Learning Experiences in Programming: The Motivating Effect of a Physical Interface

Chris Martin[1], Janet Hughes[2] and John Richards[3]
[1]*Life Sciences - CITR, University of Dundee, Dundee, U.K.*
[2]*School of Computing and Communications, The Open University, Edinburgh, U.K.*
[3]*IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 and University of Dundee, Dundee, UK*
c.j.martin@dundee.ac.uk, janet.hughes@open.ac.uk, ajtr@us.ibm.com

Keywords: Programming, Physical Interface, Screen-based Interface, Learner Motivation, Emotional Response.

Abstract: A study of undergraduate students learning to program compared the use of a physical interface with use of a screen-based equivalent interface to obtain insights into what made for an engaging learning experience. Emotions characterized by the HUMAINE scheme were analysed, identifying the links between the emotions experienced during programming and their origin. By capturing the emotional experiences of learners immediately after a programming experience, evidence was collected of the very positive emotions experienced by learners developing a program using a physical interface (Arduino) in comparison with a similar program developed using a screen-based equivalent interface.

1 INTRODUCTION

This paper describes a study designed to explore how learning experiences are affected by learning with different interfaces: a physical interface or a screen-based equivalent. A recurring issue raised by the findings from previous work (Martin and Hughes, 2011) was the extent to which the physical artefact mattered. Given that there are several practical issues related to using physical equipment in a learning setting, such as cost, maintenance of equipment and fragility, the importance of the physical artefact does need to be explored.

The Whack a Mole study described here offered a comparison between two groups engaging in isomorphic learning experiences where the only difference was in the interface of the game they were programming. Evidence from Martin and Hughes (2011) suggested that an engaging learning experience led to a measurable change in a learner's knowledge. This new study, Whack a Mole, was designed to explore this further, attempting to capture insights into which emotions were experienced by learners and in what circumstances. It explored this in the context of a comparison between physical and screen-based media, aiming to answer the research question:

How does working with a physical artefact as opposed to a screen-based artefact affect learning of computer programming?

2 BACKGROUND

Anecdotal evidence from programmers suggests that programming is an emotionally rich experience: bugs are frustrating, trapping them can be satisfying, and solving complex problems can lead to increased pride in one's abilities. A further range of emotions can be evoked via collaborative working. Meyer and Turner (2002) describe the importance of emotion in an educational context. In education in general, emotional response to learning with technology has been studied for some time. D'Mello (2013) conducted a review including 24 studies, noting that many learning contexts resulting in engagement had comparatively low reporting of negative emotions. Pekrun (1992) conducted a detailed literature review from 1974 through to 1990, which was later extended to 2002 (Pekrun et al., 2002). This review included studies attempting to establish links between emotion and learning and achievement. Their review highlighted a bias in the research towards test anxiety: in excess of 1200 studies were found in this area, with other emotions receiving single digit or tens of studies at most. This reveals that broader emotion in an

education context was an understudied area. Pekrun et al. (2002) proposed a set of nine emotions in an academic setting that are linked to achievement and learning. As well as anxiety, these include emotions that are positive and negative, and activating and deactivating: *enjoyment, hope, pride, relief, anger, hopelessness, shame*, and *boredom*. The validity of this set of emotions and their link to learning and achievement was established through a number of studies utilising complementary research methods. Their findings imply students experience a wide range of emotions in an academic setting, with positive emotions represented in similar proportions to negative ones. Their findings also argue for emotion-oriented design of learning environments (Pekrun et al., 2002).

2.1 Emotional Response to Programming

There is a more limited body of work in the literature relating to emotional response to *programming*, although some interesting work has been done (e.g. Bosch & D'Mello, 2015; Bosch et al., 2013; Good et al., 2011). Bosch et al. (2013) sought to map the emotional states a novice experiences and their relative proportion, as well as explore the co-occurrence of emotional states and the relationship between interaction events. In addition, they mapped transitions between emotional states. They used participant self-reporting at a very high frequency, sampling every 15 seconds. Following a 30-minute programming exercise, the participant was shown a web camera still of their face and the programming tool they were using at 100 random points in the session. At each of these points, they are asked to note their emotional state and asked optionally to note a second emotional state. In this study, a number of emotions were offered to participants to select from: *fear, sadness, disgust, flow/engaged, anger, confused, uncertain, surprise, natural, frustration, boredom, happiness, curiosity, anxiety*. This set has some overlap with the work of Pekrun et al. (2002).

This approach offers a rich picture of the frequency of change of emotions, although it does not capture the strength of the emotion. For example, happiness could be mild in response to a small success or intense if a substantial challenge has been overcome. This is a result of the primary research aim being to identify frequency of emotional states and transitions, rather than their intensity. Bosch also notes the limitations of the approach and the accuracy of participant self-reporting. Reflecting upon this, it would be interesting to attempt to determine the repeat validity of participants' responses, by offering them a number of situations multiple times and assessing if they report the same emotion. Although a still a young field of study, the work of Bosch and colleagues may inform the design of affective programming learning environments that can make decisions based on the learner's emotional state.

