THE EXPERIENTIAL QUALITIES OF KINEASTHETIC PROGRAMMING

Identifying the elements of the kinesthetic learning experience of a programming approach based on whole body movements.

Lenard George Swamy
School of Arts and Communication (K3)
Interaction Design, Master's Programme 2019 (Two-Year)

Email: lenardgeorge92@gmail.com
Supervisor: Henrik Svarrer Larsen
ABSTRACT

Moving the body in physical space with both a conscious and a subconscious awareness on the position of the limbs is in itself an engaging experience. This element of engagement has been one of the core reasons for turning to movement based technologies and interactions in the field of education. Using these technologies kids learn complex topics of maths and science at an improved rate of understand. However, one such activity or a subject where there is an absence of these movement based tools is Programming. Kids still use traditional interface tools such as a mouse and a computer to learn and write code.

This is a detailed case study of a 10 week design process developing and studying the interactions with a programming environment based on whole body movement for children. Through a Research through design approach, this study borrows key elements from existing visual and tangible programming tools, concepts of Kinesthetic interactions and child centric design. The investigation is further guided by the methodologies primarily influenced by the principles of Kid Centered Design and the design by movement approach. The design process is characterized by progressive cycles of conceptual design, supported by prototyping and testing. The conceptual design is further evaluated through user studied where I identify key experiential qualities that are inherent to the kinesthetic approach to programming. The aim of this these is to provide these experiential qualities as starting points for further development of tools and technologies inspired by body movements.
ACKNOWLEDGEMENTS

First and foremost, Special thanks to my supervisor, Professor Henrik Svarrer Larsen for his guidance through his explorative ideas, critiques and advice that helped me put the distinct pieces together for this thesis.

Immensely grateful to the kids who spared their Fritids time to provide knowledge for my user studies and to the teachers - Anthony Roscini and Lee Elam at Bladins international School, Malmo, for taking the time out of their busy schedule in organising the workshops. Thanks to David Cuartielles for sparing time to and helping me find an ideal starting point for the technical explorations. All those mentioned shared a willingness to participate in this study in one form or another, with curiosity for the unusual topic and its potential outcomes, making it possible for me to advance with every step of the project.

Last but not least, thanks to my family, who was actively present throughout this study. Many thanks to my classmates, without whose guidance and help I wouldn’t have been able to move forward. Finally, I thank Malmo university for providing the facilities and tools required for this research
## CONTENTS

1. DEFINITION OF KEY TERMS AND CONCEPTS 6

2. INTRODUCTION 8
   2.1. The Design Space 9
   2.2. Research Focus 11
   2.3. Research through Design 12

3. LITERATURE OVERVIEW 14
   3.1. Child Centered Design 15
   3.2. Programming Approaches 16
   3.3. A Kinesthetic perspective 20

4. ETHICAL CONSIDERATIONS 25

5. METHODOLOGY AND METHODS 27
   5.1. Children’s Roles 28
   5.2. Interaction Driven Approach 30
   5.3. Movement for Interaction 31
   5.4. Movement as Experience 32

6. DESIGN PROCESS 33
   6.1. Literature overview 33
   6.2. Methodologies and Methods 34
   6.3. A Basic model of a Programming Environment 34
   6.4. Ideation and Lo Fi Prototyping 37
   6.5. User Study Workshop 1 39
   6.6. Analysis of User Study Workshop 1 41
   6.7. Ideation and Hi Fi Prototyping 46
   6.8. User Study Workshop 2 50
   6.9. Analysis of User Study Workshop 2 51
   6.10. Final Concept 53

7. RESULTS 55

8. REFLECTIONS 57

9. CONCLUSION 59

10. REFERENCES 60
11. APPENDICES

11.1 Appendix A - Codesign workshop 1
   Aim of the workshop
   Structure of the Workshop
   Materials used for the workshop
   Overview of the workshop
      Introduction
      Warm-up
      Body storming
      Programming

11.2 Appendix B - Codesign workshop 2
   Aim of the Workshop
   Structure of the Workshop
   Materials Used for the workshop
   Overview of the workshop
      The New Approach
      Exploration
      Programming
      Informal interviews
1. DEFINITION OF KEY TERMS AND CONCEPTS

**Active learning** - Is a method of learning in which students are actively or experientially involved in the learning process (Bonwell and Eison, 1991). It is learner-centered, not teacher-centered, and requires more than just listening. Students must be doing things and simultaneously think about the work done and the purpose behind it so that they can enhance their higher order thinking capabilities (Active Learning, 2019).

**Affordance** - “An affordance is a relationship between the properties of an object and the capabilities of the agent that determines just how the object could possibly be used” - Norman (2013).

**Blocks** - In programming, blocks are considered to be a group of instruction or commands treated as one statement. A programming language that permits the creation of blocks, including blocks nested within other blocks, is called a block-structured programming language. (Blocks(programming), 2019).

**Kinesthetics** - “is a person’s awareness of the body’s position and movement and is considered as the combination of Kinesthesis and Proprioception. kinesthesis is kinetic motion, while the proprioception is the sensory faculty of being aware of the position of the limbs” - Fogtmann et al (2008).

**Programming** - Refers to a set of rules, written in one of numerous programming languages, that instructs a computer to do what a user wants it to do: perform a sequence of instructions, repeat a sequence of instructions a prescribed number of times, and test whether a sequence was performed correctly (Becker et al, 2017).

**Programming environment** - the development environment that includes the set of processes - compiling and running, programming tools used to create the program and the output of the written program.

**Somaesthetic** - Soma + Aesthetics. Critical study and meliorative cultivation of the soma (the body) as a site both of sensory appreciation (aesthetics) and creative self-fashioning. (Shusterman, 2012).

**Tangible programming** - “Tangible programming” refers to the activity of arranging physical blocks in an intended combination to build (as opposed to “write”) computer programs,
instead of manipulating virtual objects displayed on a computer screen. For example - AlgoBlocks (Suzuki and Kato, 1993).

**Visual programming** - also known as graphical programming, is any programming language that lets users create programs by manipulating graphical blocks on a screen graphically rather than by specifying them textually. For example - Scratch (Resnick et al, 2009).

**Wicked problem** - A wicked problem is a problem that is difficult or impossible to solve because of incomplete, contradictory, and changing requirements that are often difficult to recognize; they problems cannot be accurately modeled and solved using the reductionist approaches of science and engineering (Rittel & Webber, 1973)

**Designerly Activities** - “Holistic approach of integrating knowledges and theories from across many disciplines, and its iterative approach to reframing the problematic situation and the preferred state as the desired outcome of this research“ - Zimmerman et al (2010).

**Design space** - “An emergent mental construct of a world-in-spe that stipulates contours of a field of characteristics and qualities; from concrete artefacts to abstract conceptualisations as well as from hunches to analysis.” - Larsen (2015).
2. INTRODUCTION

Shusterman (2012) considers the body to be a central entity that is necessary for all our human perception, action, and even thought. It is through this body, we interact with the world around us to access the different layers of the meaning (Hummels & Overbeek, 2007). Body movement plays an important role for not just the physical displacement of the body in space but also constructing and expressing the knowledge about the world.

Body movements is also considered to be an important element that adds to the richness of the use experience with a digital products (Hummels, 2000). Today, computational artefacts and information technology (IT) services tend to have more and more ‘human-like’ interactions that uses gestures and whole body movements. (Hummels, 2000). This growing interest in movement-based interfaces and interactive systems can also be ascribed to an increasing focus on new domains such Education.

Interactive tools such as Robotangle (Duffy et al, 2017) mediates the learning of primary school math concepts for kids using technologies such as gesture recognition and projections. The River of Grass exhibit shown figure 2.1, helps kids to understand the interrelatedness of the various distinct elements of an ecosystem. A bodily experience in RobotAngle and River of Grass for instance, can be much more present and palpable when
the user can move around freely in the physical space of experience augmented by a digital interface, compared to exploring a system through an index finger (Hummels, 2000). It is this engaging aspect of movement based technologies that makes subjects like mathematics and science more interesting to learn. Besides the immediate gain of providing engaging learning experiences, movement based technologies also feeds into children’s motor and cognitive development at early stages of their growth.

However, there are certain subjects where the application of movement-based technologies have not been extensively explored and one such subject is Programming. Prominent interactive technologies such as Scratch (Resnick et al, 2009) and Lego Mindstorms (Lego.com, 2019) are used today to teach kids programming in a way that appeals to them but kids still use traditional interfaces such as mouse and keyboard to use these tools. These traditional interfaces affords only a limited set of interactions such as clicking, drag and drop and scrolling, which in my opinion doesn’t completely cater to a child’s needs.

