UNIVERSITY OF AMSTERDAM

Gaming with the Mind: creating a Virtual Reality environment for action prediction using EEG signals

Project team: Luca Simonetto, Nedko Savov, Jorik Spijkerman
Supervisor: Dr. E. Gavves

October 15, 2017

Abstract

This project aims at exploring alternative ways of sending inputs to a simulated environment, namely by using a brain-computer interface based on neural oscillations captured by an electroencephalogram (EEG). Hereby we consider the challenges caused by using this kind of input when a Virtual Reality (VR) environment is used and collecting data for its implementation. First, a Virtual Reality game environment is implemented with a predefined set of actions allowed for the user. Then, an experimental setup is implemented within the VR framework meant to be used when collecting data for the brain interface model. The Brain Computer Interface team attempts to create model finding correspondences between brain signals and actions, which to be able to be used real-time as a controller in the game. It is described in [14]. Finally, the collected data is analyzed for actions usage and correlations, which can be used to improve the action predictions.

1 Introduction

Since the 1970s, with the surge of personal computers, developers have tried to model [6] and revolutionize the way humans interact with computers. One way is to relieve the limitation of physical motor skill and use direct input from the brain. This is achieved by the creation of the Brain-Computer interface (BCI). The main interest of the BCI systems lies within its application to provide those who are motor disabled with a way of communication [16]. One of the major challenges is to have a non-invasive way of establishing a connection (or interface) between brain and computer. This is done by using an electroencephalogram (EEG) to measure brain activity in the form of neural signal oscillations. Provided that the EEG output is independent, this becomes an interesting tool as a form of input for the computer system. In recent years, the BCI approach has been applied to aid rehabilitation and recovery, as a direct quantification of the functions of the cerebrum that are otherwise concealed [11]. Additionally, the EEG has been used in combination with direct external feedback through the use of a robotic apparatus to determine the effectiveness of this motor imagery (the internal image of a person enacting motor control, without motor output) [3, 2].

The ultimate goal of BCI systems is to allow complete interaction with the environment by solely using the output of the brain. An example of this is to use EEG-controlled robots as a means of interaction. These robots can then be the manifestation of the intentions of the controller [5]. However, as these robots can directly interact with the real-world environment, an intermediate environment can be used to provide a safe setting. It is often a dependent input, namely it needs some form of motor output to support the controls of the actions [4] or through an event-related potential. Although these show user satisfaction, often times the user will be frustrated by the fact that an action did not have the intended result [8]. As the aim is to have a completely independent input basis for the BCI system the challenge arises of proper classification of the BCI system, which was already a strain on the dependent systems. Nevertheless, a simulated environment that can harbor a variety of actions is of interest when this classification strain is overcome.

To build such an environment the following functionalities have to be taken into account:

- *Communication* between the brain and computer environment.
- *Accessibility and learnability*, the ability to easily learn, operate and navigate the environment.

Differences in BCI approaches determine the implementation of the proposed platform. The intended purpose of the BCI system is to be used in an open environment where the user can be prompted to perform a specific action, for training purposes. Eventually, the user must be able to freely use any action at any time through motor imagery (Active BCI). Numerous studies have already implemented this control paradigm, both in 2D and 3D [1].

This report is organized as follows: first, in section 2 the pipeline of the project is described and the tasks associated with each team are presented. In particular, section 2.2 describes the technical details regarding the work done by the team creating the virtual reality environment. Section 3 defines the experimental setup for collecting data required for predicting the user's actions and section 4 shows the results obtained by the experiments done. Section 5 explains the intended behavior of the final system, in order to output predictions from the user thoughts, and section 6 explains the major obstacles emerging by the use of a Virtual Reality headset. Lastly, Section 7 discusses about the collaboration between the VR team and the KPMG experts and Section 8 concludes this report, summarizing the work done in this three weeks period.

2 Project setup

This project can be decomposed in two separate components, namely virtual reality environment and brain computer interface, allowing two teams to build their respective frameworks with a relatively low amount of interaction from each other. After each framework is completed ,the final merged model can be built and used for data collection, with the objective of generating a large enough dataset that can be used to train a machine learning model that is able to map in real time the signals coming from the user's brain to in-game actions. As a large amount of recordings are needed for this task, external participants have been asked to use the system under the two teams' supervision, where their brain activity and physical response can be recorded.

2.1 Brain Computer Interface

The objective of having a brain computer interface is to be able to utilize the user's brain activity as a way to decide which action the virtual reality character should perform: for this to work multiple steps are needed, namely data collection, signal analysis and action extraction.

The data collection phase requires the user to wear a wireless electroencephalogram device (EMOTIV Epoch+ [1]) that is able to detect variations in different signals emitted by the wearer's brain up to 128 times a second: the 14 channels provide a the capability of detecting variations on every region of the brain and the resolution of the readings is high enough to provide good feedback on the brain activity and user head movements.

Having recorded enough data from the EEG device, the next step is to analyze the created dataset using machine learning techniques and algorithms that are able to cluster the input signals and remove noisy readings. This step is done by using a variety of techniques ranging from clustering and dimensionality reduction to variational autoencoding, in order to create a dataset that is qualitatively adequate for the scope of the project.

[1] https://www.emotiv.com/epoc/

Finally, the last step is to build a model that is able to predict the action the user wants to be done by using real time readings, trained on the above mentioned dataset.