Good et al. (2011) have explored self-reporting of emotion to a quite different end. They conducted a study that evaluated two different approaches for students to self-report their affective state in an attempt to help students self-regulate their emotions. The study used a computer-based widget and a tangible device called Subtle Stone (Alsmeyer et al., 2008). The Subtle Stone is a physical device with buttons and the ability to illuminate itself in a range of colours to represent different emotions. The study concluded that there was a preference among students for the Subtle Stone. It had a number of advantages: it was more visible and increased the students' awareness of their emotional state. It also provided a visible representation that other students could see and respond to. The Subtle Stone can be regarded as a physical application. This is a single unambiguous artefact. The interface only does one thing but seems to do it well. In the circumstance where a desktop-based solution is used, this becomes yet another thing competing for attention on the same communication channel as other interactions. In Good's study, a set of six emotional states were used: *enjoyment, pride, frustration, boredom, nervousness*, and *confidence*. In the desktop application, the intensity of each state was also captured.

In both of the studies discussed, the restricted set of emotions is appropriate because participants were required to report emotional state multiple times. Choosing between a list of 5 items and a list of 50 items are quite different tasks for the participant. In the Whack a Mole study, emotion was sampled as an indicator of engagement and as a potentially discriminating variable between the physical and screen-based setting. Details of the study and method, both shaped by the studies described above, are next.

3 STUDY DESCRIPTION

Whack a Mole is a game found in a number of cultures, often with different names such as Splat the Rat or Simon. The essence of the game is simple: it challenges reaction time via the ability to respond speedily to a series of stimuli. In the Arduino version of the game devised for this study, each of four LEDs has a corresponding button. When the light comes on, the player must press the corresponding button to progress through the game (Figure 1). Its screen-based equivalent had a programmable interface with

representations of clickable ‘buttons’ and ‘LEDs’ that lit up (Figure 2).

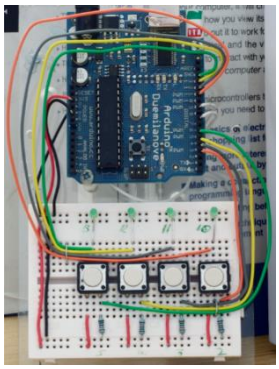

Figure 1: Physical Whack a Mole Interface.

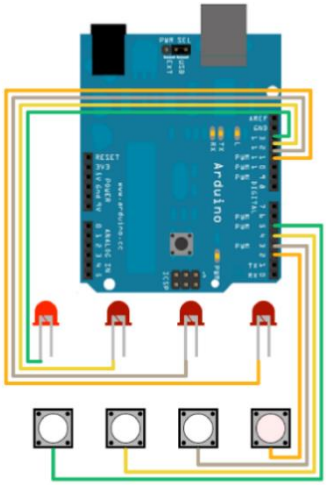

Figure 2: Screen-based Whack a Mole Interface.

In the simplest version, a light comes on at random and stays on until the corresponding button is pressed. This results in a 'playful interaction' but lacks some of the key elements that make a game. For example, it lacks user feedback: there is no indication if user is progressing other than via a subjective sense of getting quicker. There is also no defined goal at which a learner could aim. For example, if there was a goal relating to time, a player could strive to respond more quickly. The simple version of the game can be extended to introduce a timer for the light to stay on for a finite amount of time. This introduces a controllable element of difficulty. It is possible to provide user feedback when errors are made. An important feature is logging of correct pressed and incorrectly pressed buttons. This enables the player to track their performance and see how it differs from the performance of others.

Whack a Mole involved two phases for all learners. In the first stage, learners engaged in a controlled piece of tuition. The taught material was delivered via three specific worked examples. In the second phase, learners were required to demonstrate their understanding of the first stage taught material by applying it to a novel problem.

A pilot version of this study was performed with volunteer student pairs and individuals. This identified potential problems. Firstly, if the learning material was delivered by the facilitator, there was potential for different aspects of the taught material to be emphasised with different groups. Secondly, there was a risk that the tuition would become a dialogue between the facilitator and the learner, resulting in different learner experiences. Whilst dialogue is highly desirable in a typical learning situation, it was undesirable in the situation of this study, since it could result in each group of learners having a significantly different learning experience. In the wake of these insights being revealed by the pilot, a set of learning materials was developed as a series of video tutorials. These were designed to ensure that the tuition given to the learners was consistent across multiple deliveries.

