

Of Elephants and Nested Loops: How to Introduce Computing to Youth in Rural India

Unnikrishnan R, Amrita N., Alexander Muir, Bhavani Rao

AMMACHI Labs

Amrita University

Kollam, India

unnikrishnan.r; amritha.natarajan; alexander.muir; bhavani.rao @ammachilabs.org

ABSTRACT

We present Haathi Mera Saathi (My Elephant Friend), a game concept which serves as a tool for teaching programming and computational thinking to underprivileged children in rural India. It provides a metaphor and gameplay for embodied and tangible games, and creates a soft early ramp up into the conceptual and digital space of learning to code. We discuss the urgency of digital inclusion for Indian rural children, with reference to technology as an amplifier which they need to learn to direct. We contrast the grounded, embodied style of Haathi Mera Saathi with the current crop of mini-languages and coding games, with particular emphasis on the need for physicality and tangibility in the very early stages of learning to code. We further discuss our experience conducting workshops for students from the tribal and rural belts of India, where we see HMS as an effective approach for taking them from a state of having no background in computers or computing, to a state where they create interactive applications in a Java based environment. Recommendations are given for researchers interested in working with rural village children.

Keywords

Computational thinking; tangible games; ICD; ICT4D; HCI4D; Third World; Digital Divide

ACM Classification Keywords

K.3.2 [Computers and Education]: Computer and Information Science Education;

H.5.2 [User Interfaces]: User-centered design.

INTRODUCTION

The 21st Century is crashing into rural India and today's girls and boys need to understand computers, not merely as consumers but active participants in a global economy [10, 15, 16]. The stakes are high. Technology is fast arriving in

rural India, brought by NGOs, the government, private companies and villagers themselves [12, 13]. This *amplifies* [38] the intensity of the precarious socio-economic situation: healthy socio-economic processes and relationships become stronger, but broken ones become more broken.

“Technology – no matter how well designed – is only a magnifier of human intent and capacity. It is not a substitute.” Kentaro Toyama [38]

There is a clear and present danger of worsening of village life if the villagers themselves cannot manage and reprogram that computing power [40]. The ability to understand, manage and make good decisions around computing is an important capability [33, 42]. For these reasons, we feel it is crucial to teach programming and computational thinking to Indian children, and help them attain the capability of digital fluency.

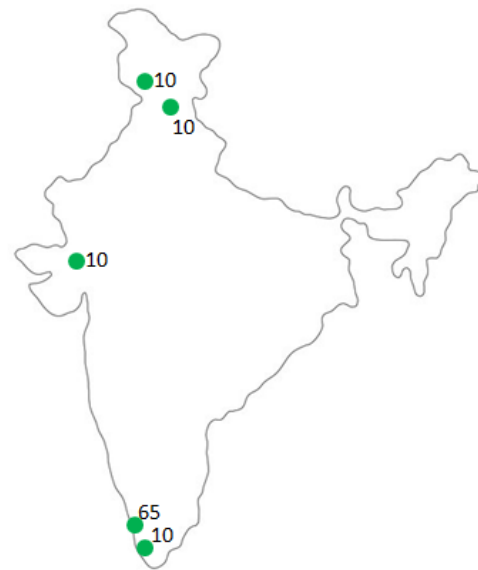


Figure 1. Map of India showing geographic spread of the HMS workshops, and number of children involved

Inner city children may use, or see other children or adults using, devices like smart phones, tablets, digital point of sales terminals, digital signage, games consoles, etc. By contrast, in a rural Indian village there may be a TV and an occasional smartphone. The gulf of evaluation and the gulf

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

IDC '16, June 21-24, 2016, Manchester, United Kingdom

© 2016 ACM. ISBN 978-1-4503-4313-8/16/06 \$15.00

DOI: <http://dx.doi.org/10.1145/2930674.2930678>

of execution [23] are therefore much larger; the children know less about what a device can do, and how to operate it.

To help Indian village children cross the gulf, we developed a game concept to teach the first steps in computational thinking: Haathi Mera Saathi (HMS), or ‘My Elephant Friend’. The child makes increasingly sophisticated commands to control an elephant as it moves around a grid containing trees (obstacles) and bananas (food). Over the last 12 months we have run workshops with over 100 children from rural Indian villages, in 5 different sites. We chose locations across India – North, Western and South – to help playtest HMS across the wide cultural differences in rural India.

Though there are many “teach children to code” tools, we found they cannot be imported as-is into the rural cultural context. For example, the highly abstract presentation in Lightbot presented challenges which are discussed in the next section. Also, since we are working with children with zero experience, we need a slow, gentle ramp; we created the HMS game concept that can be used first as a physical ‘playground’ game and then tangible board game. Yet though we start slow we do want them to learn to code; we developed a game syntax that maps more closely to Java-based programming, and creates an accurate mental model for if/then statements and nested loops. Hence the title of this paper: “Of Elephants and Nested Loops”.

This paper describes the HMS game concept, and how we have designed, validated and used it to soften the initial stages of the learning curve of programming, to help increase inclusion and participation amongst rural Indian children.

RELATED WORK

Since HMS works with children at the extreme end of the global social scale, our first section of related work places us in context of ICT for development (ICT4D). In particular, we reference work around moving from consumers to empowered creators of technology. This will help readers understand the urgency in teaching Indian village children to code.

We then focus on the field of games for teaching computational thinking, and contrast the abstract style of games like Lightbot with the earthy, embodied [32] mood of HMS.

