Note on accessibility

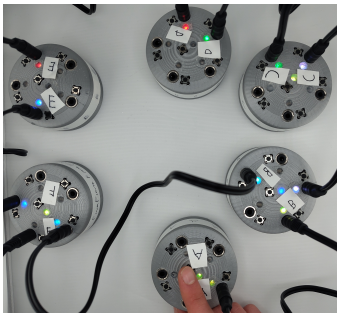
This document was produced with an early screen-reader-compatible version of LaTeX. This means that not all document features can be read correctly by a screen reader. Everything in the main body should work correctly in the latest version of Firefox or ngpdf[1], but most of the appendices are not accessible. If you require an alternative format please contact the author, who will be delighted to assist.

[1] https://ngpdf.com/loadFile

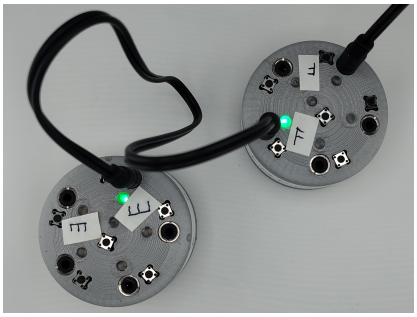
The impact of the IKEA effect on learning abstract skills

Penn Mackintosh †

(a) A single node manufactured for use as part of the study.



(b) A graph traversal being performed on a self-assembled graph, as participants did.



(c) Connections between nodes can be formed with either wires or buttons.

Figure 1: The tangible learning aid for graph theory. It was built and tested with participants to evaluate the impact of construction, rather than tangibility, on learning and attitude with abstract skills.

Abstract

Current teaching practice suffers from inefficiencies which prevent the uptake of innovations that would improve the quality of education. Technology-Based Embodied Learning (TBEL) has demonstrated significant potential for improving the learning of abstract tasks. However, TBEL is difficult to implement, so a comprehensive theory could allow transferring the benefits to easier contexts. Several theories explain improvements in learner comprehension under TBEL, but observations of improved attitudes to learning remain unexplored, despite their implications for teaching practice. The IKEA effect provides a potential reason for these changes in attitude: it states that when consumers assemble goods themselves, it leads higher valuation of the item. This dissertation hypothesizes that the IKEA effect can explain some of the improvements in attitude when introducing technology-based tangible learning aids, and investigates this using a learning aid for graph theory. An exploratory study was conducted with University students (n=9), and motivates further investigation of this topic. Overall, this dissertation proposes a novel theoretical basis for empirical improvements in learning attitudes. If demonstrated robustly, this theory could be applied to improve engagement in a classroom context.

Contents

Abstract		2
1	Introduction	2
2	Related Work	3
2.1	Tangibility and Embodiment in Education	3
2.2	Interactivity in Education and HCI	4
2.3	The IKEA effect	5
3	Method	5
3.1	Study design	5
3.2	Lesson design	7
3.3	Implementation	7
4	Results	10
4.1	Manipulation check	10
4.2	Attainment	10
4.3	Affect and Attitude	10
5	Discussion	11
5.1	Discussion of RQ1	11
5.2	Discussion of RQ2	11
5.3	Limitations	12
5.4	Future Work	12
6	Conclusion	13
A	Hardware	15
A.1	Bill of Materials	15
A.2	Schematics and Designs	15
B	Study	15
B.1	Latin square	15
B.2	Documentation	15
B.3	Data	48
C	Software	48
C.1	Host program	48
C.2	Peripheral program	54

*bath@pennmack.co.uk

1 Introduction

Tangible systems are increasingly popular in science education, most trivially for spatial learning [74], which focuses primarily on simple memorization of physical objects. Spatial learning has notably been used with chemical structures and medical anatomy [22, 42, 64] to improve retention. Recently, the integration of technology into these systems has allowed the tangible learning aids to expand into abstract skills such as language-learning and mathematics [84]. This wider field is now known as Technology-Based Embodied Learning (TBEL). It focuses on learning aids or techniques that use

technology to engage learners through inherent characteristics of the human body, and often asks learners to interact with full-body motion or feedback. TBEL has been shown to improve student attainment, and also appears to improve learner perspectives and attitude in relation to the learning task [84]. Various theories explain the improved understanding and retention observed with tangible learning aids [74]. For example, the mechanism behind TBEL is believed to be related to innate skills acquired through evolution, through a mechanism known as Embodied Cognition [71]. However, some studies have observed improved attitude to learning when using TBEL [83], which are not explained by Embodied Cognition or any related theory.

While various theories can explain the improved attainment in TBEL, none have explained the improved motivation. Zhang et al. [83] highlight the notable lack of research into the impact of TBEL on emotion, despite the evidence that it can improve learner attitudes. The lack of any theoretical development regarding this matter is especially marked given the potential benefits to understanding the theory behind these changes: deploying it in a classroom environment could improve learner motivation, concentration and self-confidence and would contribute to teaching practice. Since the primary difficulty preventing deployment of TBEL is currently cost, a theoretical approach could be used to develop a low-tech solution that achieves the same improvements as TBEL. A theory of engagement with learning aids could also be used to improve remote and asynchronous learning with potential applications to tangible and virtual learning aids and virtual reality [35].

The IKEA effect increases people's valuation of objects and influences self-confidence when performing simple assembly tasks [60]. This dissertation proposes that the IKEA effect extends to abstract tasks. It would then be one of the contributing factors to the positive impact of TBEL on attitude. The IKEA effect has been shown to increase valuation of self-assembled products across a wide range of domains [60], but has not been tested in the context of tangible learning aids. This dissertation also explores whether the IKEA effect can improve test scores in far transfer: Zhang et al.'s [83] results could suggest that changes in attitude are partly responsible for improved attainment in TBEL.

Specifically, this dissertation explores the idea that when learners encounter the IKEA effect in relation to a learning aid, they tend to assign higher value to the entire learning experience. This would then cause higher motivation to learn and stronger outcomes in test scores. Since the IKEA effect has already been shown to apply to a surprisingly wide range of contexts [60], this is a plausible extension of the existing theory. This dissertation proposes *a priori* hypotheses with the expectation that these will not be proven or disproven, but rather in the hope of further developing them from quantitative data.

RQ1: *How might the IKEA effect improve attainment while learning an abstract skill using TBEL?*

H1: *The presence of the IKEA effect in a tactile learning aid will improve learners' far transfer.*

RQ2: *How might the IKEA effect improve attitude to learning while learning an abstract skill using TBEL?*

H2: *The presence of the IKEA effect in a tactile learning aid will improve learners' confidence.*

H3: *The presence of the IKEA effect in a tactile learning aid will improve learners' motivation.*

This dissertation explores these possibilities using a tangible, interactive learning aid. The researcher chose to use traversals in graph theory, a branch of computer science, as the subject of study. These provide the opportunity to perform construction when creating connections between nodes in the graph, while keeping construction a step removed from the abstract task of finding paths through the graph. Tangible models have only been applied to graph theory to improve accessibility [10, 11, 44]. Nevertheless, the prior exploration of touch in combination with graph theory confirms that this is a topic that is well suited to tangible solutions.

Participants in an exploratory study assembled interactive models of graphs to help them understand different traversals. The extent of their participation in assembling the learning aid was manipulated by connecting nodes using either buttons (non-construction) or cables (construction). Participants completed quizzes and questionnaires to measure attainment, valuation, motivation and self-confidence across both conditions in a within-participants design.

The study could not determine whether the IKEA effect was successfully elicited due to lack of funding, so no conclusions can be drawn about causality. None of the main results were statistically significant due to the small sample size, but participants reported higher mean valuation and had more accurate estimates for their quiz scores in the construction condition. This suggests that having learners assemble learning aids may improve accuracy of self-evaluations even for abstract tasks, and provides tentative theoretical underpinnings for improving motivation, self-confidence and attainment in classrooms. Motivation was unchanged by the experiment despite changes in both attainment and estimated score. While the hypothesis was for higher confidence with the construction condition, the overconfidence of learners was actually reduced, which suggests that a novel effect may be at play in conflict with existing literature. This motivates further research into the IKEA effect in education.

Finally, I propose future directions for research into this area. These include repeating the study with more participants, testing across different contexts, and exploring other aspects of the IKEA effect in the context of abstract tasks. This dissertation contributes a novel link between the IKEA effect and education, and proposes a mechanism for the poorly understood effects of TBEL. In summary, this dissertation contributes:

- A novel theoretical basis for changes to learner attitude under TBEL, based on the IKEA effect
- An effective experimental design and tooling for investigating the IKEA effect in education
- Exploratory data to suggest that manipulation designed to induce the IKEA effect has an impact on learner attitude
- Recommendations for future work to validate and further explore the impact of the IKEA effect on learner attitudes and attainment

2 Related Work

2.1 Tangibility and Embodiment in Education

Technology-Based Embodied Learning (TBEL) integrates technology with bodily experiences for learning. The novel element of

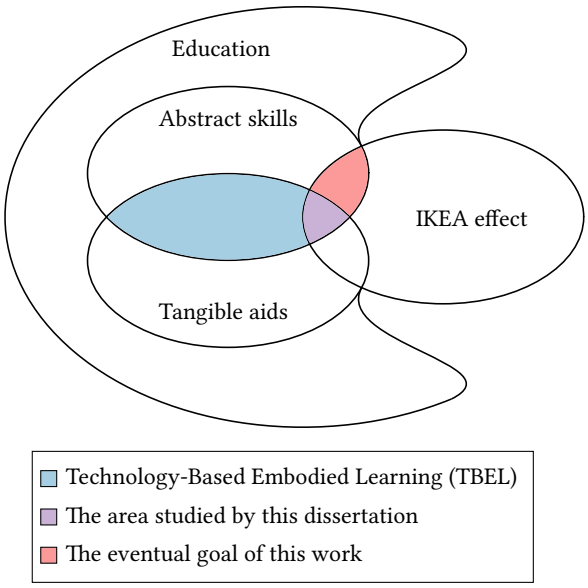


Figure 2: This dissertation proposes that TBEL improves attitude to learning due to the IKEA effect, and investigates the intersection of the IKEA effect with TBEL. This builds on prior work [83], but proposes a novel theory for the observed improvements. If the IKEA effect is responsible for the improved attitude, it is likely that this improvement can be reproduced without tangible aids. This would feed into the eventual goal of making high-quality learning more widely available.

