Modelling Achievement and Intrinsic Motivation in An Educational Game

Clara Ye

May 9, 2022

1 Introduction

Understanding the players' motivation in educational games is crucial for developers since keeping the players engaged in the game is closely related to learning from the game. Quite recently, Locke and Schattke (2019) proposed a trichotomy of motivation that differentiates among extrinsic motivation, achievement motivation, and intrinsic motivation. Extrinsic motivation is defined as "doing something in order to get some future value" (p.14). It is easiest to distinguish since its concern is in the consequences of the activity, instead of the activity itself. Achievement motivation is defined as "the recurrent concern for a standard of excellence" (p.17). It is essentially the desire to perform well in the task, and may or may not involve personal interest. Intrinsic motivation is defined as "liking or wanting an activity for its own sake" (p.8). It is usually guided by interest or fun, regardless of the performance or external reward. Among the three types of motivation, achievement and intrinsic motivation is more prominent in the context of educational games. Achievement motivation is associated with feeling happy or proud when succeeding in the game, while intrinsic motivation is associated with feeling bored or tired when repeating the game.

The current project seeks to model achievement and intrisic motivation in playing the educational game Battleship Numberline (Lomas et al., 2011), which trains the player to associate fraction numbers to positions on the number line. Figure 1 shows the interface of the submarine mode[1]. When each round begins, there is nothing visible in the ocean, and the player would read a message "Ship spotted at:" followed by a fraction number at the bottom of the screen. The fraction is randomly selected from the following 20 fractions: $\frac{1}{10}$, $\frac{1}{8}$, $\frac{1}{6}$, $\frac{1}{5}$, $\frac{1}{4}$, $\frac{3}{10}$, $\frac{1}{3}$, $\frac{3}{8}$, $\frac{2}{5}$, $\frac{3}{7}$, $\frac{1}{2}$, $\frac{3}{5}$, $\frac{5}{8}$, $\frac{2}{3}$, $\frac{7}{10}$, $\frac{3}{4}$, $\frac{4}{5}$, $\frac{5}{6}$, $\frac{7}{8}$, $\frac{9}{10}$. Then the player needs to click on the corresponding position on the number line from 0 to 1. After the click, a submarine would show up at the correct position as feedback. The players are free to continue or quit the game.

Battleship Numberline has been used to investigate the relationship between challenge, engagement, and learning in educational games among a massive pool of players (Lomas et

[1] The game also has a ship mode, where the player observes a ship at the beginning of each round and is expected to enter a fraction that corresponds to its position.

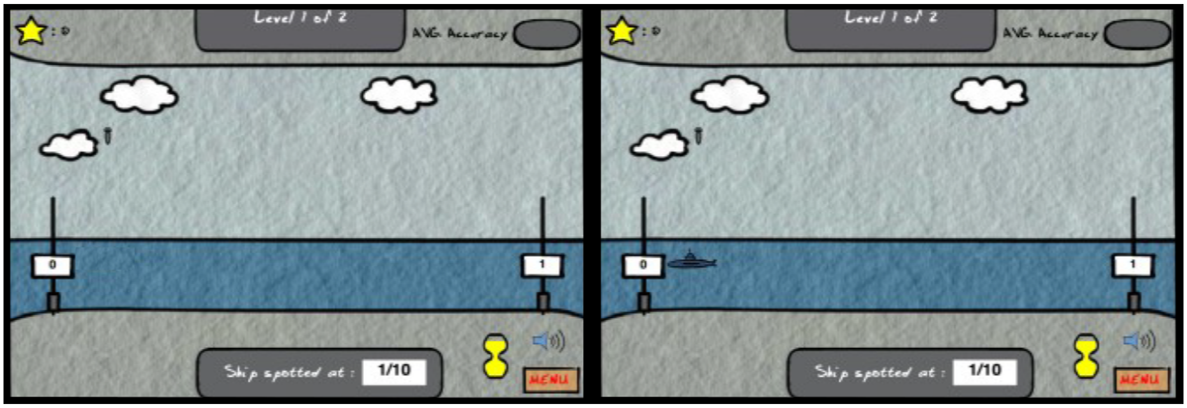

Figure 1: Game interface of the submarine mode of Battleship Numberline, at the beginning of the round (left) and after the player makes the attack (right).

