2016 3rd Asia-Pacific World Congress on Computer Science and Engineering

Gamifying foundational STEM skills

Jim Morey, John Gammack, Erik Thornquist
College of Technological Innovation, Zayed University
Abu Dhabi, United Arab Emirates
e-mail: jim.morey@gmail.com
john.gammack@zu.ac.ae
erik.thornquist@zu.ac.ae

***Abstract*: University education has always required numeracy and literacy to underpin development of higher-level knowledge and skills. In many countries, education policies have often followed transitory fashions, both in mathematics and language education, enfeebling potentially generations of their youth. The use of technological innovations can amplify or hinder the development of foundation skills, and for digital natives this is their norm. In this paper we address this issue by describing the development of a gamified approach to learning, aimed at reinforcing and developing the fundamental knowledge of basic elements, whilst enhancing more conceptual skills, such as pattern recognition and hyper search strategies. These latter are increasingly important in the digital world. We focus here on numeracy skills, illustrating both how rote learning is practiced "incidentally" as strategic pattern recognition is developed in a game context. We detail how the application can be used diagnostically and normatively as it is mapped to outcome based educational levels and standards. We then discuss the extension of the approach to language development, and show implications for educational information systems design.**

Keywords—STEM foundations, gamification, ed-tech application

I. INTRODUCTION

STEM (Science, Technology, Engineering and Maths) has always been central to University education and the technological developments that foster and inspire innovation. It is vital that these are developed throughout the school years, so that university education can build upon them, and education ministries worldwide understand this. However, at least since Hammersley (1968) much of the STEM education has actually been "enfeebled" (his word) to the detriment of learners, with implications a generation or so later, for weaker schoolteachers and ultimately the country's ability to compete on the basis of innovation.

As teachers who have collectively taught in North America, Europe, Australia, Asia and the Arab world we have noticed this as a common trend, and research backs this up. For example a few years ago Cambridge University had to adjust its first year mathematics syllabus as many school leavers were not conceptually prepared. Likewise with language skills, both anecdotal evidence and studies (e.g. Ball and Randerson, 2012) suggest (UK) school exams are easier than before, and although many students may be qualified for university by their grade profiles, an appropriately tough entrance exam will often disqualify them for university entrance.

Much of this can be attributed to educational policies that encourage weaker curricula, grade inflation, and meeting KPI targets over criterion based outcome assessments. Another trend however is the forms of STEM teaching that have become available in the digital age, and the impact of those on real learning. The calculator for example made it possible to avoid drill in mental arithmetic, and to obtain instant results with higher-level functions. For example, standard deviations are reported in the research of any discipline: we would argue that when first learning what this means, it should be calculated by hand the long way such that when it is used in later research, the origin of the numbers provided by the software is conceptually understood. We ourselves have witnessed university students resort to their phone app to compute "9×6" which occurred in the context of a weighted comparison of vendor selection criteria. The higher order understanding of the discipline-related concept was delayed or not achieved due to a more fundamental shortfall.

The old school times tables were taught by rote, perhaps with a wall chart, and those of a certain age can probably reel these off easily even today. Rote learning went out of fashion however, and focus in maths education has shifted towards more creative and conceptual preparation. This is proper and well-intentioned but Hammersley (1968) was prescient in seeing the dangers and reports "evidence that a preoccupation with abstract mathematics inhibits manipulative ingenuity". For language learning also, reinforcement of fundamental aspects of words, their structure and patterns of relations are critical in being able to read effectively, and to write productively. A problem is that, in the digital age, rote learning is potentially boring, especially if in a class setting some learners are over-reinforced ("I got it already – I am bored!") and others are under-reinforced ("I need more practice – but am bored now!"). As all current and future students will be digital natives used to fast results and app based environments, gamifying the skills needed using an app suggests a solution strategy for this problem.