Finally, we look more at the tangible aspect of HMS as a board game, and – finding few similar projects – we explain why we play a warm up game of HMS where children use their whole bodies on a playground sized board.

Confidence and digital fluency for children in the developing world

We are deeply inspired by the work by Resnick et al on the “Clubhouse Project” that aims to move youth from consumers of technology to creators of “things of significance” [29]. We assert the spirit of this approach is

important: a compassionate recognition that the global poor can be *makers* and take command of their digital destiny.

Though we all struggle with the feeling that technology controls us (rather than us controlling it), we must also recognize that the global poor have no voice where design decisions are made. They are not in the conversations, companies, conferences or forums along with those who make the technology that run, say, the system they need to claim their government grants, or bank online, communicate with family and watch entertainment. As they fall to the far end of being receivers not producers, we feel it especially important to find ways to shift this and help them become *makers*.

The work by Kentaro Toyama speaks powerfully to this: in “Geek Heresy” [39], co-founder of Microsoft Research India, takes a scalpel in one hand and machete in the other in his precise and wide reaching critique of ICT4D projects (including his own). He shows that prepackaged technological interventions mostly fail. Notably this includes the “quixotic idea that children will teach themselves digitally”, which is at the heart of the popular but now doubted Hole-in-the-Wall project by Sugata Mitra [22].

Toyama describes technology as an amplifier, which will intensify socio-economic problems as readily as strengthening solutions. Below the poverty line, socio-economic problems outweigh solutions; adding technology to village life requires fine discernment to see where its amplification effects should be applied. This point was made at the 2015 UN Academic Impact meeting [19] where humanitarian leader Sri Mata Amritanandamayi stated: “When we try to love or serve without understanding those whom we are serving, we often end up harming society and ourselves. In order for service to be beneficial, it needs to go hand in hand with discernment. This is the essence of sustainable development.”

As we are based in India and part of a team that does extensive work on the ground in villages [35] (who may be served by multiple NGOs) we have often seen the dismal non-results of efforts to empower villagers by simply giving them access to a PC and the internet. Villagers may become jaded and frustrated when they find that they cannot understand or use the technology in any meaningful way (though Hindi dance shows on YouTube are always popular). It takes hand-holding, mentoring, encouragement, good pedagogy and a space for relaxed creativity to set a solid foundation for complex skills like programming. We feel the “Clubhouse Project” by Resnick and team embodies this.

We do though, whole-heartedly acknowledge work by the Indian government [10] and others to provide access points to technology: clearly, we cannot do without this. However, we must go beyond mere access, to a sense of confidence and digital fluency. HMS is as much about empowering the

children and their communities as it is about creating programmers.

Games for teaching children to code

A well designed game with clear and achievable goals, challenging activities requiring skill, and feedback on performance enables the player to achieve what Csikszentmihalyi terms as the "Flow state of mind" [6], a state of mind most conducive to learning [30].

On a more immediate level, there is a strong field of mini-programming languages and 'teach children to code' games from which to draw inspiration, such as Scratch [28], Lightbot [4], Codemonkey [8], Code Monkey Island [7], Robot Turtles [31] and RaBit EscAPE [3]. These games are important steps on the learning journey to code, because – simply put – programming is complicated. Brusilovsky et al. in their review of mini-programming languages [5] argue that general purpose languages are ill-suited for beginners as they are too idiosyncratic, thereby making it difficult to form "strong cognitive infrastructures" that need to be developed to understand the process of programming. They state that with general purpose languages the process of program execution is hidden as the control structures are not visualized, making mastery of programming even more difficult.

MIT's Scratch is one of the most notable examples of mini languages, embodying the constructionist ideal of a learner-oriented environment offering a rich graphical programming environment. Its interface is easy-to-learn yet we feel there is a lot of scope for creating learning environments that lower the bar even further, especially for third world youth, by making it more less abstract and more concrete.

Codemonkey [8] uses a cute monkey/banana metaphor, and while we acknowledge the similarity to our elephant/banana metaphor, we take the similarity as simply accidental. 'Codemonkey' is a slang term for a programmer, and we expect their metaphor emerged from this, and the general cuteness and child-appeal of monkeys. By contrast, HMS's use of an elephant as avatar is not merely a cosmetic difference. As we will expand on in the section describing HMS, Indian children know that an elephant is controlled by a *mahout* (elephant tender). We found the children in the studies grasped the idea that just as a mahout commands the elephant, they command the computer by use of a program.

In Lightbot [4], released in 2008, the child creates short commands - using programming logic - to move the Lightbot around a grid and light up certain squares. Haathi Mera Sathi is similar in that the puzzle is how to move an avatar around a grid, using programming logic. Beyond that, though, we made different design decisions throughout the game.

We explored Lightbot in use with village children but found that commands to change the direction of the Lightbot were hard for them: turn left and turn right were frequently mistaken. Also, Lightbot's charming luminous characters are still rather abstract. As Medhi et al have shown, when

creating interfaces for low-literacy users in India, it is better to use concrete images [21]. Similarly, the classic LOGO turtle offers an abstract image. Our imperative is to meet children where they are - at ground level - and create an accessible and relatable experience, and tangible sense of space. Hence the use of a friendly elephant navigating through a jungle in search of food. We now move to a further discussion of the tangible and embodied.