TBEL is the high interactivity of the learning environment, which allows learners to receive real-time feedback or input based on their work. Many TBEL environments focus on construction of tangible objects by learners. For example, Johnson-Glenberg et al.'s [35] Chemistry study allowed learners to construct a solution by mixing the individual molecules in VR.

TBEL improves attitude and attainment, but only the latter is explained by psychological theory, despite the large potential impact. Both Cognitive Load Theory [77] and Embodied Cognition [58, 74, 84] suggest that building or interacting with a model should improve learning, but no theory has been proposed to deal with improvements in attitude to learning [83]. Nevertheless, a coherent theory could provide ways to generalize the positive impact across different modalities. Cognitive Load Theory emphasizes the principles of limited working memory and 'automation' of well-practised procedures. This automation reduces learners' extraneous working memory usage when increased effort is expended on creating 'schemas', essentially mental models of chunks of content. Since each schema only takes up one 'slot' in working memory, this frees up space for problem-solving. The creation of schemas is known as germane cognitive load. Subsequent advances include the idea that humans are evolved to specialize in certain areas and that these inherent skills can be exploited to reduce extraneous cognitive load [58]. This can apply in many cases, but of particular interest is that memorization is improved by incorporating bodily experiences, a phenomenon explained by a theory known as Embodied Cognition [see 9, 71, 74].

The IKEA effect can adequately explain improvements in attitude during both non-interactive embodied learning and TBEL. The existing evidence about the IKEA effect can be applied to embodied learning, but there is no evidence to support it applying to TBEL due to the prevalence of abstract concepts. Embodied learning was difficult to apply to more abstract tasks, but with the advent of TBEL, this has become feasible [41]. When using non-interactive embodied learning, the germane cognitive load generally stems from the assembly of new configurations, a proven case of the IKEA effect. For TBEL, the germane load is usually from interaction after construction. Since any IKEA effect applying to TBEL would apply to the constructed model, but not to its use, further investigation is required to ascertain whether this theory applies.

Understanding the interaction between attitude and attainment could have significant consequences for research and practice. Guo and Goh [23] demonstrated improvements in affect and learning outcomes when using a digital teacher with human appearance. These could be explained (among other possibilities) by affect being a mediator. Introducing another method of influencing affect using the IKEA effect would open up an opportunity to investigate whether the humanoid image, or the resulting affect, is responsible for the improved learning outcomes. For example, Fiorella and Mayer [20] tested the impact of seeing the instructor writing during learning, and conclude that the presence of the instructor, or even just their hand, improves learning. This is supported by Embodied Cognition [see 9, 71, 74], but they did not test whether attitude could be a mediator. By finding alternative ways of influencing attitude during learning, more rigorous conclusions become possible regarding the interaction between embodiment and affect. It is even possible that embodied pedagogical agents have a negative direct impact on learning but that their positive influence on attitude outweighs this, leading to the weakly positive effect on retention and transfer observed by Guo and Goh [23]. This would explain Darejeh et al.'s [16] conflicting results of increased extraneous cognitive load when using an embodied pedagogical agent.

Minogue and Jones [48] identified enhanced motivation when using haptic interaction, but a lack of evidence regarding attainment. However, they highlight that existing research could not identify whether the improved motivation was due to the involvement of conscious exploration or an innate feature of tactile experiences. Building on this, many studies have investigated the effect of embodiment and (often incidentally) haptics on attainment [39, 83], but studies regarding motivation have been more scarce. Unfortunately, while IKEA effect could apply in some of this research, the tangible and interactive elements have been seen as the key result with no link to the IKEA effect identified [e.g., 33, 73].

2.2 Interactivity in Education and HCI

Interaction lies at the intersection of the IKEA effect, Human-Computer Interaction (HCI) and education, and must be properly defined and understood. Different types and levels of interactivity can have a considerable effect on learner attitudes and attainment [36]. Research has found that participants are unaware of differences in interactivity but that they can still have a considerable

impact on attitude [76], meaning that interactivity must be well understood and controlled in the research environment.

It remains a complex word with various meanings, despite the entire field of HCI revolving around it [26, 28]. To use an example from Janlert and Stolterman [28], a piece of paper can be considered interactive in that it responds to a user's writing by displaying their text. They remove this obviously contrived example from scope by requiring that the response of the system express some internal 'will' that is capable of differing from the user's desires or expectations in a less-than-trivially predictable way—in other words, that the system appears to have agency over its responses.

A different definition is used in education, where the term can also refer to interaction between humans, as in the traditional meaning. Interactivity is considered a key tool in education, as it gives students the opportunity to explore concepts more deeply, in a relatable context, which is essential [78]. Furthermore, interactivity is a major challenge in asynchronous [72] online courses [76]. While alternative classifications exist [see 25, 36, 82], Moore [53] identifies three major types of interaction in education: learner-content, learner-instructor, and learner-learner. All of these are important [2], but the former is the least well understood [82]. The exact definition of learner-content interaction encompasses not only bidirectional 'active' engagement with the content, but also 'passive' interaction such as reading text or watching videos.

A strict categorization without the context of usage is not viable [46]. Watching a video can be likened to reading a book: just as text is only available to the learner while the book is open, audio and video are only available to the learner while a video is playing. This would suggest that watching video is not a form of active interaction. And yet a laptop playing video is certainly capable of opening a pop-up window that the user did not expect, clearly meeting an intuitive HCI-style definition of interactivity.

This dissertation uses a definition of interaction that satisfies both the HCI and educational definitions: an instance of active learner-content interaction takes place when the learner performs some action and uses the tailored response to inform learning. This definition takes context into account and focusing on 'active' learner-content interaction in an HCI setting. Hornbæk and Oulasvirta's [26] assertion that interaction requires entities to '*determine* each another's behaviour over time' is compatible with this definition.

2.3 The IKEA effect

The IKEA effect is a psychological phenomenon that produces a sense of attachment when people build or customize things themselves. In particular, it refers to 'people's tendency to overestimate the value of items they personally assembled or assisted in creating' [60]. There are at least two mechanisms for the IKEA effect: self-confidence and a sense of ownership [60].

The former contributes directly to confidence [50]. It is thought that the IKEA effect may stem from a desire to be perceived as competent (either by oneself or by others), which ties in closely with confidence. Retention and transfer of learning are both impacted by affect [23]. This indicates that the IKEA effect may provide a way to bolster learners' self-confidence and therefore motivate them to continue learning.

The contribution of the sense of ownership [43] to attitude to learning is less obvious. It was hypothesized that since ownership increases the value afforded to the learning aid by the learner, it might also result in improved valuation of the learning process.

Pelled et al. [60] call for more research into the boundary conditions of the IKEA effect. This study raises further questions about how far removed from the subject the IKEA effect can have an impact, providing a clear research direction for this work.

By investigating whether the IKEA effect influences student confidence and perceived test scores, we will learn more about results from the existing literature. There is evidence to suggest that customizing tangible models by painting them may influence student perception of quiz difficulty [42]. It remains unclear whether this is caused by the customization, which may invoke the IKEA effect [60], or an innate feature of the 3D model. If the IKEA effect does influence self-confidence, this could bring into question Manrique et al.'s [42] and Preece et al.'s [62] claims that 3D models are categorically better, as the IKEA effect may better explain their findings.

There has been little research into exploiting the IKEA effect to boost learning. Zhong et al. [84] identify five papers that detected improved attitude towards learning materials when using embodied learning. There is a clear gap in research here: none of these papers employ any student-led construction of tangible objects, despite it being known that the IKEA effect elicits feelings of attachment, which can influence motivation and confidence in learners. It is still very much possible that they inadvertently included an element that was perceived as construction by learners. The researcher was only able to identify one study that explicitly attempts to apply the IKEA effect to education [40]. Unfortunately, it lacks peer review and does not clearly demonstrate use of the IKEA effect. It also fails to test possible mediators and fails to account for several potential confounding variables: the effect of an extra lesson; novelty; a new instructor; and the inclusion of creativity in mathematics. If a more rigorous result is found, that is likely to support in-classroom investigations with higher ecological validity and larger sample sizes.

3 Method

3.1 Study design

3.1.1 Initial design. To investigate RQ1 and RQ2, the researcher designed an exploratory study. Initially, it was planned to use a between-participants design with a pre- and post-test to avoid order effects. Each participant would take part in either a construction or non-construction condition, with random assignment. The researcher planned to analyse results using analysis of covariance as this has the best power and lowest bias [30]. Ultimately, this design was not used (see section 3.1.2).