al., 2013). The authors manipulated a variety of game design factors, including the type of the target (submarine or ship), the size of the target, the time limit, and the problem presentation sequence. They also classified players into high-ability and low-ability based on whether they scored above or below the median. Considering the scope of this project, only the effect of target size and time limit on challenge and engagement among low-ability players is modeled.

2 Methods

2.1 Experimental Manipulations

Target Size The target size is defined as the proportion that the submarine occupies on the number line, and takes one of the following nine values: 2%, 3%, 4%, 5%, 8%, 10%, 12%, 15%, 20% [2] [3]. It is essentially the error tolerance rate - with larger targets, the player's estimate can deviate more from the correct position while still being judged as a success, and so the game would be less challenging.

Time Limit The time limit for each trial takes one of the following eight values: 2s, 3s, 4s, 5s, 8s, 10s, 15s, 30s. The longer the time limit, the more time the player can spend on carrying out whatever strategy they take to get more accurate estimates, and so the less challenging the game.

The nine conditions for target size and eight conditions for time limit hence lead to a 9 × 8 factorial design.

[2] In Online Experiment 2 by Lomas et al. (2013), there was a discrepancy in what was reported in the methods section and what was plotted in the results section. The values used here are taken from the plots.

[3] This is probably inappropriate for the report, but I do want to complain that I wasted so much time trying to make my model match the wrong set of human data before I noticed this...

2.2 Measures

Challenge Lomas et al. (2013) measured challenge as the success rate, which is the proportion of success trials out of all trials. To make the wordings more intuitive, this report would refer to higher success rate as lower challenge instead.

Engagement The outcome measure is player engagement, which combines total play time and number of trials played by

$$\text{Engagement} = \log(\text{Time} \times \text{Trials})$$

where log is the natural log.

2.3 Procedure

An ACT-R model is developed to play the game, the details of which will be discussed in section 3. Model performance data are collected by running 100 simulated subjects for each of the 72 experiment conditions, with the same parameter settings.

The human subject data used for model evaluation are transcribed from the figures from the study by Lomas et al. (2013) by estimating the position of the tick marks, since they did not report numerical results and only displayed results in figures [4]. The error bars are not transcribed since transcription accuracy is likely to be low.

3 Implementation

3.1 Task

The task interface for modeling purpose is a simplified abstract representation of the game interface. Figure 2 illustrates a comparison between the actual game interface and the abstract representation. The instructions are omitted and there is only the fraction at the bottom. The ocean and the number marks are abstracted away to a blue line. The submarine becomes a red line with the position and length corresponding to the specified fraction number and target size. The game procedure is exactly the same as the original Battleship Numberline.

3.2 ACT-R model

Figure 3 shows the steps by which the ACT-R model plays the game. It first reads the fraction on the screen. Fractions are represented by three slots: the *numer*ator, the *denom*inator, and the associated *posit*ion on the number line. The model has three starting fractions in declarative memory whose correct positions on the number line are known: $\frac{1}{2}$, $\frac{1}{4}$, and $\frac{3}{4}$. The positions for all other fractions need to be learned as the model progresses through the game.

The model then attempts to retrieve the fraction from declarative memory using the observed numerator and denominator. Partial matching is enabled, where the similarity between

[4] The estimation accuracy should be reasonable since the estimated curve has very similar shapes as the original curves, though the comparison is not included in the report for concision.