3.1 Tuition Phase of the Study

A set of four short video tutorials (2-3 minutes each) was produced for the screen-based and physical version of the study. The single difference between the screen-based and physical videos was in the part of the video that demonstrated a completed task. In the screen-based videos, the screen-based Whack a Mole system was shown to demonstrate the taught code working. In the physical videos, this view changed to the physical game with LEDs, buttons and the visible Arduino.

The first video contained a brief introduction to the Arduino programming environment. It outlined the workflow of programming Arduino: code, compile, upload, and test. This video also explained where the learner's code should be placed via the programming environment, as in each case there is a minimal code skeleton. The final part of the introductory video described how to use the clickable documentation, which included all the relevant Arduino functions required for the tasks and a brief description of what each did.

The second video walked the learner through the task of making a light blink (Figure 3). This is a

traditional starting point for Arduino and is considered the equivalent of a `hello world` program. Given that the Arduino has no straightforward method of displaying text, flashing an LED is the simplest program that does something observable. For both physical and screen-based groups, this task introduces digital output. Digital output requires the defining of a pin as an output. This involves making a conceptual mapping between the electrical connections on the Arduino (numbered headers) where the component is physically or virtually inserted, and the code that will control this pin and its attached component.

```
1) int led = 13;
2) void setup(){
3)   pinMode(led,OUTPUT);
4) }
5) void loop(){
6)   digitalWrite(led,HIGH);
7)   delay(1000);
8)   digitalWrite(led,LOW);
9)   delay(1000);
10) }
```

Figure 3: Code Snippet for Blink Task.

The learner must then use the `digitalWrite()` function to change the state of this pin from high (5 volts) to low (ground). This exercise shows the learner how to use a variable as an abstraction device to store the pin number. For example, if an LED is connected to pin 13, declaring an integer variable called `led` and storing the value 13 allows the variable with a descriptive name to be used in place of 13. This clarifies the code: instead of modifying the state of a pin number directly, the variable name adds meaning to the functions with which it is used. An example is `digitalWrite(13,HIGH);` as contrasted with `digitalWrite(led,HIGH);`. To control the flow of execution the `delay` function is used to introduce an interval between state changes. This example also gives learners the chance to become familiar with the structure of an Arduino sketch: the `setup()` function runs once to initialise the board and the `loop()` function iterates infinitely to carry out the interactions of the game.

The second video also walked through the code for making a momentary light switch (Figure 4). This extends the previous example to include digital input. The learner has to identify a pin to be used with the button as a digital input. The idea of using a variable to abstract the pin number is also used to reinforce the concept. The learner must use the `digitalRead()` function to retrieve pin state information. This requires understanding that a function may have a return type and at execution time, the function call can be resolved to return type. It is possible to treat the `digitalRead()` function as its return which is `HIGH` or `LOW`. When a variable is used for the pin number, this then reads as testing the state of the given component.

```
1)  int button = 2;
2)  int led = 13;
3)
4)  void setup(){
5)    pinMode(button,INPUT);
6)    pinMode(led,OUTPUT);
7)  }
8)
9)  void loop(){
10)   if (digitalRead(button) ==
HIGH){
11)     digitalWrite(led,HIGH);
12)   }else{
13)     digitalWrite(led,LOW);
14)   }
15) }
```

Figure 4: Code Snippet for Light Switch Task.

Learners were then introduced to the `if` statement, which allows them to make a decision. In this case, they can make a decision based on the state of the button. If the button is pressed (or `HIGH`) then the LED is turned on or else the LED is turned off. Embedded in the `void loop()`, this action repeats as long as the Arduino has power.

The third video introduced the concept of an array as a device to simplify having multiple physical or virtual buttons and lights (Figure 5).

```
1)  int[] button = {2,3,4,5};
2)  int[] led = {13,12,11,10};
3)  ...
9)   void loop(){
10)    for(int i=0;i<4;i++){
11)    if(digitalRead(button[i])
== HIGH){
12)      digitalWrite(led[i],
HIGH);
13)    }else{
14)      digitalWrite(led[i],
LOW);
15)    }
16) }
```

Figure 5: Code Snippet for Multiple Buttons Task.

Where before a single variable was used to abstract the button or LED pin, now an array can

conveniently handle a collection of buttons or pins. Four physical buttons in sequence connect to consecutive digital general-purpose input/output pins that can become collected as an array of integers in the code. This required learners to use array notation to specify and initialise two arrays and form the association between the physical or virtual component, IO pin and the code.

The learners also had to use a fixed loop to iterate through the array, which is a typical strategy for combining arrays that are iterated together. This example highlights how the array index can link two concepts, in this case the buttons and the LEDs. When button `i` is pressed, LED `i` will be illuminated. This is a key concept for the second stage of the study, which required learners to demonstrate their understanding of the programming concepts taught via the video tutorial supported examples.