Tangible and Embodied Games

There are physical board games to teach computational thinking, such as the elegantly crafted (but curiously capitalized) "RaBit EscAPE" [3] that uses magnetized tangibles to create a "path" for a rabbit to escape to safety from monsters, teaching abstract concepts of data properties. Code Monkey Island (not to be confused with Codemonkey) is another board game where the player guides a tribe of monkeys around an island to a banana grove using computing principles like boolean operations & assignment operations. T-Maze [41] & Tangicons [34] are both tangible environments that offer physical programming through manipulatives that connects its physical state to actions on a virtual environment. Physicality is important for learning, particularly for village culture, which is closely tied to the physicality of the land. As noted above, students could have a hard time with right and left as seen in a 2-D projection in Lightbot. However, beyond this the larger point should be understood: Indian village culture takes the physical land and the primary directions as reference points, often in ways much more powerful than a modern urbanite could feel. For example, in Haryana it is common to say "the pot is North of you" rather than "the pot is behind you". A board game provides a tangible spatial presence, and is a stepping-stone to abstract digital space.

Further, educational research supports physicality in learning in general. The well-known "Waldorf School" approach of Rudolf Steiner [24], which – incidentally – is a popular choice for children of tech execs at Microsoft and Google, strongly emphasizes the role of the body and physical movement in learning mathematics at a young age. A rich sensory web of associations helps children absorb and retain a concept, turning it into instinctively felt knowledge.

In our context, seeing the difficulty the children had with abstract changes of direction in Lightbot, we first warm them up with an embodied 'playground' style of the HMS game, where they themselves jump up and turn around in the direction required. This act of physically embodying the game before even playing the board game itself was important in teaching the children, and also appears to be fairly unique in the literature. We will return to how we sequenced and ran this embodied form of the game in the section describing HMS.

Games as a Bridge to Real Programming

Being a fun game is not enough: it has to lead to the doorstep of actual coding. HMS has design elements that map to the Java based Processing environment. Processing [26] was

chosen because of its minimal, easy to use interface originally designed to encourage artists and hobbyists to rapidly prototype interactive and visualization programs (known as 'sketches').

To return to a comparison with Lightbot: the commands it uses do not map tightly to units in a pseudo-code program. In HMS, we use commands that can easily be turned into units of pseudocode (and from there into real code). We found it possible for children to actually code programs after playing HMS. Again, we further explain this feature of the grammar and pedagogy in the section describing HMS in detail.

The Approach to Research

Turning the light on our own approach to conducting this research, a few comments will help give context. The core team is part of a research and development lab for learning technologies to support the underserved population of India, especially rural women and children [1,35]. We are based at a university in South India, but also aligned with a large team of social service projects that has relationships with schools and villages across India. The team members spend months each year in various hands-on field work projects [1]. This balance of time spent in the lab and also the field mutually support each other, and thus our approach could be called 'Action Research' in the sense that we do skills training in rural India as large part of 'day job', and we are researching how to make that more effective. This is different to, say, the excellent HCI4D work undertaken in India by Microsoft Research [33] and other research groups: in those cases, though there may be emotional ties to the work, the researchers are free and even expected to leave the user population and move on to other interests. Our organizational mandate is a multiyear commitment to work with these villages.

As working professionals in ICT4D as well as researchers into it, we have a view – this will be clear to the attentive reader so far. In our reporting here, we hope to strike a balance between our professional opinion and the rigor of objective research, without either falling into the falsity of feigned neutrality or offering a mere opinion as fact. We very much welcome feedback from the community.

Summary

A teacher wanting a tool for teaching computational thinking faces an embarrassment of riches, but a student in rural India may still have a steep learning curve with these tools and games - as we discovered when we trialed some of them with children in our workshops. HMS aims to soften that learning curve by following a mini language approach, by building a set of relatable vocabulary (commands) focusing on the most important computing principles (iteration, branching), with a grounded setting that the children can relate to.

To reach this we have drawn on existing mini languages and computational thinking games, and adapted them for the Indian context. To understand that Indian context we have

undertaken field studies and conducted workshops, inspired and informed by movements such as the Maker and Clubhouse initiatives of Resnick and others.

The need for children to gain this digital fluency and confidence to create 'things of significance' is particularly acute in rural India because 21st Century technology is arriving by wireless, by drone and – in the case of the proposed Project Loon – by hot air balloon [13]. Technology is a powerful amplifier that will speed up the social situation: what works will be more efficient, what is broken will worsen. Given the very fragile nature of third world village socio-economics, we feel it is essential that children are empowered to understand, make and create with technology.

AN OVERVIEW OF HAATHI MERA SAATHI

The Name and Metaphor

Haathi Mera Saathi (HMS) means "My Elephant Friend". It is pronounced with long-Ahs: HAH-ti meh-ra SAH-ti. This simple, rhyming, friendly name refers to something earthy, solid, familiar and a much loved part of the Indian world. Village children will mostly have seen elephants in person in festivals, and will certainly have heard stories with elephant characters. In contrast to names like Lightbot, Scratch and Process, the name directly refers to physical objects and relationships that are real in the children's worlds.

In this moment of first exposure to the world of computing, it is important to help lay the ground for a correct mental model [23]. Lakoff & Johnson in their seminal work "Metaphors we live by" [18] shed light on conceptual metaphors that can express connections between a concrete experience (*mahout* issuing commands to an elephant) and one that is relatively abstract (programmer issuing commands to a computer). Indian children are likely to know that elephants are controlled and directed by a *mahout*, and by putting the child in the place of the *mahout* we have found they easily understand the concept of the programmer of the computer. The correct relationship is established on a familiar archetype.