The two conditions were designed to manipulate the level of construction perceived by the participants without influencing interactivity, agency or tangibility. Each participant was to undertake one of two conditions:

- Construction: using wires to form connections between nodes. This condition was designed to induce the IKEA effect.

- Non-construction: using button presses to form connection between nodes. This condition was designed to avoid the IKEA effect.

This method of varying the IKEA effect is unusual: most studies compare a researcher- or peer-assembled product with a self-assembled product, but this was not appropriate. The researcher would have to configure an identical learning aid to give to the participant while also providing them with advice during the learning steps. The alternative of giving the participants a pre-configured learning aid but skipping the participants' assembly step for the non-construction condition would alter the germane cognitive load [77] and invalidate any comparison between conditions.

Care was taken to avoid altering interactivity when varying the level of construction because interactivity can affect engagement [76]. The definition of interactivity given in section 2.2 includes some types of construction where the IKEA effect might apply [for example 38]. The construction phase of the user's learning was designed to be minimally interactive and no unpredictable feedback was given at this point. It was especially important to control interactivity because post-hoc correction based on questionnaires is not effective for perceived interactivity [76].

The level of agency is tightly linked to the level of interactivity, so had to be controlled. Furthermore, the act of manipulating the content of simulations has been shown to have a significant impact on comprehension [61]. Accidental manipulation of agency was avoided by giving users exactly the same interface controls between the two conditions, and only altering the input modality between buttons and wires.

Tangible user interfaces can lead to increased motivation, which could act as a confounding variable. This effect has been replicated for learning aids [69]. This presents a possible confounding variable which was controlled for by using a tangible interface for both conditions.

3.1.2 Final design. Due to the lack of funding available for the study, issues were anticipated with recruiting enough people for a between-participants design. This was exacerbated by the fact that undergraduate students of computer science were not eligible to take part, due to already being familiar with graph theory. Given this risk, the design was adapted to a within-participants methodology. This allowed better power and more in-depth analysis, at the cost of reduced validity. Participants (n=9) undertook two conditions, each consisting of a pre-test, learning session and post-test. Sessions were designed to last one hour, but some participants chose to continue working far beyond the allocated time.

The within-participants design was piloted with an undergraduate student of computer science. They provided feedback on how long the study would take and identified a technical issue, which was resolved prior to the exploratory study. No other changes were required.

3.1.3 Measures. Measurements were conducted before and after each learning session to determine the impact of the experience on the participants' attitudes and attainment. Having one quiz before and one quiz after each session allowed comparing the change in responses resulting from the different conditions. The following measures were used:

- Quiz score
- Self-estimated quiz score
- Perceived Competence subscale of the Intrinsic Motivation Inventory
- Effort/Importance subscale of the Intrinsic Motivation Inventory
- Willingness To Pay through bidding
- Time spent practising

To measure attainment, participants undertook a quiz both before and after each learning session. The quiz was paper-based despite the learning being tangible, thus measuring 'far' transfer. Two distinct quizzes were delivered for each condition, in a counterbalanced order. The difficulty of these quizzes was validated as equal by an independent expert (a University lecturer specializing in education and mathematical foundations of computer science) to minimize order effects. Evaluating the ability of learners to perform far transfer is an better test of comprehension and results can differ compared to near transfer and retention [21].

After each of the four quizzes, participants answered a questionnaire about their performance. They provided their estimated overall score in the quiz and answered the 'Perceived Competence' and 'Effort/Importance' subscales of the Intrinsic Motivation Inventory [14, 67], a validated and widely used questionnaire. The estimated score and Perceived Competence contribute to understanding self-confidence while the Effort/Importance scale describes other aspects of motivation.

Additionally, immediately after completing the learning phase and before the second quiz, participants were asked to bid to purchase the learning aid to determine their Willingness To Pay [80], similarly to Norton et al. [57]. Due to lack of funding, the bidding was hypothetical and no actual money or goods changed hands. Participants were aware of the hypothetical nature. In an attempt to mitigate the lack of actual money or goods, participants were told they could take a photo if their bid was higher than the random valuation by the researcher [see 50]. It was also emphasized to participants that they should pick a value that they would be willing to pay personally in the context, but this was not incentive-aligned.

All questionnaires were delivered on paper. The Intrinsic Motivation Inventory questions were modified to allow a better distribution of responses by increasing them above a 7-point scale, as recommended by Bandura, quoted in [18]. Participants were allowed to reference their responses to quiz questions while completing the subsequent questionnaire, to obtain a more accurate value for how well they believed they performed on the quiz.

The Single Ease Question [68], System Usability Scale [12], Positive And Negative Affect Schedule [79], and cognitive load scales [59] were considered for inclusion in the questionnaires, but these were considered to be less important than the Intrinsic Motivation Inventory and Willingness To Pay measures. Adding more scales to the questionnaire would have prohibitively increased the study duration, as it already took over an hour for most participants. Use of the QSM/FAM scale [65] instead of the Intrinsic Motivation Inventory was also investigated, but the author's institution did not have access to this scale. The QSM/FAM would have been preferable because it was used by Skulmowski et al. [74] in a closely related work to this dissertation.

3.1.4 Procedure. Participants were greeted by the researcher and informed consent was obtained. The age and gender of participants were recorded[1]. Participants answered a pre-test for one of two algorithms. They then undertook the learning process in either the construction or non-construction condition, followed by the bidding and post-test. These steps were then repeated for the other condition, simultaneously swapping the algorithm. The order of the conditions and the algorithms were separately counterbalanced using a Latin square, shown in section B.1. Finally, participants were thanked for their participation and debriefed.

Participants were discouraged from rushing. They were timed during each learning phase. They were not timed during the quizzes to ensure they took as much time as they required.

3.1.5 Participants. Participants were recruited at the University of Bath. People who already had experience with graph theory were not eligible to participate, and no incentives were available. Recruitment of large numbers of people was not possible because the pool of potential participants was primarily people who wanted to learn about basic graph theory but had not taken the initiative to do so. It is likely that more participants could have been recruited with incentives or compensation.

Active recruitment was undertaken of undergraduates in the departments of Life Sciences, Psychology and Chemistry only for practical reasons. A bulk email (section B.2.2) was sent by a non-academic member of staff in each of these departments.

A priori power analysis for a one-sided paired-samples T-test (n=10, $|\delta| = 0.5$, $\alpha = .1$) suggested that the results would not be significant ($\beta = .589$). Since the tests to be used included analysis of variance tests to account for interaction effects, even lower power was expected. The decision was made to continue anyway since the study is explicitly exploratory and future research could still be informed by non-rigorous data. The power analysis was carried out in JASP 0.95.4 [29].

9 participants were recruited (5 female, 3 male) with a mean age of 20.5[1].

3.2 Lesson design

Participants learned graph-theoretical algorithms using a tangible model of a graph. In the construction condition, participants configured the graph themselves by connecting nodes with wires. In the non-construction condition they pressed buttons to create the connections, allowing participants to connect nodes without the sense of construction or customization.

Participants learned two graph theory algorithms: breadth-first and depth-first search. These were chosen as non-trivial but still easy tasks on the basis that task difficulty must be well-balanced. Frustration is usually seen as something to be avoided in HCI [66, p. 91], and this is the same in education. While frustration can be a side effect of positive learning experiences [7], suppressing it using affective computing [63] results in improved learning outcomes [5, 23]. Nevertheless, frustration may play a part in the IKEA effect, as it remains unclear how the possibility of failure influences the IKEA effect [60]. Although some more recent studies reduce the scope for participant failure, the role (or lack thereof) of potential failure is not proven. Since this study does not aim to test whether human error is a factor that contributes to the IKEA effect, it was important that participants were able to fail. This said, it was important not to make the task too difficult: 'labor leads to love only when that labor is successful' [57].

Prior work used Dijkstra's Algorithm and Kruskal's Algorithm to evaluate subjects' graph theoretical abilities [44], but these are unsuitable because of their complexity and the requirement for edges to have weights. All participants in Másilko and Pecl's [44] study already had a basic understanding of graph theory. Both algorithms from Másilko and Pecl require edges to have weights, which is difficult to represent clearly in a simple tangible interface. They were likely looking for diversity of algorithms, whereas for the present study, such diversity would simply introduce more scope for confounding variables. In this study, to improve validity, it is more appropriate to use two algorithms with the same goal, rather than comparing a traversal with a spanning-tree algorithm.

Participants' learning was self-directed using guided-inquiry learning as recommended by Plass et al. [61]. The full guidance, shown in section B.2.1, was presented on paper. The researcher did not advise participants on content during quizzes, but did answer participants' questions while they were learning.

Care was taken to design an effective learning aid. Relevant design features from Plass et al. [61] were implemented in the learning aid and lesson design. Colour coding, which can reduce search demands, was used to help participants check the connections between nodes and follow wires. Cueing, where learner attention is overtly directed towards relevant elements, was not needed since the changing colour of the node search status LED is salient. Information was naturally presented in both symbolic and iconic forms, which can enhance learning, through the worksheets and learning aid respectively. All three types of interactivity for educational simulations are implemented in the learning aid: participants can control progression with a button (control of pacing) [54], they can change the structure of the graph (manipulation of content), and they can change the representation of the graph by moving the nodes around (control of representation). The worksheets were designed to encourage hypothesizing by learners, but a 'hypothesis scratchpad' was not implemented because the explicit questioning of learners already encouraged hypothesis generation.