As the 9×6 example above showed, the routine application of arithmetic is subordinate to its use in a discipline based problem-solving tool (maximizing in a matrix of weighted averages). Equally STEM concepts must be the goal beyond simple arithmetic facility, and in recent years maths education has focused on generic skills such as

978-1-5090-5753-5/16 $31.00 © 2016 IEEE
DOI 10.1109/APWC.on.CSE.2016.34

estimation, problem solving, pattern recognition, set logic and so on, designed into the curriculum such that the arithmetic or simpler exercises set up conceptual constructs for later development. Professor of maths education at Stanford and a critic of rote approaches to times table learning, Boaler (2015) notes that “Visual materials prompt a different kind of brain activity to the numerical approach… they allow children to see the connections. They start with something simple and then see the ideas develop”. Her scientific work (e.g. Boaler, Chen, Williams and Cordero) explores the use of visual thinking in learning maths concepts. We adopt this philosophy for design of our app, to encourage mastery of basics as well as encouraging higher order learning and skill development.

The rest of the paper is as follows. First we describe our prototype application, illustrating its levels, scalable complexity, and mapping to leveled educational goals. We include details describing the app environment in design science and computing terms to show how the design can be implemented, or re-implemented for other base constructs, linked at http://jimmorey.com/bull/simpleBull.html. Finally we show how the design artifact can be transformed for the domain of language learning, indicating some of the analogous learning issues involved, before drawing some general conclusions.

II. PROTOTYPE

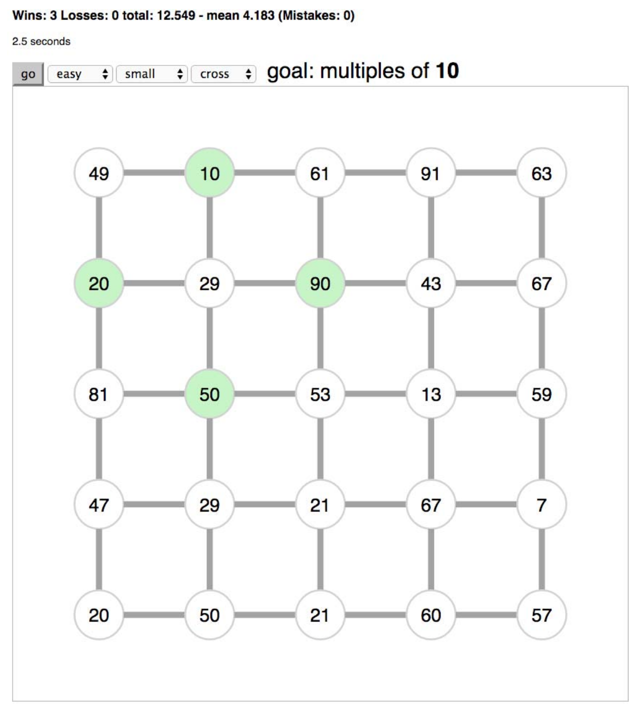

Figure 1: a screen capture of the prototype showing the solution bullseye when *being-a-multiple-of-10* was the targeted property.

The activity is a geometrical search game where the player must find a unique local pattern called a bull’s-eye in a larger network. The bull’s-eye is a node that is connected to only nodes that have the same targeted property. The bull’s-eye node may or may not have the targeted property. Figure 1 shows an example of a simple 5x5 network with its bullseye highlighted—in this case the targeted property is being-a-multiple-of-10.

One important design choice was to not enforce that the bulls-eye has the targeted property. This provides a challenging distractor. Having the central node that the player must select, which may or may not have the targeted property, presents a snag to the player: clicking on the element that might not have the property that they have been focused on adds a layer of uncertainty to the decision to click on a node.

A number of other design choices were made to balance the complexity of the searching task, aesthetic appeal, and potential for learning the targeted properties. First, the local pattern is constrained to a rigid geometry with degree four that is constant for the entire network with the exception of the edges (which may have lesser degree). Second, the bulls-eye must have degree four. And finally, the game is timed. The rigid geometry aids in the regularity of searching, helping to better focus on a player attention on node properties, rather than engaging in too much connection tracing. The choice of degree four seems to be a good balance between too many connections or two few to check—this choice also seems to be supported by the strategy game of Go’s choice of degree four. Both of those choices also have an esthetic appeal that will be demonstrated in the following sections. Time pressure through a “countdown” mechanic is a common game design element: timing the search task is intended to foster player motivation to devise efficient techniques to find the bulls-eye. Yildirim (2015) suggests that for promoting enjoyment and motivation in games there may be “an optimal time limit in which autonomy and competence are maximized.” This countdown variable can be adjusted in a class room or test setting but this requires further experimentation to establish level based norms.