A detailed description of the work done in this area can be retrieved in the companion report[14].

2.2 Virtual reality environment

After having defined the main BCI tasks and objectives, the environment specifications have been set. No existing games or simulations have been seen as appropriate, as the needs of specific timestamps recording, custom actions and user specifications restricted the search too much to find an existing work, so a completely new framework has been requested.

Developing framework and VR headset

First, as the user immersion was set as a priority, a virtual reality approach was chosen: this required the choice of what would be the most appropriate framework used to develop such environment along with the choice of what virtual reality headset to use. Regarding the developing framework, Unity [2] seemed the most appropriate, as compatible with most VR headsets, moderately easy to develop with and free of charge for non-commercial projects. The headset chosen, given the restricted time slot for this project and limited funds, was the Oculus Rift (DK 2) [3] device, as a multiple number of units were already in use by the proposing institute eliminating the need of a new purchase.

Secondly, the main aspects of the environment have been defined: a relatively simple first person game, where the user can interact with the environment using predefined actions and is able to move at his will. Due to the time restrictions imposed by this project a simple but effective type of environment has been selected, where each entity to be interacted with is a simple cube. This idea is based on the success of a popular game titled Minecraft [4], where its simplicity enables both to keep the user engaged using his creativity and eases the development of the whole game.

Player actions

As the EEG device used to record the user's brain activity has limited power and the amount of data required to classify tens of actions is huge (given the noise in the readings and moderate resolution) it was decided to set a defined number of actions that the user will be able to think about and have the virtual character execute: those actions are nine and listed below:

- *Jump*: the character jumps in the air and lands on the ground right after.
- *Walk*: the character walks forward. Other similar actions, like *Strafe Left*, *Strafe Right* or *Walk Backward* have not been used, as the conceptual similarity between them would likely result in trouble detecting the correct action to execute.
- *Run*: when paired with *walk* the character runs forward, at increased speed.
- *Crouch*: when executed, if not crouched, the character lowers the stance and stands closer to the ground.
- *Stand up*: if crouched, the character raises to normal standing position.
- *Destroy block*: while looking at a block, using this action the character is able to destroy it.
- *Place block*: while looking at a suitable spot on the game world, the character is able to place a block of choice.

[2] https://unity3d.com/
[3] https://www3.oculus.com/en-us/dk2/
[4] https://minecraft.net/en-us/

- *Turn left*: turns the character to the left.
- *Turn right*:turns the character to the right.

Controllable characters

Having defined both the tools to be used and the actions available, the implementation of the virtual reality environment can be started. Given that only one VR headset was available, not every participant may be able to sustain elongated periods using playing in VR (dizziness, discomfort etc.) and there could be the need of switch from VR to non VR mode while in game, a decision was made to create two different characters that can be controlled with and without the use of a VR headset (one at a time). This decision posed some new challenges, as ensuring switching from one character to the other at any time required precise coordination with all the rest of the components of the environment. Each character has a *player controller* that allows movement, look, jump etc., an *action controller* that incorporates the logic behind block placing and destroying along with interface updates and a *GUI module* showing on-screen information and action requests when needed.

player controller: in order to ease the development, both VR and non VR controllers were downloaded from the assets store coming with Unity and were heavily customized in order to be able to work with our framework. All actions were remapped to a common interface (as each controller was relying to a third party input manager) and actions not present in the controllers have been added (such as crouching and standing up).

action controller: this script is responsible for all the actions regarding interaction with the external world and triggers changes by sending commands to the main game controller. It is also responsible of updating the user interface, by indicating which block is selected for placing (from a list of selectable blocks), showing on screen instructions and indications on which button to press for each specific action. Regarding interface updates, more details are given in section 3.

GUI module: the main user interface, composed by four distinct elements:

- *Action request*: text displaying an action that the user is required to take.
- *Joystick mapping*: Picture of a controller with an highlighted button associated with the current action to be performed.
- *Block selection*: in order to know which block is currently selected for placing next.
- *Score*: updated when the user performs an action that gives points, to be used if a future gamification of this project is to be implemented.

The result of this implementation can be seen in Figure 2.2, where every above mentioned component is displayed.