3.3 Implementation

3.3.1 Hardware. The learning aid was designed and implemented in two main parts. The casing was created using 3D printing while the circuits were produced by soldering Arduino Nanos to perfboards.

Data transfer and power supply needs were met using 3.5mm Tip Ring Sleeve (TRS) cables. Each node had five sockets for TRS jacks, which are ubiquitous in audio systems. Each socket had a button, for use in the non-construction condition, and an RGB NeoPixel LED for visual confirmation of the edges between nodes.

Each node also had a button and NeoPixel LED in the centre. These provided the input and output for the simulation of traversals respectively. The button also provided the user with control over the progression of the simulation.

[1] A procedural error meant that the age and gender of one participant were not recorded. This participant was excluded from demographics reporting.

The use of spatially-unaware nodes meant that participants retained control of representation [61]. Conversely, systems like Sifteo [47] and SoftMod [38] use near-field communication and magnetically bonded contacts to detect neighbouring nodes. These approaches would unnecessarily constrain the physical layout of nodes and reduce the learners' control.

TRS cables were chosen because the cables needed to be familiar to the users in order to prevent any effects arising from fixation on how the connectors function. TRS is a well-known and particularly simple connector, making it especially suitable.

Figure 3: The final design of the casing for each node, rendered with SolveSpace [75].

The casing was designed in SolveSpace [75] and aimed to mask the complex electronics entirely, allowing learners to focus on the device as a learning aid rather than a technological innovation. The design had a similar appearance to standard diagrams used in graph theory: circular with a one-letter label. This helped reduce participant fixation on any perceived novelty of the hardware and refocused them on the task at hand. An image of the final design is shown in fig. 3. The casing was sliced for printing in Bambu Studio [8].

The circuit was prototyped on a breadboard while perfboards were used for the nodes in the study. Breadboards were too large to fit in a conveniently sized case, and were too easily broken, whereas perfboards are more compact and resilient. An Arduino Nano was used to control each node due to the compact form factor and ease of prototyping [56].

The electronics and the casing were mostly separable, but the buttons were soldered through holes in the casing, and the LEDs were hot-glued to ensure a smooth appearance and avoid unwanted movement.

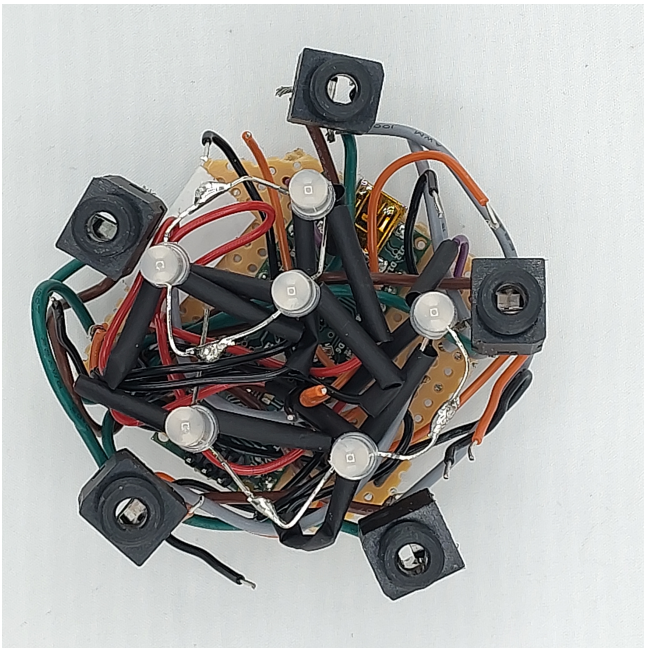


Figure 4: The soldered internals of a node. Buttons are not shown because these have to be soldered through the holes in the casing, which would obstruct the image.

To help participants keep track of the different nodes, each one was labelled with a letter.

At runtime, the nodes shared a list of neighbours over a TRS cable, which was combined to form a topology map on the controlling PC. Both the tangible graph and SoftMod [38] required an accurate real-time topology map, and both need disparate modules to communicate and cooperate, so many similar challenges appeared. Lambrichts et al. [38] describe using I2C to achieve better data transfer rates than the 1-Wire protocol. However, they are not largely constrained by number of cables or connections because they use strong magnets to connect modules. In the tangible graph, nodes had to be connected with simple cables, so this was an issue, and a custom protocol was developed in listing 4.

To ensure that TBEL was effective in the implementation, it was validated against Johnson-Glenberg et al.'s [35] taxonomy of embodiment. This ranges from the first degree, usually on a small screen with little interactivity, to the fourth degree, where the user uses their entire body to perform immersive tasks that are closely related to the content to be learned. These degrees are not strictly ordered. Rather, they can be decomposed into three components: amount of movement, similarity of movement and perceived similarity of environment. These components do not apply easily when teaching abstract skills: what would a good environment feel like? What should the movements be similar to? Nevertheless, the examples easily map this tangible implementation of a graph to second-degree embodiment:

> Second degree = Learner is generally seated, there is upper body movement; interfaces should be highly interactive, but gestural relevancy is not a given;

> with smaller display (monitor or tablet) the learner does not perceive the environment as highly immersive. [35]

This is also the joint-most common form of TBEL analysed by Zhong et al. [84]. The popularity of second-degree embodiments suggests that investigating them is appropriate to provide results that reflect the improvements observed for TBEL in general. Furthermore, a higher degree of embodiment is generally associated with higher-cost technology (such as VR being common in higher levels [84]), so a lower degree will generalize better to real classrooms. One alternative for a higher degree of embodiment would have been to implement a graph-theoretical learning aid in VR. However, this would not have helped to analyse the impact of the IKEA effect while it would have had less environmental validity to real classrooms [84].

3.3.2 Software. The main software components were a C++ program running on each Arduino and a controller Python script on the researcher's laptop. This split removed the issue of self-selecting a controller among the Arduinos and reduced the amount of computation required on each node. In turn, this allowed improved responsiveness and reliability.

The controller program maintained a list of the physical connections between nodes and received notifications for button presses. It coordinated the LEDs and traversals. Specifically, the program lit pairs of LEDs in the same colour when two nodes were connected, and it lit the central LEDs according to the traversal status. The tracking of edges in the graph differed depending on whether the study was in the construction or non-construction condition. In the former, edges were tracked by receiving the list of link neighbours from each node, while in the latter condition the controller program maintained and updated the list according to button presses. Colour for the edges were chosen randomly from a predefined set at edge creation, meaning learners could disconnect and reconnect a pair if they wanted a new colour. Communication with the hardware took place over the USB serial connection to a single Arduino, which relayed the messages over the TRS cable. An overview of the code is shown in listing 1. The Python program also rendered a visual display of the graph for debugging. Development used *PyCharm* [31], *NetworkX* [24] and *Matplotlib* [27].

Listing 1: Pseudocode for the controller's main functionality.

```
when started
  connect to Arduino
  pairing_state = 0
  connection_graph = {}
end when
when message received
  if message is button press
    if button is middle button
      start algorithm -- either DFS or BFS according to configuration
    else if connection mode is buttons
      if pairing_state = 0
        pairing_state = (message sender, button id)
        flash LED (message sender, button id)
      else
        if pairing_state and (message sender, button id) connected in
            connection_graph
          remove connection between pairing_state and (message
              sender, button id) from connection_graph
        else if (message sender, button id) in connection_graph
          -- do nothing, the pin is already in use
        else if message sender = pairing_state[0]
          -- do nothing, cannot connect a node to itself
        else if (message sender, *) connected to (pairing_state[0], *) in
            connection_graph
          -- do nothing, nodes are already connected
        else
          add connection between pairing_state and (message sender,
              button id) to connection_graph
        end if
        pairing_state = 0
      end if
    end if
  else if message is node connectivity and connection mode is wires
    for each (pin, connected node)
      if (message sender, a) connected to (connected node, b) in
          connection_graph
        replace edge with (message sender, pin, connected node, b)
        reset edge timer
      else
        add connection between (message sender, pin) and (connected
            node, unknown) to connection_graph
      end if
    end for
  end if
end when
when (node1, pin1) and (node2, pin2) connected in connection graph
  pick unique random colour from predefined list
  set colour of (node1, pin1) to colour
  set colour of (node2, pin2) to colour
  if connection mode is wires
    set a timer to remove the edge after 3 seconds
  end if
end when
```

The C++ code reported the state of individual nodes using a custom protocol. Each node communicated with its neighbours using a lightweight half-duplex version of the Sony SIRC protocol [34]. This was chosen due to the simplicity and reliability of SIRC. It was modified with the addition of a startup negotiation phase where the initial direction of the half-duplex connection is negotiated. An overview of the code is shown in listing 2. Development took place in the *Arduino IDE* [4] using *digitalWriteFast* [6], *Bit Twiddling Hacks* [3] and the *Adafruit NeoPixel Library* [13].

To ensure reliability, communications used a self-synchronising clock which was offset between each port on the node. This guaranteed that the device would respond within the required time-frame to avoid spurious disconnection of nodes. No special handling was implemented to avoid cycles but in practice this was tested and did not pose an issue. Messages are broadcast rather than routed, so routing cycles are not an issue as long as there is sufficient bandwidth, which practical testing revealed to be the case.