The above design choices were also aimed at allowing for variety and novelty in search activities. One key method of maintaining engagement with games is to continually add new challenges and features allowing the players to master tasks while encouraging them to improve. This is commonly referred to as flow. The current prototype allows for the player to simply select the activity by three drop down menus that can be seen at the top of Figure 1. These menus determine three aspects of each activity: a) network style, b) network size, and c) content of nodes. In a final version of this game, these aspects will be determined though game play and access to the different activities with be the result of unlocking levels and leveling up through game achievements.

A. *Network Style*

The network style is defined by its local geometry and global structure. So far, there are four local geometries and seven global structures (using these four local geometries) implemented into the prototype. The simplest style is the one from Figure 1 that has a 90° cross as its local geometry that connects to create a square grid. Of its 25 nodes, only 9 of the nodes could potentially be bulls-eyes—the rest do not have degree four. The rest of the global structures come from local geometries that are a subsets of a 60° asterisk. Figure 2 shows a global structure created by local X geometry. Note that this structure contains 12 out of 30 nodes that can be bulls-eyes. One key difference between this pattern and a square grid is that there are three possible orientations for each node. This difference adds a layer of complexity to the search task.

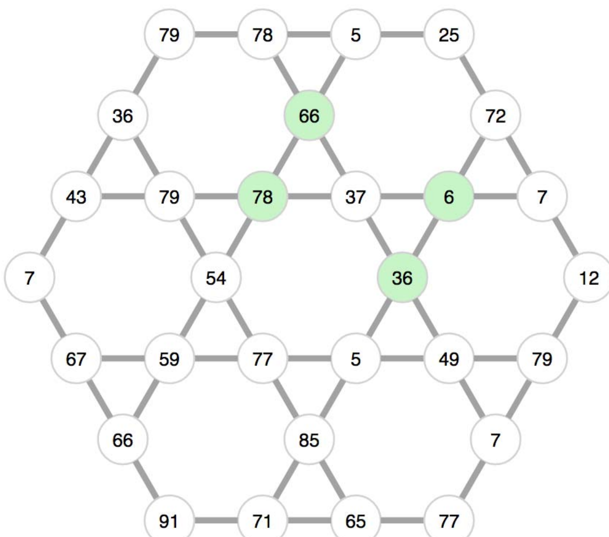

Figure 2: a global structure generated with an *X* pattern with a labelled multiple of 7 bullseye.

Another local geometry is a K pattern, which admits many possible structures as can be seen in figures 3,4, and 5. The global structure in figure 3 is isomorphic to the structure of figure 2, which means it also has 12 out of 30 nodes that can be bulls-eyes. As with the X pattern, there are three orientations of the local geometry present in the global structure.

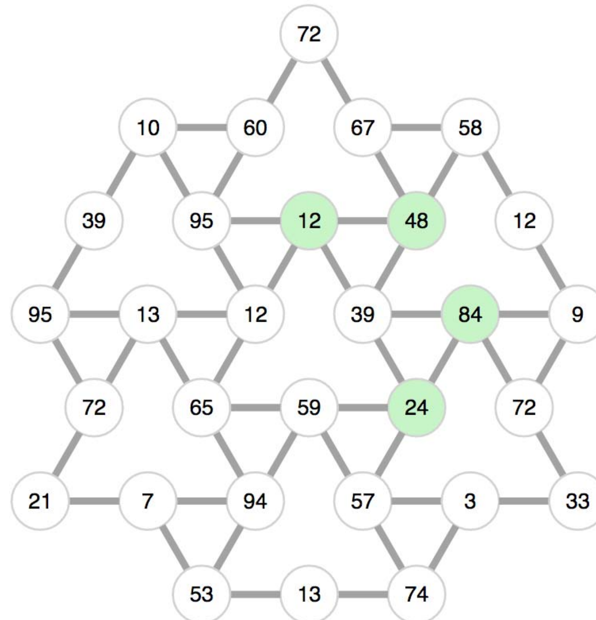

Figure 3: a global structure generated with a *K* pattern with a labelled multiple of 12 bullseyes