This absence of movement-based technologies to teach kids programming was the motivating force behind the undertaking of this design research. In this case study I take a closer and a critical look at various existing tools used by kids, Child centric design and the role of the body in a learning environment. Following this overview of aforementioned domains, I set out to explore and identify a set of elements that contributes to the user experience of a novel programming approach using whole body movements.

2.1. The Design Space

The proliferation of various educational tools such as Scratch, Lego Mindstorms, Cosmo (Hu et al, 2015), Tern (Horn and Jacob, 2007), Mr Wagon (Chawla et al, 2013) and Algoblocks (Suzuki and Kato, 1993) have brought down the learning barrier to programming for kids. The child friendly design of these tools engage kids in an active learning process where they write code to create stories, control the behaviour of a robot or a virtual character and develop simple games. This child-centric approach taken towards the design of these tools not just makes programming fun and meaningful for kids but also help in the understanding of a certain topic form multiple perspectives (Papert, 1980), (Resnick, 1998). Figure 2.2 shows some of these tools in use.
figure 2.2 also reveals nature of interactions with these different programming tools. The programs on Scratch and Lego Mindstorms are written on a digital screen using a mouse and a keyboard or through one’s finger. In the Osmo programming environment, kids arrange physical blocks in a logical sequence to write code. These interactions are primarily Influenced by a tool’s physical or virtual form. Based on these two different modes of interactions there are two different approaches taken towards programming - tangible and graphical programming.

Despite the differences in their representations, both the tangible and the virtual tools provide similar learning experiences which is attributed to the block programming style. In this style of coding, kids problem solve by arranging blocks in a logical order. The manipulation of blocks happens collaboratively or individually. Figure 2.3 illustrates the design space of some of these programming tools forming the two categories and the underlying similarities between the two.
2.2. Research Focus

Interaction with TUI’s relies on the sense of touch and hand movements. The primary modality of interaction is here is tactile as it involves only the hands and not the other parts of the body. One can hold these tangible blocks, move them and operate their functional parts - knobs and buttons embedded in the blocks.

However, These standard set of interactions rather limit the degrees of freedom with which one can explore programming. Kids between the ages 7-15 like to move around and are physically active (Stewart, 2018) and designing something that employs only single mode of interaction isn’t really ideal as it fails to engage most of their motor skills.

What if the tangible blocks in tangible programming languages required the use of the whole body than just its isolated parts such as hands and fingers? what kind of interactive technologies will be required to support this new third approach? More importantly, what are the elements of the user experience this approach needs to make it a meaningful learning experience for kids? These were the initial set of questions that heightened my curiosity to explore a uncharted area o within this vast design space of interactive technologies used to teach programming for kids.
This third approach to programming (figure 2.4) is fairly an unexplored territory in terms of knowledge relevant to interaction design. It does borrow elements from tangible programming however, in order to identify the elements of the user experience of this approach one must iteratively design and develop tools that would allow one to study the interactions and experiences that emerge when kids use this tool. This requires an explorative design process where kids actively participate at different stages to inform the designer. It also requires an iterative approach to reframing the research focus and the preferred state as the desired outcome. As there’s no straight path from the problem framing to the solution, this wicked problem was handled using the principles of Research-through-Design.

2.3. Research through Design

Zimmerman et al (2010) talks about using Research through Design (RtD) as a research approach to wicked problems that employs methods and processes from design practice as a legitimate form of enquiry. Theoretical and empirical grounding, ideation, iteration and reflection are the elements of a practice used to approach messy situations with unclear or even conflicting agendas (Zimmerman et al, 2007).

These methods and process also lead to creation of conceptual frameworks, Design implications and several other outcomes, and Zimmerman et al, ( 2010) calls this as the Theory for Design. This theory is developed with the intention for improving the practice of design through the creation of research artefacts (Zimmerman et al, 2010). Here research artefacts are concrete embodiments of theory and technical opportunities and produces knowledge for the research and practice communities (Zimmerman et al, 2007). Höök & Löwgren (2012) term the knowledge produced as Intermediary Knowledge as it resides at an...
intermediate level that are more abstracted than particular instances and does not aspire to the generality of theory.

Amongst the various categories of intermediary level knowledge (Höök and Löwgren, 2012) this study produces knowledge that falls under the category of experiential concepts. Löwgren (2007) defines experiential qualities to be concepts that identify desirable properties in genres of products in such a way that other designers can appropriate the concepts and use them to design new products with the same qualities. The experiential qualities in this study are more concerned with the overall kinesthetic experience of a novel programming approach.

Using the aforementioned Research-through-Design approach, I explore the distinct landscapes of kinesthetics, Tangible Programming and Child centered design. Using one of Löwgren's (2007) strategies of designing prototypes with the intention to study the qualities of the new design ideas in use, I employ a generative and evaluative mode of working to produce valuable knowledge.
3. LITERATURE OVERVIEW

This section aims to unpack critical theoretical elements and perspectives to build the necessary scaffolding required for guiding this research through design process. Redström (2017) States that “Theories give a direction, an orientation, a purpose and this also determines what needs to be accounted for and what can be considered the matter of something else”. Under this light, this section explicates the interwovenness of different fields of child computer interaction, the programming approaches, the design of digital tools for this whole body movement as a modality (Figure 3.1).

A user-centric design process requires the designer to first lay down a set of design principles highly relevant for the users. Children being the ones using this approach to learn and understand programs, familiarising myself with critical aspects of Child Friendly design was the first step taken in this design process. Child Centered Design section borrows important elements of experience from the Hourcade’s (2015) comprehensive overview of Child Computer interaction (CCI) and underlying issues with the designing of technological tools.

One of the most important elements of design ability is to build a repertoire of abstracted examples that is to spawn ideas in new design situations (Höök & Löwgren, 2012). In Programming Approaches I chart out the move from text based to graphical and finally to the use of tangible programming languages. In that process, I critically analyse the drawbacks and the success of these similar yet distinct approaches. I also provide an account of how...
certain elements, crucial for the learning experience, can be borrowed from each of these approaches for the design of the third programming approach.

Whole body movement approach to programming requires certain theoretical frameworks that addresses the role of the body in the learning process and the emergent learning experience. Bakker et al (2012), emphasis the use of embodied knowledge while learning abstract concepts. Fogtmann et al. (2008) presents Kinesthetic Interaction as a unifying concept for describing the body in motion as a foundation for designing interactive systems. In Kinesthetic perspective, I describe the confluence of these two approaches which informs the design of the third approach.

3.1. Child Centered Design

A central tenet for user centred design practices is that there is no one design that fits all, but rather design should be driven by knowledge of the target users (Markopoulos & Bekker, 2003). Children being the target user group for this thesis project, it is important to have a review of research on the elements of children’s experiences with the use of technologies concerned with learning (Hourcade, 2015). One must also be aware of the fact that as children grow up their skills, needs, knowledge and their relation to technology change (Markopoulos & Bekker, 2003). In the following sections I highlight a few important elements of the learning experience that was considered for this study.

Child Centered Design

A sensible goal when designing technologies for children is to make them “child friendly” or “child centered”, but what exactly does this mean? what are the feelings that matter in a child’s experience of using a certain technology? Hourcade (2015) states that the goals for technology design should on the positive development across physical, social, intellectual, and emotional dimensions. Striving towards addressing these different dimensions is what adds to the quality of the experience that the child will value based on the context of the experience.

The 4 different dimensions stated by Hourcade (2015), is relatable to the components of Csikszentmihalyi’s (1990) Optimal experiences. Csikszentmihalyi, defines an optimal experience to be a combination of elements like difficulty levels, novelty, sense of control and the merging of action and awareness. Having the right level of difficulty makes an activity challenging at the same time motivates the child to learn a certain task. Striking a good balance between a challenging and an easy activity poses a challenge for designers (Hourcade, 2015). Another component that could help in sparking the curiosity of the kids is to add a novel way of interacting with a technology. Novel interactions along with a
sense of control especially for interactive technologies can sustain the kids at high engagement levels that can lead to rich experiences.

Learning Theories

Piaget’s theories of development and its extension by Papert have had a significant impact on the field of child-computer interaction (Hourcade, 2015). Piaget thought that learning occurs through a process of adaptation, in which children adapt to their environment. He saw this adaptation as an active process in which children construct knowledge structures by experiencing the world and interacting with it. This idea is also referred to as constructivism, where children actively construct their own knowledge through experiences. Seymour Papert later expands on Piaget’s ideas with his proposal for constructionism (Hourcade, 2015) by placing greater emphasis on the social and motivational aspects of learning, as well as on the importance of providing children with more opportunities to modify their environment, instead of just experiencing it (Hourcade, 2015). This is particularly clear in the emphasis on providing children with technologies with which they get to be authors, rather than experiencing worlds and situations that are pre-scripted, or absorbing facts provided by a computer (Hourcade). This learning-by-doing style of exploring a particular topic is a prominent feature in the underlying design of the various tangible and virtual programming tools used today.

