Designing Engaging Learning Experiences in Programming

Book Section

How to cite:

Martin, Chris; Hughes, Janet and Richards, John (2018). Designing Engaging Learning Experiences in Programming. In: Escudeiro, Paula; Costagliola, Gennaro; Zvacek, Susan; Uhomoibhi, James and McLaren, Bruce M. eds. Computers Supported Education. Communications in Computer and Information Science, 865. Springer, pp. 221–245.

For guidance on citations see FAQs.

Version: Accepted Manuscript

Link(s) to article on publisher's website:
http://dx.doi.org/doi:10.1007/978-3-319-94640-5_12

Designing Engaging Learning Experiences in Programming

Chris Martin[1], Janet Hughes[2] and John Richards[3, 4]

[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, U.S.A.
[4]University of Dundee, Dundee, U.K.

c.j.martin@dundee.ac.uk,janet.hughes@open.ac.uk,
ajtr@us.ibm.com

Abstract. In this paper we describe work to investigate the creation of engaging programming learning experiences. Background research informed the design of four fieldwork studies to explore how programming tasks could be framed to motivate learners. Our empirical findings from these four field studies are summarized here, with a particular focus upon one – Whack a Mole – which compared the use of a physical interface with the use of a screen-based equivalent interface to obtain insights into what made for an engaging learning experience. Emotions reported by two sets of participant undergraduate students were analyzed, identifying the links between the emotions experienced during programming and their origin. Evidence was collected of the very positive emotions experienced by learners programming with a physical interface (Arduino) in comparison with a similar program developed using a screen-based equivalent interface. A follow-up study provided further evidence of the motivation of personalized design of programming tangible physical artefacts. Collating all the evidence led to the design of a set of 'Learning Dimensions' which may provide educators with insights to support key design decisions for the creation of engaging programming learning experiences.

Keywords: Motivation, Programming, Learning Dimensions

1 Introduction: Motivating Learning

Motivation is defined in the Oxford English Dictionary as "a reason or reasons for acting in a particular way" [32]. If it changes desire into will and subsequent action, then it can be an important element to cultivate in the learner when designing an engaging learning experience. This paper is about creating a context for motivation and rich engagement with learning to program in particular.

The paper begins with a brief description of related literature, including work highlighted in detailed literature reviews on motivation and learning [9, 12]. Continuing with the theme of published literature relating to engaging learning experiences, section 2 gives a review of a selection of studies relating to evidence of emotion generated by learning experiences. Section 3 introduces the main study of the paper, detailing the Whack a Mole study which compared the engagement resulting from screen-based programming work with that resulting from a matched piece of work that used a tangible physical artefact instead of a conventional screen. Sections 4-6 provide the study design, results and discussion. Section 7 then describes a follow-up study which further sets the learning of programming in the context of using tangible physical artefacts. All the lessons learned from the literature and the studies described here are then used to derive a set of Learning Dimensions, described in section 8, which it is hoped will prove useful to educators seeking to design engaging learning experiences in programming.

1.1 Categories of Motivation

[33] described the ability to modify behavior through reinforcement in animal experiments. It was shown that if a reward was given in response to a desired behavior, this desired behavior by the animal was likely to recur. This reward is referred to as a task-extrinsic reward and its effect can be described as extrinsic motivation. Complementing this, [36] described a number of studies that demonstrate animal behaviors which were not purely functional but may be driven by intrinsic motivation, i.e. ‘motivation from within’ or to engage in an activity for its own sake without directly perceivable payoff to the participant.

[2] offer a good description of elements of task-extrinsic rewards and intrinsic motivation. The former may take any of the following forms: competition, evaluation, recognition and tangible incentives. Elements of the latter include self-determination, competence, task involvement, curiosity, enjoyment and interest, all of which are desirable behaviors for a learner engaged in an activity.

1.2 Extrinsic Motivation

[4] explored the effect of financial incentives on both central and peripheral learning tasks. An experimental group showed an increased performance (measured as time to complete a task) for the central task when offered an additional financial incentive. However, no improvement was observed with the peripheral task, suggesting that a financial reward narrows the focus of the participant and reduces the chances of incidental learning.

Extrinsic motivators are commonplace in education at all levels. Competency is typically measured with a summative assessment that results in a grade for the learner. This is an important aspect of assessment in education, though this risks fostering the type of motivation that encourages goal-oriented strategies in learners: [9] described extrinsic rewards as increasing speed and quantity of effort at a reduced quality. In addition to encouraging greater throughput of work at a reduced quality, extrinsic rewards increase focus at the expense of engagement in peripheral topics. In a learning situation, this is likely to foster shallow learning where the learner focuses only on activities that are clearly related to the examination at the end of the learning. From a review of 43 studies, [14] conclude that extrinsic rewards undermine intrinsic motivation where the reason for their delivery is poorly defined in the eyes of the recipient.

1.3 Intrinsic Motivation

[3] observed the effects of money, awards and verbal feedback on intrinsic motivation in kindergarten children engaged in a drawing activity. Only verbal feedback had a positive impact on the learner's motivation and time on task. [12] also found evidence that positive feedback can enhance intrinsic motivation. They also found that that tangible rewards undermined intrinsic motivation and were in some cases seen as methods of control that may forestall self-regulation. In a learning situation in particular, [10] robustly locate the importance of personalization and choice for increasing intrinsic motivation in learners. Their study of 72 fifth grade learners (10 to 11 year olds) found that there was a powerful learning benefit observed in the personalized choice condition. Learners displayed a deeper engagement as well as increased motivation.

In conclusion, intrinsic motivation is inarguably a desirable feature to encourage in any learning experience, with evidenced learning benefits described above. The role of extrinsic motivators is a little more uncertain: tangible rewards have been shown to narrow thinking and reduce the depth at which learners engage. However, positive feedback has been observed to enhance intrinsic motivation. For a review of motivation in a broad educational context, see [29], who identifies three over-arching themes in motivation in education. This work, however, focuses on motivation in computer science education, particularly with respect to programming. Understanding intrinsic incentives, and how they relate to the instructional paradigms, may give pointers to the design of engaging learning experiences. Understanding how to obtain evidence of engagement is a further challenge.