Figure 4 shows another global structure with an identifiable repeating structure (although currently, its edges are not as aesthetically pleasing as they will be in the final version of the game—which will be discussed in the next sub-section). In this structure, there are six different orientations of the K. In Figure 5, the final global structure based on a K pattern can be seen as a limited version of the Sierpiński Triangle. In this example, all but the three corners can possibly be a bulls-eye, which is 39 out of the 42 nodes. As well, notice that there are only three different orientations of its Ks.

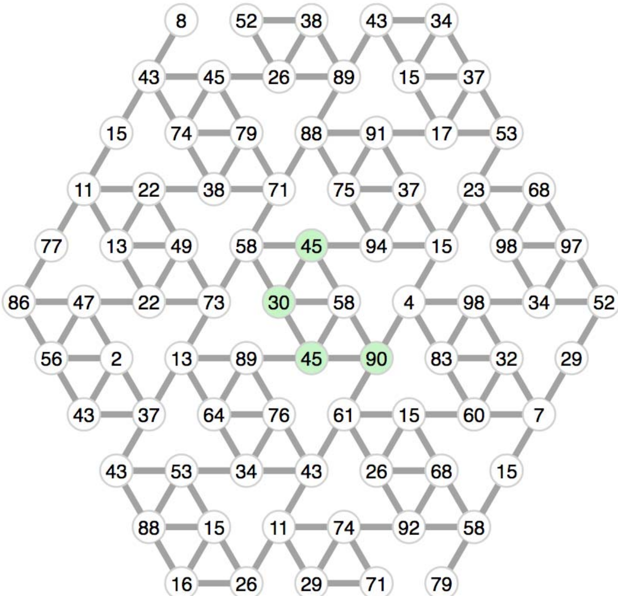

Figure 4: a global structure generated with an *K* pattern with a labelled multiple of 15 bullseye.

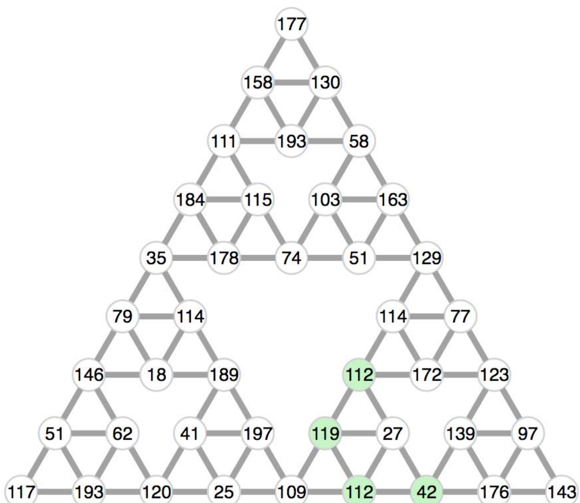

Figure 5: a fractal global structure generated with a *K* pattern with a labelled multiple of 7 bullseye.

The last of the local geometries implemented in the prototype is one that looks like the peace symbol ☮. Figure 6 shows a pattern constructed with six different orientations of the peace symbol and has 21 of 39 nodes that can be bullseyes. Figure 7 shows the last implemented global structure, which has only four different orientations of the peace symbol.