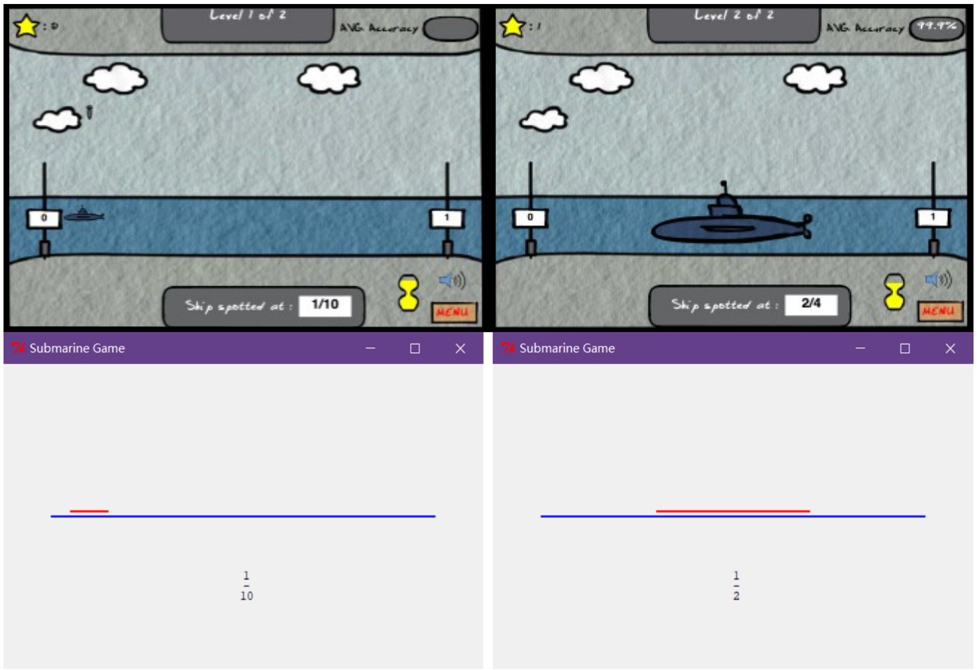

Figure 2: Comparison of the real game interface (top) and the abstract task representation (bottom), with different combinations of target sizes and positions (left: 5% and $\frac{1}{10}$; right: 20% and $\frac{1}{2}$).

numbers are set as

$$\text{Similarity}(a, b) = -\frac{|a-b|}{10}.$$

If the retrieval is successful, then the model will click on the position corresponding to the value recorded in the *posit* slot. If it fails to retrieve anything, then it will simply click at the middle of the number line. Cursor noise is enabled to simulate estimation errors. The default target width used in Fitt's Law is set to the width of a submarine with 10% size.

After the model makes the click, there are two aspects of processing the solution. First, it encodes the midpoint of the displayed target in the *posit* slot. More importantly, the model will receive reward based on whether the click position is correct and the number of times it has seen the fraction before. The relationship is given by

$$\text{Reward} = \text{Baseline} - 6 \times \text{Repetitions}$$

where the baseline reward is 10 for a correct click and -5 for an incorrect click. That is, the model receives a positive reward for being correct and a negative reward for being wrong, which reflects achievement motivation, and gets additional negative reward for repeatedly encountering a problem, which reflects intrinsic motivation.

The framework for handling the choice of whether to continue playing is inspired by the work of Nagashima et al. (2020). After the model finishes processing the solution, there are always two matching productions - the continue production and the end production - that the model will resolve based on their utilities. The continue production leads to a key press

