Drops and Kinks: Modeling the Retention of Flow for Hour of Code Style Tutorials

Alexander Repenning
University of Applied Sciences and Arts Northwestern
Switzerland (FHNW)
Windisch, Switzerland
alexander.repenning@fhnw.ch

Ashok Basawapatna
SUNY College at Old Westbury
Old Westbury, NY
basawapatnaa@oldwestbury.edu

ABSTRACT

It can be difficult to evaluate Hour of Code activities for outcome measures such as motivation. Participation levels, for example, might be more indicative of marketing effectiveness and give little insight into longitudinal user engagement. By imagining these activities as a series of steps, we can develop a survival function model based on simple Markov chains. The student-retention this model predicts can be compared to empirical retention data gathered from traditional step-by-step and puzzle based programming tutorials. Retention of Flow is an affective evaluation [1] instrument that compares empirical student retention data to this model to better understand student motivation throughout the activity and beyond. This paper discusses two specific aspects of this Retention of Flow analysis. Drops, or sharp declines in retention, indicate a loss of motivation resulting from cognitive, practical and technical challenges. Kinks in retention indicate more gradual shifts in activity motivation. This paper uses data from a puzzle and a tutorial-based Hour of Code activity to show how understanding the Retention of Flow as a mathematical model can help with the evaluation and the design of programming tutorials.

Keywords

Evaluation; computer science education; retention; Flow;

1. INTRODUCTION

Interest in computer science education has reached a critical tipping point. Events such as the European CodeWeek and CSEdWeek have participants numbering in the hundreds of millions according to Code.org's website. Furthermore, initiatives such as US Computer Science for All aim to "empower a generation of American students with the computer science skills they need to thrive in a digital economy." The idea of providing extremely low threshold programming activities reaching millions of students is most prominently captured by the Hour of Code annual event that started in 2013. However, without proper evaluation it is not clear what motivational and educational consequences the participation has. If, for instance, participants' levels of motivation towards the end of an hour long activity are significantly fading, then their perception of programming to be "hard and boring" may actually get reinforced [2].

In the context of the Scalable Game Design project we have developed research instruments to assess cognitive [3, 4] and affective [5] measures related to Computational Thinking [6, 7].

Scalable Game Design is a curriculum teaching Computational Thinking by introducing students to gradually more sophisticated game design projects resulting in CT skills that can later be leveraged to create STEM simulations [8]. With over 20,000 subjects, Scalable Game Design includes the largest formal study in US middle schools exploring a strategy on how to systematically [9] and sustainably [10] introduce Computer Science education in public schools. Recently, we have developed a means of evaluating Hour of Code type activities, through a method entitled the Retention of Flow. Retention of Flow analysis has been employed to gauge motivation of a 3D Frogger Hour of Code activity [11] and to compare localized versions of the same tutorial used in different countries (3D Frogger used in the USA, Mexico and Switzerland [12]).

Retention of Flow is an evaluation framework to assess motivation in Hour of Code type activities as an affective measure. This is achieved by comparing the percent of students retained through each step of the activity with a mathematical model describing how student participation should theoretically decline through the activity. Specifically, Retention of Flow is used to analyze programming projects created with the use of a tutorial in a context where participation is voluntary and can be terminated at any point in time. Stopping points collected can be aggregated easily into a retention distribution representing a survival function. This survival function, in turn, can indicate how many people advanced from one step in a tutorial to the next one. Prominent inflection points found in actual retention data can be highly indicative of cognitive, technical and practical challenges that reduce student motivation to continue with the activity [12].

There are a number of existing instruments available to evaluate motivation including pre/post questionnaires, observational approaches [13], and even physiology-based approaches, for instance assessing engagement and Flow [14] through eye tracking. Common to many of these approaches is a relatively large effort to collect and analyze data even with small participation numbers. Retention of Flow has the opposite problem. It can be computed completely automatically but does require large numbers, e.g., thousands, of participants. One appealing characteristic of a Retention of Flow analysis is it is particularly simple to implement on large numbers of participants, for example cloud-based computing environments.