2 Evidence of Engagement in a Learning Experience

Anecdotal evidence from programmers will confirm that bugs are frustrating, finding and removing them is satisfying, and reaching a successful conclusion can lead to pride. In education in general, emotional response to learning with technology has been studied for some time, e.g. [23].

2.1 Emotion and Learning

D'Mello's review [13] of 24 studies noted that many learning contexts resulting in engagement had comparatively low reporting of negative emotions. [25] conducted a detailed literature review from 1974 through to 1990, which was later extended to 2002. It 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. [25] 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 utilizing 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 [25].

2.2 Emotion and programming

Emotional response to *programming* has a more limited body of work in the literature, although some interesting work has been done [e.g. 6, 7 and 15]. [7] sought to map the emotional states a novice experiences and their relative proportions, and explore the co-occurrence of emotional states and the relationship between interaction events. In addition, they mapped transitions between these states. Participants self-reported 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 [25]. The approach offers a rich picture of the frequency 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. [7] also note 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. The studies conducted by Bosch and colleagues may yet inform the design of affective programming learning environments that can make decisions based on the learner's emotional state.

2.3 Emotion and programming tangible devices

[15] explored self-reporting of emotion in 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 the Subtle Stone [1]. The Subtle Stone device had buttons and the ability to illuminate itself in a range of colors to represent different emotions. The study concluded that there was a preference among students for the Subtle Stone rather than the computer-based widget. It had a number of advantages: it was more visible and in-

creased 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. It is a single unambiguous artefact; the interface only does one thing but seems to do it well. Where a desktop-based solution is used, this becomes yet another thing competing for attention on the same communication channel as other interactions. In [15], 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, a restricted set of emotions was appropriate because participants were required to report emotional state multiple times.

Robot Dance. "How do you teach a Robot to Dance" was developed as part of an outreach activity to engage and enthuse possible future computing students. The workshop, reported in [19 and 22], was also designed to explore how working with physical robots can support introductory programming learning. Middle-school students were given the challenge of programming pre-fabricated Arduino-based differential drive robots to choreograph a robot dance. There were two constraints: the dance must last for exactly 20 seconds and the robot must not fall off the stage (a 1 square meter mat or equivalent table top for added suspense). Dance was partly chosen as a context that may make computer programming more appealing to female participants. This strategy (often seen in narrative tools as powerful) places a task in a context that has cultural relevance or 'coolness'. After the students have been given a relatively short development time of 10 - 15 minutes, they upload their final program and gather round the dance floor to watch each of the performances. [5] note that performance has a more wide-reaching motivational effect than competition in this area.

An increased understanding of the programming concepts of sequence and syntax was demonstrated by these learners when measured using a pre- and post-test approach. One aspect more directly related to programming motivation was having a performance opportunity for their work. The dance element of the workshop offered an open problem for which prior knowledge was largely irrelevant: the task was creative, subjective and without endpoint. This had a pitch-levelling effect: a number of teachers commenting that students who previously had been disinterested or failed to show aptitude for programming now had engaged in the workshop successfully.

Robot Dance in the Community. A follow-on workshop used a less constrained structure without the confines of a classroom, as described in [22]. Learners were given a greater degree of independence: the learning experience was organized to be drop-in, situated in a public shopping center. Following a brief introduction to Arduino, learners were given a skeleton program to extend and then were "walked-through" the program required to make the robot move forward a short distance. The challenge presented then was to create 20 seconds of dance moves. Participant observation recorded four different groupings: single child, child pairs, child parent pairs and multiple children and parents. All learners demonstrated an observable emotional response to the performance they had programmed. Irrespective of the small audience, learners showed an observable pride in their creation.

Robot Dance in the Community observations confirmed that programming has an emotional dimension. The main study described next (Section 3) investigated this further: it explored how to get a more detailed picture of the learner's emotional response to programming tasks. The Whack a Mole study used an alternative physical artefact for learners to work with and sampled emotion as an indicator of engagement. Details of the study and method are given next.

3 Whack a Mole Study Description

Whack a Mole is a simple game that challenges reaction time via the ability to respond speedily to a series of stimuli. In the simplest version, a light comes on at random and stays on until a corresponding button is pressed. This results in a 'playful interaction' although it lacks some of the key elements that make a game. For example, it lacks user feedback or performance tracking with respect to the performance of other players.
In the Arduino version 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 (Fig. 1). Its screen-based equivalent had a programmable interface with representations of clickable 'buttons' and 'LEDs' that lit up (Fig. 2).

Whack a Mole involved two stages. In stage one, learners were given taught material via three specific worked examples. In stage two, learners had to demonstrate their understanding of the first stage taught material by applying it to a novel problem. A pilot version of this study performed with volunteer students 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 emphasized 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. Given those insights, a set of learning materials was developed as a series of video tutorials to ensure that the tuition given to the learners was consistent across multiple deliveries.

3.1 Video tutorials

A set of four 2-3 minute video tutorials was produced. The single difference between the screen-based and physical artefact 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 artefact videos, this view changed to the physical game with LEDs, buttons and the visible Arduino.

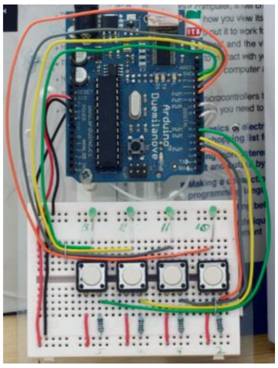

Fig. 1. Physical Whack a Mole Interface [21].

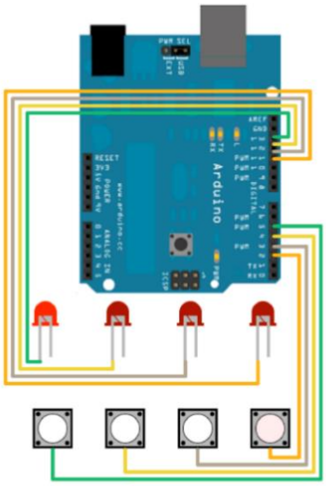

Fig. 2. Screen-based Whack a Mole Interface [21].