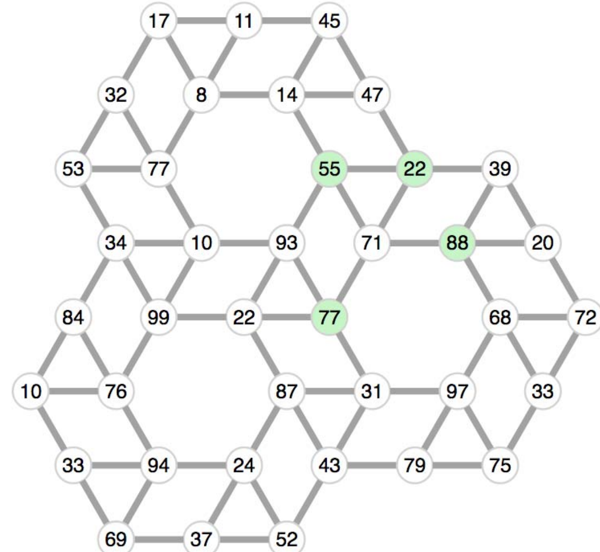

Figure 6: a global structure generated with a *peace* pattern with a labelled multiple of 11 bullseye.

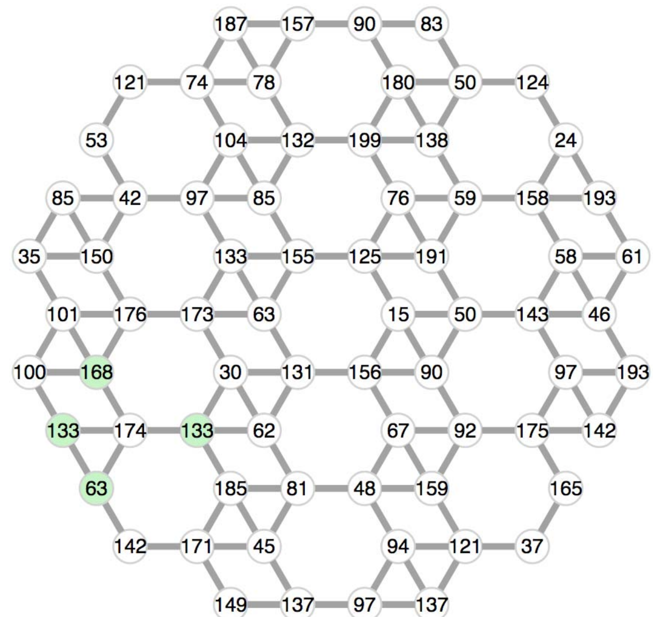

Figure 7: a global structure generated with a *peace* pattern with a labelled multiple of 7 bullseye.

These seven global structures shown in the figures 1-7 suggest a great potential for variety in activities even when limited to these 2D structures.

B. Network Size

Another source of variety in activities is the size of the structure that is being searched. These structures can be seen as highly repetitive and easily extended to larger structures. Although it would be possible to hand code these examples, it is much more flexible to create them using a specialized finite state machine (FSM). In all but the fractal example, the prototype uses the approach described in [CLA] where the language generated by a finite state machine is projected into space to make the network. In these examples (excluding the fractal one), each different orientation represents a state in the FSM. In general, that won't always be the case (these cases are not as complex as they could be).

This approach is augmented to use a priority queue to apply metrics that have the network grow in the prescribed way. These metrics essentially define growth of the network to fill concentric polygons. Figure 8 shows an example of the sequencing of the nodes' addition to the structure.