3.2 Challenge Phase of the Study

The challenge was for learners to devise an algorithm for a Whack a Mole game that (i) demonstrated understanding of the concepts that had been taught and (ii) used some additional features found in the documentation, such as the random function. Possible extensions were hinted at but not prescribed or described in detail. The algorithm for the completed Whack a Mole game is given in Figure 6.

```
1)  int[] button = {2,3,4,5};
2)  int[] led = {13,12,11,10};
3)  int turnOn=0;
4)
5)  void setup(){
6)    ...
7)    turnOn = random(4);
8)
digitalWrite(led[turnOn],HIGH);
9)  }
10)
11)  void loop(){
12)
if(digitalRead(button[turnOn] ==
HIGH){
13)
digitalWrite(led[turnOn],LOW);
14)      turnOn = random(4);
15)
digitalWrite(led[turnOn],HIGH);
16)    }
17)  }
```

Figure 6: Example Code for Whack a Mole Game.

It consists of turning on a random light, waiting until the corresponding button is pressed and then picking another random light. This requires learners to demonstrate all the taught skills in context and integrate them into an application.

4 STUDY DESIGN

The Whack a Mole study ran as part of an undergraduate module in Physical Computing. This module is taught to Level 1 (first year) applied computing, computing science, product design and interaction design learners in the University of Dundee. The class was organised into small practical groups of three or four. To ensure an optimal staff to learner ratio, the class was separated into two separate sittings. The two lab groups alternated between taught sessions and independent sessions. In one week, group A would have a taught lab while group B would engage in an independent lab assignment. The following week, the sittings were reversed. Learners were assigned randomly to either group A or group B at the start of semester and these groupings were used in the delivery of the Whack a Mole study. In the first week of the study, the taught group received the physical Whack a Mole intervention. This group had 22 participants of which 14 were male and 8 were female. The following week, the groups switched around and the taught group received the screen-based Whack a Mole intervention. This group had 16 participants, 15 of whom were male.

As this study involved human participants, ethical clearance was sought and obtained from the ethics committee of the School. Two methods were designed to capture appropriate data. Firstly, a paper-based questionnaire was designed to test knowledge and understanding of arrays. Secondly, a method was devised and piloted to capture a learner's emotional response to programming. These two methods are described in the next section.

4.1 Knowledge and Understanding

A paper-based questionnaire was designed to measure changes in knowledge and understanding of arrays. The first parts of the questionnaire contained questions to test the participant's knowledge of arrays. The final part of the questionnaire required responses to questions associated with given code snippets that demonstrated array use within a small program.

Before the lab teaching began, participants were given the questionnaire to complete independently under exam conditions, i.e. without conferring with peers and without external resources. After

completing the study, participants were asked to complete the post-test questionnaire. Participants were also given the emotions questionnaire (described next) and advised how to complete it.

4.2 Emotional Response

The method designed to measure emotion was minimally disruptive for the learners. The decision was made to design a post-test questionnaire that learners could fill out as a reflective process. The studies discussed earlier involve multiple sampling, identifying the points at which an emotion occurred and any transitional states. A high frequency of samples requires a small set of possible participant responses and ideally the reconciliation of similar emotions, such as *calm* and *content*. The approach taken for Whack a Mole was the opposite. As the response from the participant was sought once at the end of the study, a broader range of emotions could be included. The instrument was not designed to measure when the emotion occurred in relation to other emotions. Instead, it was designed to capture *why* a state of emotion occurred. With more time available and without repeat sampling fatigue, participants were able to respond to a larger range of emotions and offer contextual information about what they were doing and why the emotion occurred. Where similar emotions were present, this provided several opportunities for a subtly different trigger to elicit feedback from learners. Amusement, elation and pleasure all fall under the heading of *positive lively* but may be attributed to different activities. For these reasons, a new method to obtain emotion data was designed, based on an ontology of emotional states: the Reflective Emotion Inventory.

The Reflective Emotion Inventory (REI) was designed to capture emotional response in individuals. It is a reflective tool, designed to be delivered at the end of a session. It encourages learners to think back over their experience and indicate if they felt any of a range of emotions. The list of emotions used for the REI was derived from the HUMAINE project (Petta et al., 2011). HUMAINE's 'Emotional Annotation and Representation Language' proposed a core of 48 different emotions arranged into 10 sub-categories: negative and forceful, negative and not in control, negative thoughts, negative and passive, agitation; positive and lively, caring, positive thoughts, quiet positive, and reactive. Figure 7 gives the components of these sub-categories that were used for the study.