Video 1 contained a brief introduction to the Arduino programming environment, which uses a C-style textual language. It outlined the workflow of programming Arduino: code, compile, upload, and test. It also explained where the learner's code should be placed, as in each case a minimal code skeleton is given. The final part described how to use the clickable documentation which included all the relevant Arduino functions required for the tasks together with a brief description of what each function did.

Video 2 walked the learner through the task of making a light blink (Fig. 3). This traditional starting point for Arduino is considered the equivalent of a `hello world` program: since Arduino has no straightforward method of displaying text, flashing an LED is the simplest program that does something observable. For both physical artefact and screen-based groups, this task introduces digital output that 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.

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 `digital-Write(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 initialize the board and the `loop()` function iterates infinitely to carry out the interactions of the game.

```
int led = 13;
void setup(){
  pinMode(led,OUTPUT);
}
void loop(){
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```

Fig. 3. Code Snippet for Task 1 Blink [21].

Video 2 also walked through the code for making a momentary light switch. This extended the previous example: 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 can be `HIGH` or `LOW`. When a variable is used for the pin number, this test the state of the component. 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 whilst the Arduino has power.

Video 3 introduced the concept of an array as a device to simplify having multiple physical or virtual buttons and lights. 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 the use of an array notation to specify and initialize two arrays and form the association between the physical or virtual component, the 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 Whack a Mole Challenge

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 code for the game (Fig. 4) consists of turning on a random light, waiting until the corresponding button is pressed and then picking another random light. Learners had to demonstrate the taught skills in context and integrate them into an application.

```
int[] button = {2,3,4,5};
int[] led = {13,12,11,10};
int turnOn=0;