We acknowledge the metaphor would break in the real world case of uncontrollable elephants going on a destructive rampage. Though ironically it is also true that badly designed software can apparently lead a computer to go on a destructive rampage through the forest of one's work - perhaps the metaphor holds after all.

Gameplay

The game play is designed to mimic the process of program execution in Processing, and in general purpose languages more broadly. The game takes place on a grid, where the game master has laid out trees (obstacles) and bananas (food), and placed the elephant. The more obstacles, the more complex the game. The player has a set of commands from which she/he can create a "program" that will cause the elephant to navigate around the trees and eat all the bananas. The challenge is to solve the puzzle of moving the elephant

correctly, using the concepts of turn, go forward, if/then and iterations (loops).



Figure 2. Laxmi is one of our local elephants; here she is tended by her *mahout* (who is talking on a flip phone)

The player has control only over the "program board" and is not allowed to move the elephant or any of the on board elements. This is to expose the feature of programming environments where the programmer has control only over the APIs and other interfaces afforded to him/her. Players can also see the effects of their commands only once the execute card is shown as this acts as a means to understand the "Run/Debug" feature of compiled programming languages. In fact, the execute card has a symbol of the "Run" button that is found in the Java based Processing environment.

The command cards use both visuals and text to show their meaning. We found that using local language phrases was not necessary for simple English phrases like eat, turn left/right, go forward etc., which our children could understand.

1. The game master sets the trees & bananas on the board.
2. The player looks at the board and creates a set of commands that will direct the elephant to eat all the bananas. The commands are placed on the "program board".
3. Once the player is satisfied of the solution, he/she shows the "execute" card to the game master.
4. The game master moves the elephant according to the commands in the player's 'program'.
5. The player wins the game if all the bananas in the board are "eaten".
6. If not all the bananas are eaten, the player creates another program. They get as many chances as time and situation allow.

Figure 3. The steps in a round of Haathi Mera Saathi

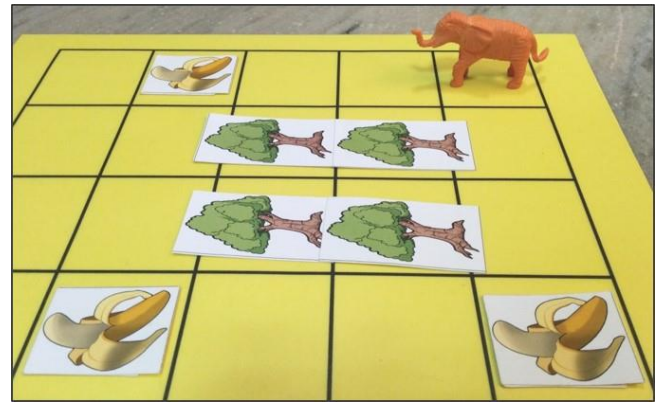


Figure 4. An example of the HMS game board with an arrangement of trees and bananas

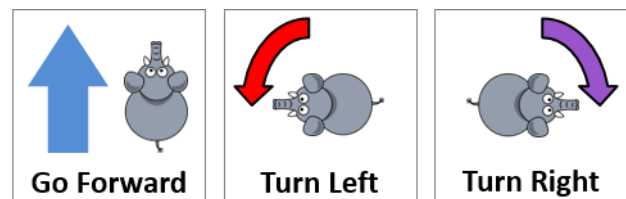


Figure 5. Examples of HMS command cards, using concrete images, abstract images and text

In the case of the basic movement cards (Fig 5) we see both concrete images (elephant) and abstract images (colour coded arrows) as well as the text. We iterated the design until we found the children understood the cards without difficulty.

Creating More Sophisticated Programs with loops

If we take the board shown in Fig 4 as an example, a basic programming approach is to move the elephant step by step. This is shown in the following program on the HMS "program board" (Fig 6).

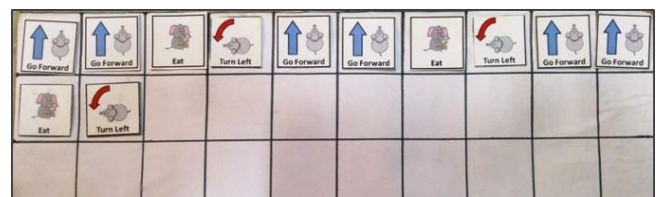


Figure 6. Simple program on the HMS "program board"

A key step is to introduce the idea of loops. We demonstrate to the children how the elephant can do the same thing three times, in order to eat the bananas. This is done by having them place command cards in sequence as shown in Fig 6 and then they are shown how the process can be simplified with the use of the loop cards. Fig 7 shows a more compact version of the program in Fig 6 using the loop construct. A series of challenges are given to the children after the initial demonstration.

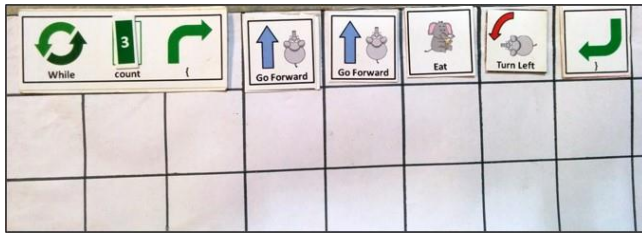


Figure 7. Program using loops

The loop command cards are as follows:

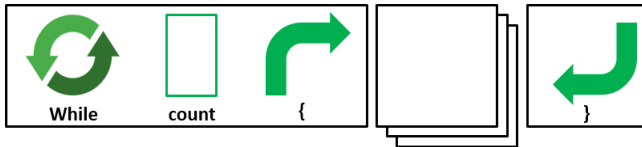


Figure 8. Program using loops

The syntax for while-loops contain color coded abstract symbols and text, as seen before in the basic movement cards. Also we now introduce the { } symbols (tokens), which is a key element of the specialized vocabulary of programming (Fig 8).