3.2. Programming Approaches

The origins of various programming approaches that are available today can be traced back to Papert and Resnick’s work at the Massachusetts Institute of Technology (MIT) on making programming accessible to children.

Text based programming

Papert saw that the power of the computer was its universality and its power to simulate. It was his children’s thinking machine – a machine that enabled children to be builders of their own learning and thinking (Papert, 1980). Based on this idea of a ‘children’s thinking machine’ he developed the first text based programming language called Logo.

Logo was the first educational programming tool that was developed by Papert that allowed kids to control an onscreen turtle to draw geometrical shapes and patterns (Figure 3.2(a)). By arranging a set a set of movement and drawing commands, kids were able to explore, problem solve, experiment – and become immersed in an environment in which they take charge of the computer (Hourcade, 2015). The idea was to create a virtual playground where kids can create and test their ideas through programs. This idea of a virtual playground later
moved to the physical world where tangible objects such as robot were controlled using the same programming language as shown (Figure 3.2(b))

![Figure 3.2. The turtle programming language was used to (A - left) - draw geometric patterns using a turtle and (B - right) to control the movement of a tangible robot.](image)

Logo programming was primarily text based where kids had to type in instructions on to the computer. Typing in lengthy instruction sets was one of the primary shortcomings of text based programming as it places high emphasis on the syntax. This approach made Logo programming a cognitively complex task that required instructional guidance (Hourcade, 2015).

Visual programming

Visual programming languages brought down this cognitive barrier in learning to write code by reducing the amount of typing by the children through the use of easily recognizable programming constructs called *blocks*. These blocks help kids to easily recall concepts (Hourcade, 2015) and in the reduction of syntax errors. Perhaps the best-known current example of a visual programming is Scratch (Resnick et al., 2009), which enables children to select from categorized sets of instructions that can be dragged into a programming area and attached to other instructions. The Scratch programming environment (Figure 3.2 A) allows kids to create animations and interactive stories using the various blocks available.
The design of the Scratch programming language is based on the design of Block Programming. In this type of programming each block of instruction has a certain shape that lays down constraints on its attachment to other blocks. As Hourcade (2015) says, this facilitates the understanding of where new instructions can be placed which subsequently makes it intuitive for the kids to create programs.

Kids interact with these blocks using traditional computer interface tools such as a mouse, keyboard or even a finger depending on the type of the 2 dimensional interface (figure 3.3 B). The inherent problem with the 2 dimensional interface is its inability to engage multiple users at the same time. They often restricts only a single user to interact with the code while others passively observe. As a result Scratch fails to address the social aspects to take turns between different users in a collaborative learning environment. Kids won’t be able to collaboratively tinker with code because of this restriction tools.

Tangible programming

Tangible programming was designed to address this very problem of engaging multiple users which was crucial for a collaborative learning experience. In a tangible programming language such as the one shown in figure 3.4, physical objects are used to represent various programming elements, commands, and flow-of-control structure (Horn and Jacob, 2007). Students arrange and connect these objects to form physical constructions that describe computer programs.
The physical blocks of the tangible programming language are embedded with computation making them capable of helping kids to explore complex topics and a broader set of concepts. Resnick (1998) refers to these digital blocks as *Digital manipulatives*. He identifies these objects as manipulative materials that enable children to explore mathematical and scientific concepts through direct manipulation of these objects. With digital manipulatives, children can help young children learn concepts that were once too advanced for them, especially subjects like programming Resnick (1998). The blocks of AlgoBlocks can be considered as a digital manipulatives based on his definition.

Towards more ‘Tangibility’ in Programming

Horn et al’s (2009) comparative study of graphical and tangible programming tools shows that tangible interfaces offer additional educational benefits as compared to more conventional interaction techniques provided by graphical programming languages and user interfaces. Their study reveals that children are more actively involved in the tangible condition which was a result of the tangibility of the interface (Horn et al, 2009). The tangible interface was more inviting for the children, more supportive of active collaboration, and more child-focused than the traditional mouse-based graphical interface. They also observed that girls were significantly more likely to use the exhibit in the tangible condition and
suggest that tangible programming can be another approach to create more gender-neutral computer programming activities for both formal and informal education. It was the unique nature of the user interface that made tangible languages support more turn-taking that was absent in the graphical programming languages.

3.3. A Kinesthetic perspective

Movement has always been a component of computer interaction. In the early history of HCI, movement was viewed primarily as a means of inputting information into a system using a mouse and a keyboard. With the recent advances in tangible and ubiquitous technology, researchers have developed a variety of techniques that utilize movement for interaction (Levisohn and Schiphorst, 2011). These methods include gestural interaction, full body kinesthetic interaction, as well as the use of specialized augmented objects and controllers have transformed the way that HCI practitioners and researchers view movement (Levisohn and Schiphorst, 2011).

Levisohn and Shiphorst, (2011) identify two aspects of movement that one must keep in mind when design interactive technologies - the experiential and the functional aspects. The experiential aspects of movements looks into the role the body plays in the construction of a lived experience and places emphasis on the sensory and the tacit knowledge. The functional aspects are more concerned with the body’s contribution to linguistic, mathematical and conceptual thinking. An equal consideration of these two aspects of movements in computational interaction has the potential to improve user experience, enhance the fidelity and quality of communication, and produce heightened engagement for users (Levisohn and Shiphorst, 2011). Figure 3.4 shows how these two aspects are related to both understanding programming and the use of technologies to write programs.
The Functional Aspects

The functional aspects of movement addresses the role of movement in the development and structuring of cognition, the acquisition and use of skilled movement, and the semantic nature of movement (Levisohn and Schiphorst, 2011). As this study involves the creation of digital manipulatives used to learn and understand programming, I look into the role of the body in the development and the structuring of cognition, more specifically at embodied cognition.

Embodied cognition emphasises the role of the body in the development and support of human thought processes. In HCI, one of the primary theoretical frameworks incorporating the theory of embodied cognition is the theory of embodied schemata (Levisohn and Schiphorst, 2011). Embodied Schemata is a cluster of knowledge representing a particular generic procedure, object, percept, event, sequence of events, or social situations generated by the perceptual interactions and bodily movements within our environment (Johnson, 2013). These schematic structures that make it possible for us to experience, understand and reason about the world. These structures can also be considered as pre-linguistic constructs.
based on bodily movements, physical orientation, and interaction with objects (Johnson, 2007; Lakoff and Johnson, 1980).

Building on this concept of an embodied schemata, Lakoff and Johnson (1980) introduce the idea of Embodied metaphors. A metaphor is the human ability to project the structure of an embodied schemata onto a conceptual domain (Lakoff and Johnson, 1980). A metaphor allows us to understand or experience one concept (target domain) in terms of another (source domain) (Bakker et al, 2012). When the source domain involves metaphors that have arisen from bodily experience we call them embodied metaphors (Bakker et al, 2012). Figure 3.5 illustrates the relation between bodily experiences, embodied schemata, and embodied metaphors by using the IN - OUT schema. It shows how we apply this IN-OUT schema to reason about abstract domains such as “I am in love” which is more like a metaphor (Bakker et al, 2012).

![Figure 3.5 the relation between bodily experiences, embodied schemata, and embodied metaphors by using the IN - OUT schema (Bakker et al, 2012).](image)

The Experiential Aspects

Our bodies are the foundation for the manner in which we experience and interact with our surroundings (Fogtman et al, 2008). Our sensory feedbacks determine an adequate response to the surrounding environment that provides an awareness of the position and the movement of body in space (Floyd and Thompson, 2009). This kinetic motion and the sensory awareness of the position of limbs is called Kinesthesis. Its this kinesthetic sense that culturally and socially grounds our everyday actions in the world as moving bodies that influences our Kinesthetic experience (Floyd and Thompson, 2009).
Expanding on this basic definition of kinesthetics and the emergent kinesthetic experience that it dictates, Fogtman et al (2008) introduces the foundation for defining for Kinesthetic interactions. They view Kinesthetic interactions (KI) as a unifying concept for understanding the bodily potential in interaction design comprising of physiological definition of kinesthetics, kinesthetic experience and interactive technologies. They give a broad definition of Kinesthetic Interaction as when the body in motion experiences the world through interactive technologies.

The physiological definition of kinesthesis is the awareness of the position and the movement of the body in space (Rasch and Burke, 1971). Movements are produced by the motor system and how well a movement is performed depends on a person’s ability to coordinate and control muscular movement (Fogtman et al, 2008). The motor system comprises of two motor skills - fine and gross motor skills that control these movements. Fine motor skills are defined by the coordination of small muscle movement, which occur in the fingers, eyes, mouth, feet and toes. Interactions involving whole body movements addresses the gross motor skills which is a fairly new and unexplored (Fogtman et al, 2008). It’s also this kinesthetic sense that conditions the manner in which we experience the world in framing our embodied actions, by providing a sense of spatiality and bodily-motor potential in our relation to the physical and socio-cultural world (fogtman et al, 2008). The social and cultural contexts also shapes our movements and our acting in the world which is mediated by the motor memory (Merleau-Ponty, 1945). It is this motor memory that forms the kinesthetic experience by mediating our experience of a cultural and social world, which in turn manifests itself as motor skills and potential for action.