void setup(){
  ...
  turnOn = random(4);
  digitalWrite(led[turnOn],HIGH);
}

void loop(){
  if(digitalRead(button[turnOn] == HIGH){
    digitalWrite(led[turnOn],LOW);
    turnOn = random(4);
    digitalWrite(led[turnOn],HIGH);
  }
}
```

Fig. 4. Example Code for Whack a Mole game [21].

Further examples of code extracts as provided in the various videos are provided in [21]. Ethical approval for the study design, which is described next, was given by the School's ethics committee.

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 already organized into small practical groups of three or four. To optimize the staff to learner ratio, the class was arranged 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 session 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, Group A 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, with Group B receiving the taught lab session. This group had 16 participants, 15 of whom were male. 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. Both methods are described next.

4.1 Knowledge and Understanding

A paper-based questionnaire was designed to measure changes in knowledge and understanding of arrays. The first part contained four questions to test the learner's general level of knowledge of arrays, independently of any context or specific situation. The second part was designed to explore the learner's level of understanding of arrays, for instance asking the learner to "*describe in your own words (and pictures) what an array is in the context of computer programming*". Constructing a description of a concept and externalizing it in words and images requires a good understanding of the concept. It deliberately lacks a pre-defined framework into which learners could slot their knowledge. There also is no opportunity for learners to guess the answer. The third and final part of the questionnaire requires the learners to respond to three questions associated with given code snippets that demonstrate array use within a small program.

Before the lab teaching began, participants were given the questionnaire to complete independently under exam conditions. 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 selected to measure emotion had to satisfy three criteria. Firstly, it had to enable learners to reflect on the emotions they experience as a result of learning to program. Whilst a record of a particular emotion is interesting, it is even more valuable to have the learner’s explanation of the context of this emotion. Secondly, it had to be practical to deploy as part of a regular laboratory class activity that had a high volume of concurrent participants. Approaches such as that devised by [7] where participants review high volumes of still images to self-report emotion would not have been practicable. Finally, as this was an early stage in the exploration of emotional response to programming, it was considered premature to seek high volume time-series data such as would derive from automated facial recognition software. However, an interesting approach for future research would involve using that technology coupled with a reflective learner interview that discussed the emotions that had been captured.

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 in Whack a Mole is the opposite. As the response from the participant is sought once at the end of the study, a broader range of emotions can be included. The instrument is not designed to measure when the emotion occurred in relation to other emotions. Instead, it is designed to capture *why* a state of emotion occurred. With more time available and without repeat sampling fatigue, participants are 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 are present, this provides several opportunities for a subtly different trigger to elicit feedback from learners. For example, 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 that was minimally disruptive for the learners. It was based on an ontology of emotional states: the Reflective Emotion Inventory.

The Reflective Emotion Inventory (REI) has been 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 [28] which proposed a core of 48 different emotions arranged into 10 sub-categories (Table 1).

Table 1. HUMAINE Emotion Categories [28].

Negative	Positive
Negative and forceful: anger/annoyance/contempt/disgust/irritation	*Positive and lively:* amusement/delight/elation/excitement/happiness/joy/pleasure
Negative and not in control: anxiety/embarrassment/fear/helplessness/powerlessness/worry	*Caring:* Affection/empathy/friendliness/love
Negative thoughts: doubt/envy/frustration/guilt/shame	*Positive thoughts:* courage/hope/pride/satisfaction/trust
Negative and passive: boredom/despair/disappointment/hurt/sadness	*Quiet positive:* calm/content/relaxed/relieved/serene
Agitation: shock/stress/tension	*Reactive:* interested/politeness/surprise

The REI questionnaire captures three things (Table 2). (a) The learners are first invited to scan through the list of emotions and indicate if they have experienced any of them. (b) Learners can indicate the degree of arousal or intensity for each of the experienced emotions on a four-point unipolar Likert scale [11]. 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) Learners can offer some contextual information in a free-text response space. The purpose of this is to describe why they experienced the given emotion. An example response might be: Annoyance, 3, "Getting the wires in the correct place".

Table 2. Components of Reflective Emotion Inventory

(a)	(b)	(c)
Which emotion has been experienced	Degree of intensity for each: 4-point Likert scale	Why and when was this emotion experienced

The two parts to the study thus captured change in knowledge and any emotional response to programming, in order to identify any difference between the two groups.

5 Results

5.1 Knowledge and Understanding

Table 3 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%. The physical artefact group scored a mean pre-test score of 57% and a mean post-test score of 61%.

Table 3. Screen-based and Physical artefact group scores (%).

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

Each group had three distinct categories of learners. Some learners improved their performance, some showed no change and some performed worse in the post-test. The physical artefact 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. No difference was statistically significant.

5.2 Emotional Response

Emotional response data was collected by the Reflective Emotion Inventory (REI) described in the study design. The results are presented at the level of 10 sub-categories, with the first five being negative and the latter five positive. The intensity scale for each emotion ranged from 0 to 3, where 0 indicated no emotion and 3 indicated the emotion occurred intensely. There is also a space for contextual response about when or why the emotion was experienced. Fig. 5 gives the mean response from each sub-category of emotion for both groups.

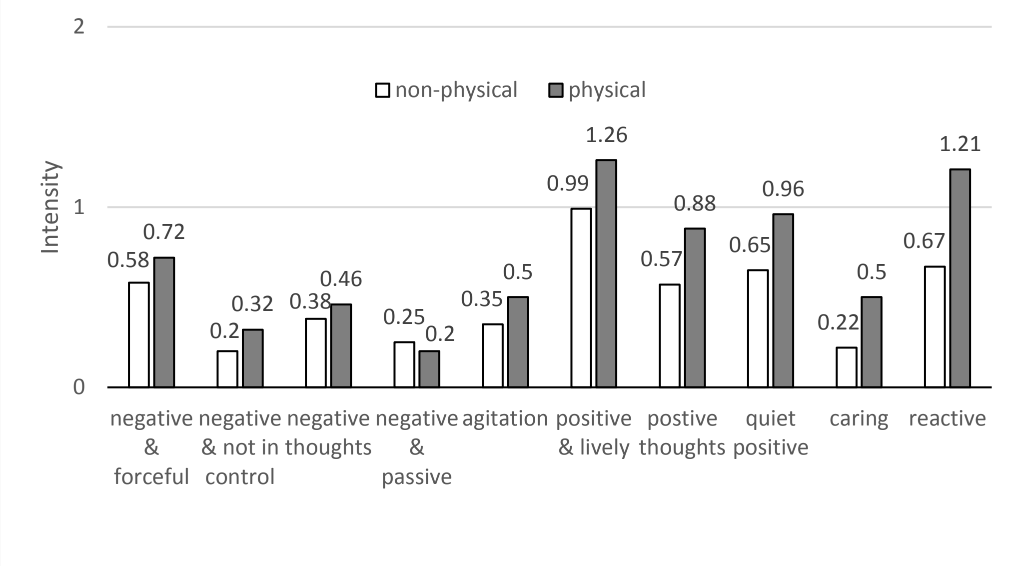

Fig. 5. Strength of Emotional Responses [21].

Screen-based group. This group were less ready to offer free text comments to contextualize emotions than the physical artefact group. They expressed *negative forceful* emotions as the result of problems with the code: "code errors" or "sorting some issues with the program". Several participants intimated feeling envy when other groups had their program working before they did. Several 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" whereas other groups indicated a sense of boredom at being finished early. Positive emotions were largely because of task completion and "getting it working". This was recorded by many participants as feelings of amusement, joy and happiness. The contextualizing of positive emotions was as frequent as that of negative emotions. However, the reasons cited for a positive emotion were far less diverse.