Do children confuse the green arrow with the turn command? Across all our workshops, we found they did not: the absence of the elephant, the color coding, the label and the context all combined to make it clear for the children.

Variables for counting the progress of the loop are also introduced here: small number cards are placed in the space about the label “count”. The game master keeps track of iterations by manually displaying the state of the *count* 'variable' on a piece of paper, as they execute the program. It is admittedly slightly fiddly, but we felt it unproblematic in playtesting.

Conditional branching

The conditional branching structure enables the player to issue commands to the elephant when certain conditions are met. This structure has three kinds of command cards each one having color coded symbols and realistic programming tokens specified at the bottom, to facilitate a transition to programming environment later on. The labels in Figure 9 are: if *condition* { } else { }.

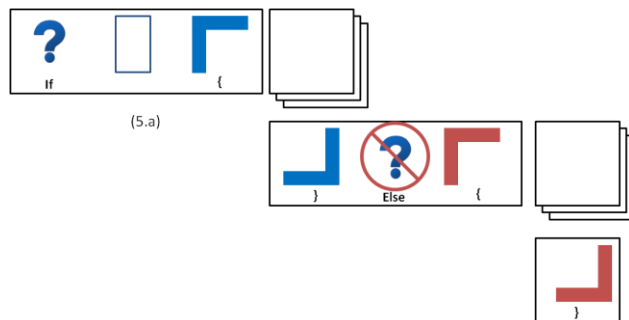


Figure 9. HMS conditional branching cards

The reader can now see that quite elaborate programs are available by combining direct movement with loops (and nested loop) and conditional branching.

Sequence of Activities in an HMS Session

Before playing the HMS board game we recommend playing it as a ‘playground’ game. Then afterwards to have children write down the pseudo code for one or more programs they created. Finally, for children who show interest, to move into actual programming projects. This sequence moves from the most physical (embodied HMS), through the tangible board game with its progression from simple to sophisticated compound programming concepts, to a trial run of writing something very close to actual programming (pseudo code) (Fig 11).

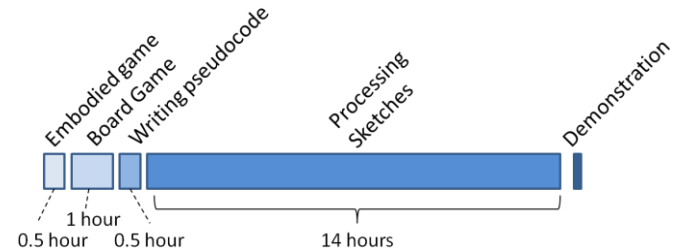


Figure 10. The flow of an c-HMS session

HMS session flow:

1. Embodied HMS as a playground game
2. HMS as a board game
3. Transposing HMS commands into pseudo code
4. Actual programming

Through our playtesting, we felt this an effective way to move children from zero. The following figure illustrates the flow and the time taken for a complete HMS session.

Embodied Haathi Mera Saathi

In an early workshop we realized that some of the children had trouble telling their right apart from their left. To help, the children were brought out into the open and asked to perform the basic moves of HMS (go forward, turn right/left) on a 3x3 grid drawn on the ground. To make sure they understood some volunteer students were made to stand on one of the squares. A second volunteer would then show poster sized command cards for everyone to see and call out the command, which is followed by the student. See Figure 10.

Beyond the need to help remind children of right and left, we feel there are other benefits of opening a session with this form of the game. Firstly, in terms of simple class management, having a physical activity can raise energy levels and engage the group in something fun. Secondly, before inviting the children into the conceptual, abstract and ‘heady’ world of programming we want to seed a solid basis of visceral understanding of key concepts of movement.



Figure 11. Embodied HMS

Writing down pseudocode

Once the board game sessions are finished, the students are asked to write down the programming tokens written at the bottom of the command cards they have arranged to solve the given scenario (Table 1). They are asked to add "();" to the end of each basic command and to eliminate spaces. This serves to cement the relation of the commands in the tangible space to the form of text (as function calls) that they can type and use in the Processing environment.

HMS's game elements to programming elements is in contrast to the strategy used in the popular code.org [9] series of games where players program is translated to code after the game. The design of the HMS cards exposes the functionality during the gameplay itself.

Command sequences	Child writes...
	<pre>while(count<3) { goforward(); }</pre>
	<pre>if(tree) { } else { eat(); }</pre>

Table 1. Converting HMS gameplay into pseudocode

Here we also see a limitation in some programming tools and mini-languages: they do not allow for a simple transposing to pseudocode. This means they require a greater jump to go from their own language to a general purpose language. By contrast, with HMS we aim to provide stepping stones to help make the transition from the game's syntax to Java as used in Processing.