Listing 2: Pseudocode for the Arduino software's main functionality.

```
loop forever
  for each pin
    if data available from USB and SIRC connected on pin 0
      send data over SIRC on pin 0
    else if data available over SIRC on pin 0 and USB connected
      send data over USB
    else if button pressed
      send button press notification over SIRC on all pins
      ignore button presses for 500 milliseconds
    else if data received on SIRC
      if data is an LED change and data is addressed to this node
        update LEDs
      else
        add neighbour node to deny–list for packets from the message
            sender for 1 second
        forward packet to non–deny–listed nodes
      end if
    else if half–duplex mode is transmit
      either forward message or transmit neighbour state on round–
          robin basis
    end if
  end for
end loop
```

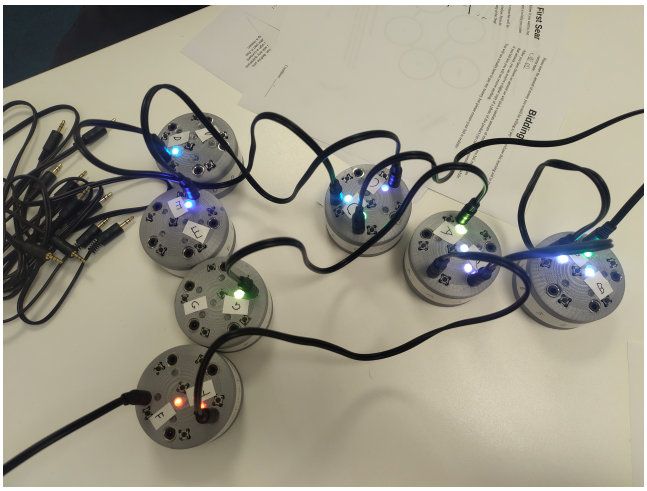


Figure 5: The tangible graph in use by a participant, in the construction condition. Note how the participant has constructed a binary tree and has chosen to use their agency in representation format to arrange the nodes approximately in the standard shape of a binary tree, with the root at the right.

4 Results

Differences between conditions were tested using 2x2 repeated measures analyses of variance (RMANOVA). This test is able to detect interaction effects, such as the interaction between time and condition. In this study design, both conditions should have been the same in the pre-test but a difference between groups was expected in post-test, so an interaction effect test was required. The small sample size meant that most results were not significant even at an exploratory 0.1 significance level, as discussed in section 3.1.5. Significant results are marked with a *. Results were analysed using JASP 0.95.4 [29].

4.1 Manipulation check

Only three participants reported different values for Willingness To Pay between the two conditions. Of these, one participant reported being willing to pay £90 for the Wires condition versus £50 for the Buttons condition. The other two participants reported £15 vs £20 and £20 vs £15. Difficulties were expected in the bidding process because of the lack of incentives and the use of a within-participants design. The mean increased in the construction condition (£67.50 vs £63.06, $SD_{construction}$ = £78.72, $SD_{non\text{-}construction}$ = £78.42). I assume that the manipulation was successful and the IKEA effect applies, despite the lack of evidence ($F(1, 88.89) = 0.966$, $p = .354$), due to the difficulties faced implementing an effective Willingness To Pay measure.

4.2 Attainment

Participants had significantly different attainment in the post-test than the pre-test ($F(1, 69.444) = 8.133$, $p = .021^*$), and this was an increase, as expected[2].

Participants had a slightly (but not significantly) higher improvement in quiz scores in the construction condition (Condition * Time, $F(1, 3.361) = 0.398$, $p = .546$), with a post-test mean of 6.056/9 (SD = 2.663) versus the non-construction group with 5.056/9 (SD = 3.395). The changes are shown in fig. 6.

This improvement in attainment is not caused by the amount of time spent learning: spending longer on the task was positively correlated with higher scores($r = 0.375$, $p = .062^*$), but participants spent an average of 6.11 minutes *less* on the construction condition ($F(1, 168.06) = 1.831$, $p = .213$)[2].

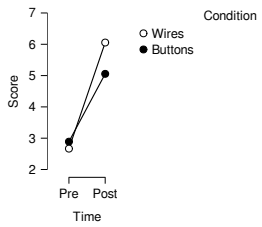


Figure 6: The increase in score was considerably larger for the construction condition. This could align with the hypothesized results of better attainment when participants assemble learning aids themselves.

4.3 Affect and Attitude

The accuracy of the estimated quiz score was considerably better in the construction post-test with an error of -0.072 (SD=2.732), as opposed to 1.267 (SD=4.825) in the non-construction condition,

[2]. Post-hoc testing was not required because there were only two groups for this variable so a t-test would have been equivalent to the RMANOVA conducted.

as shown in fig. 7. This was not significant (Condition * Time, $F(1, 1.381) = 0.073$, $p = .794$).

Pooled, both conditions saw a significant increase in estimated score from pre- to post-test ($F(1, 58.141) = 16.034$, $p = .004$*. The increase was slightly larger for the construction condition due to noise (see fig. 8).

The estimated score and perceived competence were strongly correlated ($r = 0.922$, $p < .001$*). Therefore, no tests were run for perceived competence. Estimated score was preferred over perceived competence because it can be compared with the actual score achieved, to evaluate overconfidence.

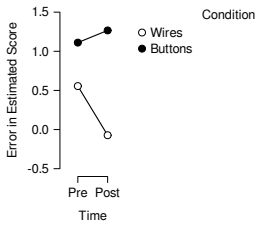


Figure 7: Error in estimated score decreased in the construction condition but increased in the wires condition. This could suggest that participants overestimate less when they build something, rather than simply being more confident, contrary to expectations.

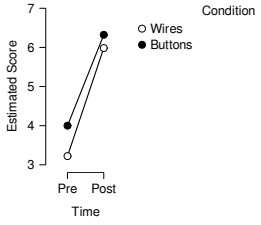


Figure 8: Estimated score was the same across both conditions, contrary to the expectation of higher estimates in the construction condition.

The Effort/Importance scale did not show a significant difference between conditions (Condition * Time, $F(1, 441.0) = 2.109$, $p = .184$). However, it was very similar between the conditions in post-test ($t(8) = 0.493$, $p = .635$) despite had a sizeable difference in the pre-test ($t(8) = 1.502$, $p = .171$). This is because some participants were less invested in the second pre-test, having realized during the first learning session that they had got a very low mark. Therefore, the difference in effort/importance in the pre-test can be ignored, suggesting that the different conditions had no effect. The increase from pre- to post-test was significant ($F(1, 7685.4) = 21.527$, $p = .002$*)[2].

5 Discussion

Despite the small sample size and lack of significant results, there is useful data to draw on. This section evaluates the *a priori* hypotheses and further develops the research questions to inform future work.

5.1 Findings around how the IKEA effect might improve attainment while learning an abstract skill using TBEL (RQ1)

As expected, difficulties were encountered with the manipulation check for the IKEA effect. The within-participants design meant that participants were more likely to try to rationalize their bids, and the lack of incentive alignment was also problematic. The difference in average bid is not even approaching significance, but the limited data suggests that IKEA effect may have been involved and requires further research with funding for a more effective measurement of value.

The results demonstrate that both types of teaching session significantly improved participants' understanding of graph theory. This indicates that the study design is a good foundation for future research.

The results do not demonstrate the hypothesized increase in score in the construction condition (**H1**), but a trend in that direction was observed. Assuming that the manipulation of the IKEA effect was successful, this is a promising suggestion that it may be partly responsible for an increase in far transfer scores, and merits further investigation.

Alternatively, it remains possible that the IKEA effect was not successfully elicited at all, and something else caused the improved attainment. A failure to elicit the IKEA effect could be explained by the 'rebuild' context under which the study takes place: participants assembled and disassembled the graph multiple times while following the worksheets, but the IKEA effect does not apply when participants 'unbuild' the product [57].

Overall, the results demonstrate promising learning under both conditions and suggest that there may be an improvement in score caused by the IKEA effect, which would affirm **H1**.

5.2 Findings around how the IKEA effect might improve attitude to learning while learning an abstract skill using TBEL (RQ2)

The results do not indicate any difference in learner confidence or effort between the two conditions. This could suggest that the IKEA effect does not explain improvements in attitude to learning, or it could point to a more nuanced situation.

It is possible that the two conditions had a similar average estimated scores because the IKEA effect did not apply, or because it has no impact on self-confidence (rejecting **H2**). If this were the case, a higher perceived competence and estimate would be expected in the construction condition, corresponding to the slightly improved actual scores observed. This would be expected even without a statistically significant effect because the variance in data comes from differences between participants and from order effects, rather than from measurement errors, but the quiz and estimate were taken immediately in sequence.

A better explanation is that the more accurate estimated scores in the construction condition are explained by the IKEA effect. The IKEA effect can be understood as a mechanism for people to express their competence [50] and it is possible that having expressed this through another channel reduces overconfidence in relation to the quiz. This would suggest that **H2** should be rejected and replaced by

'The presence of the IKEA effect in a tactile learning aid will improve the accuracy of learners' confidence.' It is worth noting that Mochon et al. [50] use an unusual measure of 'feelings of competence' (based on pride and a desire to show off one's creations), which lends additional weight to this theory because the researcher observed the participants' attempts in this dissertation.