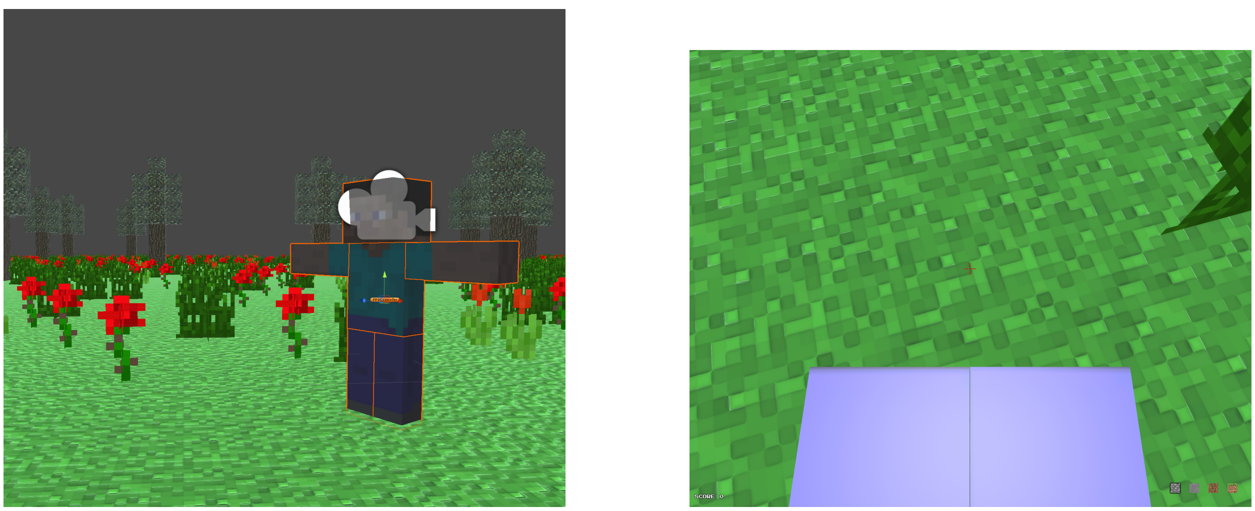
Player model

The effectiveness of using games and a virtual reality has proven significant for rehabilitation [12, 13]. In order to facilitate the potential of rehabilitation through mirror box therapy [9], a character model was implemented. At first a character was tried that was consistent with the environment that was established, namely *"Steve"* as shown in figure 2a. However, as the goal is to have a high level of immersion, this model was unsuited because of the top-down view provided little visual feedback (figure 2b). Additionally, the implementation had difficulties animating the model, for which both Blender [5] and 3ds Max [6] were used to create and improve the character animation (to little success). Unity's asset store provided a large library of motion capture data, which naturally tends to be realistic humanoid motion and animation[7]. Using this

[5] `https://www.blender.org/`
[6] `https://www.autodesk.com/products/3ds-max/overview`
[7] `https://www.assetstore.unity3d.com/en/content/19991`

Figure 1: Implemented environment, showing the interactive environment and the on-screen GUI.

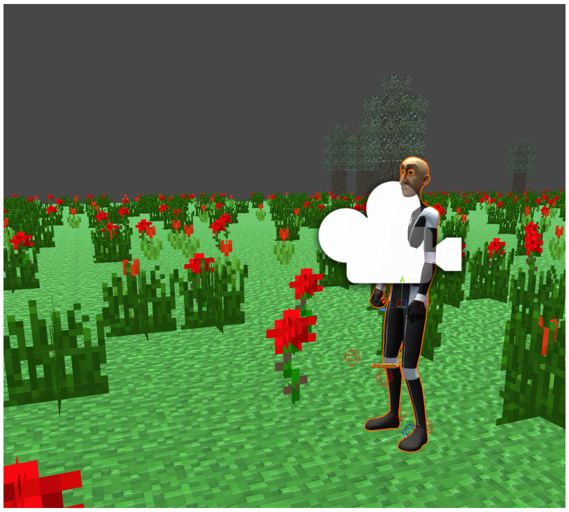
library provided the immersion that was intended. The model itself is more representative of a human, unlike *Steve*, as can be seen in figure 3a. While also having proper visual feedback when looking down, as can be seen in figure 3b.



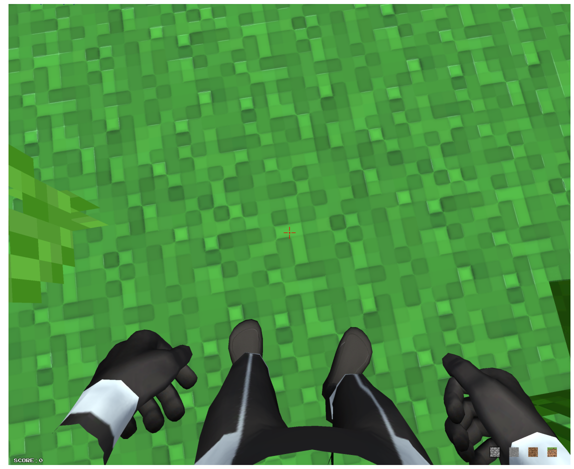
(a) Minecraft Steve character.

(b) Looking down from the Steve perspective.

Figure 2: Minecraft consistent character.



(a) Motion capture character.



(b) Viewing down from the motion capture character perspective.

Figure 3: The motion capture character.

3 Experimental setup

Here are presented the details about setting up the framework for the experiments and data collection used by both teams.

The major necessity of the model to be built by the brain computer interface team is having a large amount training data available. This data is in the form of EEG readings of participants collected while they are executing specific actions within the VR game. To address this need, a framework for the experiments and data collection has been implemented. One of the main issues that arise is synchronization between the collected EEG signal and the actions performed. This is done by storing timestamps at the key moments, which can be then used to slice the EEG readings and isolate only the data of interest.

The data collection process consists of three separate parts, each of which was implemented as a separate mode using the implemented VR game in a different way.