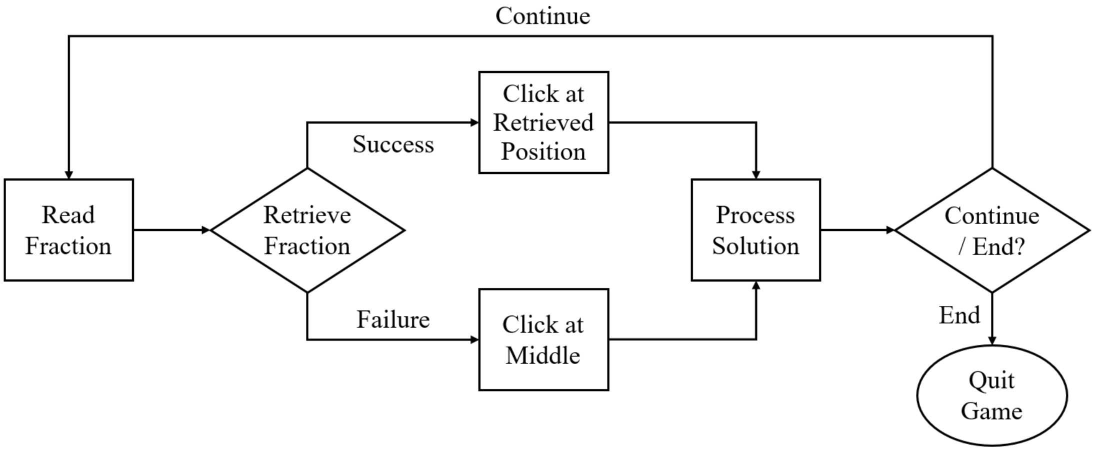

Figure 3: Flowchart of how the ACT-R model plays Battleship Numberline.

that starts another game trial, while the end production leads to a key press that quits the game. The continue production has higher initial utility than the end production, so the model will choose to continue in the first few trials. However, as the model plays more trials, it will encounter more repeating fractions and hence receive more negative rewards. This would eventually cause the utility of the continue production to fall below that of the end production, at which point the model will choose to end the game.

A final note on the model is that production compilation is enabled so that the model will make faster responses as it encounters repeating fractions.

4 Results

4.1 Target size

When varying target sizes, the model's performance matches human data closely, as shown in Figure 4 and Table 1. The correlation is 0.986 and the mean deviation is 0.021. For lower ship sizes, the model's performance tends to be slightly better than human performance. This might be because the model keeps the exact correct position in declarative memory and only makes mistakes by cursor noise if it successfully retrieves the fraction. For larger ship sizes, the model performs slightly worse than human, likely due to the naive representation of fractions which would be addressed further in Section 5.2.

Table 1: Average success rates of human and model with varying target sizes.

Size	2%	3%	4%	5%	8%	10%	12%	15%	20%
Human	0.125	0.145	0.170	0.200	0.250	0.275	0.298	0.338	0.383
Model	0.158	0.161	0.169	0.185	0.225	0.251	0.279	0.320	0.356

In terms of engagement, the model's behaviors are also reasonably close to human data (see Figure 4 and Table 2). The correlation is 0.978 and the mean deviation is 0.092. The relationship between target size and engagement seems mostly linear for the model, while for