The fact that participants spent more time on the non-construction condition could suggest that the use of wires was simply more intuitive for participants. However, that would not explain the lower quiz scores, as participants were not rushed and therefore should have had time to attain a satisfactory level in both conditions. This could suggest that any differences impacted the participants' attention or concentration, rather than task importance. If construction directly increased task importance, this would have been expected to result in more time spent practising. This also aligns with the unchanged Effort/Importance subscale, suggesting that the participants did not have increased intrinsic motivation in the construction condition (**H3**). This suggests that the IKEA effect may not have a direct impact on motivation, and instead controls the amount of time that participants take to expend a fixed amount of perceived effort.

The strong positive correlation between estimated score and perceived competence suggests that it may be possible to speed up participation in future work by asking participants to estimate their performance on the test task and removing the questions on perceived competence.

If the IKEA effect improves learning by preventing overconfidence and encouraging concentration, this would be a powerful tool to incorporate into teaching. Furthermore, it would suggest that other approaches to moderating confidence could have similarly positive effects at a lower cost, given the right circumstances. While this is not certain, it would be impactful in practice if proven true.

These results point to better alternatives to both **H2** and **H3**. For **H2**, the IKEA effect might improve accuracy of self-confidence, rather directly increasing confidence. For **H3**, the IKEA effect does not appear to impact self-reported motivation, but it may improve concentration or quality of work.

5.3 Limitations

Within this study, the researcher only recruited 9 participants, because the inclusion criteria were strict (computer science students were not eligible) and no incentives were available. The resulting within-participants design led to difficulties drawing statistically significant conclusions about interaction effects. It also led to the issue of identical reported Willingness To Pay across conditions. However, the researcher worked independently, including developing the concepts, experimental protocol, hardware, and software systems, and performing data analysis and synthesis.

The lack of compensation meant that participants were not asked to pay actual money in the bidding process, which may have further impacted its validity. Since the bidding process was ineffective, it was not possible to evaluate whether the IKEA effect was actually produced by the manipulation of the level of construction, although interesting results were still observed.

The design and production of the hardware was time-consuming and this could limit future research, especially if participants are allowed to purchase the learning aid. It would be beneficial to have a simpler design, mass-producible PCB, or VR version of the learning aid.

This study did not investigate the implementation of a tangible learning aid in an actual classroom environment. This was explicitly out of scope but could be significantly different [44].

5.4 Future Work

5.4.1 Replication work. This study should be repeated with funding to allow a more robust manipulation check and a larger sample size with a between-subjects design. In particular, participants should pay real money in the bidding step to ensure accurate results. It would also be beneficial to develop an alternative validated evaluation methodology for the IKEA effect to reduce the need to compensate participants. Another alternative would be to mislead participants that they will be allowed to purchase the learning aid, as this would avoid needing to produce hundreds of copies of the hardware, without compromising validity.

5.4.2 Exploratory variations. Future work should investigate how the IKEA effect could differ in a classroom environment. Interactions between learners, teachers and content all differ in classrooms, so this could reveal interesting results. It may also be beneficial to test it across different cultures and age groups.

There could be significant changes in retention and transfer over time, as has been demonstrated before with psychomotor rather than abstract skills [32]. Investigating that in a longitudinal study would give a more complete view of the IKEA effect's impact on learning.

The experimental data highlighted the possibility that the IKEA effect gives learners means to express their competence and therefore reduces their overconfidence in test scores [see 50]. This should be explored further in future research, as it could be applied to personalize teaching when learners show signs of overconfidence. To investigate this hypothesis, future work should try manipulating the learners' ability to express their confidence, for example by having them work unsupervised. For example, the study could be conducted online with physical devices delivered to participants, or the study could be conducted in VR or on-screen.

This study did not attempt to identify which of the two mechanisms could contribute to the IKEA effect's impact on learning abstract skills. Future work should investigate this, as this could help induce the IKEA effect more effectively and economically.

5.4.3 Improvements to relevant literature. A clearer categorization of degrees of embodiment for abstract tasks would be beneficial for future reviews in this area. Currently, Johnson-Glenberg et al.'s [35] criteria do not apply clearly to abstract skills. Future work should develop a clearer taxonomy of TBEL in a manner that is applicable to all learning tasks.

Qualitative data could help to enhance the theory in this space. This study did not gather qualitative data due to the already-long duration of the study, but with a larger, between-participants design, this would be possible. Future work should consider using mixed methods, as this is especially lacking in TBEL research [84].

Finally, it would be worthwhile to simultaneously pursue other possible explanations for TBEL's improvements in attitude. The researcher is not aware of any alternative proposed explanations,

so an initial research direction would be a qualitative exploratory study that repeats an existing TBEL study which was known to improve attitude. This could then guide future research based on participant insights.

6 Conclusion

In conclusion, this dissertation proposes and explores a novel theory for improving attitude to learning, based on interactions between the IKEA effect and TBEL. Results of the exploratory study suggest that learner-led construction may cause higher attainment without directly affecting confidence. This unexpected result could point to the opportunity for self-expression and creation being intrinsically linked to attainment regardless of effort and motivation. If demonstrated, these effects might allow exploiting more benefits of TBEL in a classroom setting, without the barriers of full-body interaction. Further work is needed to validate the hypotheses and explore them in more depth across more participants and contexts.

Word count: 8123

References

[1] 2021. 3.5mm Audio Jack (Male). Components101. (20th Aug. 2021). Retrieved 14th Nov. 2025 from https://components101.com/connectors/35mm-audio-jack.

[2] Ahotovi Thomas Ahoto, Mark Bunji Mbaye and Emmanuel Anyigbah. 2022. The Impacts of Learner-Instructor Interaction, Learner-Learner, Learner-Content Interaction, Internet Self-Efficacy and Self-Regulated Learning on Satisfaction of Online Education of African Medical Students. *OAlib (Online)*, 9, 9, 1–16. doi:10.4236/oalib.1109202.

[3] Sean Eron Anderson. [n. d.] Bit Twiddling Hacks. Retrieved 5th Dec. 2025 from http://www.graphics.stanford.edu/%7Eseander/bithacks.html#ParityParallel.

[4] [SW] Arduino, Arduino IDE version 2.3.7, 18th Apr. 2026. URL: https://github.com/arduino/arduino-ide.

[5] Marta Arguedas, Thanasis Daradoumis and Santi Caballé. 2024. Measuring the effects of pedagogical agent cognitive and affective feedback on students' academic performance. *Frontiers in Artificial Intelligence*, 7, (16th Dec. 2024). doi:10.3389/frai.2024.1495342.

[6] [SW] Armin, digitalWriteFast 24th Nov. 2025. URL: https://github.com/ArminJo/digitalWriteFast.

[7] Ryan S. J. d. Baker, Sidney K. D'Mello, Ma. Mercedes T. Rodrigo and Arthur C. Graesser. 2010. 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*, 68, 4, (1st Apr. 2010), 223–241. doi:10.1016/j.ijhcs.2009.12.003.

[8] [SW] Bambu Lab, Bambu Studio version 2.4.0 Public Release, 18th Nov. 2025. URL: https://flathub.org/en/apps/com.bambulab.BambuStudio.

[9] Lawrence W. Barsalou. 1999. Perceptual symbol systems. *Behavioral and Brain Sciences*, 22, 4, (Aug. 1999), 577–660. doi:10.1017/S0140525X99002149.

[10] Cristian Bernareggi, Christian Comaschi, Giancarlo Dalto, Piero Mussio and Loredana Parasiliti Provenza. 2008. Multimodal Exploration and Manipulation of Graph Structures. In *Computers Helping People with Special Needs*. Vol. 5105. Klaus Miesenberger, Joachim Klaus, Wolfgang Zagler and Arthur Karshmer, editors. Springer Berlin Heidelberg, Berlin, Heidelberg, 934–937. ISBN: 978-3-540-70540-6. doi:10.1007/978-3-540-70540-6_140.

[11] Cristian Bernareggi, Christian Comaschi, Andrea Marcante, Piero Mussio, Loredana Parasiliti Provenza and Sara Vanzi. 2008. A multimodal interactive system to create and explore graph structures. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*. CHI '08: CHI Conference on Human Factors in Computing Systems. ACM, Florence Italy, (5th Apr. 2008), 2697–2702. ISBN: 978-1-60558-012-8. doi:10.1145/1358628.1358747.

[12] John Brooke. [n. d.] SUS - A quick and dirty usability scale.

[13] [SW] Phil Burgess, Adafruit NeoPixel Library version 1.15.2, 15th Oct. 2025. URL: https://github.com/adafruit/Adafruit_NeoPixel.

[14] Center for Self-Determination Theory. 2025. Intrinsic Motivation Inventory (IMI). (2025). Retrieved 18th Nov. 2025 from https://selfdeterminationtheory.org/intrinsic-motivation-inventory/.

[15] cfr. 2025. Answer to "Problem with tagging non-floating lstlisting". TeX - LaTeX Stack Exchange. (4th Dec. 2025). Retrieved 4th Apr. 2026 from https://tex.stackexchange.com/a/755795/284347.

[16] Ali Darejeh, Nadine Marcus and John Sweller. 2022. Increasing learner interactions with E-learning systems can either decrease or increase cognitive load depending on the nature of the interaction. *L'Année psychologique*, 122, 3, (13th Sept. 2022), 405–437. doi:10.3917/anpsy1.223.0405.