The REI questionnaire captures three things. (a) The learners are first invited to scan through the list of emotions and indicate if they have experienced any of them. (b) Following this, they can indicate the degree of arousal or intensity for each of the experienced emotions on a four-point unipolar Likert scale (Cummins and Gullone, 2000). A unipolar Likert scale was selected for two reasons. Firstly, given that the REI contains many emotions, there was a preference for a unipolar scale because it is easier for users to respond to than a bipolar scale that places opposites at either end of the scale. Secondly, the REI is intended to be a reflective tool that captures emotions experienced over a period. It is therefore quite possible that opposing emotions will be experienced at different times throughout the event. (c) Once the learners have noted emotions they have felt and the degree of arousal, they are encouraged to offer some contextual information in a free-text response space. The purpose of this is to understand why they experienced the given emotion. An example response might be: Annoyance, 3, "Getting the wires in the correct place".

This three-part design offered the ability to capture change in knowledge and emotional response to programming. Results were then analysed to obtain insights into any difference between groups.

Anger, Annoyance, Contempt, Disgust, Irritation, Anxiety, Embarrassment, Fear, Rage, Worry, Doubt, Envy, Frustration, Guilt, Shame, Boredom, Despair, Disappointment, Hurt, Sadness, Shock, Stress, Tension, Amusement, Delight, Elation, Excitement, Happiness, Joy, Pleasure, Affection, Empathy, Friendliness, Love, Courage, Hope, Pride, Satisfaction, Trust, Calm, Content, Relaxed, Relieved, Serene , Interest, Politeness, Surprise

Figure 7: Reflective Emotion Inventory.

5 RESULTS

5.1 Knowledge and Understanding

Table 1 presents descriptive statistics for test performance. The screen group scored a mean pre-test score of 76% and a mean post-test score of 79%, with an improvement of 3%. The physical group scored a mean pre-test score of 57% and a mean post-test score of 61%, resulting in an improvement of 4%.

In each group, there were three distinct classes of learners. Some learners improved their performance, some showed no change and some performed worse in the post-test. The physical group performed slightly better than the screen-based group across all aspects, with a greater percentage of the group improving and fewer reducing their pre- to post-test performance – but no difference was statistically significant.

Table 1: Screen and Physical Group Scores (%).

	Pre-test		Post-test		Sig.
	Mean	**SD**	**Mean**	**SD**	**2-tailed)**
Screen	75.63	14.59	78.75	10.25	>0.05
Physical	57.27	20.74	60.90	19.50	>0.05

5.2 Emotional Response

Figure 8 gives the mean response from each sub-category of emotion for both groups. The screen-based group did not offer free text comments to contextualise their emotions as readily as the physical group did. They expressed *negative forceful* emotions that were cited as being the result of problems with the code: "code errors" or "sorting some issues with the program". Several participants in the screen-based group intimated feeling envy when other groups had their program working before they did. Several learners also expressed a feeling of friendliness as a result of working in a group. One group noted a feeling of worry "if they could complete the task on time". In contrast, other groups indicated a sense of boredom at being finished early. Positive emotions for the screen-based group were largely cited because of completion of the task and "getting it working". This was attributed by many participants to a feeling of amusement, joy and happiness. The contextualising of positive emotions was as frequent as that of negative emotions. However, the reasons cited for a positive emotion were far less diverse.

The physical group offered a number of comments for each sub-category of emotion. Negative emotions were attributed to a range of features of creating the Whack a Mole game. One of the most frequently cited situations resulting in negative emotions was wiring. Many participants just stated the single word "wiring", while others elaborated. Responses include "when the wires fall out", which is a common problem if jumper wires are not cut long enough or well organised. As with programming, there is often a tendency by the novice to get "stuck in" to the task and not plan their actions well. "Getting the wires in the right places" was also expressed as a problem by some (the pitch of the breadboards used is one hole every millimetre, which can be problematic). Specific components were mentioned by some: "getting the LED the right way" was noted by one participant, with another noting "wiring up resistors". LEDs have a polarity and require both the signal and ground voltage wires to be in the correct position. Resistors, on the other hand, do not have polarity but are very small, and placing them into breadboards can be problematic.

These type of difficulties were most prevalent under the *negative forceful* category, with learners frequently associating these difficulties with feeling anger and annoyance. This category was the most strongly reported negative emotion in the physical group. To a lesser extent, these difficulties also appeared under the *not in control* category, such as rage. Several participants cited negative thoughts related to whether their build would work or not. Also in the negative thoughts category, frustration was related to wiring-up of the build. Interestingly, frustration was also cited in response to poorly specified compiler errors. It is fair to say that the Arduino IDE provides much more novice-friendly compiler errors than an industry standard IDE such as Eclipse or Visual Studio. Nonetheless, there are inevitably situations where there is disconnect between the error, the specific line of code and the description offered in the IDE. One or two of the learners expressed passive emotions such as boredom, at being finished early. Being stressed was also noted by several individuals in response to the system as a whole (wires and code) not working, or being unsure as to whether they would complete the build on time or not.