Physical artefact group. This group offered comments for each sub-category of emotion. Negative emotions were attributed to a range of features of creating the game; one of the most frequently cited reasons was wiring. Responses include "when the wires fall out", which is a common problem if jumper wires are not cut long enough or well organized, and "getting the wires in the right places" (the pitch of the breadboards used is one hole every millimeter, which can be problematic). Specific components were mentioned, such as: "getting the LED the right way" and "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 common in 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 artefact 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 some industry standard IDEs. 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 contextualized 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, particularly because of completing the build: "when it worked", and when engaging with their product: "playing the game". People also noted a feeling of happiness at getting their task completed. The second most strongly reported emotional category was *reactive*. This was noted as interest in "learning new things". One participant noted interest in the logic they had arrived at in developing the Whack a Mole algorithm.

Screen-based compared to physical artefact group. When considering the screen-based group's emotional responses grouped together, 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. However in the physical artefact group there was a difference in intensity of positive and negative emotional categories. With the exception of *caring,* all the positive emotions were more intensely experienced than the negative ones. This matched the rich contextual data offered by the physical artefact group. Where learners worked with the physical artefact, they had a strongly positive experience. Two of positive emotions reported by the physical artefact group are notably greater than that of the screen-based group: *positive & lively* and *reactive*. Reinforcing this, an observable difference was noted in the degree of engagement of different groups with the finished artefact. Several participants in the physical artefact group were seen taking pictures and videos to share on social media, indicating a degree of pride and a desire to share their work that was not observed in the screen-based group.

6 Discussion

6.1 Knowledge and Understanding

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 the fact that when both the screen-based and the physical artefact groups are ordered for performance, the top two-thirds of both groups had very high pre-test scores, leaving little room for improvement. It is therefore likely that this study took place too late in the teaching period and offered a substantially reduced opportunity for the interventions to create a change in knowledge and understanding. The groupings for this study also proved problematic. The pre-test data for the screen-based group showed a tight normal distribution centered on a very high mean. The physical artefact group had a slightly skewed distribution in pre- and post-tests, with several particularly weak scores. The two randomly allocated groups thus had different abilities or levels of experience. This may have reduced the sensitivity of the test to detect improvements between groups.

Alternatively, the lack of measured improvement in performance in the Whack a Mole study may also have related to the method of content delivery: video tutorials were used in preference to direct delivery by a tutor. This was in response to findings from previous studies (e.g. [19]) which contrasted the benefits and pitfalls of tightly delivered content versus a much looser learner-led approach. The intention was that the video tutorials would enable learners to control the pace of content delivery. While this was effective up to a point, it was susceptible to the same problem as giving someone a list of directions rather than a map: if they fail to act upon one of the directions, they can become lost. A list of directions also offers no contextual information for exploring different routes or other points of interest along the way. One of the key advantages of facilitating a session directly is the ability to identify and respond to spontaneous learning opportunities. The instructional videos served as narrow routes for learners to take. In addition, using video content to support delivery left the tutor far more removed from the process than when tight cycles of delivery and consolidation were used.

6.2 Emotional Response

Firstly, a low response was noted for the free text component of the REI. This is hardly surprising, given the additional effort required by learners to verbalize 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.

The screen-based group noted envy and friendliness. These emotions represent a competitive situation well, particularly if a tight group has formed which was keen to demonstrate its success relative to the other groups they are working with. The REI also established different emotions about completing the test on time. Differing rates of task completion is evidence that the screen-based group, despite having a high knowledge and understanding pre-test score, still contained a range of abilities.

The physical artefact group experienced almost all of the positive emotions with greater intensity than the negative ones. Most often, the anecdotal references to programming and emotion were focused on negative feelings. The free-text contextualization presented here shows that participants frequently experienced causes of irritation that are often reported in the literature, including obscure compiler and syntax errors. The physical elements mimic many 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 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 programming, compiler error messages are offered to assist the learner. Unfortunately, no such support exists when wiring breadboards. As a result, errors in electrical circuits are often very difficult to identify. It seems counter-intuitive therefore that combining programming and electrical prototyping activities could improve the emotional response to the programming experience. The dominance of positive emotions being reported suggests that this did happen in the Whack a Mole study. This suggests that creating a functioning physical game presented a challenge: completing the task generated an emotional response that outweighed the ‘pain’ endured in working through the task.

One possibility is that this resulted from the different bandwidth of interaction offered by the two systems. In the screen-based group, learners could only interact with a single device via its three components, namely the PC with its mouse, keyboard and screen. Using the physical system, learners interacted with **two** devices, namely the PC via its three components ***and*** with the Arduino via its buttons and LEDs. This may have contributed to the richer more positive emotional response from learners in the physical artefact group. Furthermore, a learner with a low ratio of negative emotions to positive emotions may signify someone who will do well in programming. The ability to take greater pleasure from the competed task than displeasure experienced by the challenges on the road to success may be an important attribute that resonates with Perkin’s findings of movers and stoppers [26].

6.3 Whack a Mole Summary

The Whack a Mole study aimed to explore how learning with a physical device differed from learning with a screen-based equivalent. The original research question posed was: *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 neither measurably contributed to nor hindered learning. The study also indicated that video-based teaching materials do not offer the opportunity for the interaction and subtle response that a tutor can provide in probing areas of difficulty for the learner. 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 artefact group noted a greater intensity of positive emotions associated with the learning experience.

One of the main tenets of constructivism is that learners should engage in projects that are relevant to them and the world around them [35]. If this session were to be adapted to enable a greater degree of flexibility, for example allowing learners to design their own interface for the Whack a Mole 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.