Fogtmann et al’s (2008) describe 7 design parameters that can be used to both inform and evaluate the design of a kinesthetic Interactions. Out of the The 7 design parameters, the 5 most relevant ones are:

- *Engagement* describes KIs that engage users in a kinesthetically memorable manner, and facilitate interested exploration through the body in motion.
- *Sociality* relates to designing for a body among other bodies. By designing Kinesthetic Interaction, the interaction often moves into a collaborative and social place, where others are invited to take part in the interaction, actively or as spectators.
- *Explicit motivation* means that the system tells the users explicitly how to interact with the system. The range of movements is restricted, and there is a direct motivational invitation to react.
- *Movability* is central for an understanding of whether the body can move freely, or is physically restricted while interacting with the system.
- *Kinesthetic empathy* is where specific and controlled movement patterns are affected by the relation to other people, and stimuli from the surrounding environment.
Kinesthetic empathy is achieved when the system opens up for the possibility of the users being able to read, decode and react on each others’ movements.

Further, the choice of technology is closely related to how the system can motivate use in terms of facilitated interaction forms. When interacting with interactive systems there will always be constraints in how the body is involved in the interaction, however, these constraints do not necessarily dictate the design process. Instead, they can act as a substantiating factor in exploring the bodily possibilities.

Combining the two aspects

Bekker et al, (2012) suggest that designing interactive learning technologies, especially the ones that involves body movements, should include a metaphorical link between the body movement and the abstract concepts. Since the learning experience greatly benefits from interaction models based on embodied metaphors, it is crucial to identify the embodied metaphors that underlie how we structure and reason about the targeted abstract concepts. The embodied metaphors contributes to an interaction model that makes it intuitive and natural. This interaction model is further enhanced by applying the design parameters of the Kinesthetic interaction framework. The kinesthetic perspective on interactive technologies makes visible some of the possibilities for more directly addressing the bodily potential for interactive learning systems.
4. ETHICAL CONSIDERATIONS

The design of the programming environment in this study requires the involvement of kids in design activities. Regardless of how involved they are, it is important to consider ethical issues whenever children participate in research or contribute to the design of technology. Some of the critical ethical questions that were considered during course of this research were the selection of subjects, protection of personal data, ownership of ideas, the impact of the design on children and the issues with technologies used by them. These ethical issues have been classified into three broad categories based on values, children’s participation and the issues with designing technologies.

Value Centric

The CHECK tool by Read et al. (2013) design helped me clarify my values and also consider the values of children. My desire to design a third approach to programming wasn’t solely based on the desire to create a fancy piece of interactive technology but based on the importance of movement to the physical, emotional, intellectual and social development of children. This research was also undertaken with a belief that this work will provide a starting point for other designers and technologists to foster the creation of tools that will use body movements as a modality to learn programming.

Participation of children

Participation of children were influenced by being transparent about my motivations and most importantly receiving full consent from both children and their parents. I ensured that they were aware of how their actions and opinions were used so that they make a well-informed decision to participate (Read et al., 2014). It was also important to be observant about any signs of discomfort especially for younger children because they may not always voice their discomfort in front of an authority figure.

Issues with Technology

As this study explores the use of interactive technologies for children it is important to be aware of the potential risks involved as technologies do not always provide advantages to children and in fact may harm them (Hourcade, 2015). Hourcade (2015) describes the physical, intellectual and emotional considerations a designer has to take into account when designing interactive tools for kids. The physical considerations recommendations helped in avoiding the use of objects with sharp edges, toxic materials, choking, squeezing or strangulation hazards. The intellectual considerations raised concerns regarding what type of media they could access, how they accessed it, and for how long they access it. The
amount of time children spend with the technology, and how they experience it (e.g., alone or with a parent providing guidance and feedback) are factors that will all have a significant impact on outcomes. As whole body movements are considered to be the primary mode of interaction it one should be aware of violent and aggressive behaviors that might arise and their harmful consequences.
5. METHODOLOGY AND METHODS

This generative and evaluative study that seeks to identify the experiential qualities of a learning approach to programming that lies at the intersection of kinaesthetics, Programming and Child Centric design. Figure 5.1 shows the relationship of the body to code and the digital manipulatives used to write code informed by theory. This section briefly describes the appropriate methodologies and the corresponding methods that will guide the empirical study in the production of knowledge relevant to the aforementioned areas.

![Figure 5.1. relationship of the body to code and the digital manipulatives used to write code informed by theory](image)

The child centric design of the digital manipulatives is significantly informed by the involvement of children during the design process. *Children’s Roles*, looks into the various roles that children take in the design of technologies that support learning and how these roles in turn will have different influences on the design of the experience. Based on Druin’s (2002) description of the various roles children take during a design process, I identify and discuss the ones more relevant for this design research.

To design a digital manipulative that Kinesthetically engage the Kids, one must start the process by thinking about the various of interactions these manipulatives can foster. In *Interaction Driven Approach*, I briefly talk about the underlying design guidelines of the
existing digital manipulatives, the inherent shortcoming with one specific guideline and how this shortcoming can be addressed when one takes an interaction driven approach to product development.

When designing for a learning experience based on body movements, one has to be mindful about the role of the body in both constructing the knowledge- the functional aspect, and its influence on the experience - the experiential aspect. Levisohn and Schiphorst, 2011 identify two distinct approaches that addresses these two aspects. Movement for interaction addresses the functional aspects of the body based on the theory of Embodied Cognition using the user-centric iterative approach by Bakker et al (2012). Movement for experience describes the use of the Kinesthetic Interaction Framework by Fogtmann et al (2008) in addressing the experiential aspects of Kinesthetic Programming.

5.1 Children’s Roles

User-centeredness is often more challenging when thinking about children, especially at young ages (Hourcade, 2015, p. 38). Along with the awareness of child centered design and learning theories, the consideration of how children will be involved in the design process is also important for the design of technologies (Hourcade, 2008). Druin’s (2002) classification of the various levels of participation of children in the design process helped me determine The degree to which they were involved in the design process. In this section I describe the rationale behind involving children as user’s, tester’s, informants in the design of the third new approach to programming.

Children’s roles as users

The inclusion of users from the very beginning of a design process is a key element to user centered design. Children’s participation as users at the beginning of the design process can reveal insights into their needs, abilities, preferences and help the researcher better understand situations in which technologies could aid children (Hourcade, 2015, p. 39). Druin (2002) suggests observation as a primary method for identifying patterns of activity, and general user concern in regards to the performance of a certain task. She suggests the use of Video cameras can to capture data such as gestures, movements and other subtle behaviours for later analysis.

Children’s roles as testers

Towards the end of the design process, children can play the role of a tester, where children test prototypes of emerging technology (Druin, 2002). As testers children can inform the designer about usability issues for the revision of prototypes. While testing prototypes often
brings up the idea of testing a fully developed technology, this need not be the case. Children can test low-fidelity prototypes (e.g., paper sketches), high-fidelity prototypes (e.g., interactive, but not fully functioning), and fully functional technologies at each design iteration (Hourcade, 2015, p. 39). Techniques such as Wizard of Oz prototyping (Molin, 2004) can also be used to test early ideas and prototypes. Qualitative surveys such as interviews were conducted to reveal, dislikes, difficulties, and interest areas after the use of technology. This helps to clarify children’s motivations and pinpoint specific reactions to particular content.

Children’s roles as informants

The main drawback of the above mentioned approaches is that children do not directly affect the design of the technology as it is being designed and provide no feedback until the work is completed. Building on the notion of informant design (Scaife et al, 1997) Druin, 2002 considers children playing the role of informants. As informants, children share ideas and opinions with the design team acting as consultants at key points in the development and design process. This role provides a compromise that enables children to contribute their ideas to the design process and at the same time is flexible enough that it works for short-term projects. There are numerous ways to bring children as informants into the design process. At the start of a project or product design, teams may decide to observe children using existing technologies. In this way, design directions may not necessarily be expressed directly by children, but may be implied by their actions. Children can also give ideas and share opinions after testing a certain technology that can affect the design process. Once the technology is developed, children may again offer input and feedback.

Children as design partners

There can also be partnerships with children who are experts in a particular topic (Druin, 2002). Druin (2002) states, “As design partners, children can be a part of the research and design process throughout the experience. With this role, children are considered to be equal stakeholders in the design of new technologies”.