Positive emotions were contextualised with free text comments less richly than negative emotions. However, positive emotions were given greater intensity than negative emotions. *Positive and lively* was the most strongly reported emotional category of all. This was heavily noted by participants because of completing the build: "when it worked", and when engaging with the product of their work: "playing the game". Participants also noted a feeling of happiness at getting their task completed. The second most strongly reported emotional category was *reactive*. This was cited as interest in "learning new things". One participant noted interest in the logic they had arrived at in developing the Whack a Mole algorithm.

When considering the screen-based group's emotional responses grouped together as positive or negative, there was a noticeable difference between the positive and negative emotions reported. Positive emotions were experienced by all participants to a greater extent than negative emotions. It is notable that in the physical group there was a difference in intensity of positive and negative emotional categories as a whole. For the physical group, all the positive emotions had greater intensity than the negative ones, with the exception of *caring*. This matched the rich contextual data offered by the physical group. Where learners worked with the physical artefact, they had a strongly positive experience. Three of positive emotions reported by the physical group were notably greater than that of the screen-based group: *positive & lively*, *quiet positive* and above all *reactive*.

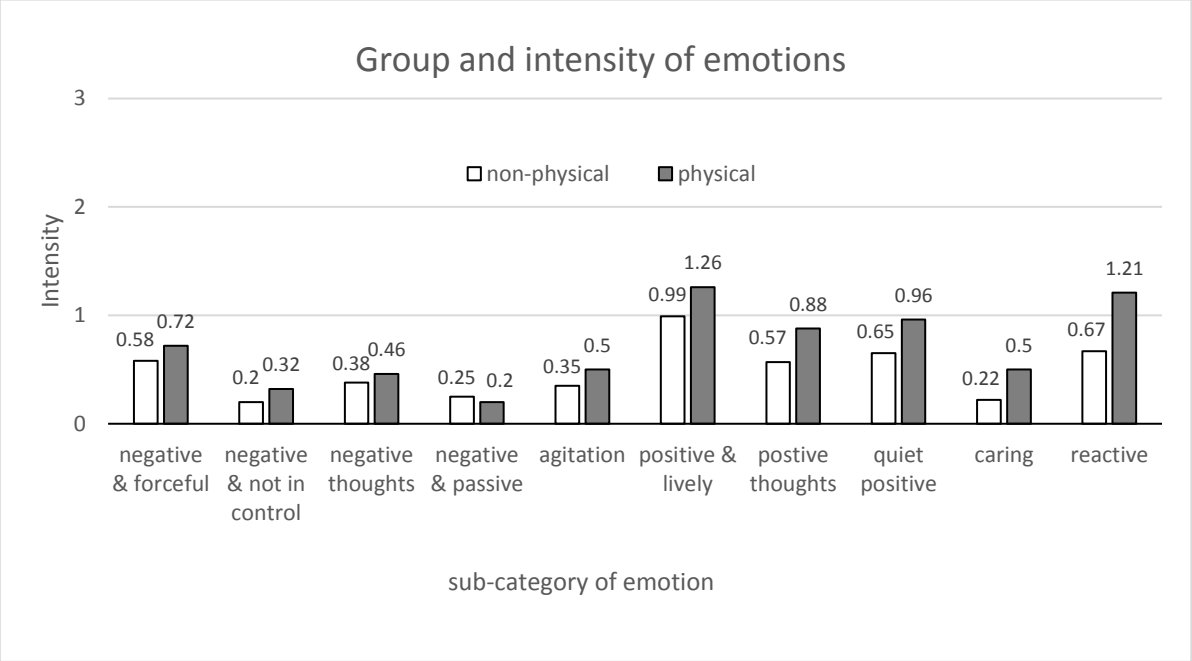

Figure 8: Strength of Emotional Responses.

6 DISCUSSION

6.1 Knowledge and Understanding

It is striking that in both groups, around two-thirds of learners showed no change in knowledge or understanding about arrays and associated strategies. The most likely explanation for this is that when both the screen-based and the physical groups were ordered for performance, the top two-thirds of both groups had very high pre-test scores, leaving little room for improvement. It is likely that an earlier point in the teaching period for the study would have offered a greater opportunity for the interventions to create a change in knowledge and understanding.

The groupings for the study also proved problematic. The pre-test data for the screen-based group showed a tight normal distribution centred on a very high mean. The physical group had a slightly skewed distribution in pre- and post-tests, with several particularly weak scores. The two randomly allocated groups thus had different academic abilities or levels of experience. This may have reduced the sensitivity of the questionnaire to detect improvements between groups.