This paper introduces a simple mathematical model based on a Markov chain to match the retention data from two different Hour of Code tutorials. It contrasts the nature of these tutorials, presents the associated retention data and briefly discusses potential consequences of the model for the interpretation of student motivation throughout these tutorials.

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. *WiPSCE* '16, October 13-15, 2016, Münster, Germany © 2016 ACM. ISBN 978-1-4503-4223-0/16/10...\$15.00DOI: <http://dx.doi.org/10.1145/2978249.2978260>

2. RETENTION OF FLOW MODEL

Considered to be a survival function, retention data collected from tutorial participants describes how quickly, on average, interest in an activity fades over time. Figure 1 depicts a simple Markov chain used to model the survival function. A tutorial is conceptualized as a sequence of discrete steps. Each step involves a human decision leading to one of two possible outcomes. If participants enjoy the activity they will move to the next step of the tutorial. If their levels of motivation fall below a certain threshold they will stop their participation and give up. These state transitions can be expressed as probabilities $P(cont)$ to continue from the current step to the next one. $P(stop)$ is the probability to stop with $P(stop) = 1 - P(cont)$. These probabilities reflect how much participants have enjoyed an activity so far combined with their subjective prediction of how challenging the next steps appear to be.

Our *Retention of Flow conjecture* attempts to relate Csikszentmihalyi's notion of Flow to student activity retention [16]. For example, a tutorial might try to keep participants in the Flow by carefully balancing challenges posed and skills acquired to follow instructions. To achieve this, two extremes of unbalanced challenges and skills need to be avoided. On the one hand, highly challenging instructions for users lacking essential skills will result in participant anxiety. On the other hand, participants with skills exceeding challenges, will result in participant boredom. One might design a tutorial such if the participant can do step one, they can do all the subsequent steps. For instance, in some IKEA furniture and Lego construction assembly guides, it makes sense to keep all the instructions, roughly at the same level of challenge. The Retention of Flow conjecture suggests that devoid of any marked increase in challenges, all the continuation probabilities at each step of the tutorial are more or less identical. Given this to be the case, it is sufficient to conclude that the resulting survival function would be a negative exponential function with a $P(cont)^n$ probability to reach the endpoint of an n -step tutorial. Figure 1 depicts a Markov chain model based on this conjecture as well as the resulting retention plot predicted by the model. These kinds of negative exponential functions are not unique and have been found in diverse contexts including the participation in MOOCs [15].

Figure 1: Retention of Flow model based on Markov Chain with identical probabilities.

How closely will real tutorials follow this theory? Of course there are many factors that could influence the probabilities of the Retention of Flow Markov chain model significantly. For instance, is the motivation to participate intrinsic or extrinsic [16]? An example of an extrinsic motivation would be that teachers force students to participate in an activity. Sometimes the boundary between intrinsic and extrinsic motivation can be

difficult to assess. For instance, the idea of wanting a chair may be intrinsic motivation to buy this chair at IKEA but the motivation necessary to assemble the chair is likely to be extrinsic. Unlike with the LEGO construction kit example, the IKEA user may only put up with a tutorial in order to save money or get a very specific product. These factors may have profound consequences on continuation probabilities. A low continuation probability would result in a quickly decaying negative exponential function. A badly designed tutorial, for instance five trivial chair assembly instructions followed by an instruction that is really difficult to execute, may result in a wide range a probabilities reflecting a certain unevenness of the challenge versus skills balance over time. Csikszentmihalyi's notion of Flow is the description of a psychological state experienced "in the moment" [17]. This notion of Flow only provides a limited account for Flow over time as it would be relevant to most educational activities. Many tutorials have an educational goal. To understand the efficacy of a tutorial it would be necessary to have instruments that allow designers to measure the probabilities. The Drops and Kinks framework is a mechanism to determine these probabilities.