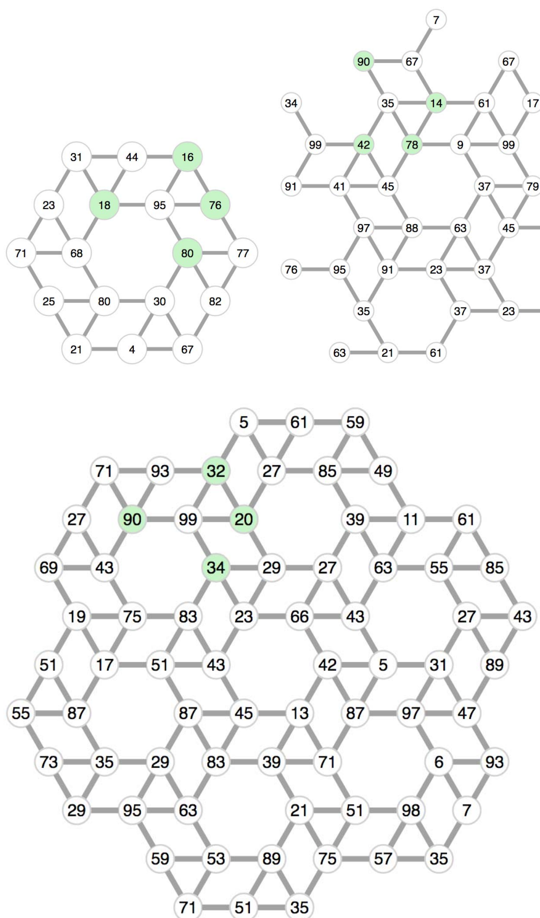

Figure 8: three different constructions of the first peace pattern with 18, 39, and 72 nodes.

In this case, a node's weighting in the priority queue is calculated by examining the distances from six lines going through the origin (a point in the center of the first hexagon). The middle network in Figure 8 does not match Figure 6 because for the generation of Figure 6 (the medium version with 39 nodes) the origin is set to the inside of its central triangle, making a much more appealing end result. On a related note, Figure 4's edges are not appealing because the metrics have not been honed for its unique pattern. The versatility of this method of construction is demonstrated by the total number of lines of code of the prototype which is under 1000 lines (currently 850 excluding JQuery and d3 libraries).

C. Node Content

The final aspect of an activity has to do with the node content. The prototype focuses on numbers and their factorizations, which can be deconstructed into a list whose properties are of the form that the node is a multiple of some number n. Depending on n, the property can vary in difficulty, and the bulls-eye thus harder to ascertain. Content difficulty is classified as easy, normal, hard, and expert. The easy-content properties relate to multiples of 2, 5, 10, and 11. This has to do with isolating attention down to easy-to-spot conditions. For n equals 2, 5, or 10, only last digit in the node needs to be checked (i.e. even for 2, 0 or 5 for 5, and 0 for 10). As well for nodes containing numbers 2 through 99, spotting multiples of 11 is also trivial—doubled digits: 11, 22, 33, … , or 99. So easy-content in the prototype has 2 digit nodes with 2, 5, 10, and 11 as the targeted multiple. Examples with easy-content can be seen in figures 1, 6, and 8 (having corresponding multiples 10, 11, and 2). The normal-content properties relate to multiples of 20, 3, 9, 6, 4: 20 ends in a 0 and has the tens digit even; 3 requires the sum of the digits is divisible by 3; 9 requires the sum of the digits is divisible by 9; 6 has to be both a multiple of 3 and of 2; and finally 4 requires that half the last two digits is even (4 is hardest normal-content). An example of normal-content is seen in figure 2 (multiple of 6). The hard-content properties relate to multiples of 18, 12, 15, and 7: 18, 12, and 15 rely on 2 previous skills whereas 7 has no shortcuts. An example of normal-content is seen in figure 3 (multiple of 12). Finally, the expert-content combines all the previous multiples but changes the possible code content from 2-99 to 11-199. Examples with expert-content can be seen in figures 5 and 7 (having corresponding multiples 6 and 7).

One of the benefits of using numbers and their factorizations as the content is that is easy to generate all of the content and associated information. While randomly generating the activities, it is important to structure the content so that interesting patterns occur in the nodes. Equivalency lists created with respect to the targeted multiple where the numbers in each list share what is common with the targeted multiple. With numbers, the commonality is just the lowest common multiple (LCM). An example of the lists is shown in Table 1. The lists are then used to create the activity. Notably, the bottom list makes bulls-eyes and the all of the lists are then used to fill in the rest of the nodes careful ensuring that no new bulls-eyes are created. The lists capture the connection all of the numbers have with the bulls-eye so more than patterns can be created within the activity for potentially varying the level of challenge. For example, the numbers relatively prime to 12 could be mostly avoided to make the activity more challenging. Currently, lists are only used to more evenly distribute the classes of numbers.