6.2 Emotional Response

Firstly, with regard to the REI, a low response was noted for the free text component of the REI. This was unsurprising, given the additional effort required by learners to verbalise the contexts in which they felt a given emotion. Secondly, considering the two different groups, the REI did establish different responses from the two groups. With the physical group, all but one of the positive emotions had greater intensity than the negative ones. Indeed there was an observable difference in the degree of engagement of different groups with the finished artefact. Several of the participants in the physical group were seen taking pictures and videos to share on social media. This indicated a degree of pride and a desire to share their work that was not observed in the screen-based group.

Most often, the anecdotal references to programming and emotion are focused on negative feelings. The free-text contextualisation presented here shows that participants frequently experienced many causes of irritation that are well reported in the literature, including unintelligible compiler errors and syntax errors. Programming is inherently an error-prone activity and the activity in the study reflects

this. This particular study did not look for insights into strategies to overcome any difficulties of debugging.

It is interesting that the physical element in many respects confounds many of the areas of programming difficulty. Breadboarding with electrical components is an inherently finicky task requiring good eyesight and a steady hand. It also has many of the same problematic features as programming, such as the error-prone nature, requiring high degree of detail, tracing of routes through a connected network and a one-to-many mapping from problem to solution. In addition to these problems, whilst programming offers compiler errors to assist the learner in trapping errors, there is no such support when wiring breadboards. As a result, errors in electrical circuits are often very difficult to identify. It seems counter-intuitive therefore that placing programming and electrical prototyping activities together can improve the emotional response to the programming experience. The dominance of positive emotions being reported suggests that this happened in the Whack a Mole study. The results suggest that creating a functioning physical mole game presented a sufficient challenge for most participants across a range of skills. The resultant completion of the task generated an emotional response that outweighed the 'pain' endured in working through the task. One theory to propose is that this resulted from the different bandwidth of interaction offered by the two systems. In the non-physical group, learners could only interact with a single device, namely the PC being used to program the virtual game, giving a screen to offer feedback to the user and a mouse and keyboard to accept input. In constructing a Whack a Mole game with the physical system, the Arduino, buttons and LEDs used to make the tangible game all increased the bandwidth of interaction. This may have contributed to the richer more positive emotional response from learners in the physical group. The implications of Mayer's cognitive theory of multimedia learning (2002) would be an interesting subject for future work in this respect.

Having a low ratio of negative emotions to positive emotions may signify a learner who will do well with programming. It resonates with the 'movers and stoppers' findings of Perkins et al. (1986). A stopper is categorised as person who is halted abruptly by an error or difficulty and does not have the inclination to tackle the problem independently. A stopper appears to have abandoned all hope of solving the problem on their own, the emotional response to being confronted with a bug or compiler error being crucial. A novice who becomes very frustrated by unforeseen problems is likely to become a 'stopper'. In contrast, a 'mover' is a learner with enthusiasm who views an error as a challenge rather than an obstacle (Perkins et al., 1986). The ability to modify and adapt programs effectively in response to errors is likely to reinforce a mover's ability to self-support his or her problem solving and progress. Therefore an important attribute for an aspiring programmer may be the capacity to take greater pleasure from the competed task than displeasure experienced by the challenges on the road to success.

7 LIMITATIONS

One limitation of the Whack a Mole study resulted from the composition of the screen-based and the physical groups. It is not ideal to have two groups of different sizes and of different academic abilities. Neither was the gender balance of the groups ideal. A solution to the problem would be to administer the pre-test and then create groups based on the score. Unfortunately, it could be problematic to implement paper tests in a single study and in this case, it would have disrupted the established groups within the class.

Secondly, the approach adopted had a qualitative focus and identified descriptive statistics. An alternative approach, to evaluate whether the difference between the two groups was statistically significant, would have required much more statistical testing around the repeated validity of participants' responses and the instrument in general. Since the sample size was small and the instrument new, the former approach was preferred to provide early insights into the phenomenon.

One of the challenges with a pre/post-test methodology is pitching the test difficulty correctly to ensure maximum sensitivity to the phenomena being researched, which in this case related to knowledge and understanding of arrays. The pre-test knowledge results suggest that in many cases an understanding of arrays has developed prior to the study. As a result, for many of the learners the measure had limited sensitivity. Despite these difficulties, the Whack a Mole study offers some valuable insights into the differences observed in novice programmers working with screen-based and physical media.

8 CONCLUSIONS

The Whack a Mole study aimed to explore how learning with a physical device differed from learning with a screen-based equivalent. The main findings of the study can be summarised by referring to the research question posed:

How does working with a physical artefact as opposed to a screen-based artefact affect learning of computer programming?

There was no noticeable difference in learning effect measured between the two groups, indicating that the physical interface did not measurably contribute to or hinder learning. However, there was a difference in emotional response to the learning experiences. Both groups described a range of negative emotions with similar levels of strength and for similar reasons. Both groups also noted a similar range of positive emotions. However, the physical group noted a greater strength of positive emotions associated with the learning experience.

