable to contexts which involve emotional and expressive behavior.

These results suggest that two key factors in participants' preference for using their own movements were personal ownership and emotional engagement. These factors would seem to carry over well to game customization. Personal ownership is a key motivation for customizing game characters, and therefore, increasing personal ownership by using players' own movements is likely to enhance the customization experience. Emotional engagement is also highly prized in video games, and so the benefits here are also likely to carry over.

Interestingly, the two negative responses about physically playing the game were linked to participants not being very expressive in their movements. P5 said he *"would be happier with a predefined set because they would be more extreme or more cartoon like than my own."* However, P7 remarked, *"In the game I wasn't so expressive but in the game I wanted to see more expressive animations. If I did it over, I would be more expressive in the game - more theatrical" – P7.* He went on to say, *"In the beginning I was interested in describing what I felt by playing. In the end, [after creating the game] I realized I enjoyed seeing more expressive animations".* This suggests that there is a learning curve to using the system (i.e., the players need to determine beforehand what sort of body expressions they want to portray and what sort of game character they want to customize), and that changes would be made with subsequent usage.

Customizing Responses

The above results suggest that customizing by doing is likely to increase player engagement. However, this could be done simply by customizing animation. Does it also help customizing a character's responses? This is a harder task that relies on understanding both the character's actions and the factors that trigger those actions. Does performing the movements themselves help participants with this task?

One of the main reasons for the positive response to the embodied interaction is that it helped the participants think about the game. Indeed, 17 out of 20 reported that physically playing the game helped them plan their responses: *"It helped me get better reactions" – P7* and *"I knew what I was feeling and I knew how I wanted the system to react" –P9.* P9 further commented that she started to work out what she wanted the game to do while she was recording her movements. This shows that the participants seemed to be engaging in a cycle of 'thinking through doing' in which their understanding of the task influenced their embodied action and this action in turn deepened their understanding of the task. This is evident in the quote from P13: *"it was from that one that I decided how does it work and what we needed to do."*

As a result of learning through playing, the majority of the participants (13 out of 20) answered that physically playing the game made a difference in how they ultimately used the software to create the character's responses. The process allowed them to do and experience while designing, as deemed a requirement according to Hummels et al [8]. For instance, P11 stated, *"I knew what the moves were for. […] when I was doing the motion capture, I had the game in mind, and how I wanted it."* According to P13, *"from that one [physically playing the game] I understand how to use the system".*

Moreover, the participants found the process easier to understand when they performed the motions themselves. One reason for this is that they had a clearer understanding of their own movements than they did of the movements that had been supplied to them: *"[I was] a bit more certain about what I was doing because it was easier for me when it was my own movement, and also I remembered what I was doing at that time. I could visualize it a bit more" – P2.* Similarly, P9 remarked, *"I knew what I wanted to do because it was me - they were my movements […] I just knew what I wanted and it worked from the first time"* and P4 stated, *"you already know what you've done in terms of the motions that were captured so it's easier to put them in slots [categorize] without checking them".* P16 said, *"with [the exisitng motion task], I had one goal/ one strategy. But after I experienced the motion capture [...] I slightly changed my strategy by looking at myself, reacting to how I won and how I lost."*

These results suggest that the act of performing movements in context does support participants in understanding the complex process of designing responsive characters. Playing the game and performing the movements seems to have given participants a clearer idea of the types of responses that they wanted to create than they would have had doing it in the abstract. Customizing by doing may therefore support players in better understanding the process of customizing characters' responses.

5.2. Interactive Machine Learning for Customization

The above results show that playing the game and performing the motions could help players become more engaged with customizing their characters and better understanding the customization process. However, the question still remains of whether the IML system described effectively supports this form of customization. The results below raise some issues with this form of software.

Data labeling, manual editing and learning

The concept of labeling movement clips with classes was fairly simple and used by 17 out of 20 participants. Participants were able to create their own classes and apply these to the movement clips. This shows that participants were able to relate their own movements to classes of responses and that this is a basic task that game players could readily understand and use when customizing characters.