Controller input instructions mode: the first mode consists of performing a sequence of random actions drawn from a uniform distribution of the possible 9 actions. This sequence is generated at the beginning of the experiment together with specified randomized time periods used by timers. These parameters makes it also possible to calculate roughly the amount of time needed for all the actions, which helps to coordinate with a provided as parameter total time for this part of the experiment. The participant is asked to perform the action appearing in an on-screen message. The participant thinks about the action and performs it by a keypress. The data collected by this mode, along with EEG recordings, is in the form of three timestamps - when the message appears, when the key is pressed and when it is released. The period between the second and third timestamps is considered the one when the person actively thinks about the action and performs it, while the period between the first two is when the person makes sense of the action and starts thinking about it. Both can be used to provide valuable information. In this mode each action is executed only once, so only one keypress is recorded while giving a time limit for holding the key. After key release there is a wait period before the next action, It has a randomized length in seconds, determined by a given range - for the performed data collection 3 seconds of waiting were used every time. The purpose of this is for the participant to stop thinking of the previous action, putting his mind in a neutral state. This would result in prevention of EEG readings of brain activity caused by overlapping action thoughts. On Fig. 2.2 is shown a screen from playing in this mode - instructions are visible on top.

Input simulation mode: this second part aims at collecting data which is as close as possible to the EEG data that would be read in real-time with a working BCI controller for the game. To achieve that, a realistic simulation of a working BCI controller was implemented, that performs the on-screen instructions without the need of external input. Here the participants are again asked to execute an action from a randomly generated sequence of actions in turn. However, they are told that they are working with a working BCI controller and in this part they will only use their thoughts to execute the action from the instructions. A few seconds after the shown instruction, the displayed action is automatically performed for a specified amount of time. The automated actions are implemented by simulating key presses from the keyboard and mouse mappings of the actions. However, to make the controls more believable, a component for error generation was implemented. At every ten frames, it is decided with a chosen probability if an error should be initiated or not. If so, it is decided with another specified probability if nothing should be done or a wrong action should be performed. In the latter case a simulated keypress for a random wrong action, drawn from a uniform distribution, is performed. In any case, the time for the error is randomly chosen between 100 and 500 ms. After some tuning, the probabilities used in the data collection were 0.3 for an error and 0.5 for a wrong action. Here timestamps are collected when the message appears, when the simulated action starts and when it ends (errors are included in this time period).

Free play mode: with the first two experiments, the player receives instructions for simple actions that he has to execute. However, this is not the usual case during typical gameplay: the player usually gets to choose his own actions and is not bounded by requests. This is why a third experiment is performed, in which a player can choose to do whatever he likes in the game. The data collection from this mode has been implemented by collecting two timestamps for each action executed, along with the type of the action. Here, a combination of actions can be performed, and all of the actions will be recorded separately. This could cause difficulties with the classification of EEG data, but would also provide more realistic data readings. We also use the data collected from this mode to determine correlation and usage of actions, since it reflects the most an actual gaming session.

4 Experimental results

Here an analysis of data obtained from the free play mode is performed. It has been collected from 22 subjects where each one of them performed two sessions of 2 and 5 minutes. First, a more broad analysis is done by checking actions usage, followed by a more in depth study about the correlation between actions.

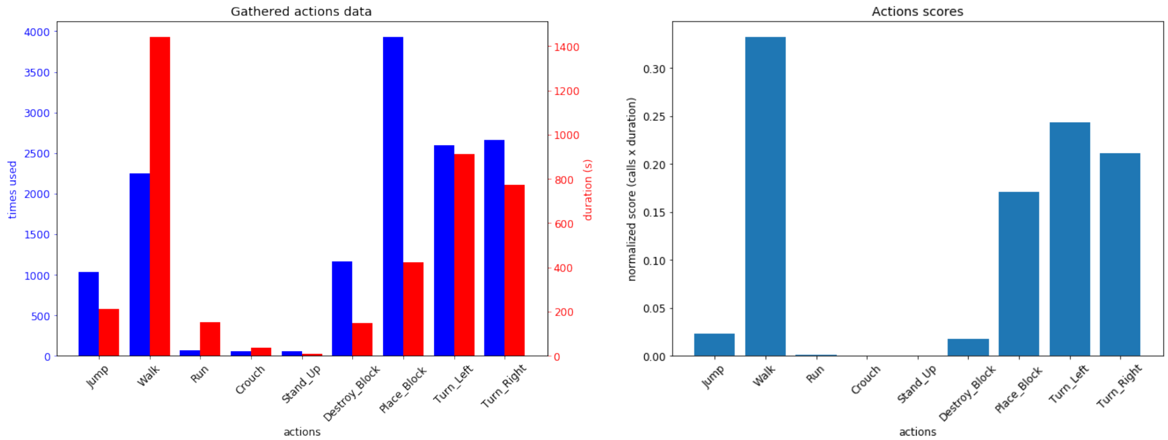
Actions usage

An interesting analysis that can be done on the gathered data is determining the usage of each input given by the user, corresponding to one of the 9 actions allowed: as the actions number and types were chosen without prior studies, some important results could arise, suggesting future improvements in the created framework.

This study has been done by merging all the recordings gathered from the free play sessions in a list of actions with the respective timestamps. Grouping this data by action results in the plots in Figure 4, where on the left frequency and duration of actions is plotted and on the right a normalized score for every action (normalized frequency $\times$ duration) is calculated.

Looking at the Fig 4a, it can be seen that:

- The *Walk* action is the one done for the most time, totaling almost than half of the combined time of the recordings.
- *Run*, *Crouch* and *Stand Up*, on the contrary, are the three actions that are done the least amount of times and for the lowest amount of time. If a model is trained on this dataset, the quality of the predictions for this three actions will be very poor, resulting in the need of more of this specific data or different tasks required from the user during data collection.
- *Place block* is the action done the largest amount of times, indicating that the user is concentrated on building when in free play mode, something to be expected.