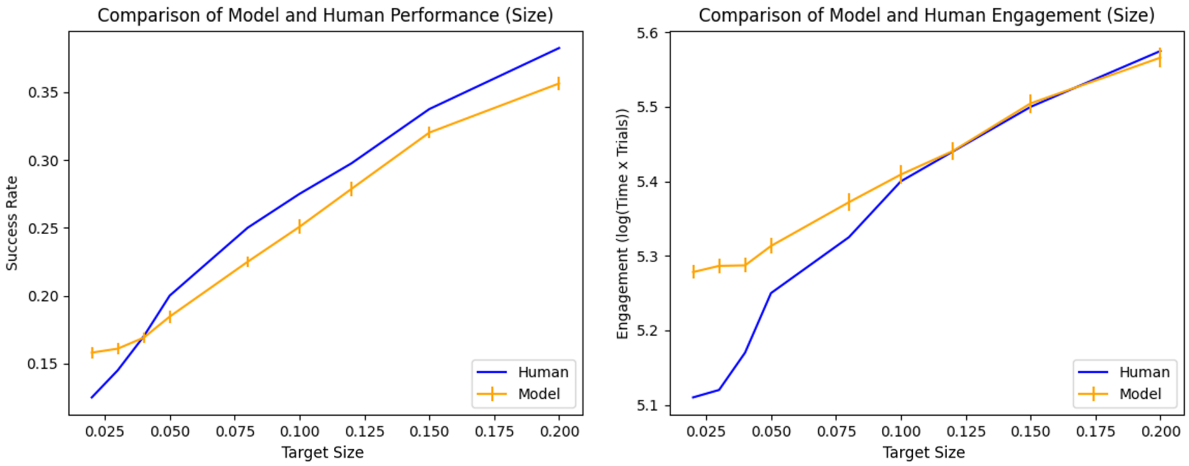

Figure 4: Comparison of the performance (left) and engagement (right) of human and model on varying target sizes. Error bars are standard errors.

human data the slope is sharper for smaller targets. Consequently, the model's data deviate considerably from human data for smaller target sizes. This again might be due to the high accuracy in the model's representation of positions.

Table 2: Average engagement scores of human and model with varying target sizes.

Size	2%	3%	4%	5%	8%	10%	12%	15%	20%
Human	5.110	5.120	5.170	5.250	5.325	5.400	5.440	5.500	5.575
Model	5.278	5.287	5.287	5.313	5.372	5.409	5.441	5.505	5.566

4.2 Time limit

Unfortunately, the model's performance does not match human performance when varying time limits. The correlation is 0.604 and the mean deviation is 0.071, which seem reasonable, but as Figure 5 and Table 3 demonstrate, the actual data patterns are very different. The model's success rate is very low when the time limit is 2, jumps to around 0.28 when the time limit gets to 3, and remains roughly constant for all the longer time limits. This means the model performance is similar as long as it has enough time to complete the trials, and would fail almost all the time otherwise. On the other hand, for human subjects, although there is also a jump in success rate between time limits of 2s and 3s, the gap is much smaller, and success rate keeps increasing as time limit further increases.

Table 3: Average success rates of human and model with varying time limits.

Time	2s	3s	4s	5s	8s	10s	15s	30s
Human	0.158	0.185	0.218	0.230	0.253	0.265	0.280	0.320
Model	0.000	0.281	0.262	0.266	0.263	0.268	0.261	0.269

The model's engagement shows a similar problem, where engagement is low when time limit is 2, and jumps to a higher constant value for longer time limits (see Figure 4.2 and Table

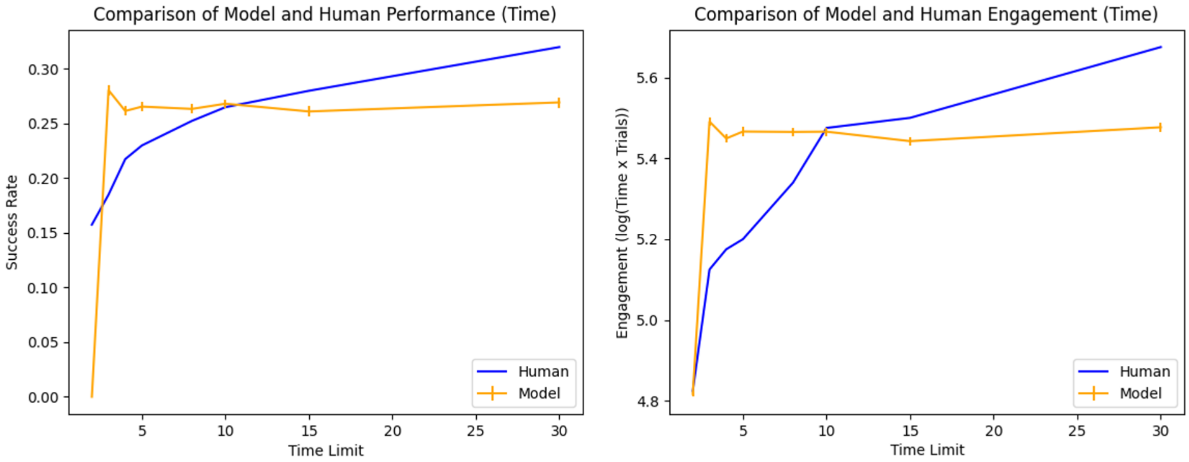

Figure 5: Comparison of the performance (left) and engagement (right) of human and model on varying time limits. Error bars are standard errors.