The manual editing of the decision tree was conceptually much more difficult for the participants to grasp. For many, when they used the manual part, it was without understanding exactly what they were doing and why. For example: *"I found the graphical interface [manual part] too complicated to sort of understand the embedding of rules and stuff like that, so in the end I just grouped the animations" – P5.* The participant went on to say that he found labeling much easier than using the manual part. Similarly, P17 felt *"the [labeling] is quite easy to understand how it works but [the manual part] I couldn't use it in a controlled way"*. Two participants admitted to trying to create a visually pleasing pattern. However, others were clearly able to use it and one participant in particular remarked that *"it was very quick to catch on to because you only explained it to me and within 5 minutes I was able to use it" – P4.* Some participants commented that labeling the data helped when manually editing the tree. For example, *"I started by labeling the animations first, because then I sort of knew what the animation was and then it was easier to see, like, my classes and the color of each class […] because I knew already that my animations were put in a certain class, so then I didn't have to look at each animation, I could just look at the color of the class" – P9.*This would suggest that automatic learning (described to participants as "auto create") would be an easier approach to designing responses. However, participants rarely used this function, with 16 out of 20 participants using data labeling as a precursor to manual editing rather than learning.

When auto create was used, it was sometimes without an understanding of what it produced. *"I did [use auto create] but not very rationally" – P17.* P8 used auto-create only one time, *"mainly just to see what it was."* This may be because the automated training resulted in a very complex decision tree, and the visualization looked intimidating to participants. P18 remarked that auto-create created a tight set of rules which made it difficult to see what was happening in the model. However, he still considered it easier to use than manual editing alone. When further describing his strategy and how he used the automated training, P18 stated with enthusiasm, *"it felt like that was a savior. I don't think I would have been able to have much chance. It was just too confusing and frustrating not to use that [auto-create] because at least that sets you up with an initial [model] and then you can sort of change it a little bit"*.

The decision tree visualization therefore had an overall negative effect on the customization process. Not only was it too complex for most participants to perform manual editing effectively, but the complexity seemed to actively discourage participants from using the potentially easier process of automatic learning. This is an important issue for game customization, where players are unlikely to want to spend time learning a complex interface.

Iterative design

In IML, it is important to be able to label the training data and use an iterative process to create the model [3], [4]. However, using statistical machine learning, Patel et al [17] found that non-experts had difficulty in applying an iterative approach. The results of our study indicate that labeling was quite intuitive, but iterating was less so, regardless of level of programming experience. The participants varied considerably in the number of iterations they performed on the model, ranging from 1 to 15 iterations for both conditions. There was an average 5.03 (SD = 4.23) iterations for the task with their own motions and an average 4.24 (SD = 3.27) iterations for the task with existing motions. These results are similar to the findings of Fiebrink et al [5]. Some did indeed seem to be using an iterative testing strategy, reporting testing as many as 10 to 15 times. However, others did not, only testing once or twice. Indeed, P2 remarked that he performed only one iteration because he took the time to carefully label each animation first. Another participant, P9, performed only three iterations, stating *"this one took more"* (referring to the task using her own motions in which she performed only one iteration). There was no significant difference in these results between those with a programming background and those without.

Training set size

The amount of data the participants were able to record was small compared to typical machine learning systems, up to about 20. Smaller data sets have the advantage that they require less effort to create and label and that it is possible to create simpler visualizations of them. However, this raises the question, is it possible to effectively create decision trees with such small datasets? In fact, it seems that at least six of the participants actively chose to further reduce the data sets they were using. One participant deactivated movement clips *"because there were so many" – P13.* P18 said *"I realized I didn't need the largest variety of emotions, like I think I could enjoy this game with probably like five different characterizations or movements and it seemed like there were far too many and some were too indistinct. I found I had much more satisfaction playing it when I could differentiate between the responses. Even if it was the same one that was repeated, it was still more satisfactory to have that sense of difference between them"*. Some participants deactivated clips because they felt there were too many redundant clips**:** *"I found that some of the motions were very similar to others so they weren't really needed" – P4*.

P16 chose to focus on the most important clips: *"I was just thinking about the game play. If someone was to play this, would they enjoy that clip, or would they enjoy this clip. […] I was thinking about the user as well as the creator."* These findings support the results of Fiebrink [4] who found that composers creating new musical instruments recorded small sets of training data and were happy with the models they created. She argues that this was possible because the systems are built through human interaction, and in particular that users focused on the key, most informative data items that were most informative when building the model. Many of the participants seemed to use a similar strategy.