Having children acting as designers requires them to be engaged in the design just as much as the designer as it takes time for children to realize their ideas can actually be included in real products. Another challenge in including children as design partners is the time needed to develop a multi-generational design team. This it may take time for some children to fully understand what they are supposed to do in specific activities (Druin, 2002). In most cases, children do not make valuable contributions on a regular basis until they have been part of a design team for several months (Druin, 2002). In light of these challenges, children weren’t involved as designers in the process.
5.2 Interaction Driven Approach

The research on digital manipulatives by the Lifelong Kindergarten group is guided by three underlying principles:

- Encouraging design projects where students can create external artefacts and products that they can share and discuss with others
- Leveraging new media that will help students rethink and create new representations and formulations of knowledge or a subject
- Facilitating personal connections that will help learners form new relationships between the tools and the learner's interest, passion and experiences.

Amongst the three design principles, it's quite evident that the design projects and the new connections that will be formed is dependent on the design of the digital manipulative. Therefore, this study looks closer into how leveraging new tools and media affect a child's interaction with these digital manipulatives without disregarding the purpose of these manipulatives.

Maeng et al, (2012) identify two different starting points to an interactive product development process - Technology driven and Interaction driven product development process. According to Maeng et al, (2012) in the technology driven process the starting point of the design is new technology. The ideation sub patterns that emerge depending on the characteristics of the technology, are namely function related or interface related. In a function related subpattern, the ideation process was linked to the development of ideas around the functions of the product. Where as in an interface related pattern, the technology forms an interface (Touch screen, flexible displays) and the ideations are about the product domains and needs (Figure 5.2. (A)).

On the contrary, an interaction driven product development process considers the movements of interactions as the starting point for designing an interactive product (Maeng et al, 2012). In comparison with the previous, technology driven approach, movements created the greatest amount of possible ideation subpatterns, such as definitions of movements, contexts related to actions, interactions of different products, possible form factors (physical structure) and etc.
The design principle of leveraging new tools and media means that the design of the digital manipulatives will take a technological driven development process. Without identifying detailed movements of interaction and merely considering the broad domain of technology will cause complications in extracting new ideas on how these digital manipulatives will foster whole body interactions. A more appropriate starting point for generating more ideas of the different form and function of these digital manipulatives would be the movements. With the interaction driven approach one could generate a wide array of new interaction style. The interaction driven approach will determine the product element to apply movement and will also help me identify various programming elements that will influence movement and vice versa.

5.3 Movement for Interaction

The Moving Sounds Tangibles by Bakker et al, (2012) case study is similar to this case study in terms of using embodied metaphors. It looks into the design and implementation of interactive learning systems with embodied metaphor-based mappings (Bakker et al, 2012). Due to the limited availability of guidelines for the identification and translation of these metaphorical mappings into an interaction model, identifying these mappings used by children in their understanding of the targeted abstract concepts as well as translating them effectively into interaction models was not a straightforward process (Bakker et al, 2012).

An interaction model specifies the mappings between input action and output response. Bakker et al (2012) present a people-centered, 5 stage iterative design approach to the design of embodied-metaphor based interaction models.

1. *Enactment studies* to identify applicable embodied metaphors
2. *Creating low-fidelity prototypes* based on embodied metaphors, to explore the input design space
3. *Evaluating low-fidelity prototypes* to validate the input design space in terms of affordances which support embodied schematic movements
4. Creating high-fidelity interactive prototypes with suitable affordances, to explore the mapping between the input design space and metaphorically linked output responses.

5. Evaluating high-fidelity interactive prototypes, to validate the input design space, embodied interactional mappings and output responses.

Using the above approach as a backbone, I structured my own approach more relevant to the context of this study. More specific details on how this approach was used is described in detail in the User Study Workshops 1 & 2 sections of the Design process.

## 5.4 Movement as Experience

The approach used to address various aspects of kinaesthetic experience is influenced by Hummels & Overbeeke’s (2007) work on movement based interactions. They suggest that if one likes to design for movement-based interaction, one has to be or become an expert in movement, not just theoretically, by imagination or on paper, but by doing and experiencing while designing. Incorporating movements as a part of our design process helps us see interaction design as choreography, putting together a sequence of events in a dynamically infolding experience (Hummels & Overbeeke, 2007).

Márquez et al’s (2016), embodied Ideation methods such as Bodystorming and Embodied Sketching helped in sketching the experiential aspects. Body storming is an activity where the designer brainstorms on the different movements possible with an artefact or a physical space (Márquez et al, 2016). Where as embodied sketching can be considered as an activity that happens in a situation that closely resembles the design context thats been addressed Bodystorming was used frequently (Márquez et al, 2016). Bodystorming was mostly used by the designer in an early ideation phase, much before the creation of hi fi prototypes. Embodied Sketching was frequently used with children during the user study workshops.

Márquez et al’s (2016) embodied ideation methods are more appropriate for stages before the development of fully functional prototypes however they don't address how one can continue to use these methods through the development of the hi fi prototypes. Buxton (2010) talks about sketching as exercising the imagination and understanding through the materials used. By materials he means any physical forms such as 3D or sculpture and suggest that they can take on even more extended forms such as technologies that will be used. Combining Buxton’s (2010) notion of Sketching with Márquez et al’s methods one can develop hifi prototypes based on body movements.
6. DESIGN PROCESS

A design is always a *compositional assembly* - in other words, made up of unifying relationships and connections between elements (Nelson and Stolterman, 2003). Nelson and Stolterman (2003) state that to design is to be creative and innovative; to relate and connect the individual elements that makes it to stand together as a unified whole - a compositional assembly.

The design knowledge produced as a result of this design research is a compositional assembly of various theoretical frameworks and methodologies. Schön(1992) considers design knowledge as knowledge in action that is both generative and reflective, revealed in and by actual designing. Based on this notion, each stage of the design process is a reflective conversation with the materials- real, physical, virtual, theoretical and the design situation. This section of the report describes the various critical points of the journey taken towards the research goal of producing knowledge on the experiential qualities of Kinesthetic programming.

![Figure 6.1. Overview of the design process model](image_url)

6.1. Literature overview

This stage gave me a good theoretical grounding from the fields of Child computer interaction, kinesthetics and programming approaches. The research first started off by identifying the core elements of user experience appropriate for children by taking a closer
look at Csikszentmihalyi’s (1992) *Optimal experiences* and Papert’s *Learning theories*. Based on a detailed analysis of programming approaches and on Horn et al’s (2007) comparative study of tangible programming, I borrowed important elements of tangible programming due its engaging and collaborative nature. These elements of tangible programming are incorporated in the digital manipulatives of the kinesthetic approach. The role of the body in learning and creating the lived experience is informed by theoretical frameworks of Embodied cognition (Johnson, 2013) and Kinesthetic Interactions (Fogtmann et al, 1993).

### 6.2. Methodologies and Methods

Followed by the literature review, identifying the relevant methodologies and methods was crucial to the design process. Determining the involvement of children was based on Druin’s (2002) classification on children’s roles. The children played the role of an *user* for the first user study and as a tester for the second user study. They were also encouraged to share their opinions and ideas, playing the role of informants in both the user studies. Certain important design decisions were based on their inputs from both the user studies.

The design approach to developing the digital manipulatives was influenced by Maeng et al, (2012) interaction driven approach in combination with Design by movement approach by Hummels and Overbeeke (2007). Several Embodied engagement methods such as body storming, embodied sketching were used right from the beginning till the end of the design process. Through the design-by-movement approach I gained a good understanding of children’s body language, interaction style, emotions and sensations with respect to different aspects of programming and the interactive technologies used.

### 6.3. A Basic model of a Programming Environment
Based on a technical analysis of previous programming approaches, I extracted a general characteristic structure of a programming environment prevalent across all the existing approaches. Having an idea of this underlying structure was important as it had its own set of important constraints for both the designer and kids. This programming environment also influences the context by dictating what a child could do and cannot do. A programming environment can be divided into two broad sections - input space and the output space.

The Output Space

The final output representing the collective behaviour of a set of instructions manifests itself in the output space. This manifestation takes various forms, mainly tangible and virtual. From the previous approaches, kids write programs to control objects in the virtual space (Figure 6.2 B) or in the tangible space (6.2 A). This study focuses more on controlling an object in the virtual space primarily due to its ease of development with limited resources.
Figure 6.2 A (Top) Mr Wagon project by Chawla et al (2013) and its constituent blocks. B (Bottom) The Osmo Programming environment by Hu et al (2016).

The Input Space

This space consists of the blocks that controls an object in the output space. These blocks can be of the virtual or the tangible form (figure 6.2). The different behaviours of an object in the output space often determines the different functional blocks required. For example, the Osmo programming environment, has the blocks “run,” “jump,” and “grab,” to guide a tiny monster named Awbie on his eternal quest for more strawberries (Hu et al, 2015).