(a) frequency and duration of actions

(b) normalized action scores

Figure 4: Usage analysis of the implemented actions.

- *Turn Left* and *Turn Right* are balanced between each other, indicating that there is not prevalence of one action w.r.t. the other.

Fig. 4b shows even more clearly how there are three actions that have almost non-existent importance on the overall dataset. With this in mind, it can be concluded that the current environment and data collection approach is not optimal as the disparity in actions usage is very high, resulting in a very unbalanced dataset: the first most straightforward way of solving this issue would be to completely remove the almost non-occurring entries from the list of allowed character actions, reducing the number of classes to predict along with having a more balanced dataset. In the case that those actions are needed for the final user, the environment or the free play mode need to be changed, involving the user to do these actions more frequently.

Actions Correlation

Correlation between actions can be a useful tool to get insight on what the typical combinations of actions are and what action can be expected given the previous ones.

Again, the timestamp data collected from Free Play was used. This data can be thought as a timeline with a channel for each action. An example is given on Fig. 4. Each action is treated as a variable. We look for correlations between all the 9 variables. To compute it many observations of the variables are needed. To get these observations the data timeline is sampled at specific time positions. At each, a state of the actions is defined as a binary array, having 0 for an action which is not observed at this moment and 1 otherwise.

It is important how the states are sampled, since this directly determines the observation data which is correlated. A possibility would be uniform sampling. However, then the interval over which sampling is done is very important. If it is too big, fast actions like creation and destroying blocks would be skipped over. If it is too small then long actions like walking would have a larger influence in the correlation. To remove the influence of the length of an action, a different approach is taken. Only the states where there was a change from previous states are chosen. They are visualized as vertical lines on Fig. 4.

First, an attempt was made to find correlations between simultaneous actions. In specific, the Spearman rho correlation was used. It is a measure of rank correlation, which doesn't assume continuous variables for the source data or a specific distribution which generates it [7]. Since our observations are counts and there is no distribution to assume, this is a relevant choice. Goodman and Kruskal's Gamma was also considered, since it is meant specifically for categorical data. However, it's calculation with large amounts of observations

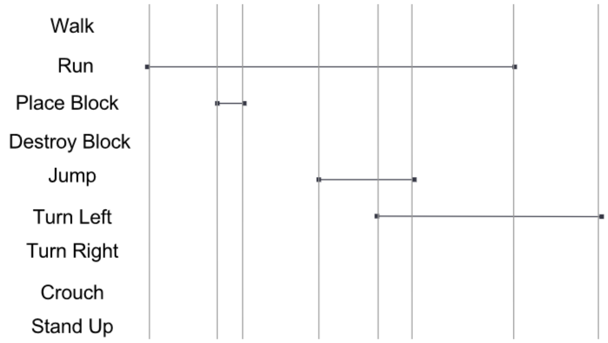


Figure 5: Free Play data for placing and jumping over a block viewed as a timeline. Vertical lines show the sampling chosen for getting observations used to estimate correlation matrices.

turned out to be infeasible and we turned our attention to Spearman rho. Individual states were treated as observations. Results are visible on Fig. 4.

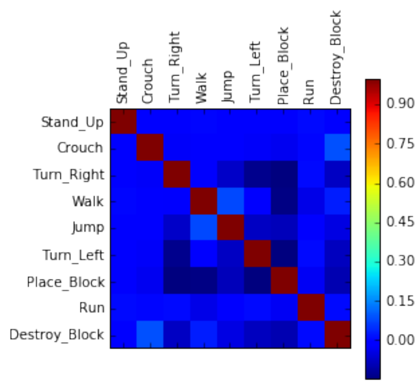
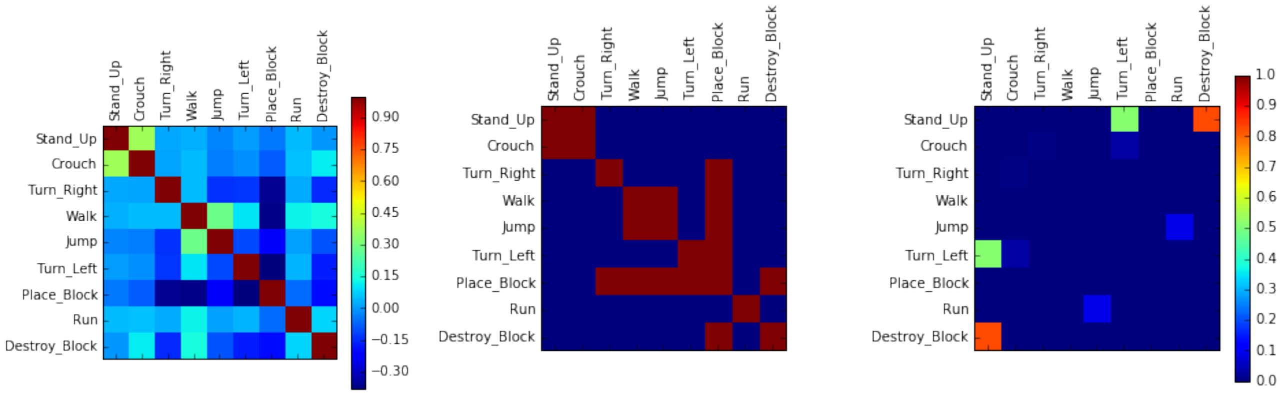


Figure 6: Spearman rho correlation matrix for simultaneous actions.