If the Whack a Mole study were to be adapted to enable a greater degree of flexibility, for example allowing learners to design their own interface for the game, there would be no additional programming overhead to create a physical game. All that would be required would be longer wires for the buttons and LEDs that could be embedded in any number of craft materials. For the same to be done with a screen-based solution, additional skills would need to be taught, adding to the complexity of the session.

Re-considering the literature, it is worth noting that the sample task learners engaged in for the study by Bosch et al. (2013) was a traditional CS1-style maths based problem. Although this problem type is valid, it represents what Robins et al. (2003) argue is a knowledge-driven approach to programming education. One can argue for an approach to programming education that is more stimulating and framed within a context of value to the learner. The results of this study suggest that the powerful affordances of physical computing, i.e. the ability to take intangible things and make them physical (such as when using an LED to indicate state) can lead to very positive emotions without jeopardising learning.

The difficulties of learning to program have been studied for nearly 50 years and many challenges identified years ago endure to this day. The essence of programming remains unchanged. It requires a programmer to take a problem and describe its solution in sufficient detail, without ambiguity, such that a machine can reliably follow the instructions. What has moved forward considerably is the set of tools used to support learning to program. Just as commercial development tools have matured from rudimentary text editors to powerful interactive development environments, so too have educational tools, which have benefited from years of research and from the increased capacities of modern computers. Desirable features for education programming tools and learning experiences are increasingly being recognised as those relating to the motivation of the learner, such as personal, social and contextual elements rather than purely technical ones.

The study described here demonstrates that a physical interface can provide a more positive emotional experience than a screen-based equivalent. Designers of learning experiences may wish to include consideration of this insight when planning the introduction of new programming concepts or creating programming laboratory exercises and assessments. Designing an engaging learning experience is not a routine process that can be governed by a set of rules to be followed dutifully to guarantee consistent results; it is a much more creative task. It requires reflection and consideration not just of what is to be learned but also of who is learning and how they can best be motivated to succeed.

REFERENCES

Alsmeyer, M., Luckin, R. and Good, J., 2008, April. Developing a novel interface for capturing self reports of affect. In *CHI'08 Extended Abstracts on Human Factors in Computing Systems*, pp. 2883-2888. ACM.

Bosch, N. and D'Mello, S., 2015. The Affective Experience of Novice Computer Programmers. *International Journal of Artificial Intelligence in Education, 27*(1), pp.181-206.

Bosch N., D'Mello S., Mills C., 2013. What Emotions Do Novices Experience during Their First Computer Programming Learning Session?. In: Lane H.C., Yacef K., Mostow J., Pavlik P. (eds) Artificial Intelligence in Education. AIED 2013. *Lecture Notes in Computer Science, 7926*. Springer, Berlin, Heidelberg.

Cummins, R. A. and Gullone, E., 2000. Why we should not use 5-point Likert scales: The case for subjective quality of life measurement. In *Proceedings, second international conference on quality of life in cities*, pp.74-93.

D'Mello, S., 2013. A selective meta-analysis on the relative incidence of discrete affective states during learning with technology. *Journal of Educational Psychology*, 105(4). pp 1082-1099.

Good, J., Rimmer, J., Harris, E. and Balaam, M., 2011. Self-Reporting Emotional Experiences in Computing Lab Sessions: An Emotional Regulation Perspective. In *Proceedings of the 23rd Annual Psychology of Programming Interest Group Conference*.

Martin, C. and Hughes, J., 2011. Robot dance: Edutainment or engaging learning. In *Proceedings of the 23rd Annual Psychology of Programming Interest Group Conference.*

Mayer, R. E., 2002. Multimedia learning. *Psychology of Learning and Motivation, 41*, pp. 85-139.

Meyer, D. K. and Turner, J. C., 2002. Discovering emotion in classroom motivation research. *Educational psychologist, 37*(2), pp.107-114.

Pekrun, R., Goetz, T., Titz, W. and Perry, R.P., 2002. Academic Emotions in Students' Self-Regulated

Learning and Achievement: A program of Qualitative and Quantitative Research. *Educational psychologist*, *37*(2), pp.91-105.

Perkins, D.N., Hancock, C., Hobbs, R., Martin, F. and Simmons, R., 1986. Conditions of learning in novice programmers. *Journal of Educational Computing Research*, *2*(1), pp.37-55.

Petta, P., Pelachaud, C. and Cowie, R., 2011. Emotion-Oriented Systems. *The Humaine Handbook*, pp.978-3.

Robins, A., Rountree, J. and Rountree, N., 2003. Learning and teaching programming: A review and discussion. *Computer Science Education*, *13*(2), pp.137-172.