The basic version of the programming environment will consist of a Start, Move and Repeat blocks (figure 6.3). The Move blocks will allows a certain character to move or jump in a certain direction by a unit. The Repeat blocks will repeat a set of Move blocks placed between Start Repeat and End Repeat for n times. The Start block compiles all the instructions and starts to run the program once they have all been arranged.
The Act of Writing a Program

Children first think about the problem presented to them before they start manipulating the blocks. Some manipulate the blocks as they think. They arrange the blocks (both move and repeat blocks) in a logical order for the intended output or solution. Once they have arranged the blocks, they instruct the computer to compile the blocks and run the program.

6.4. Ideation and Lo Fi Prototyping

Based on the above model, a basic version of the programming environment was created along with a set of tangible objects for the input space. This programming environment was built for the user study in the next step of the process.

Programming environment V1

The first version of the programming environment was emulated using simple online games and lofi paper blocks. Objects from two online games such as the yellow block from Temple Trouble and Pacman from the Pacman world (figure 6.4) served as characters to be manipulated in the output space. Temple Trouble had a 3 Dimensional game environment, whereas Pacman had a 2D game environment.
For the input blocks, I analysed the different movements and manipulations of the characters from two games and extracted a set of behaviours and actions to control the characters. Figure 6.5. represents a set of basic blocks required for first version of the programming environment.

The Lo-fi Prototypes

The lo fi prototypes consisted of random objects of various shapes - cylinders, cubes, ropes and bendables; and materials - styrofoam and cardboards (Figure 6.6). While I chose these objects I also ensured that they engaged the gross motor skills through a quick body storming session in relation to the programming environment. As I body stormed, I had to be mindful
about engaging the different parts of the body. The light weightedness of these objects also mattered so that they could be moved around easily especially in classrooms for small sizes. For the free flow of interaction ideas I avoided using any technology with these prototypes.

6.5. User Study Workshop 1

The main goal of this workshop was to Identify the embodied metaphors in relation to the programming environment and exploring the input design space using lo-fi prototypes.
Structure of the Workshop

The workshop was divided into 4 sequential stages - Introduction, sensitising, body storming and programming (Figure 6.7). The first half focused on sensitising the kids for effective participation and involvement in the second half of the workshop. The second half was based on Bakker et al’s (2012) first two stages of their 5 stage design that was composed of enactment activities with the lo fi prototypes. This helped in the identification and mapping of the embodied metaphors used to manipulate the characters of the programming environment.

Figure 6.7. The Structure of the User Study Workshop 1.

Overview

After a quick introduction of the research study, the kids were asked to move around, stretch and say one word in relation to programming as they moved. They were then introduced to bodystorming along with a few programming concepts and were asked to enact movements to represent these concepts using movement cards and the lo fi prototypes. In the programming stage, they wrote programs using the lo-fi prototypes and the paper-blocks to manipulate the objects in Temple run and Pacman. The wizard of Oz technique was used here to decode the program they have written and move the objects in the output space. The kids not only played the role of users but they also shared their opinion on the lo fi prototypes. They were observed for patterns of activity an the performance of certain movements using video cams. Figure 6.8 shows some of the activities from this user study.
6.6. Analysis of User Study Workshop 1

The videos were analysed based on Bakker et al.’s (2012) 4th stage of their design approach for behaviours (sequences of actions) that enacted the schematic origins of these metaphors. This analysis was also evaluated using the kinesthetic interaction framework (Fogtmann et al., 2008) along with the elements of Child-Centered design. This section describes the reflections for each stage of User Study 1.
Sensitising

The sensitizing phase showed the extent to which the kids were aware of programming as they looked for connections and relations. Some kids who have already had some previous knowledge in coding already know more than what I knew when I was their age. The ones who didn't have previous knowledge in coding knew some programmed artefacts. The sensitizing activity also helped in the transfer of some basic knowledge from the kids who knew coding to the ones who didn't know.

Body storming

When they body stormed without movement cards, all of their movements were the same as one kid influenced the other which affected their creativity in coming up with movements. This changed when they were given the movement cards (figure 6.9 A) as it made them more engaged and aware of their movements (figure 6.9 B). The lo-fi prototypes helped them to better express their move adding to the level of engagement. Through these movement cards I discovered how some parts of the body - elbows, knees, shoulder made it difficult for the kids to come up with different movements and to express themselves. Even though some enjoyed performing these movements some were disappointed as they weren't able to do the movements.

![Figure 6.9. A (left) - Movement cards highlighting the parts of the body that needs to be moved. B (right) - A boy ideating a movement using his knees and a lofi prototype.](image)
The design of the lo fi prototypes should support movements that are easier for the kids to perform. This ease of performing the movements was an important element that emerged at this stage.

The stretchable objects were appropriate in a plethora of ways by kids due to flexible form (figure 6.10). This flexibility allowed them to express themselves more freely than other rigid objects. There weren’t any set of characteristic metaphors that emerged from using these objects.

Figure 6.10. Children appropriating a flexible objects in different ways

Some objects can have some harmful consequences and this can only be discovered when they are in use by kids. This could be considered as a fun movement, but not when kids have to move around in space. It’s better to avoid using objects like this in the first place, than instructing them.
Programming

The ‘sequential execution’ of programming was represented in the way kids arranged themselves. When they were given a problem to solve they discussed amongst themselves and arranged themselves in a logical order, sideways (Figure 6.12).

The nature of the output space that they tried to control affected the way they moved. For example the movements for the “move up” block for Temple Trouble were different from the movements of the same block for Pacman. The sideways movement (Move Left and move Right) remained the same for both the environments. This revealed that the orientation of the object in the output space affected how the kids oriented their movements (Fig 6.X). The
embodied metaphors were more connected to the way the objects moved in the programming environment orientation.

Figure 6.13. The movement direction of the Move Up block along the different axes for (A) (top) - Temple Trouble and (B) (Bottom) - PacMan

On playing Pacman, the kids wanted to move around in the space around them in order to control the object. They were disappointed when they realised that this wasn’t considered in the design of these blocks. The idea of spatial movement became another important design element that was included in the next iteration of the programming environment.
The Embodied metaphors

The left and right directional movement were the same, where the kids moved sideways in space or just a single body part sideways in a certain direction. (Figure 6.14) The *Move Up* and *Move down* action was dependent on the orientation of the object in the output space (Figure 6.13). However, they were unclear about the repeat action once it came to programming the object to move in the output space.

![Figure 6.14. Two different ways in which kids implemented the Move Block to 'write' move left.](image)

6.7. Ideation and Hi Fi Prototyping

This section describes the reiteration of the output space and the translation of certain embodied metaphors into the input design space of programming environment V2.

Programming Environment V2

The output space has a close resemblance to Pacman due its nature to provoke spatial movement. Sid the spider was a game where the character Sid had to be programmed to reach the fly along a predefined path shown in grey (Figure 6.15). In some levels Sid had to
make a “cobweb” to trap the fly. There were about 8 different levels with increasing level of
difficulty.

Figure 6.15 Two different levels of Sid the Spider

The directional movement of Sid could be controlled using a Move block. To repeat a set of
instruction, the repeat block was used. The functionalities of these blocks remained the same
as the initial proposed programming model.

The Hifi Prototypes

The Move and Repeat blocks were sketched through bodystorming and rapid development of
technologies (Figure 6.16). The embodied metaphor of the Move action was taken into
consideration as the Hifi prototype of the Move block was developed. The sideways
movement was easily translated to an interactive digital manipulative using an
accelerometer and an arduino. To set a direction to the block, the child needs to move the
block in a particular direction. Four LEDs were used to show the direction the block has been
set to by the user’s movement. A piezo was included as an additional sound feedback to
notify the user of a change in movement.
As there wasn’t any prominent embodied metaphor revealed for the abstract concept of looping or with the use of the repeat block, I used the metaphor of a circular motion as it was highly appropriate to the action of a loop. This circular motion was combined with idea of moving in space that the kids wished for in the previous user study.

The repeat block consists of set of two blocks that needs to be moved between the Start & End Repeat block. To perform a Repeat operation, the kid needs to first enclose a set of Move blocks with the Start and End repeat blocks (Figure 6.17). For the move blocks to be repeated twice, the kids go around those blocks in circles exchanging the repeat blocks between the Start and End repeat blocks twice. Figure 6.X illustrates the development of the repeat block using a similar technological setup as the move block.
Figure 6.17 A () The sketching process of the Repeat blocks
6.8. User Study Workshop 2

The aim of the second workshop was to evaluate the high fidelity representations of the kinesthetic programming environment. The evaluations focused on the embodied metaphorical mappings between the input and output design space and identifying the elements of the emergent kinesthetic experience.