FIELDWORK AND WORKSHOPS

The work to develop HMS has involved rural village children from the outset. User research with children has challenges but also is easier than with village adults, as children typically show their interest level and comprehension unguardedly. In this way we feel that rural village children in India are much the same as children elsewhere: you can quickly tell if they like something or not. More than a teacher, the role of the facilitator in these kinds of engagements assumes that of a motivator and of a trusted friend [17] and over the course of the workshops, we had the unique opportunity to glimpse into their hopes and dreams.

The general format of workshops was to conduct an embodied 'playground' style of HMS, followed by the board game, followed by some actual hands on use of computers for a creative task chosen by the children themselves.

The promise of getting to play on a computer is bright motivation for many children, and so winning their participation was usually easy. It should be emphasized that for many children this was their first time being in physical proximity to a computer, let alone actually operating one.

In the spirit of honesty let us admit that we could wrap labels around our approach (action research using rapid iterative development with a child-friendly participatory design mindset) because it is true that we developed and refined the game with the ongoing feedback of end users. Yet it is closer to the truth to simply say we brought different versions of the game to the children, taught them how to play it, and played it with them. It was easy to tell where it was fun, where students succeeded, and where they stumbled. Success can be measured by how many levels (variants of the arrangement of trees and bananas) were accomplished within an hour. Also, the number and types of question and confusion the children raised gave us an indicator of when the gameplay as "dialed in".

The workshops happened within the 'magic circle' of play [14] – a time set aside from 'ordinary life' both by the fact that we were playing (not learning) and that the mysterious and precious computers were about to be made available. This mood of play sets it apart from the more somber types of participatory design accomplished with grownups. It is also different in mood to research on children who use computers on a regular basis and for whom they are nothing special. Our children were dancing at the edge of totally new experiences, and our task was to understand how the game could take them towards confidence and digital fluency.

Here we define two types of engagement depending on the logistical constraints: a complete HMS (c-HMS) session described in Fig 10 and an introductory HMS (i-HMS) session which includes only playing with the board game. While c-HMS would take a child up the arc of introductory programming, i-HMS would teach basic computational concepts through a combination of the board game with

supporting activities like the embodied game and writing pseudocode.

In the following discussion, we define “Processing (Basic)” as the skill of understanding & recreating the basic Processing sketch structure including the essential *size*, *background*, *setup* and *draw* methods. “Processing (Intermediate)” involves understanding the Processing coordinate system, basic graphics primitives (*rect*, *ellipse*, *line*) and keyboard event handling. “Pseudocode” involves the child writing down the commands used in solving any of the HMS scenarios.

Our research team has the advantage of being part of a network of social service activities [35] that has relationships with various rural schools across India. The Indian fieldwork team was known to the supervising adults in the schools, and so we were free to conduct sessions of several hours at a time.

Would it be possible for a Western research team to work with children? As long as there is a trusted Indian adult and translator in the room, we would anticipate relatively few difficulties in getting rapport with the children. However, there would be significant gatekeeper challengers in finding the right school, the right principal and the right teachers who could give access to the right children. Bureaucratic, logistical, linguistic and relational requirements for foreigners to conduct research in India are non-trivial. Working with children is the easy part.

The workshops were conducted in 7 instances in 5 locations: Amritapuri in Kerala, Paripally in Kerala, Nani Borwai in Gujarat, Pandori in Jammu and Kashmir and Indpur in Himachal Pradesh.

The following sections are summarized from research diaries, and give some highlights that illustrate the mood, experiences and success cases with the game.

Nani Borwai, Gujarat

We pilot tested the board game in sessions with 10 children of ages 12-16. We had conducted parallel sessions with another set of 10 students with Lightbot and observed that the children who played HMS did not show any confusion involving tasks that needed mental rotation while the majority found the Lightbot turn tasks confusing. This could have been simply because of their unfamiliarity with abstract digital representations [21]. The board game was followed by asking the students to write down the code equivalent of their game moves after each session, the most persevering of whom ended up writing the “code” after observing a scenario on paper before using the command.

Amritapuri, Kerala (A1, A2)

We scaled up and ran one-on-one sessions with 65 children from an orphanage of ages 12-16, most of whom are from tribal communities of Meghalaya and the Western Ghats. We brought them over to our campus to spend a week filled with activities ranging from digital literacy classes to storytelling workshops. In the digital literacy classes, they were

introduced to productivity applications like Microsoft Office along with an introduction to the Internet. After the 65 sessions, we picked 10 of the most enthusiastic children for the next stage of developing creative IT projects. Since we had almost complete access to these 10 children for a few days during their stay at Amritapuri, they constituted the only group that went through the complete HMS arc (c-HMS).

Their choice of applications to work on was mostly motivated by their personal interests. Girls ‘S’ and ‘A’ want to become police officers. ‘S’ had an abusive father (note that she is now at an orphanage). Together they worked on an interactive ‘criminal location viewer’. They started out working together but eventually split up and worked to create applications on their own. ‘J’ and ‘K’ are accomplished *Chenda* percussion players and expressed a mutual love for vehicles of all kinds. Both want to become bus drivers – and it turns out it was because both of them knew a driver from the school very well. They made an interactive car application where a car that they made on their own on MS Paint was loaded onto a Processing Sketch and upon each keystroke, was made to travel from one house to another (which they also drew with MS Paint). The other groups made a cooking tutor with a series of images to instruct on cooking healthy food and a rehabilitation game where one is encouraged to touch keys at opposite ends of the keyboard to encourage exercise of the arm. In the end, they presented their work to each other. Throughout the workshop we found that the metaphors introduced in the HMS board game were crucial in explaining various features of the Processing language including keyboard/mouse event handling and the use of braces, both of which would have been much tougher to explain had the children not been exposed to the HMS game.