Whack a Mole Limitations. One limitation of the study resulted from the composition of the screen-based and the physical artefact groups: different sizes, different academic abilities and with a non-ideal gender balance. A solution to the problem would be to administer the pre-test and then create groups based on the scores. In this case, that would have disrupted the established groups within the class. The sample size was small and the assessment instrument was new, neither of which allowed repetition and hence more formal statistical analysis of the responses.

One of the challenges with a pre/post-test methodology is pitching the test difficulty correctly to ensure maximum sensitivity to the phenomena being observed, 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 had 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 insight into the differences observed in novice programmers working with screen-based and physical media.

Re-considering the literature, it is worth noting that the sample task learners engaged in for the study by [7] was a traditional CS1-style mathematics-based problem. Although this problem type is valid, it represents what [30] 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 jeopardizing learning. Despite its limitations, therefore, the study prompted further investigation of the effects of motivation upon learning to program. The follow-up study is described next.

7 Digital Makers Study Description

This follow-up field study included additional learner ownership, personalization and purpose. In the Robot Dance and Whack a Mole studies, learners had been constrained to solve a puzzle devised by the educator. In Digital Makers, design decisions were intended to make the product less constrained for the learners. This made it possible for learners to apply their newly acquired programming skills to solve a problem of their own. The Digital Makers study summarized here is reported fully in [20]. There were two stages to the study, which engaged 48 young learners (83% male) from across the country. In the morning, the learners had several iterations of short demonstrations of

Arduino programming followed by enactment by learners to complete the following task: make an LED blink, using a potentiometer to control the blink rate and using a button to make the LED blink when pressed. Then, learners were guided through an idea-generation session.

7.1 Idea generation

Equipped with post-its and marker pens, learners were asked to identify three things that make them excited and note them down concisely on the post-it wall. Learners were then encouraged to bring their post-its to the front and stick them on a predetermined part of the wall that was visible to the group throughout the day: the ‘wall of situations’. The ideas gathered together on the wall served as an information radiator [31] for use later in the day. This process was repeated for things that make them cross and for things that make them stressed. The wall of situations served to identify physical app ideas to focus around later that day. Bringing all the ideas together allowed learners to react to each other’s experiences and stimulated memories and new ideas.

The learners were finally guided through some additional Arduino output devices: servo, speaker and red green blue (RGB) LED. The final example the learners constructed was a red, green and blue color mixer. With a single RGB LED and three potentiometers, a physical color mixer was constructed. In a natural progression, the learners were shown how to group this now quite complex program into a single user-defined function and how to alter this so that the color of LED was specified by three parameters passed to the function. Extending this further and utilizing the random function and bringing in some sound with loudspeakers, playing beeps of a program-specified tone, the learners created a light and sound show.

7.2 App generation

In the second stage, learners were given the chance to self-select groupings and build a physical app utilizing the morning’s teaching. Groups revisited the post-it wall of situations that excite, irritate and stress them to pick several post-its they could relate to and expand upon them. Groups were given three hours to build a physical app based on one of the ideas from the selected post-its, with support available as required. The study ended with groups sharing their idea and resultant physical app with all other groups. Participants were asked then to complete the emotional response questionnaire.

7.3 Reported emotions

The most striking result was that positive emotions were reported as being far more intensely experienced than negative emotions (Fig. 6). It is clear that the physical apps session evoked a rich emotional response from the participants. The contrast between the extent to which negative and positive emotions were experienced is strong. Negative emotions experienced do tie into many of the problems reported in the literature about novice programming. Interestingly, many of the error-prone features of coding do match with those of physical prototyping, with breadboarding being particularly error-prone. Nonetheless, the minor irritations of an error-prone medium were outweighed by the strength of the positive emotions that learners reported. Many positive emotions stem from a sense of overcoming challenges to produce something that works and it is important to note that each participant succeeded in producing something that he or she was able to present to the group. The strength with which learners expressed a sense of pride was further evidence of cultural impact: participants were very proud of their work.

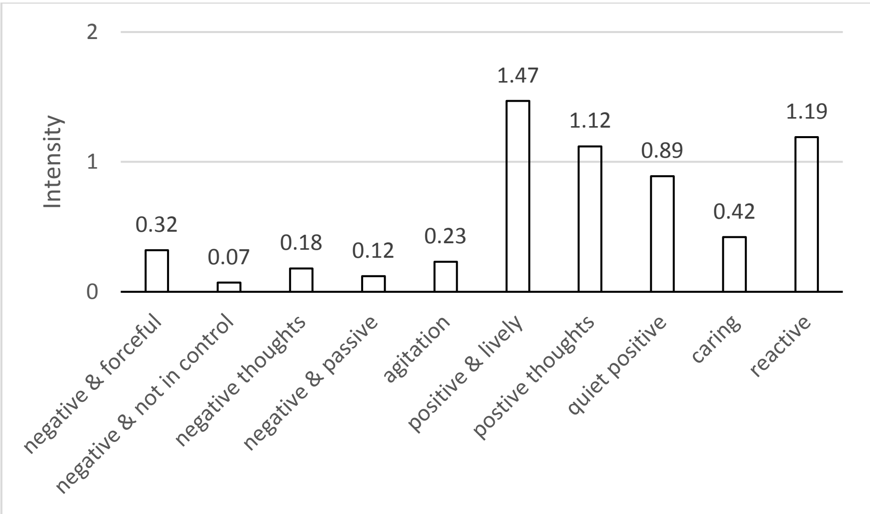

Fig. 6. Strength of Emotional Responses

7.4 Digital Makers Summary

In summary, the Digital Makers study used ownership, personalization and purpose to create a highly engaging learning experience that resulted in strong positive emotional responses from learners. The ideas for physical apps reflected the culture the learners belong to: examples include notifications relating Facebook or personalizing their product by creating toys for younger siblings (police car with lights and sound). The complexity of the builds varied from relatively simple extensions of the demonstrations to complex compositions with multiple sensors and actuators. The only learners not to engage fully with the rich context were learners who had come to the session with an existing knowledge of Arduino and who had premeditated plans for what they wanted to explore. For these learners, the rich context was a distraction to their intentions.

The next section describes how insights generated from Whack a Mole and Digital Makers in particular were synthesized as a set of ‘Learning Dimensions’. The Learning Dimensions follow the style that [16] proposed in their ‘Cognitive Dimensions of Notations’ framework, in which they outline a common vocabulary and reference point for the design and discussion of notations. The Learning Dimensions are a first attempt at capturing and describing key areas for design and decisions relating to learning experiences. They are not intended to be used formulaically, but rather to serve as a source of inspiration and information for educators who are designing or critically evaluating a learning experience.

8 Learning Dimensions

The Learning Dimensions (LDs) bring together findings from the literature and new empirical work that can guide the creation of engaging learning experiences for computer programming. The aim is to provide a resource for computer science educators that can be used either in the design of new learning experiences or as a reflective toolkit for the review and improvement of existing learning experiences. The Cognitive Dimensions framework by [16] has served as a very successful nucleus for a great deal of research relating to notations of many forms including code, sketching, algorithm visualization and musical staff notation. Cognitive Dimensions provided a much-needed common vocabulary that enabled researches to share and discuss insights. It is hoped that the LDs fulfil a similar role for educators in the design and evaluation of learning experiences. As a resource, the LDs are intended to be lightweight, accessible and easy to use. The intention of the LDs is not to present a new pedagogy or theory that tackles all or even most of the aspects of the creation of computer science learning experiences. Instead, the LDs are a set of insights and knowledge from which educators can select to add value and to make informed decisions about their practice. The eight LDs address three high-level aspects: (a) **design and delivery** of learning task, (b) **rhythm or tempo** of the learning experience and (c) **practicalities**. The first of these high-level aspects is described here (the remaining aspects and further detail of each learning dimension is reported in [22]). **Design and Delivery** describes learning dimensions that relate to the design of activities or tasks that make up a learning experience. It consists of four dimensions: Closed versus Open, Cultural Relevance, Recognition and Space to Play. *Closed versus Open* describes the relative merits of designing learning tasks with or without a lot of detail and structure. *Cultural relevance* describes the affordances presented by locating learning tasks within the learner's culture. *Recognition* describes opportunities that arise from enabling learners to share their work. *Space to Play* describes the impact of designing learning tasks that encourage iterative experimentation, for example with peers, and self-directed discovery of knowledge and skills.