4). The correlation is 0.699 and the mean deviation is 0.206. Notably, though, the model's engagement is very close to human data at time limit of 2, meaning that the number of trials is roughly correct since the model play time is always capped at 2s and human play time should not be too much shorter due to the tight limit.

Table 4: Average engagement scores of human and model with varying time limits.

Time	2s	3s	4s	5s	8s	10s	15s	30s
Human	4.825	5.125	5.175	5.200	5.340	5.475	5.500	5.675
Model	4.817	5.491	5.449	5.466	5.465	5.466	5.442	5.477

5 Discussion

5.1 Summary

This project implements an ACT-R model that simulates playing the educational game Battleship Numberline (Lomas et al., 2011) and compares the model with human subject data collected from a massive study (Lomas et al., 2013). The model is affected by varying target sizes similarly as human subjects, such that both performance and engagement increases uniformly as the target size increases. However, the effect of varying time limits on performance and engagement is not simulated well. For the model, there exists a boundary where it either never finishes the trial or is always able to finish the trial, while the transition is mostly smooth for human subjects.

5.2 Limitations

One limitation of the current project is that the representation of fraction numbers by the model is very naive. The model treats the numerator and denominator as independent numbers and matches them separately during retrieval. This causes issues when judging the relative sizes of the fractions. For example, with the current representation, the model treats $\frac{3}{8}$ as much closer to $\frac{5}{8}$ than to $\frac{1}{2}$, while in fact the latter is closer. Although this project aims to model the behaviors of low-ability players, such mistakes might still be too naive, since a lot of them involve misjudgment of relative sizes compared to a half, which at least some of the low-ability players should be able to do. These naive mistakes may in part explain the slightly worse performance of the model than human subjects.

The model also has only a single strategy to solve the problems, and therefore does not learn to respond to different time limits. For particularly short time limits, the model still always attempts to retrieve the fraction and never manages to make any mouse click. In this situation, even clicking completely randomly might actually be a more effective strategy. An intuitive way to implement the switch of strategies is to have a guess production parallel to the retrieval production and let the model resolve between them based on their utilities. This should fix the near-zero success rate when the time limit is 2, which could also slightly close the performance gap of various target sizes.

Finally, this project only implements a small subset of the manipulations used by Lomas et al. (2013). To fully replicate their findings, an additional task (the ship mode) and an additional model (high-ability players) need to be implemented, and the current model also needs to be adjusted to be able to play the ship mode. An interesting question that follows is whether and how much transfer there exists between the submarine mode and the ship mode, since the knowledge of fractions presumably affects the performance in both modes.

References

Locke, E. A. and Schattke, K. (2019). Intrinsic and Extrinsic Motivation: Time for Expansion and Clarification. *Motivation Science, 5*(4), 277–290.

Lomas, D., Ching, D., Stampfer, E., Sandoval, M., and Koedinger, K. (2011). Battleship Numberline: A Digital Game for Improving Estimation Accuracy on Fraction Number Lines. in Proceedings of the Society for Research on Educational Effectiveness.

Lomas, D., Patel, K., Forlizzi, J. L., and Koedinger, K. R. (2013). Optimizing challenge in an educational game using large-scale design experiments. in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 89–98.

Nagashima, K., Morita, J., and Takeuchi, Y. (2020). Modeling Intrinsic Motivation in ACT-R: Focusing on the Relation Between Pattern Matching and Intellectual Curiosity. in Proceedings of the 18th Annual Meeting of the International Conference on Cognitive Modelling.