Indpur, Himachal Pradesh

Our group here consisted of ten girls of age 16 years or older, from a local computer training center. Some of them reported attending the computer course to increase their marriage prospects. They went through an HMS board game session. The embodied game was not played as all of them had and took turns becoming the game master and setting challenges for their fellow students. They then moved to Processing where they learned basic Sketch structure, drawing using primitive functions and the coordinate system using which each of them went on to draw their own hut in Processing.



Figure 13. Child playing HMS board game

Other workshops (P1, A3, P2)

We followed an i-HMS approach in other workshops conducted in Amritapuri & Paripally in Kerala and Pandori in Jammu & Kashmir. The i-HMS workshop in Amritapuri had 18 participants from North India of ages 15-17 who came over to attend vocational training workshops. The workshop at Paripally orphanage had 10 children from ages 14-16. The workshop conducted in the village of Pandori in Jammu and Kashmir had 10 children who were students at a local tuition center ranging in age from 9 to 16. Participants in both the i-HMS sessions in Amritapuri and Pandori completed all of the fundamental HMS board scenarios covering sequential commands, loops and conditional branching.

Summary

We have played the HMS game with over a hundred rural village children in the North, South and West of India. We find they can successfully play the game, i.e. create commands to direct the elephant around the board using simple directions, loops and branching. Naturally, some children are more enthusiastic than others – and these we have invited to make playful efforts at writing their own application in Processing. The fact that we can move from little or no IT experience to actual programming within an afternoon indicates that this approach can be effective for smoothing the early curve of the learning journey to code.

CHALLENGES, LIMITATIONS AND FUTURE WORK

We absolutely shun any notion that HMS is a prepackaged game that can be dropped into a rural village and automagically enable children to program. The success of the sessions is based on a warm mood of mentoring and encouragement, and the experience of the facilitators and their assistants. That said, there is no magical quality our field team possess that a similarly motivated group of facilitators or researchers would not also have. We merely underline the easily ignored fact that technology *per se* cannot bridge the digital divide and increase inclusion amongst the rural poor: it requires human relationships, warmth and discernment to wield the tools.

Working with children can be tiring, especially when ideas become complex and feel less like play. Classroom orchestration is tough on the facilitator especially considering the high level of distractions in a rural school. There are frequent intrusions into the classroom from outside forces, such as siblings, parents and goats. Children may be called away on chores, and electricity is likely to come and go as it pleases. This puts a burden on the facilitators. To those teachers used to more sanitized and orderly classrooms, who may become rattled by the intrusion of large quadrupeds, we simply suggest: just flow with it.

A pedagogical and HCI challenge is that feedback is not immediate. Here a digital tool like Lightbot is stronger: a program in Lightbot may take less than 2 seconds to complete while a loop in HMS would take around half a minute. The game master has to read the cards and act them

out on the elephant. This also means human error is possible, if the game master misreads or misunderstands the program, or simply makes a slip and places the cards wrongly. To improve in this aspect, we are currently playtesting a digital version of HMS, and will compare the differences in learners' experience between the tangible and digital.

SELECTION AND PARTICIPATION OF CHILDREN

The children were selected informally from the local tuition center run by our parent organization. Though we do not have the exact figures, most of the children are drawn from families who belong to the lowest rung of India caste spectrum, the scheduled caste and tribes. In most cases, we talked to the children at the tuition center and informed them that if they wished they could play a game with us. In a few instances we also mentioned the purpose of the game (to teach computational thinking).

ACKNOWLEDGEMENTS

We wish to thank Mata Amritanandamayi Devi, world-renowned humanitarian and chancellor of Amrita Vishwa Vidyapeetham University for her guidance and constant inspiration and support. We also wish to thank Amrita University and fellow collaborators at AMMACHI labs for all their support and advice.

REFERENCES

1. Akshay, N., Sreeram, K., Anand, A., Venkataraman, R., & Bhavani, R. R. (2012). MoVE: Mobile vocational education for rural India. In *Technology Enhanced Education (ICTEE)*, 2012 IEEE International Conference on (pp. 1–5). IEEE.
2. Antle, A. N., Droumeva, M., & Corness, G. (2008). Playing with the sound maker: do embodied metaphors help children learn? In *Proceedings of the 7th international conference on Interaction design and children* (pp. 178–185). ACM.
3. Apostolellis, P., Stewart, M., Frisina, C., & Kafura, D. (2014). RaBit EscAPE: a board game for computational thinking. In *Proceedings of the 2014 conference on Interaction design and children* (pp. 349–352). ACM.
4. Armor Games.Lightbot. Retrieved from <http://armorgames.com/play/2205/light-bot>
5. Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A., & Miller, P. (1997). Mini-languages: a way to learn programming principles. *Education and Information Technologies*, 2(1), 65–83.
6. Chen, J. (2007). Flow in games (and everything else). *Communications of the ACM*, 50(4), 31–34.
7. Code Monkey Island. Code Monkey Island. Retrieved from <http://codemonkeyplanet.com/>
8. CodeMonkey Studios Ltd. Code Monkey. Retrieved from <https://www.playcodemonkey.com/>
9. Code.org.. Retrieved from <http://code.org>
10. Department of Electronics & Information Technology, Govt of India. (n.d.). Digital India: Power to Empower. Retrieved from <http://digitalindia.gov.in/>