Even though a small correlation between turning and placing block is present, all the values are too close to zero to differentiate a significant correlation. The reason is most likely the small amount of different values that can be taken by observations.

Next, a correlation is estimated between the number of actions within a close proximity of each other. Observations in this case are counts of actions estimated in a window of 10 actions. From one window one observation for each of the actions is obtained. To obtain multiple observations, a sliding window is used, leading to 27 236 observations for each action. It is important to note that the window is slided through each Free Play session and then concatenating the accumulated data. This prevents from correlating action from the end of one session and beginning of a new one. The accumulation of states solves the problem encountered in the estimation of the previous correlation. A small window size is chosen as to address the temporal locality which is desired, so as to improve predictions given from the BCI. Results are shown on Fig. 7a.

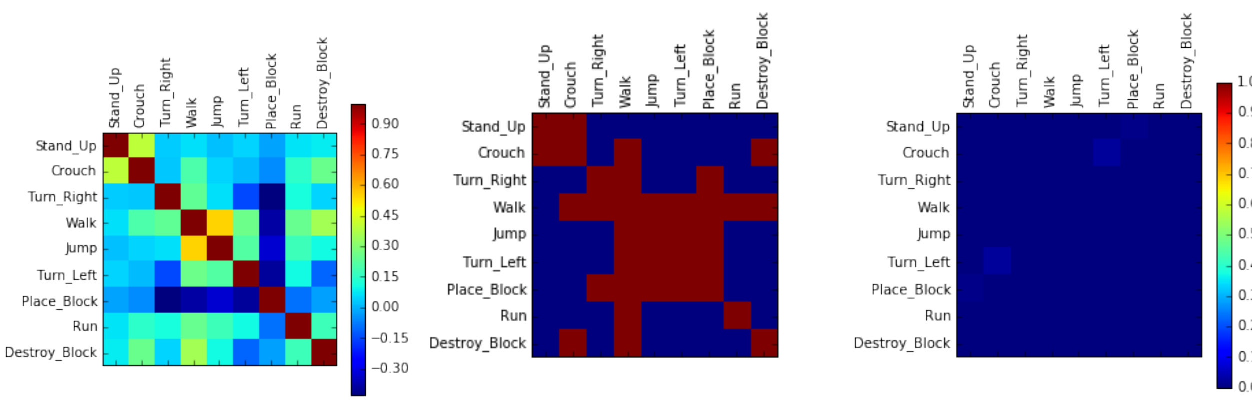
It can be seen that a medium correlation is present between Stand Up and Crouch and between Walk and Jump. Also, weaker correlation is found between Place Block and Walk, Turn Left and Turn Right, Jump and Destroy Block. They are better seen on Fig. 7b where the correlation values are thresholded with the values 0.2 and -0.2, which supposes at least a weak (positive or negative) correlation between the actions. The correlated actions are marked with red. To examine how certain the found correlation coefficients are, their p-values were found. They are shown on Fig. 7c. The larger the p-value, the more uncertain that there truly exists a correlation between the variables despite a large correlation coefficient found. It can be



(a) Spearman rho correlation matrix (b) In red are weakly correlated actions, as opposed to blue ones (c) Correlation p-values

Figure 7: Spearman rho correlation matrix for a narrow window of 10 actions.

seen that the correlation of Stand Up and Destroy Block, also Stand Up and Turn Left is very uncertain. Additionally, a long term temporal correlation was estimated by just increasing the size of the window. Results, as shown on Fig 8 suggest significant increase in the reported values in a longer considered time period. However, short periods are more suitable for our purposes and so we stick with the previous results for the next section.



(a) Spearman rho correlation matrix (b) In red are weakly correlated actions, as opposed to blue ones (c) Correlation p-values

Figure 8: Spearman rho correlation matrix for a wide window of 100 actions.

5 Improving Action Prediction

The final game is planned to work by first using the EEG headset to make readings, then passing the data to a classification model, designed by the BCI team. This model would return a list of probabilities representing the user's action choice - one probability for each action. In the game, the action with the highest probability can be then chosen to be performed.

However, the classification model does not consider the in-game environment. This means that the predictions can be further improved by using the correlation coefficients found (the ones estimated on close proximity actions - small window size). More precisely, each probability can be multiplied by the correlation coefficient of its corresponding action and the previously executed action. By doing this , the less likely actions in combination with the previous one are penalized while the more likely ones have more advantage of being chosen. Additionally, environment based restriction of actions can be applied. Space position information can be used to determine if an action would make sense in a specific situation. For example starting to run while being in front of a wall is highly unlikely and the action with the next highest probability can be chosen.

The time provided for the project proved to be insufficient for getting to a stage where a well working classifier could be finished by the BCI team. The lack of a model prevented us from implementing an input connection to the VR game with such a model and also directly applying this prior information, which still remain theoretically promising.