6. Conclusion

The first result of our study is that when recording their own movements, participants were more engaged with the process and reported a more personal connection with the result. This is itself an important result as it demonstrates the potential for customizing by doing in video games, where engagement and personal connection are important factors. This result supports the use of players' own motions for customizing both animation and responses.

The study also showed that customizing by doing helped participants with the harder task of customizing responsive behavior. Players found that it helped them understand both their own responses to the game and the process as a whole. This seems to indicate that they are going through the process that Klemmer et al [10] call "thinking through doing", in which situated physical action is a key component of cognitive understanding. They propose that this is a key benefit of embodied interaction and our results seem to support this. Customizing by doing may therefore allow game players to customize more complex aspects of the character than simply appearance and animation.

While participants' feedback on customizing by doing was almost entirely positive, some comments do suggest a potential issue. Embodied design is about playing a game in order to design the character's responses. However, this does not mean that it is just about playing the game. Playing the game normally would result in a poor design. The behavior would not be expressive enough and the number of winning or losing movement clips is likely to be unbalanced. Embodied design also requires a certain distance from playing. The designer needs to keep in mind, as she plays, that she is designing. The fact that some participants did not spontaneously produce expressive behavior suggests that there are individual difference in the suitability of customizing by doing. This approach may be less suited to players who are not naturally expressive in their body movements.

A limitation of the study relates to the use of visualization in IML. Participants found the visualization of the decision tree hard to use and it seemed to actively inhibit them from using automatic learning, as the resulting visualization was highly complex. This could well be a problem with the visualization itself and a better visualization could be designed that solves this problem. However, it is also possible that the problem was also with the innate complexity of decision trees. The decision tree visualizations generated were complex because the trees were complex, and decision trees are relatively simple compared with other machine learning algorithms. One conclusion that we could draw is that visualizations are not useful in IML because of the innate complexity of the methods. Alternatively we could conclude that we should choose machine learning algorithms that lend themselves to simpler and more understandable algorithms.

There were two other issues with the way the participants used the system, both of which were compatible with Fiebrink et al.'s results [5]. Firstly, many participants did not use an iterative strategy. This is a big problem as it suggests that players are unlikely to adapt their customization if it does not work the first time, resulting in dissatisfaction with the experience. This indicates that players need support in the process of IML. This result supports the first conclusion in that the purpose of the graphical interfaces should be to guide players in using an iterative process, rather than visualizing the end result. The other issue is that participants used relatively small numbers of training samples. This again is a problem for most machine learning algorithms which rely on large amounts of data. This supports the second conclusion in that we may need to use different machine learning algorithms; ones which require less data, such as nearest neighbor methods, but which may also be easier to visualize. Our results therefore suggest at least two avenues for future research in further developing IML.

In summary, customizing by doing is a promising approach to customizing video game characters that can support players in simple customization tasks such as altering animation but potentially also in more complex tasks such as changing characters' responses to diverse situations. Our study has shown the potential of the approach while identifying issues that suggest fruitful avenues of future research.

ACKNOWLEDGMENTS

This research was supported by the UK Engineering and Physical Science Research Council grant "Performance-based Expressive Virtual Characters". We would like to thank the staff and students of the Embodied Audio-Visual Interaction laboratory and the Goldsmiths' Digital Studios.