8.1 Closed versus Open

LEGO Mindstorms [18] combined micro-controllers and programming with a very adaptable construction kit. The current generation of LEGO Mindstorms has a huge community to support its use. There are globally organized events such as RoboCup Junior (RCJ), in which teams of schoolchildren compete in three subject areas using robots almost exclusively developed using LEGO Mindstorms.

[27] presented empirical evidence for 'back door learning' taking place whilst young children compete in an RCJ event. Interviews with teams were followed up by a detailed case study with one team. Two topics that arose from their inductive analysis were motivation and evidence of learning. One frequently reported reason for being motivated was the 'openness' of the task. Although each event culminated in a competition that identified the best robots, there was no final definitive end point:

> *"you can always improve it and you never have it perfect" Participant [27],*

The *Closed versus Open* Learning Dimension encapsulates the extent to which these activities have a well-defined structure, route and end point. A good example of a closed problem is programming a robot to follow a line. The task defines the answer: there is little scope for the learner to take ownership. Towards the open end of the dimension would be a free choice activity where learners are able to demonstrate competency in a given skill through the creation of a piece of work that is not constrained by the educator, such as creating a robot dance.

8.2 Cultural Relevance

[10] robustly locate the importance of personalization and choice for increasing intrinsic motivation in learners. Their study engaged 72 fifth grade learners (10 to 11 year olds) who were randomly assigned to one of five groups, in a 2x2 factorial design with one control, with the first factor being personalization and the other factor being choice. They found that there was a powerful learning benefit observed in the personalized choice condition. Learners were observed to have not only increased motivation but also displayed a deeper engagement in the task.

Often part of a learning experience involves creating a product of some kind, such as code or a sketch. The *Cultural Relevance* dimension considers where this product sits within the learner's culture. It prompts consideration of whether or not the tasks they are asked to perform are authentic and relevant to their daily life experience. Ownership, personalization and purpose are key aspects of creating a learning experience that will have high cultural relevance for the learner. If the learning experience is divorced from the world the learner inhabits, the cultural relevance will be low.

8.3 Recognition

Many educational programming tools enable learners to contribute to vibrant online communities of learners [8]. Learners can be inspired and informed by the work of others and in equal measure provide the inspiration and support for those who follow them. There are significant motivational affordances to be found in sharing work and observing it being valued by a community of other learners. [27] noted a similar effect: a number of participants identified placing the task in a social context as a factor contributing to motivation. The ability to share ideas and the pride associated with demonstrating expertise was also reported to be important.

> *"It's interesting meeting new people and showing how good you can be" [27].*

It is typical for a learning experience to result in the generation of a product. It may be a program, a sketch or a concept. The *Recognition* dimension considers the potential for the learner to share the product of their learning. As early as nursery school, learners seek recognition from their teachers, peers and parents. A good example of this is pleasure gained from the displaying of work on the walls of the learning environment for all to see. In the Learning Dimensions context, a model of *recognition* has three parts: (a) mode of the interaction, (b) audience size, and (c) response or result. Each of these will have an effect on the learner's engagement and motivation.

8.4 Space to play

The Contributing Student Pedagogy (CSP) [17] enabled learners to have a prominent role in their learning experiences. In CSP, learners co-create knowledge and understanding. This pedagogy moves beyond variations of peer-assisted learning, e.g. [34], and empowers learners to participate actively in all aspects of the learning. In this situation, the educator takes the role of a guide rather than that of a gatekeeper or provider of information. One advantage argued by its proponents is that it mirrors informal workplace learning, in which it is common for co-workers to support each other.

The *Space to Play* dimension seeks to break down the traditional view of a teacher-learner relationship. It encapsulates the extent to which a learning experience offers and encourages learners to explore independently, experiment and iterate over aspects of the learning experience. This idea is rooted in constructivist learning theory [24]: learning takes place best when learners engage in project work that results in an artefact that is relevant to the learners, as described in the *Cultural Relevance* learning dimension. *Space to Play*, however, addresses the fact that space and independence may be intimidating for certain learners. Furthermore, it acknowledges the tension between the learner as an individual and a need to cover a particular amount of content with a group of learners. Where space can be intimidating, direction, constraint and facilitation can be catalysts to creativity and learning.

9 Conclusions

One overarching theme that emerges from the literature is that learning is most fruitful in credible engaging learning experiences that place the learner at the center. Desirable features for education programming tools and learning experiences are increasingly being recognized as those relating to the motivation of the learner, such as personal, social and contextual elements, rather than purely technical elements. Examples include the capacity to tap into and contribute to a community of like-minded learners, and the ability rapidly to make a thing that the learner values. This section presents an overview of the field studies conducted by the authors to uncover and describe features that can stimulate engaging learning experiences in programming.

9.1 Engaging learning experiences re-visited

Robot Dance explored supporting introductory programming learning with programmable robots. A small learning effect was measured as a result of the workshop. Learning in Robot dance was supported by the delivery of several small demonstrations followed by space for learners to experiment with the new skill. This ensured that new skills were always applied to avoid becoming inert [26]. Working towards a performance at the end of the session was a motivator for many of the learners: it showed the importance of recognition.

Robot Dance in the Community observed learners working with no structure imposed on how they arranged themselves. Learners were observed self-organizing into a number of differently groupings with evident emotional engagement in the task of programming the robot. The learners appeared to value the artefact they were working on and be highly motivated by taking control of this small object.