[17] Cory A. Dunnick et al. 2023. Dermatology Tactile Learning Tool: Development and student evaluation of an interactive 3-dimensional skin lesion model for medical student education. *Journal of the American Academy of Dermatology*, 88, 5, (1st May 2023), 1113–1115. doi:10.1016/j.jaad.2019.10.071.

[18] EDU, Imperial College London. 2018. Guidance for developing questionnaires using self-efficacy scales. (2018). Retrieved 16th Nov. 2025 from https://www.imperial.ac.uk/media/imperial-college/staff/education-development-unit/public/Handout_Guidance-on-self-efficacy-questionnaires.pdf.

[19] egreg. 2016. Answer to "Inter-sentence space without a period?" TeX - LaTeX Stack Exchange. (3rd Mar. 2016). Retrieved 29th Oct. 2025 from https://tex.stackexchange.com/a/297226/284347.

[20] Logan Fiorella and Richard E. Mayer. 2016. Effects of observing the instructor draw diagrams on learning from multimedia messages. *Journal of Educational Psychology*, 108, 4, (May 2016), 528–546. doi:10.1037/edu0000065.

[21] Isabel Sophie Fitton, Elizabeth Dark, Manoela Milena Oliveira da Silva, Jeremy Dalton, Michael J Proulx, Christopher Clarke and Christof Lutteroth. 2024. Watch This! Observational Learning in VR Promotes Better Far Transfer than Active Learning for a Fine Psychomotor Task. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (CHI '24). Association for Computing Machinery, New York, NY, USA, (11th May 2024), 1–19. ISBN: 979-8-4007-0330-0. doi:10.1145/3613904.3642550.

[22] Simon Ford and Tim Minshall. 2019. Invited review article: Where and how 3D printing is used in teaching and education. *Additive Manufacturing*, 25, (1st Jan. 2019), 131–150. doi:10.1016/j.addma.2018.10.028.

[23] Yan Ru Guo and Dion Hoe-Lian Goh. 2015. Affect in Embodied Pedagogical Agents: Meta-Analytic Review. *Journal of Educational Computing Research*, 53, 1, (Sept. 2015), 124–149. doi:10.1177/0735633115588774.

[24] Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart. 2008. Exploring Network Structure, Dynamics, and Function using NetworkX. *Python in Science Conference*, (6th May 2008). doi:10.25080/TCWV9851.

[25] Daniel C. A. Hillman, Deborah J. Willis and Charlotte N. Gunawardena. 1994. Learner-interface interaction in distance education: An extension of contemporary models and strategies for practitioners. *American Journal of Distance Education*, 8, 2, (1st Jan. 1994), 30–42. doi:10.1080/08923649409526853.

[26] Kasper Hornbæk and Antti Oulasvirta. 2017. What Is Interaction? In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17: CHI Conference on Human Factors in Computing Systems. ACM, Denver Colorado USA, (2nd May 2017), 5040–5052. ISBN: 978-1-4503-4655-9. doi:10.1145/3025453.3025765.

[27] J. D. Hunter. 2007. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9, 3, 90–95. doi:10.1109/MCSE.2007.55.

[28] Lars-Erik Janlert and Erik Stolterman. 2017. The Meaning of Interactivity—Some Proposals for Definitions and Measures. *Human–Computer Interaction*, 32, 3, (4th May 2017), 103–138. doi:10.1080/07370024.2016.1226139.

[29] [SW] JASP Team, JASP version 0.95.4, 2025. URL: https://jasp-stats.org/.

[30] Megan A. Jennings and Robert A. Cribbie. 2022. Comparing Pre-Post Change Across Groups: Guidelines for Choosing between Difference Scores, ANCOVA, and Residual Change Scores. *Journal of Data Science*, 14, 2, (4th Aug. 2022), 205–230. doi:10.6339/JDS.201604_14(2).0002.

[31] [SW] Jetbrains s.r.o., PyCharm version 2025.2.5, 27th Nov. 2025. URL: https://www.jetbrains.com/pycharm/.

[32] Crescent Jicol et al. 2021. Effects of Emotion and Agency on Presence in Virtual Reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–13. ISBN: 978-1-4503-8096-6. doi:10.1145/3411764.3445588.

[33] Yunju Jo, Seok-Ju Chun and Jungwoo Ryoo. 2021. Tactile Scratch Electronic Block System: Expanding Opportunities for Younger Children to Learn Programming. *International Journal of Information and Education Technology*, 11, 7, 319–323. doi:10.18178/ijiet.2021.11.7.1529.

[34] Timothy Johnson. 2018. SONY SIRC Protocol. (June 2018). Retrieved 2nd Dec. 2025 from https://faculty-web.msoe.edu/johnsontimoj/Common/FILES/sony_sirc_protocol.pdf.

[35] Mina C. Johnson-Glenberg, David A. Birchfield, Lisa Tolentino and Tatyana Koziupa. 2014. Collaborative Embodied Learning in Mixed Reality Motion-Capture Environments: Two Science Studies. *Journal of educational psychology*, 106, 1, 86–104. doi:10.1037/a0034008.

[36] Insung Jung, Seonghee Choi, Cheolil Lim and Junghoon Leem. 2002. Effects of Different Types of Interaction on Learning Achievement, Satisfaction and Participation in Web-Based Instruction. *Innovations in Education and Teaching International*, 39, 2, (1st Jan. 2002), 153–162. doi:10.1080/14703290252934603.

[37] Todd R. Kelley and J. Geoff Knowles. 2016. A conceptual framework for integrated STEM education. *International journal of STEM education*, 3, 1, 1–. doi:10.1186/s40594-016-0046-z.

[38] Mannu Lambrichts, Jose Maria Tijerina and Raf Ramakers. 2020. SoftMod: A Soft Modular Plug-and-Play Kit for Prototyping Electronic Systems. In *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied*

Interaction (TEI '20). Association for Computing Machinery, New York, NY, USA, (9th Feb. 2020), 287–298. ISBN: 978-1-4503-6107-1. doi:10.1145/3374920.3374950.

[39] Zhiwei Liu, Haode Zuo, Yan Zhao and Yongjing Lu. 2025. The effect of embodied learning on students' learning performance: A meta-analysis. *Frontiers in Psychology*, 16, (22nd Aug. 2025). doi:10.3389/fpsyg.2025.1658797.

[40] Einav London. 2022. Changing students' attitudes toward mathematics through creative steam projects designed and executed by them (smart program). *Acta Didactica Napocensia*, 15, 2, (31st Dec. 2022), 269–277. doi:10.24193/adn.15.2.18.

[41] Laura Malinverni and Narcis Pares. 2014. Learning of Abstract Concepts through Full-Body Interaction: A Systematic Review. *Educational Technology & Society*, 17, 4, 100–116. doi:10.30191/ETS.201410_17(4).0007.

[42] Martha Manrique et al. 2024. Haptic experience to significantly motivate anatomy learning in medical students. *BMC medical education*, 24, 1, 946–10. doi:10.1186/s12909-024-05829-w.

[43] Lauren E. Marsh, Patricia Kanngiesser and Bruce Hood. 2018. When and how does labour lead to love? The ontogeny and mechanisms of the IKEA effect. *Cognition*, 170, (1st Jan. 2018), 245–253. doi:10.1016/j.cognition.2017.10.012.

[44] Lukáš Másilko and Jiří Pecl. 2014. Making Graph Theory Algorithms Accessible to Blind Students. In *Computers Helping People with Special Needs*. Vol. 8547. Klaus Miesenberger, Deborah Fels, Dominique Archambault, Petr Peňáz and Wolfgang Zagler, editors. Springer International Publishing, Cham, 549–556. ISBN: 978-3-319-08596-8. doi:10.1007/978-3-319-08596-8_86.

[45] Mattman944. 2023. Answer to "How should I wire my circuit onto a perfboard?" Electrical Engineering Stack Exchange. (27th Nov. 2023). Retrieved 15th Nov. 2025 from https://electronics.stackexchange.com/a/690706/343789.

[46] John McCarthy and Peter Wright. 2017. Commentary: Making Interactivity Meaningful for Contemporary HCI. *Human–Computer Interaction*, 32, 3, (4th May 2017), 148–152. doi:10.1080/07370024.2016.1254048.

[47] David Merrill, Emily Sun and Jeevan Kalanithi. 2012. Sifteo cubes. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '12). Association for Computing Machinery, New York, NY, USA, (5th May 2012), 1015–1018. ISBN: 978-1-4503-1016-1. doi:10.1145/2212776.2212374.

[48] James Minogue and M. Gail Jones. 2006. Haptics in Education: Exploring an Untapped Sensory Modality. *Review of Educational Research*, 76, 3, (1st Sept. 2006), 317–348. doi:10.3102/00346543076003317.

[49] Frank Mittelbach. 2012. Answer to "How to automatically hyphenate within \texttt?" TeX - LaTeX Stack Exchange. (13th Feb. 2012). Retrieved 18th Nov. 2025 from https://tex.stackexchange.com/a/44362/284347.

[50] Daniel Mochon, Michael I. Norton and Dan Ariely. 2012. Bolstering and restoring feelings of competence via the IKEA effect. *International Journal of Research in Marketing*. Special Issue on Consumer Identities 29, 4, (1st Dec. 2012), 363–369. doi:10.1016/j.ijresmar.2012.05.001.

[51] moewe. 2020. Answer to "Apostrophe s when citing". TeX - LaTeX Stack Exchange. (10th Apr. 2020). Retrieved 5th Nov. 2025 from https://tex.stackexchange.com/a/537765/284347.