11. Evans, G. E., & Simkin, M. G. (1989). What best predicts computer proficiency? *Communications of the ACM*, 32(11), 1322–1327.
12. Facebook. Free Basics. Retrieved from <https://mobile.freebasics.com/desktop>
13. Google. Loon - Balloon Powered Internet for Everyone. Retrieved from <http://www.google.com/loon/>
14. Huizinga, J. (1949). *Homo Ludens*.
15. Int. Telecom. Union. (2014). *The World in 2014: ICT Facts and Figures*. Int. Telecommun. Union Geneva, Switzerland.
16. Irani, L., Vertesi, J., Dourish, P., Philip, K., & Grinter, R. E. (2010). Postcolonial computing: a lens on design and development. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1311–1320). ACM.
17. Jenkins, T. (2001). Teaching programming—A journey from teacher to motivator. In *2nd Annual LTSN-ICS Conference*.
18. Lakoff, G., & Johnson, M. (2008). *Metaphors we live by*. University of Chicago press.
19. Mata Amritanandamayi. (2015). *Mata Amritanandamayi Speech at United Nations Academic Impact meeting on Sustainable Development*. Retrieved from https://www.youtube.com/watch?v=0SNsHbI4_xc
20. Mazlack, L. J. (1980). Identifying potential to acquire programming skill. *Communications of the ACM*, 23(1), 14–17.
21. Medhi, Indrani, S. Raghu Menon, Edward Cutrell, and Kentaro Toyama. Beyond strict illiteracy: abstracted learning among low-literate users. In *Proceedings of the 4th ACM/IEEE International Conference on Information and Communication Technologies and Development*, p. 23. ACM, 2010.
22. Mitra, Sugata, Ritu Dangwal, Shiffon Chatterjee, Swati Jha, Ravinder S. Bisht, and Preeti Kapur. Acquisition of computing literacy on shared public computers: Children and the "hole in the wall". *Australasian Journal of Educational Technology* 21, no. 3 (2005).
23. Norman, D. A. (2013). *The design of everyday things: Revised and expanded edition*. Basic books.
24. Palumbo, L. (2004). Math in the Kindergarten. *Gateways: Professional Journal of Waldorf Easrly Childhood Association of North America*, (47). Retrieved from <http://www.waldorflibrary.org/journals/15-gateways/202-fallwinter-2004-issue-47-math-in-the-kindergarten?highlight=WyJtYXRoIiwYm9keSIslidib2R5IiwYm9keSdzIiwYm9keSciLCJib2R5JywXQ==>
25. Parikh, T. S., & Ghosh, K. (2006). Understanding and designing for intermediated information tasks in India. *Pervasive Computing, IEEE*, 5(2), 32–39.
26. Processing Foundation. Processing. Retrieved from <https://processing.org/>
27. Reason, P., & Bradbury, H. (2005). *Handbook of action research: concise paperback edition*. Sage.
28. Resnick, Mitchel, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner et al. Scratch: programming for all. *Communications of the ACM* 52, no. 11 (2009): 60–67.
29. Resnick, Mitchel, and Natalie Rusk. The Computer Clubhouse: Technological Fluency in the Inner City. *High technology and low-income communities: prospects for the positive use of advanced information technology* (1999).
30. Rieber, L. P., Smith, L., & Noah, D. (1998). The value of serious play. *EDUCATIONAL TECHNOLOGY-SADDLE BROOK NJ-*, 38, 29–36.
31. Robot Turtles LLC. Robot Turtles. Retrieved from <http://www.robotturtles.com/>
32. Rohrer, T. (2007). The body in space: Dimensions of embodiment. *Body, Language and Mind*, 1, 339–378.
33. Sambasivan, N., Cutrell, E., Toyama, K., & Nardi, B. (2010). Intermediated technology use in developing communities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 2583–2592). ACM.
34. Scharf, Florian, Thomas Winkler, Claudia Hahn, Christian Wolters, and Michael Herczeg. Tangicons 3.0: an educational non-competitive collaborative game. In *Proceedings of the 11th International Conference on Interaction Design and Children*, pp. 144–151. ACM, 2012.
35. Sheshadri et al. (2015). Training India's first female toilet builders: An argument for improving sanitation through women empowerment and social inclusion. Presented at the International Conference on Sustainable Development.
36. Slater, D., & Tacchi, J. A. (2004). *Research on ICT innovations for poverty reduction*. UNESCO.
37. Thinkersmith (TM). (2013). My Robotic Friends. Presented at the CSED Week, CSED. Retrieved from <https://csedweek.org/files/CSEDrobotics.pdf>
38. Toyama, K. (2010). Can technology end poverty. *Boston Review*, 36(5), 12–29.
39. Toyama, K. (2015). *Geek Heresy: Rescuing Social Change from the Cult of Technology*. PublicAffairs.
40. Walsham, G., & Sahay, S. (2006). Research on information systems in developing countries: Current landscape and future prospects. *Information Technology for Development*, 12(1), 7–24.
41. Wang, Danli, Cheng Zhang, and Hongan Wang. T-Maze: a tangible programming tool for children. In *Proceedings of the 10th International Conference on Interaction Design and Children*, pp. 127–135. ACM, 2011.
42. Zheng, Y., & Stahl, B. C. (2011). Technology, capabilities and critical perspectives: what can critical theory contribute to Sen's capability approach? *Ethics and Information Technology*, 13(2), 69–80.