Whack a Mole explored the importance of working with a physical product as opposed to a screen-based simulation. Two similar groups of learners were exposed to identical learning tasks; one group produced a physical computing game while the other group programmed a screen-based equivalent. There was a significantly different emotional response, with physical artefact groups being more engaged with the task and experiencing a greater degree of positive affect.

Finally, Digital Makers explored a more sustained learning experience that afforded learners greater control over the product of their learning and the groups they worked in. There was empirical evidence that the majority of learners engaged deeply in the rich context in which the learning was situated. The emotional profile reported by the learners was similar to that of Whack a Mole. There was a more pronounced demonstration of positive emotions resulting from the physical computing programming.

9.2 Learning Dimensions

Learning Dimensions are distilled insights from existing literature and these four studies. The motivation was to support educators but also present a useful framework for researchers who seek to investigate the learning of programming. A subset of the Learning Dimensions is presented here as version one of an open set of key decision areas: *Closed versus Open, Cultural Relevance, Recognition, and Space to Play*. Future work is needed to provide additional research-driven detail to each of the Learning Dimensions and to identify additional Learning Dimensions.

The difficulties of learning to program have been studied for nearly 50 years and yet many challenges still remain. The essence of programming remains unchanged: it requires that the solution to a problem is described in sufficient detail, without ambiguity, such that a machine can follow the instructions. What has moved forward considerably is the set of tools used to support learning to program: educational tools have matured considerably as a result of the years of research and from the increased capacities of modern computers. It is now recognized that desirable features for education programming tools and learning experiences increasingly relate to the motivation of the learner, such as personal, social and contextual elements, rather than purely technical issues.

The work described here demonstrates that a physical interface can provide a more positive experience than a screen-based equivalent. Designers of learning experiences may wish to include consideration of this insight, and the others as summarized via the Learning Dimensions, when planning the introduction of new programming concepts or creating programming laboratory exercises and assessments. Creating an engaging learning experience is not a mechanical process that can be governed by a set of rules to be followed dutifully to guarantee consistent results. 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

1. Alsmeyer M, Luckin R and Good J (2008) Developing a novel interface for capturing self reports of affect. In CHI'08 Extended Abstracts on Human Factors in Computing Systems ACM pp 2883-2888
2. Amabile TM, Hill KG, Hennessey, BA and Tighe, EM (1994) The Work Preference Inventory: assessing intrinsic and extrinsic motivational orientations. Journal of Personality and Social Psychology, 66 (5), pp 950-967
3. Anderson R, Manoogian ST and Reznick JS (1976) The undermining and enhancing of intrinsic motivation in preschool children. Journal of Personality and Social Psychology, 34 (5), pp.915-922.
4. Bahrick HP, Fitts PM and Rankin RE (1952) Effect of incentives upon reactions to peripheral stimuli. Journal of Experimental Psychology, 44(6), pp 400-406
5. Blank DS and Kumar D (2010) Assessing the Impact of Using Robots in Education, Or: How We Learned to Stop Worrying and Love the Chaos. In AAAI Spring Symposium: Educational Robotics and Beyond
6. Bosch N and D’Mello S (2015) The Affective Experience of Novice Computer Programmers. International Journal of Artificial Intelligence in Education, pp 1-26
7. Bosch N, D’Mello S and Mills C (2013) What emotions do novices experience during their first computer programming learning session? In Artificial Intelligence in Education Springer Berlin Heidelberg pp 11-20
8. Brennan K, Monroy-Hernández A and Resnick M (2010) Making projects, making friends: Online community as catalyst for interactive media creation. New directions for youth development (128), pp 75-83
9. Condry J (1977) Enemies of exploration: Self-initiated versus other-initiated learning. Journal of Personality and Social Psychology, 35(7), 459-477
10. Cordova DI and Lepper MR (1996) Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice. Journal of Educational Psychology, 88(4), pp 715-730
11. Cummins RA 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
12. Deci EL, Koestner R and Ryan RM (1999) A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation. Psychological bulletin, 125(6), pp 627-668
13. D’Mello S (2013) A selective meta-analysis on the relative incidence of discrete affective states during learning with technology. Journal of Educational Psychology, pp 1082

14. Eisenberger R Pierce WD and Cameron J (1999) Effects of reward on intrinsic motivation – Negative, neutral, and positive: Comment on Deci, Koestner, and Ryan (1999)
15. 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 Psychology of Programming Interest Group PPIG 2011
16. Green TRG and Petre M (1996) Usability analysis of visual programming environments: a 'cognitive dimensions' framework. Journal of Visual Languages & Computing, 7(2), pp 131-174
17. Hamer J, Cutts Q, Jackova J, Luxton-Reilly A, McCartney R, Purchase H, Riedesel C, Saeli M, Sanders K and Sheard J (2008) Contributing Student Pedagogy. ACM SIGCSE Bulletin, 40(4), pp 194-212
18. LEGO. 2010. LEGO Mindstorms [Online]. Available: http://mindstorms.lego.com/en-us/default.aspx.
19. Martin C and Hughes J (2011) Robot dance: Edutainment or engaging learning. Proceedings of the 23rd Psychology of Programming Interest Group PPIG 2011.
20. Martin C and Hughes J (2017) Digital Makers: Ownership, Personalisation and Purpose. Proceedings of the 28th Psychology of Programming Interest Group PPIG 2017.
21. Martin C, Hughes J and Richards J (2017) Learning Experiences in Programming: The Motivating Effect of a Physical Interface. In Proceedings of the 9th International Conference on Computer Supported Education, SCITEPRESS, pp 162–172
22. Martin C, Hughes J and Richards, J (2017). Learning Dimensions: Lessons from Field Studies. In: Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education - ITiCSE '17, ACM, pp 299–304
23. Meyer DK and Turner JC (2002) Discovering emotion in classroom motivation research. Educational psychologist, 37(2), pp 107-114
24. Papert S and Harel I (1991) Situating constructionism. Constructionism, 36, pp.1-11
25. Pekrun R, Goetz T, Titz W and Perry RP (2002) Academic Emotions in Students' Self-Regulated Learning and Achievement: A program of Qualitative and Quantitative Research. Educational psychologist, 37(2), pp 91-105
26. Perkins DN, 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
27. Petre M and Price B (2004) Using robotics to motivate 'back door' learning. Education and Information Technologies, 9(2), pp 147-158
28. Petta P, Pelachaud C and Cowie R (2011) Emotion-Oriented Systems. The Humaine Handbook, pp 978-3
29. Pintrich, PR (2003) Journal of Educational Psychology, 95(4), pp 667–686
30. Robins A, Rountree J and Rountree N (2003) Learning and teaching programming: A review and discussion. Computer Science Education, 13(2), pp137-172
31. Sharp H, Robinson H and Petre M (2009) The role of physical artefacts in agile software development: Two complementary perspectives. Interactive Computing, 21, pp 108-116
32. Simpson JA and Weiner ES (1989) Oxford English Dictionary, Clarendon Press Oxford.
33. Skinner BF (1953) Science and human behavior, Simon and Schuster
34. Topping K and Ehly S (eds) (1998) Peer-assisted learning, Routledge
35. Vygotsky LS (1980) Mind in society: The development of higher psychological processes. Harvard University Press
36. White RW (1959) Motivation reconsidered: the concept of competence. Psychological review, 66(5), pp 297-333