6 Virtual Reality related issues

One major feature that separates this project from the others is the use of a virtual reality environment: allowing the user to freely look around without the press of a button results in a great immersion, much more enjoyable than the common on-screen play. In this way the user is more involved in the experience and the degree of interaction should be much higher: this has been one of the main requisites. However, after the full project has been completed, major issues stood up regarding VR.

One big limitation of this work is the need of using a virtual reality headset: a significant amount of time has been spent in the development of a virtual reality environment, that poses multiple problems related to the use of the headset. The need of having two separate characters in order to use the environment not in VR required strong communication between the various controllers implemented, resulting in a lot of work just to have this feature working.

Another side effect of using VR is the large amount of noise in the EEG readings resulting from the muscular activity of the head when turning: this problem is alleviated by asking the participant to limit the head movement during the experiment but nullifies the need of having a VR headset for increased immersion. A change that will allow greater performance of the overall system would be to completely remove the virtual reality aspect from the environment, at the cost of a much less immersive experience.

Ultimately, a lot of users experienced discomfort and dizziness during the data gathering phase, mostly associated to the fact that it was their first time using a VR headset. Lack of optimization done in environment to assure velocity in animations and sufficient framerates also contributes for that.

7 Collaboration with KPMG

The following section expands upon the collaboration between the BCI-VR development and the KPMG experts, to establish footholds on the pillars of the KPMG control framework.

7.1 Data governance

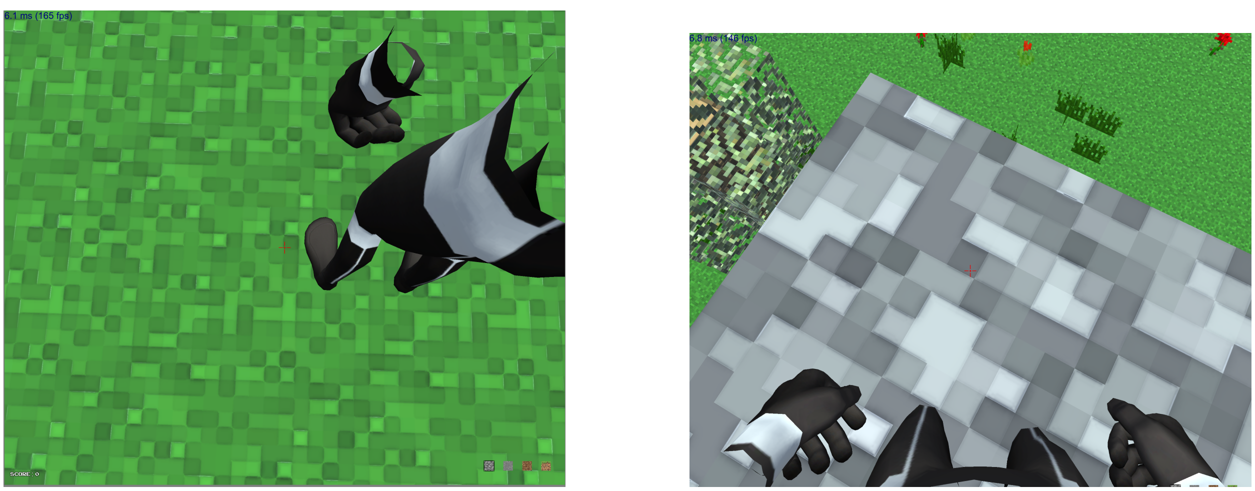
Data governance can be split up into multiple domains, i.e., data principles, data quality, data access, meta data, and the data lifecycle [10]. The main focus lies within the data quality in this project and will be elaborated on, lightly touching upon some of the other domains. More specifically data quality becomes the determination of requirements to set for the data that is needed for the project. In combination with the KPMG experts it became apparent that in order to preserve data quality it is necessary to determine the quality of the data pipeline itself as well. Furthermore, predefined data selection rules aid in specifying the range of the data, having a more controlled input to the pipeline.

In accordance with the ISO standards[8], inherent data quality has five attributes, i.e., accuracy, completeness, consistency, relevancy, and timeliness [15]. As is the case with this project timeliness and relevancy are intrinsically dealt with, as the data is self-acquired it is up-to-date and its relevancy as being necessary for the overall BCI. However, accuracy, completeness, and consistency are not as straightforward. Accuracy is relatively easy to determine, the actions performed provide data directly about this measure namely the action itself and every input always provides the corresponding output. However, when a model is to be trained on this set, it is important to note the non-uniform distribution of actions. This might lead to a bias towards actions that are relatively frequent as opposed to those that are not. As stated in section 4 actions can have a duration and a frequency. Together with the time stamps and action label, these features encompass what it means to perform an action. All of the data of the free play sessions was stored successfully, which accounts for the completeness of the data. The consistency of the data can only be checked in reference to intuition, namely that the data attributes do not contradict one another.The previously mentioned attributes are distinct but do support one another making these attributes coherent and thus they are consistent.

7.2 Effectiveness

To understand how well the system or virtual environment performs the system can be tested for sensitivity. The system was built from the ground up and actions and functionalities were incrementally added. Each of the added actions then had time to be tested and fixed if necessary.

Two behaviors were observed that were counter intuitive but not detrimental, namely the placement range of the blocks and the placement of blocks underneath the player that might represent the block being place on a space where the player would occupy space as well. However, with increased frequency of the block placing action underneath the player, while jumping repeatedly, might actually entomb the player in the placed blocks, leaving the player to only be able to look around and no longer move. This behavior was not exhibited by any of the participants. Additionally, repeatedly issuing the same movement tends to misalign the character model with the camera, shifting the character model in that direction.