Structure of the Workshop

The structure of this workshop was primarily influenced by the last stage of the 5 stage development process used by Bakker et al. (2012) that consisted of three sub stages - explorations, testing and interview. An introductory stage was added to introduce the new programming approach to the kids before moving on to the following stages.

Overview

The introduction phase was followed by a 10 min exploration phase where the kids tried out the hi Fi prototypes of the Move and Repeat blocks. As they explored they tried out writing simple programs that made Sid move along a specific path. Once they gained a basic
understanding of the programming environment, they were asked to solve 7 programs. The workshop ended with an informal interview session where they spoke about what they liked and disliked. The kids played the role of both testers and informants throughout the workshop.

6.9. Analysis of User Study Workshop 2

The analysis and reflection of outcomes in workshop 2 against the backdrop of the kinesthetic interaction framework (Fogtman et al, 2008). This section describes the reflections for each stage of User Study 2.

Explorations

When the kids first explored the Move block without any connection to the output in the game, they appropriated the block in different ways and there was hardly any similarities in their movements. Some turned it, some held it with one hand and a few moved sideways. This changed when they were asked to move the blocks based on the character in the Output space and solve two sample levels of the game, is when the movements matched and a pattern started to emerge. Figure 6.19 A illustrates the pattern of interaction with the Move left block.

The kids realised the need for the repeat block, however they didn’t know how to use it initially. They quickly grasped how it worked after showing them how it supposed to work. More importantly they liked this aspect of spatial movement while interacting with the repeat block, however they didn’t necessarily go in circles as it was intended. Figure 6.19B Shows two kids moving around with the repeat block.
The interaction with the move block was intuitive as the embodied metaphor was directly translated into the interaction model. Even though the repeat block was the least intuitive block, it didn’t mean that the kids didn’t like the interaction. They were fond of the spatial movement using the repeat blocks that even made them use the repeat block often. This Element of novelty in the interaction motivated the kids to move and solve more programs without getting bored.

Writing Programs
As the size of the programs were small, the kids had enough space to arrange themselves and perform the movements freely without any space limitations. When a few kids wrote
programs there was more space to move and perform the movements. However, more use of blocks means more kids participating which can lead to the reduction of physical space around them to move.

Difficulty levels matter a lot when it comes to keeping kids engaged throughout the programming session. For levels that are easy, kids not only perform the movements quickly but they also idly wait until the program finishes executing the instruction. For levels that are challenging, they move while they discuss and problem.

Kids can get easily distracted if the digital manipulatives have too many features built on to them. For example, the LEDs in the repeat block were made to blink constantly when they were moved from the placeholder. This blinking sequence caught the attention of the kids and reduced the rate of movements as the kids started fidgeting with digital manipulative.

6.10. Final Concept

The experiential qualities of the kinesthetic programming approach is a fusion of elements from child-centric design and the kinesthetics experience. Csikszentmihalyi (1992) Optimal Experience and Papert’s notion of Learning-by-doing governs the child-centeredness of this approach. The ‘kinesthetic’ perspective to this programming approach can be further evaluated using the design parameters - Engagement, Movability, Kinaesthetic empathy & sociality stated by the Fogtmann et al’s (2008) framework. The combinations of the aforementioned frameworks was used to evaluate the user studies based on which I arrived at a final concept of the program environment.

The programming environment

The programming environment consists of a 2D virtual output space (figure 6.20 (A)) and some Move, Repeat and Start blocks to manipulate the character in the virtual space. The Move instructions are written by moving the digital manipulatives in a certain direction. The Repeat instructions are written by moving the Repeat block between the Start and End repeat blocks (Figure 6.20 (c)). Kids arrange themselves in a logical order to write a complete program. Once they have finished ‘writing’ they use the Start block to compile the instructions and run the program.
Figure 6.20. Components of the programming environment; A (Top left) - The output space; B (Top Right) - the Move, Repeat and Start blocks of the input space; C (Bottom) Illustrations showing a set of kids writing code in a kinesthetic programming environment.
7. RESULTS

I initially set out to explore the area of kinesthetic programming with the intention to create an approach to learn programming using whole body movements. During this process is when I realised that it was both the design of digital manipulatives and the identification of experiential qualities of an approach that would provide a meaningful learning experience for children. It is this design knowledge and the research artefact through which this knowledge was produced, that are the valuable contributions of this thesis. This section describes the elements of user experience that have been identified through a RtD approach taken in the exploration of Kinesthetic programming for kids.

Elements of the User experience

Movability
The ‘Movable’ aspect of the interactive digital manipulatives was the key to make this approach appealing to the kids making it a critical element of the experience. Kids moved not when they wrote code and even when they were thinking about how to problem solve. Besides the movements that physically restricted them, they were also fond of moving around in space to enact a certain concept of programming or to control the character in the programming environment. This spatial aspect to movement also made kids enthusiastic towards solving more programming challenges.

Engagement & Difficulty levels
The ability to engage kids in a kinesthetically memorable manner was dependent on both the intuitive nature of the digital manipulatives and the increasing difficulty levels of the programming environment. With this gradual increase in the difficulty of problem solving, kids got more involved and interested to identify solutions by using their bodies.

Novel Interactions
When a new programming concept and a movement was added as they leveled up, they were more eager to use the new concept just to try out the new move. This happened with the use of the Repeat block that required them to move spatially. Their fondness towards moving around in space made them use repeat block often. This element of novelty in combination with heightened their interest and engagement levels no matter how challenging the level was.
Sociality
As kids discussed to arrange themselves in a logical order to solve a problem, they became more social. As it required multiple kids to write a program, they frequently discussed on how to solve the problem. Having this collaborative approach to coding also made it a fun learning experience.

Kinesthetic Empathy
As there were a limited range of meaningful movements such as directional moves for the Move block and circular spatial movement for the Repeat block kids read, decoded and reacted based on each others’ movements quite easily. This kinesthetic empathy amongst the kids was also a key element that helped kids share a common language based on which the collaborations happened.

Explicit motivation
The problem in each levels determined what movements that kids were supposed to do. The design of the digital manipulatives formed a second layer of restrictions in how the kids could perform these movements.
8. REFLECTIONS

The design knowledge produced was a result of a Rtd approach taken towards the production of research artefacts. The research artefact that’s produced here is the programming environment through which the interactions were studied. It was also a result of combining various theoretical frameworks and empirical methods. Therefore one can also say that the way this research artefact was produced was an important factor in determining the knowledge outcomes of the design process.

The design of the interaction model of the programming environment, was the combined result of two different frameworks - Embodied Cognition and Kinesthetic Interactions. Embodied Cognition addresses the functional aspects of the body whereas Kinesthetic interactions addresses the experiential aspects. For designing a concept that mainly helps in learning a specific subject or a skill through the involvement of the body, one must give equal consideration to both these aspects. As designer it is easy to solely focus on the experiential aspects of Kinesthetic Programming and leave out on the functional aspects. But from the results we could see that it was the underlying embodied metaphors that influenced the concept design of Kinesthetic Programming. One can also end up exploring the cognitive aspect, and leave out on the kinesthetic experience. To strike a good balance in the use of both these aspects, the designer had to constantly shuffle between these two keeping the end users of this approach in mind.

Another critical design decision taken towards the development of the digital manipulatives was to start with ideating about the interactions - the different whole movements, that will be used to interact with these manipulatives. This interaction driven approach helped me identify a set of meaningful movement patterns by contrasting them with the movements that could possibly be inappropriate or even potentially harmful especially for kids. These inappropriate or harmful movements tend to emerge early on in the ideation phase were kids have the freedom to explore different movements without any restrictions. If not for the interaction driven approach it would have been difficult to identify these movements during the design process.

Initially I had stated that the exploration of a new Kinesthetic programming approach required the active participation of children throughout the design process. This meant that children were meant to be involved as designers (Druin, 2002) in the process. However this changed over the course of this study due to practical reasons and children could only play the role of users, testers and rarely as informants. Had it been design partners, they would have been involved in the design process just as much the designer. This change in role
would have caused subtle effects on the outcomes of this case study. One of the places where children could have influenced the process is during the selection of lo-fi prototypes instead of a selection based on my subjective opinion. There could have also been a design workshop where kids could make their own lo-fi prototypes through body storming. Another phase were they could have played a role as designer is during the ‘sketching’ of the Hi-fi prototypes along with the methods for live prototyping. Their involvement in the aforementioned stages could have had an influence of the emergent experiential qualities of Kinesthetic programming.
9. CONCLUSION

In today’s highly digital world, programming has become a form of literacy that provides a comprehensive knowledge required for the design and development of products, tools, platforms and services. To better prepare learners from a young age, an increasing number of school leaders and technologists are making the case for embedding coding into K–12 curricula to help students build and develop this literacy (Becker et al, 2017). The literacy can be taught in a plethora of ways using the existing traditional interface tools, but a more appropriate one that would be rather appealing to the kids are using tools that requires bodily engagement.