TABLE I
EQUIVALENCY LIST FOR 12

LCM	#	Equivalent numbers w.r.t 12
1	0	5 7 11 13 17 19 23 25 29 31 35 37 41 43 47 49 53 55 59 61 65 67 71 73 77 79 83 85 89 91 95 97
2	1	2 10 14 22 26 34 38 46 50 58 62 70 74 82 86 94 98
3	1	3 9 15 21 27 33 39 45 51 57 63 69 75 81 87 93 99
4	2	4 8 16 20 28 32 40 44 52 56 64 68 76 80 88 92
6	2	6 18 30 42 54 66 78 90
12	*3*	12 24 36 48 60 72 84 96

Pedagogically the teacher can make decisions about the appropriate level of game, the order in which particular factors are reinforced, whether a timer is used and how much time should apply, while allowing faster learners to self pace through levels, perhaps under test conditions. The effect of drill and practice is retained, but subordinated to a game context, rather than a recital. At the same time visual and pattern strategies are developed, which are essential in a multidimensional world of digital hyper information.

III. Extending the Application

In design research, a core principle of IT artifact design is artifact mutability. Although a nuanced and elusive concept (Pöppelbuß & Goeken, 2015) inasmuch as theory underpins the artifact as a research contribution, the changes implies as it undergoes implementation or redesign should be considered. We have taken the view that foundational concepts are best instructed through practice and exposure to examples, but embodied this in a visual game rather than a recital context, with optional timing for motivation and testing. We believe other mathematics concepts, such as estimation, may be taught using versions of this game, but in this section we wish to consider the adaptation of the design artifact for learning basic language patterns.

No matter the geometry and global structure shown above, the game itself is simple: it focuses on identifying patterns. In one example above, the game player must be able to identify multiples of ten: these numbers need not correspond to the number in the "bulls-eye". Thus, without a cue within the network, a player must consider the prompt, think about the patterns that are possible, then seek to find those patterns. The time element allows players to test themselves based on pattern recognition speed -- against him or herself, other players, or a norm based test clock. This suggests: if these patterns can be used to learn, say, factors of a given number, could they also be used to learn patterns in language? Learners of any language must learn patterns; such patterns are the foundations of language and communication. Understanding the root of a word, or its stem, from which other words are derived, is vital in this, and forms the basis of traditional stemming based information retrieval. Patterns are not just at the word or sentence level but implicit in the act of reading itself. Indeed, Goodman (1967) noted how reading is a series of successful predictions that are either confirmed or proven false.

Moving from the orthography to whole words, from words to phrases and sentences, from sentences to paragraphs, and from one paragraph to another, readers can discern the meaning of a given text. Moreover, at the macro level one text is compared to another. At any of these levels, the pattern of reading captured by Goodman is the same: readers recognize something, predict what comes next, and confirm that prediction. If the reader makes a mistake, he or she must correct it, or terminate the exercise. The timer – and, for example, scoring points awarded due to finishing a task quickly, – helps gamify the task.

At the same time, not all prompts nor would just any linguistic pattern work well within the prototype. Full texts would not work in the current form of the app. Gamified activities, according to Chou (), function best when there is a sense of 'development' and 'accomplishment'. Thus more bottom-up, discrete skills would work best within the prototype.

Working within a two-minute timer of the prototype, there are several possible crossover applications for language learners. To start with, within a given program, it might be most effective to identify the patterns that could be taught or reinforced. At the lowest level, learners could learn to identify consonants or vowels (a, e, i, o, u). Moving up the linguistic ladder, new readers could learn to identify words that start with a given letter. Further still, the game could focus on past tense verb endings either regular (looked, cooked, tricked, kissed) or irregular (bought, written, spoken, ran) and progressive verbs (looking, running, jumping, thinking). At the lexical level, the same game can help learners identify synonyms (looked, watched, glanced, stared) or collocations (for the prompt 'vehicles' - car, bus, taxi, truck) based on a given prompt. All of these are possible developments within the games current formats.

IV. Conclusions

This paper has sought to show an application designed to augment student abilities to discern patterns in mathematics, whilst reinforcing basic learning through playing a game. It is our hope that act of helping students with lower order processing (in this case, the times tables) will enable students to then move on to higher order processing through development of strategies in search and pattern recognition that transcend the simple task. The application also seeks to teach students this in a way that neither boring nor rote but fun and challenging.