(a) Mis-aligned character model.

(b) Stuck player.

Figure 9: Unintended behavior.

8 Conclusions

This project displays how the problem of building a virtual environment to be used for action prediction has been tackled in the span of three weeks, the problems encountered and the solutions adopted. Some

[8]ISO 25012

insights are given regarding the type and number of actions that the user is allowed to perform, allowing future implementations to be less prone of overlooking some important problems that would arise after full development.

It has been discovered that the addition of a virtual reality headset is not to be recommended for the use of this kind of BCI, as the downsides of using it are greatly exceeding the upsides. In order to be able to fully utilize this technology prior research has to be done, and the results of our work can help future projects.

Finally, it has been proven that a small team of students are able to produce viable results if the correct tools are provided and used, implying that future work on this area could direct most work in developing a suitable prediction framework to be used in the implemented environment.

References

[1] Minkyu Ahn, Mijin Lee, Jinyoung Choi, and Sung Chan Jun. A review of brain-computer interface games and an opinion survey from researchers, developers and users. *Sensors*, 14(8):14601–14633, 2014.

[2] Kai Keng Ang, Karen Sui Geok Chua, Kok Soon Phua, Chuanchu Wang, Zheng Yang Chin, Christopher Wee Keong Kuah, Wilson Low, and Cuntai Guan. A randomized controlled trial of eeg-based motor imagery brain-computer interface robotic rehabilitation for stroke. *Clinical EEG and neuroscience*, 46(4):310–320, 2015.

[3] Kai Keng Ang, Cuntai Guan, Kok Soon Phua, Chuanchu Wang, Longjiang Zhou, Ka Yin Tang, Gopal J Ephraim Joseph, Christopher Wee Keong Kuah, and Karen Sui Geok Chua. Brain-computer interface-based robotic end effector system for wrist and hand rehabilitation: results of a three-armed randomized controlled trial for chronic stroke. *Front. Neuroeng*, 7, 2014.

[4] Abdelkader Nasreddine Belkacem, Supat Saetia, Kalanyu Zintus-Art, Duk Shin, Hiroyuki Kambara, Natsue Yoshimura, Nasreddine Berrached, and Yasuharu Koike. Real-time control of a video game using eye movements and two temporal eeg sensors. *Computational intelligence and neuroscience*, 2015:1, 2015.

[5] Luzheng Bi, Xin-An Fan, and Yili Liu. Eeg-based brain-controlled mobile robots: a survey. *IEEE transactions on human-machine systems*, 43(2):161–176, 2013.

[6] AA Clarke. A three-level human-computer interface model. *International Journal of Man-Machine Studies*, 24(6):503–517, 1986.

[7] Atila Göktas and Öznur Isçi. A comparison of the most commonly used measures of association for doubly ordered square contingency tables via simulation. *Metodoloski zvezki*, 8(1):17, 2011.

[8] Elisa Mira Holz, Loic Botrel, Tobias Kaufmann, and Andrea Kübler. Long-term independent brain-computer interface home use improves quality of life of a patient in the locked-in state: a case study. *Archives of physical medicine and rehabilitation*, 96(3):S16–S26, 2015.

[9] Neethu Jose. Mirror box therapy. *Int. J. Adv. Nur. Management*, 2(2):97–99, 2014.

[10] Vijay Khatri and Carol V Brown. Designing data governance. *Communications of the ACM*, 53(1):148–152, 2010.

[11] Floriana Pichiorri, Giovanni Morone, Manuela Petti, Jlenia Toppi, Iolanda Pisotta, Marco Molinari, Stefano Paolucci, Maurizio Inghilleri, Laura Astolfi, Febo Cincotti, et al. Brain–computer interface boosts motor imagery practice during stroke recovery. *Annals of neurology*, 77(5):851–865, 2015.

[12] Gustavo Saposnik, Robert Teasell, Muhammad Mamdani, Judith Hall, William McIlroy, Donna Cheung, Kevin E Thorpe, Leonardo G Cohen, Mark Bayley, et al. Effectiveness of virtual reality using wii gaming technology in stroke rehabilitation. *Stroke*, 41(7):1477–1484, 2010.

[13] Joon-Ho Shin, Hokyoung Ryu, and Seong Ho Jang. A task-specific interactive game-based virtual reality rehabilitation system for patients with stroke: a usability test and two clinical experiments. *Journal of neuroengineering and rehabilitation*, 11(1):32, 2014.

[14] Nuno Mota Sindi Shkodrani Fabrizio Ambrogi Tom Pelsmaeker, Robin Bakker. Gaming with the mind: classifying brain activity of different mental actions recorded with eeg in a virtual reality environment. 2017.

[15] Kristin Weber, Boris Otto, and Hubert Österle. One size does not fit all—a contingency approach to data governance. *Journal of Data and Information Quality (JDIQ)*, 1(1):4, 2009.

[16] Jonathan R Wolpaw, Niels Birbaumer, Dennis J McFarland, Gert Pfurtscheller, and Theresa M Vaughan. Brain–computer interfaces for communication and control. *Clinical neurophysiology*, 113(6):767–791, 2002.