The absence of these tools denotes an ignorance to the role of body in learning programming resulting in a gap in knowledge required for the design of movement based technologies for teaching programming. This case study contributes to that gap in knowledge that lies at the intersection of kinesthetics, Child computer interactions and Programming. The main knowledge contributions of this research are the experiential qualities of a novel learning approach that emerged by a RtD process of developing an artefact - a concept. Löwgren (2007) considers design knowledge as form of scientific knowledge provided that it satisfies the criterias of novelty, relevance, groundedness and criticizability. Elaborating on Löwgren’s (2007) four criterias I assess the academic value of this knowledge contribution in interaction design research.

The experiential qualities of a novel approach to programming is a new contribution to the body of knowledge relevant to this approach as there aren't any traces of work that does the same. The contributions are highly relevant to the field of Interaction Design more specifically to the subfields such as Child Computer Interaction, Kinesthetic Interactions and learning experiences. This contribution is theoretically grounded on the frameworks of Embodied Cognition, Kinesthetic Interactions and Child Centric Design and at the same time empirically grounded through the user studies. For the knowledge to be used by other members of the community, this thesis report explicate the production of this knowledge in a transparent and in a easily comprehensible manner making it critizable.
10. REFERENCES


Buxton, B., 2010. Sketching user experiences: getting the design right and the right design. Morgan kaufmann.


conference on technology in mathematics teaching (Vol. 13). 13th International Conference on Technology in Mathematics Teaching (ICTMT 13).


Horn, M.S. and Jacob, R.J., 2007, February. Designing tangible programming languages for classroom use. In Proceedings of the 1st international conference on Tangible and embedded interaction (pp. 159-162). ACM.


Image Sources

Figure 2.1 (A) (Left) Scratch Programming. (B) (Center) Osmo Tangible Programming. (C) (Right) Lego Mindstorms.
Sources:
Figure 2.3. Interactions with the physical block of various Tangible programming languages.
Sources:
https://projectbloks.withgoogle.com;
http://hci.cs.tufts.edu/tern/

Figure 3.2. The turtle programming language was used to (A - left) - draw geometric Â patterns using a turtle and (B - right ) to control the movement of a tangible robot.
Sources:
https://blogs.msdn.microsoft.com/user_ed/2015/01/26/small-basic-the-history-of-the-logo-turtle/

Figure 3.3. (A)(left) The Scratch programming language interface (B)(right) A child interacting with the programming Blocks of Scratch
Source:
11. APPENDICES

11.1 Appendix A - Codesign workshop 1

Aim of the workshop
The main goal of this workshop was to identify the embodied metaphors related to the abstract and concrete concepts of programming - sequential flow, looping and branching, and exploring the input design space using lo-fi prototypes.

Structure of the Workshop
The workshop was divided into 4 sequential stages - Introduction, sensitising, body storming and programming (Figure 11.1). The first half - Introduction and sensitising focused on sensitising the kids for effective participation and involvement in the second half of the workshop. The second half was based on Bakker et al’s (2012) first two stages of their 5 stage design. The first two stages are composed of enactment activities with the lo-fi prototypes that help in the identification and mapping of the embodied metaphors used in understanding abstract concepts.

![Figure 11.1. The Structure of the Codesign workshop 1.](image)

Materials used for the workshop
A short Presentation was prepared to for kids to help them be aware of the stage of the workshop they are in Figure 11.1. This presentation also helped the researcher to structure the different activities and methods used in a manner that was appropriate to the different stages of the workshop.
Other materials included as a part of this toolkit were, movement cards (figure 11.3. (A)) they were used to help the kids on a certain part of the body during the body storming sessions. Paper Blocks (figure 11.3. (B)) were given to kids asking them to ‘be’ the block that controlled the behaviour of the object in the output space.

Figure 11.3. A (left) - Movement cards used for the ideation of body movements. B (Right) - Paper blocks used to write code.
Overview of the workshop

There were 6 kids who took part for this workshop. There were 4 girls and 3 boys. The total time for the workshop was about 1 hour and was divided into four sections - Introduction, sensitising, body storming and programming.

Introduction

In the introduction phase, the kids were introduced to the overall goals of the study in simple language that could be understood by them. They were then given a introduction about the objective of the workshop and the different activities they will take part in.

Warm-up

Followed by the Introduction stage, the kids were asked to warm up by moving around the classroom and do some stretches. After a few minutes, they were asked to go around and throw a ball to a random person in their group. The person who catches the ball was asked to say a word that’s related to programming. This activity was repeated for the next 5 mins of the workshop and served as a good warm-up session both physically and in bringing certain aspects of programming to the foreground.

Body storming

After the warm up stage - the kids were introduced to the idea of body storming. After a brief introduction, they were first asked to repeat the warm up activity, but for this step, the one who caught the ball was supposed to do come up with a random body movement. Also the movement they were supposed to ideate was in relation to abstract programming concepts such as sequential flow, looping and Branching. Once they all took part and came up with at least two different movements they were asked to use the movement cards to ideate 2 more movements. Once they were done, they used the movement cards along with tangible objects to ideate a few more movements. The body storming stage heThe kids enjoyed bodystorming as this was something new and it involved one to go wild with imaginations while moving the body.

Programing

The kids were asked to stick the paper blocks on themselves to become the blocks. They were then given tangible objects to enact out the movements as the manipulated the character in the output space. They wrote programs by arranging themselves in a logical order. This order was dictated by the researcher but came out naturally. The designer acted as the ‘compiler’ to compile the program which was wizard of Oz as i didn’t have the time to develop a fully functional prototype.


11.2 Appendix B - Codesign workshop 2

Aim of the Workshop
The aim of the second workshop was to evaluate the high fidelity representations of the kinesthetic programming environment. The evaluations focused on the embodied metaphorical mappings between the input and output design space and identifying the elements of the emergent kinesthetic experience.

Structure of the Workshop
The structure of this workshop was primarily influenced by the last stage of the 5 stage development process used by Bakker et al, (2012) that consisted of three sub stages - explorations, testing and interview. An introductory stage was added to introduce the new programming approach to the kids before moving on to the following stages.

![Figure 11.4. The Structure of User Study 2](image)

Materials Used for the workshop
Evaluate the embodied metaphors present in the interaction model along with the experiential aspects based on the kinesthetic Interaction framework using the high fidelity prototypes. For this workshop I used the Hi Fi prototypes of the output space and the input space (figure 11.6).
Overview of the workshop

The same 6 kids who took part for the previous workshop took part in this workshop as well. There were 4 girls and 3 boys. The total time for the workshop was about 1 hour and was divided into four sections - Introduction to the new approach, exploration, writing programs and informal interview.

The New Approach

The children were given a short introduction to the stages of workshop 2 followed by an introduction to the programming environment. They were told about how they were supposed to spider to its bait using the three different blocks - start, move and repeat. They were excited and enthusiastic about trying the new approach as they were curious on how it would work.

Exploration

After a brief 5 min introduction, the kids were asked to explore the programming environment on their own. This was done using 3 different and easy levels of the game that involves the use of 2 blocks of code. The children were asked to try out the prototypes to manipulate the spider in the game world. No explanation were given to them regarding how the artefacts worked. When the child used the move artefact the designer moved the spider accordingly based using the wizard of technique. The kids easily identified this direct mapping between the input space and output space for the move block. However for the Repeat block a simple demonstration was given before they could start using.
Programming

After the exploration phase the kids were asked to solve the 7 different levels of the game. As there needs to be more than one move block required for the writing programs, the move blocks used here were low fidelity prototypes that resembled the high fidelity move block (figure 11.6). Even though the shapes were different both the blocks were appropriated in the same way. However the HiFi prototype of the Repeat block was used in this programming stage.

Figure 11.6. The output space and input space of the programming environment.

My involvement at this stage was limited and observed them as they fidgeted, moved and, When the kids were given the repeats blocks they were able to put the pieces (Start and end pieces) at the right position however, they were clueless on how they were supposed to “repeat” themselves while performing a certain looping operation. As the number of blocks required to complete each stage of the program were different, the kids took turns when they wanted to participate.

Informal interviews

Q. How did they like this new programming approach?
Most of them found it novel and interesting and they liked the game approach to writing programs. They liked the ability to control SID the spider and suggested a few improvements to the game like adding animations.
Q. What artefact they liked the most? Most of them liked the repeat block cause they were allowed to go around. They also liked it for the fact that it was Blinking and it made strange sounds. Two of the kids liked the move block as it sensed their movement.

Q. What was the most enjoyable activity in the last two activities?
They liked combining programming with movements it made it interesting more interesting than the usual tool they were using at school.