3. ANGRY BIRDS VERSUS 3D FROGGER

The Retention of Flow model was applied to retention data from two radically different Hour of Code tutorials. Angry Birds [20] is a puzzle-based tutorial introducing participants gradually to key programming concepts through twenty fill in the blank challenges. The first challenge, for instance, makes participants fill in a move block to make the bird get to the pig. 3D Frogger is our own tutorial, based on AgentCubes online [2, 18, 19], that was used during the 2013 Hour of Code by nearly a quarter million participants (this study was done on a more modest subsample of 5,000 participants from 2016). The step-by-step tutorial of 3D Frogger makes participants draw their own 3D objects, e.g., a frog, and program them to create a complete game. 3D Frogger was designed to be a cliff hanger tutorial meaning that at the end of the one hour a working part of the game was created but it was hoped that participants would be sufficiently motivated after the hour to continue programming.

Figure 2: Angry Birds retention versus 3D Frogger retention

In spite of the philosophically opposing approaches – puzzle-style versus step-by-step tutorial and 3rd party high production value art versus draw your own – both approaches result in retention distribution that is matched surprisingly closely (Figure 2) by the same negative exponential trend line capturing the Retention of Flow model. In [19] Harms found higher degrees of motivation

for puzzle-based tutorials compared to step-by-step ones but explored tutorials built by the same authors.

The Angry Birds distribution based on data analyzed by Piech [20] has fewer retention data points as retention was only measured after each of the twenty different puzzle challenges. The 3D Frogger retention data is finer grained capturing the addition of each Line of Code (LOC) by participants. Both distributions are normalized in terms of magnitude and time. In Angry Birds the participants who have completed the first puzzle are considered 100%. In 3D Frogger the participants who wrote the first line of code are considered 100%. In Angry Birds participants were assumed to solve 20 puzzles in one hour. In Frogger 3D participants were assumed to create the first part of the game, a Frog able to move around to a goal state in one hour resulting in 59 lines of code.

Three main features indicative of Flow, or lack thereof, were identified in the retention data:

- **Survival Rate:** This is the survival function rate at which participants complete a certain number of tutorial steps. Because both activities were billed as Hour of Code tutorials it makes sense to look at survival rates after the number of steps typically followed in one hour. Amazingly, in spite of the radically different approaches the survival rate of both tutorials was virtually the same (64%).
- **Drops:** Sharp drops, e.g., the drop at 45 LOCs in our 3D Frogger tutorial, are distinct inflection points suggesting cognitive, technical or practical challenges that have been discussed in depth elsewhere [11]. The fact that these drops are sharp suggests highly localized challenges that can be traced back precisely to the tutorial instructions responsible. For instance, the drop at 45 LOCs is cognitive challenge where some participants misinterpreted the instructions resulting in a program that would no longer work. This caused significant frustration. The drop at 59 LOCs was a practical challenge resulting from the tutorial chapter ending aligned with the one-hour activity mark. A technical challenge was indicated by a drop most pronounced by retention data collected from participants in Mexico [12]. Drops are simple to spot and can be correlated precisely to the instructional materials such as the time stamp in a video. The drop simply documents that there is a challenge but does not provide explanations on what is causing the challenge or how the challenge could be addressed. Combined with A/B testing [21] Retention of Flow can be used to pinpoint and even fix usability issues.
- **Kinks:** Kinks are more gradual shifts in motivation resulting in changes to the continuation probabilities. They are much harder to spot by looking at the retention function. In essence kinks are deviations from the negative $P(i) < 1.0$ exponential decay. A better way to spot kinks is to compute the probability distribution by deriving it from the retention distribution. The probability distribution $P(i)$ is simply the ratio of the current retention $R(i)$ to the retention of the next step $R(i+1)$: $P(i+1) = R(i) / R(i+1)$. Figure 3 combines the 3D Frogger retention distribution also found in Figure 2 with the continuation probability.