REFERENCES

1. Bianchi-Berthouze, N., Kim, W.W., and Patel, D. Does body movement engage you more in digital game play? And why? In *Proc of the Int'l Conf on Affective Computing and Intelligent Interaction,* (2007), 102-113.
2. Dourish, P. *Where the Action is: The Foundations of Embodied Interaction*. MIT Press, Cambridge, MA, USA, 2001
3. Fails, J.A. and Olsen, Jr., D.R. Interactive machine learning. In *Proc of the Int'l Conf on Intelligent User Interfaces,* (2003), 39-45.
4. Fiebrink, R. *Real-Time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance.* PhD thesis, Princeton University, Princeton, NJ, USA, January, 2011.
5. Fiebrink, R., Cook, P.R., and Trueman, D. Human model evaluation in interactive supervised learning. In *Proc CHI 2011,* ACM Press (2010), 147-156.
6. Gillies, M. Learning finite state machine controllers from motion capture data. *IEEE Transactions on Computational Intelligence and AI in Games, 1,* (2009), 63-72.
7. Gratch J. and Marsella, S. Tears and fears: modeling emotions and emotional behaviors in synthetic agents. In *Proc of the 5th Int'l Conf on Autonomous agents*, ACM Press, New York, NY, USA 2001, 278–285.
8. Hummels, C, Overbeeke, K. C. and Klooster, S. Move to get moved: A search for methods, tools and knowledge to design for expressive and rich movement-based interaction. *Personal Ubiquitous Comput. 11*, 8 (2007), 677-690.
9. Kapoor, A., Lee, B., Tan, D., and Horvitz, E. Interactive optimization for steering machine learning. In *Proc of the SIGCHI Conf on Human Factors in Computing Systems,* (2010), 1343-1352.
10. Klemmer, S. R., Hartmann, B. and Takayama.L. How bodies matter: Five themes for interaction design. In Proceedings of the 6th conference on Designing Interactive systems. ACM Press, New York, NY, USA 2006, 140-149.
11. Kwon, T., Cho, Y.S., Park, S.I. and Shin, S.Y. Two-character motion analysis and synthesis. *IEEE Transactions on Visualization and Computer Graphics, 14,* 3, (2008), 707–720.
12. Lee, J. and Lee, K.H. Precomputing avatar behavior from human motion data. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, (2004), 79–87.
13. Sergey Levine, Philipp Krähenbühl, Sebastian Thrun, and Vladlen Koltun. 2010. Gesture controllers. In ACM SIGGRAPH 2010 papers (SIGGRAPH '10), Hugues Hoppe (Ed.). ACM, New York, NY, USA
14. Moustakis, V. Do people in HCI use machine learning? *HCI, 2,* (1997), 95-98.
15. Neff, M and Kim, Y. 2009. Interactive editing of motion style using drives and correlations. In Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '09), Dieter Fellner and Stephen Spencer (Eds.). ACM, New York, NY, USA, 103-112.
16. Han Noot and Zsófia Ruttkay. 2005. Variations in gesturing and speech by GESTYLE. Int. J. Hum.-Comput. Stud. 62, 2 (February 2005), 211-229
17. Patel, K., Fogarty, J., Landay, J., and Hariison, B. Examining difficulties software developers encounter in the adoption of statistical machine learning. In *Proc 23rd AAAI Conf on Artificial Intelligence,* AAAI Press 2008, 1563-1566.
18. Prendinger, H., Ullrich, S., Nakasone, A. and Ishizuka, M. MPML3D: Scripting agents for the 3D internet. *IEEE Transactions on Visualization and Computer Graphics, 17*, 5, (2011), 655-668.
19. Quinlan, J.R. Induction of decision trees. *Machine Learning, 1,* 1, (1986), 81-106.
20. Taylor, G. W. and Hinton, G. E. 2009. Factored conditional restricted Boltzmann Machines for modeling motion style. In Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09). ACM, New York, NY, USA, 1025-1032.
21. Talbot, J., Lee, B., Kapoor, A., and Tan, D.S. EnsembleMatrix: Interactive visualization to support machine learning with multiple classifiers. In *Proc of the SIGCHI Conf on Human Factors in Computing Systems,* (2009), 1283-1292.
22. Vilhjalmsson, H. Augmenting Online Conversation through Automated Discourse Tagging. In *Proc of the 6th Annual Minitrack on Persistent Conversation at the 38th Hawaii Int'l Conf on System Sciences*, (2005)
23. Vinayagamoorthy, V., Gillies, M., Steed, A., Tanguy, T., Pan, X., Loscos, C. and Slater, M. Building expression into virtual characters. In *Eurographics Conference State of the Art Reports,* (2006).

VITAE

1.Andrea Kleinsmith

Andrea Kleinsmith is a post-doctoral researcher in the Department of Computing at Goldsmiths, University of London, UK. Dr. Kleinsmith received a PhD degree in computer science in the area of affective computing from UCL, UK in 2010. Her main research interests are in affective human computer interaction and automatic affect recognition systems with a focus on body posture and movement.

2.Marco Gillies

Marco Gillies is a Senior Lecturer in the Department of Computing at Goldsmiths' University of London where he is a founder and co-director of the Embodied Audio-Visual Interaction research group. Before joining Goldsmiths' he obtained a PhD from the University of Cambridge and was a post-doctoral research at University College London. His research centers on animated virtual characters and full body interaction.