The students' lack of background in maths fundamentals, their preference for screen-driven learning in the digital age, and basic principles of gamification have all provided impetus to the inception of this game. It is assumed that even as STEM education is becoming more valued, students have continued to struggle with these fundamentals.

Next, the paper served to shed light on the essential choices involved in design. Design choicess from the degree 4 pattern to the lack of a restriction on "bulls-eye" content to a countdown timer, serve to focus the gameplayer on the salient aspects to be reinforced by the game. The game is scalable in both size and content, provided that the essential architecture of the game remains the same. In addition, the application can remain under these same constraints and use the same architecture whilst having possible uses outside the STEM domain, as suggested for basic language structure learning.

To pursue this application further, several additional items must be reviewed and/or tested. At this point the application is a prototype; further evaluation under various circumstances is required. Secondly, to register the effectiveness of the application, student times must be logged and measured. If possible, in a controlled environment, it would be prudent to study how the lower-order skills taught and reinforced by the app can affect higher-order processing. The logging and measuring must be measured first, and we would want to set up a way of empirically measuring any effects the application had on higher order processing as well. Along with this, perhaps eye-tracking software could assist to distinguish the differences between superior, average, and below average players of the game. For instance, if superior players had certain eye patterns that the other two groups did not, this could be measured, assessed, and eventually taught to all players. We have applied this in reading diagnostics in previous work [7]. Finally, cross-platform applications should be set up and tested so that the game would act in the same way and yield the same result, regardless of platform. The html 5 and Javascript basis of the design is intended to facilitate such cross platform usage.

As further work, we would like to take the prototype and extend it for linguistic purposes. Just as math has its lower-order processing (times tables, sums, etc.), the field of language has its own bedrocks on which the rest of learning is built. In particular, we wish to explore discrete, lower-order processing at the lexical level, which would provide an analogous situation to the one in the original artifact. We believe this basic design allows leveling and application for a range of fundamental STEM concepts, and can be extended to language learning elements as well.

V. References

[1] Hammersley JM (1968) On the enfeeblement of mathematical skills by modern mathematics and by similar soft intellectual trash in schools and universities EducationalStudies in Mathematics 1(1):17 · May 1968

[2] Ball J and Randerson J (2012) Have GCSEs got easier? https://www.theguardian.com/politics/reality-check-with-polly-curtis/2012/jun/21/have-gcses-got-easier

[3] Boaler J (2015) In Stanford P. 'Times tables are not how you teach maths' Daily telegraph (London) http://www.telegraph.co.uk/education/educationopinion/12045884/Times-tables-are-not-how-you-teach-maths.html

[4] Boaler, J Chen L, Williams C & Cordero M (2016) Seeing as Understanding: The Importance of Visual Mathematics for our Brain and Learning. Stanford University. https://drive.google.com/file/d/0Bz8cVS8LoO7OU0pZdnVGQnh3Wm8/view

[5] Yildirim I.G. (2015) *Time Pressure As Video Game Design Element And Basic Need Satisfaction* MSc thesis. Middle East Technical University http://etd.lib.metu.edu.tr/upload/12618994/index.pdf

[6] Morey, J., Sedig, K., Mercer, R.E., and Wilson, W., Crystal Lattice Automata, in Proceedings of the Fourth International Conference on Implementations and Applications of Automata, Pretoria, South Africa, July, 2001 (Lecture Notes Computer Science, Watson, B.W. and Wood, D., Eds., Berlin: Springer, 2002, vol. 2494, pp. 214–220.

[7] J. Morey and J. Gammack, "A data visualisation for horizontal eye-movements," *2015 2nd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE)*, Nadi, 2015, pp. 1-4.

[8] Pöppelbuß, J., and Goeken, M. 2015. “Understanding the Elusive Black Box of Artifact Mutability,” Wirtschaftsinformatik Proceedings. online: http://aisel.aisnet.org/wi2015/104/