The continuation probability graph helps with the identification of all three main features to assess Flow. The one hour survival rate can be computed as $P(cont)^n$ with n being the average number of steps completed in one hour and $P(cont)$ being the average probability. Higher values for average $P(cont)$ suggest higher degrees of Flow. While $P(cont)$ may not necessarily be a direct

measure of Flow, e.g., students may have been coerced into participation, it can be considered a strong indicator. $P(cont)$ also depends on the granularity of the retention data. With 3D Frogger the $P(cont)$ average for the first hour is 0.989, i.e., there is a ~99% chance for a participants to continue after each instruction. A linear trend line in Figure 3 ($y = -1E-05x + 0.9898$) suggests a very low gradual loss of continuation probability over time.

Figure 3: 3D Frogger Retention of Flow (blue, 0-100%) and Continuation Probability (red, 0.9-1.0) graphs as function of Lines of Code (0-200).

Drops are more accentuated in the continuation probability graph. They can be witnessed as low negative values with the most extreme drop at 45 LOC corresponding to a continuation probability of 0.918. Less severe drops can be found all the way to the end of the graph representing participation of many hours. These drops are much more pronounced in the continuation probability graph compared to the retention graph.

Kinks in the continuation probability can be witnessed as positive and negative trends of probability values. For instance, from the beginning of the activity to the first drop there is a positive trend (Figure 3: +Kink) possibly suggesting increasing Flow among participants. Then after the first drop there is a negative trend (-Kink) suggesting a gradual loss of Flow. This part in the tutorial deals with the creation and testing of four IF statements to control the frog to move in four different directions. Up to this point the 3D Frogger tutorial retains significantly higher percentage of participants than the Angry Birds activity. The drop resulting from the introduction of the IF statement and the following kink may be due to how early the IF statement was introduced in the 3D Frogger tutorial compared to Angry Birds one.

4. CONCLUSIONS

The design and evaluation of effective programming tutorials is difficult and time consuming. With the advent of online activities integrating tutorials into cloud based programming environments it is becoming relatively simple to collect data from large groups of participants. Using the Retention of Flow model based on Markov chains it is effortless to analyze the efficacy of instructions. Continuation probability, that is the probability to advance from one instructional step to the next one, can be indicative of Flow over time. Retention of Flow not only predicts negative exponential retention functions but it can also use inflection points such as drops and kinks as indicators of challenges. Retention of Flow was used to compare two very

different Hour of Code tutorials, Angry Birds and 3D Frogger, and found that in spite of diametrically opposed philosophical approaches the retention after one hour was essentially identical.

5. ACKNOWLEDGEMENTS

This work is supported by the Hasler Foundation, the Swiss National Science Foundation under grant CRAGP2_158545, and the National Science Foundation under Grant Numbers 0833612, 1345523, and 0848962. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of these foundations.

6. REFERENCES

- [1] R. Baker, S. D'Mello, M. Rodrigo, and A. Graesser, "Better to Be Frustrated than Bored: The Incidence, Persistence, and Impact of Learners' Cognitive-Affective States during Interactions with Three Different Computer-Based Learning Environments," *International Journal of Human-Computer Studies*, vol. 68, pp. 223-241, 2010.
- [2] A. Repenning, D. C. Webb, C. Brand, F. Gluck, R. Grover, S. Miller, H. Nickerson, and M. Song, "Beyond Minecraft: Facilitating Computational Thinking through Modeling and Programming in 3D," *IEEE Computer Graphics and Applications*, vol. 34, pp. 68-71, May-June 2014.
- [3] M. Bienkowski, E. Snow, D. Rutstein, and S. Grover, "Assessment Design Patterns for Computational Thinking Practices in Secondary Computer Science: A First Look," SRI International 2015.
- [4] K. H. Koh, H. Nickerson, A. Basawapatna, and A. Repenning, "Early validation of Computational Thinking Pattern Analysis," presented at the Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education (ITICSE), Uppsala, Sweden, 2014, 213-218.
- [5] R. W. Picard, P. S., W. Bender, B. Blumberg, C. Breazeal, D. Cavallo, T. Machover, M. Resnick, D. Roy, and C. Strohecker, "Affective Learning — a Manifesto," *BT Technology Journal*, vol. 22, pp. 253-269, 2004.
- [6] J. M. Wing, "Computational Thinking Benefits Society," in 40th Anniversary Blog of Social Issues in Computing vol. 2014, J. DiMarco, Ed., ed. <http://socialissues.cs.toronto.edu/2013/01/40th-anniversary/>: University of Toronto, 2014.
- [7] J. M. Wing, "Computational Thinking," *Communications of the ACM*, vol. 49, pp. 33-35, 2006.
- [8] A. Repenning, D. C. Webb, K. H. Koh, H. Nickerson, S. B. Miller, C. Brand, I. H. M. Horses, A. Basawapatna, F. Gluck, R. Grover, K. Gutierrez, and N. Repenning, "Scalable Game Design: A Strategy to Bring Systemic Computer Science Education to Schools through Game Design and Simulation Creation," *Transactions on Computing Education (TOCE)*, vol. 15, pp. 1-31, 2015.
- [9] A. Repenning, D. Webb, and A. Ioannidou, "Scalable Game Design and the Development of a Checklist for Getting Computational Thinking into Public Schools," presented at the SIGCSE 2010, The 41st ACM Technical Symposium on Computer Science Education, Milwaukee, WI, 2010, 265-269.
- [10] K. H. Koh, A. Repenning, H. Nickerson, Y. Endo, and P. Motter, "Will it Stick? Exploring the Sustainability of Computational Thinking Education Through Game Design," presented at the ACM Special Interest Group on Computer Science Education Conference (SIGCSE 2013) Conference, Denver, Colorado, USA, 2013, 597-602.
- [11] A. Repenning, A. Basawapatna, D. Assaf, C. Maiello, and N. Escherle, "Retention of Flow: Evaluating a Computer Science Education Week Activity," presented at the Special Interest Group of Computer Science Education (SIGCSE 2016), Memphis, Tennessee, 2016.
- [12] N. Escherle, S. Ramirez-Ramirez, A. Basawapatna, D. Assaf, A. Repenning, C. Maiello, Y. Endo, and J. Nolasco-Florez, "Piloting Computer Science Education Week in Mexico," presented at the Special Interest Group of Computer Science Education (SIGCSE 2016), Memphis, Tennessee, 2016.
- [13] D. C. Webb, S. B. Miller, H. Nickerson, R. Grover, and K. Gutiérrez, "Student Centered Observation Protocol for computer-science Education (SCOPE)," University of Colorado at Boulder 2014.
- [14] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*. New York: Harper Collins Publishers, 1990.
- [15] C. Coffrin, L. Corrin, P. de Barba, and G. Kennedy, "Visualizing patterns of student engagement and performance in MOOCs," presented at the Fourth International Conference on Learning Analytics And Knowledge, 2014, 83-92.
- [16] M. Vansteenkiste, Lens, W., & Deci, E. L., "Intrinsic Versus Extrinsic Goal Contents in Self-Determination Theory: Another Look at the Quality of Academic Motivation," *EDUCATIONAL PSYCHOLOGIST*, vol. 41, pp. 19-31, 2006.
- [17] M. Csikszentmihalyi, *Finding flow in everyday life*. New York: BasicBooks, 1997.
- [18] A. Repenning, "Making Programming Accessible and Exciting," *IEEE Computer*, vol. 18, pp. 78-81, 2013.
- [19] A. Ioannidou, A. Repenning, and D. Webb, "AgentCubes: Incremental 3D End-User Development," *Journal of Visual Language and Computing*, vol. 20, pp. 236-251, 2009.
- [20] C. Piech, M. Sahami, J. Huang, and L. Guibas, "Autonomously Generating Hints by Inferring Problem Solving Policies," presented at the Proceedings of the Second (2015) ACM Conference on Learning @ Scale, Vancouver, BC, Canada, 2015, 195-204.
- [21] P. Hynninen and M. Kauppinen, "A/B testing: A promising tool for customer value evaluation," in *Requirements Engineering and Testing (RET)*, 2014 IEEE 1st International Workshop on, 2014, pp. 16-17.