[52] Joi L. Moore, Camille Dickson-Deane and Krista Galyen. 2011. E-Learning, online learning, and distance learning environments: Are they the same? *The Internet and Higher Education*. Web Mining and Higher Education: Introduction to the Special Issue 14, 2, (1st Mar. 2011), 129–135. doi:10.1016/j.iheduc.2010.10.001.

[53] Michael G. Moore. 1989. Editorial: Three types of interaction. *American Journal of Distance Education*, 3, 2, (1st Jan. 1989), 1–7. doi:10.1080/08923648909526659.

[54] Roxana Moreno. 2007. Optimising learning from animations by minimising cognitive load: cognitive and affective consequences of signalling and segmentation methods. *Applied Cognitive Psychology*, 21, 6, 765–781. doi:10.1002/acp.1348.

[55] Jonathan Mumm and Bilge Mutlu. 2011. Designing motivational agents: The role of praise, social comparison, and embodiment in computer feedback. *Computers in Human Behavior*. 2009 Fifth International Conference on Intelligent Computing 27, 5, (1st Sept. 2011), 1643–1650. doi:10.1016/j.chb.2011.02.002.

[56] [n. d.] Nano | Arduino Documentation. Retrieved 15th Nov. 2025 from https://docs.arduino.cc/hardware/nano/#features.

[57] Michael I. Norton, Daniel Mochon and Dan Ariely. 2012. The IKEA effect: When labor leads to love. *Journal of Consumer Psychology*, 22, 3, (July 2012), 453–460. doi:10.1016/j.jcps.2011.08.002.

[58] Fred Paas and John Sweller. 2012. An Evolutionary Upgrade of Cognitive Load Theory: Using the Human Motor System and Collaboration to Support the Learning of Complex Cognitive Tasks. *Educational Psychology Review*, 24, 1, (1st Mar. 2012), 27–45. doi:10.1007/s10648-011-9179-2.

[59] Fred Paas, Juhani E. Tuovinen, Huib Tabbers and Pascal W. M. Van Gerven. 2003. Cognitive Load Measurement as a Means to Advance Cognitive Load Theory. *Educational Psychologist*, 38, 1, (1st Jan. 2003), 63–71. doi:10.1207/S15326985EP3801_8.

[60] Ayellet Pelled, Stefanie Z. Demetriades and Nathan Walter. 2025. Labor Leads to Love, Right? A Meta-Analysis of the IKEA Effect. *Psychology & Marketing*, n/a, n/a, (14th Oct. 2025). doi:10.1002/mar.70064.

[61] Jan L. Plass, Bruce D. Homer and Elizabeth O. Hayward. 2009. Design factors for educationally effective animations and simulations. *Journal of Computing in Higher Education*, 21, 1, (1st Apr. 2009), 31–61. doi:10.1007/s12528-009-9011-x.

[62] Daniel Preece, Sarah B. Williams, Richard Lam and Renate Weller. 2013. "Let's Get Physical": Advantages of a physical model over 3D computer models and textbooks in learning imaging anatomy. *Anatomical Sciences Education*, 6, 4, 216–224. doi:10.1002/ase.1345.

[63] Ramkumar Rajendran, Sridhar Iyer and Sahana Murthy. 2019. Personalized Affective Feedback to Address Students' Frustration in ITS. *IEEE Transactions on Learning Technologies*, 12, 1, (Jan. 2019), 87–97. doi:10.1109/TLT.2018.2807447.

[64] Melissa V. Ramirez and Claire L. Gordy. 2020. STEM BUILD: An Online Community To Decrease Barriers to Implementation of Inclusive Tactile Teaching Tools. *Journal of Microbiology & Biology Education*, 21, 1, (10th Apr. 2020), 10.1128/jmbe.v21i1.1963. doi:10.1128/jmbe.v21i1.1963.

[65] Falko Rheinberg, Regina Vollmeyer and Bruce D. Burns. 2001. FAM: Ein Fragebogen zur Erfassung aktuller Motivation in Lern- und Leistungssituationen. *Diagnostica*, 47, 2, (Apr. 2001), 57–66. doi:10.1026//0012-1924.47.2.57.

[66] Rosalind W. Picard. 2000. Applications of Affective Computing. In *Affective Computing*. The MIT Press, Cambridge, MA, 85–111. ISBN: 978-0-262-28158-4. Retrieved 12th Nov. 2025 from https://direct.mit.edu/books/monograph/4296/chapter/180323/Applications-of-Affective-Computing.

[67] Richard M. Ryan, Valerie Mims and Richard Koestner. 1983. Relation of reward contingency and interpersonal context to intrinsic motivation: A review and test using cognitive evaluation theory. *Journal of Personality and Social Psychology*, 45, 4, (Oct. 1983), 736–750. doi:10.1037/0022-3514.45.4.736.

[68] Jeff Sauro and Joseph S. Dumas. 2009. Comparison of three one-question, post-task usability questionnaires. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '09). Association for Computing Machinery, New York, NY, USA, (4th Apr. 2009), 1599–1608. ISBN: 978-1-60558-246-7. doi:10.1145/1518701.1518946.

[69] Konrad J. Schönborn, Petter Bivall and Lena A. E. Tibell. 2011. Exploring relationships between students' interaction and learning with a haptic virtual biomolecular model. *Computers & Education*, 57, 3, (1st Nov. 2011), 2095–2105. doi:10.1016/j.compedu.2011.05.013.

[70] Amer Sehic and Qalbi Khan. 2024. Traditional Hands-On Puzzle Method versus Fully Digital Approach in Teaching Tooth Morphology: A Comparative Study of Educational Outcomes. *Dentistry Journal*, 12, 6, 162. doi:10.3390/dj12060162.

[71] Lawrence Shapiro. 2019. *Embodied Cognition*. (2nd ed.). Routledge, London, (7th May 2019). 304 pp. ISBN: 978-1-315-18038-0. doi:10.4324/9781315180380.

[72] Vandana Singh and Alexander Thurman. 2019. How Many Ways Can We Define Online Learning? A Systematic Literature Review of Definitions of Online Learning (1988-2018). *American Journal of Distance Education*, 33, 4, (2nd Oct. 2019), 289–306. doi:10.1080/08923647.2019.1663082.

[73] Ishu Singhal and B. S. Balaji. 2022. Open-Source, Tactile 3D Printed Interlockable Tiles Incorporating Valency, Bonding, and Hybridization for Molecular Representation for Sighted and Visually Impaired Students. *Journal of Chemical Education*, 99, 4, (12th Apr. 2022), 1708–1714. doi:10.1021/acs.jchemed.1c01278.

[74] Alexander Skulmowski, Simon Pradel, Tom Kühnert, Guido Brunnett and Günter Daniel Rey. 2016. Embodied learning using a tangible user interface: The effects of haptic perception and selective pointing on a spatial learning task. *Computers & Education*, 92–93, (1st Jan. 2016), 64–75. doi:10.1016/j.compedu.2015.10.011.

[75] [SW] SolveSpace contributors, SolveSpace version v3.1, 18th May 2022. URL: https://solvespace.com/index.pl.

[76] Jui-ni Sun and Yu-chen Hsu. 2013. Effect of interactivity on learner perceptions in Web-based instruction. *Computers in Human Behavior*. Including Special Section Youth, Internet, and Wellbeing 29, 1, (1st Jan. 2013), 171–184. doi:10.1016/j.chb.2012.08.002.

[77] John Sweller, Jeroen J. G. van Merrienboer and Fred G. W. C. Paas. 1998. Cognitive Architecture and Instructional Design. *Educational Psychology Review*, 10, 3, (1st Sept. 1998), 251–296. doi:10.1023/A:1022193728205.

[78] Azunka Ukala, Chinyere C. Ukala, Martin Thomas and Eugene Ogbodo. 2023. Engaging Young Minds: A Novel Approach to E-Learning in STEM Subjects. *European Modern Studies Journal*, 7, 4, 213–217. doi:10.59573/emsj.7(4).2023.19.

[79] David Watson, Lee Anna Clark and Auke Tellegen. 1988. Development and Validation of Brief Measures of Positive and Negative Affect: The PANAS Scales. *Journal of personality and social psychology*, 54, 6, 1063–1070. doi:10.1037/0022-3514.54.6.1063.

[80] Klaus Wertenbroch and Bernd Skiera. 2002. Measuring Consumers' Willingness to Pay at the Point of Purchase. *Journal of Marketing Research*, 39, 2, (1st May 2002), 228–241. doi:10.1509/jmkr.39.2.228.19086.

[81] Nan Xiang, Hai-Ning Liang, Lingyun Yu, Xiaosong Yang and Jian J. Zhang. 2023. A mixed reality framework for microsurgery simulation with visual-tactile perception. *The Visual Computer*, 39, 8, (Aug. 2023), 3661–3673. doi:10.1007/s00371-023-02964-1.

[82] Junhong Xiao. 2017. Learner-content interaction in distance education: The weakest link in interaction research. *Distance Education*, 38, 1, (2nd Jan. 2017), 123–135. doi:10.1080/01587919.2017.1298982.

[83] Rong-Ling Zhang, Shu-yu Yu, Rui-feng Gao, Li-fei Wang and Xiang-chun He. 2025. How does technology-based embodied learning affect learning effectiveness? – Based on a systematic literature review and meta-analytic approach. *Interactive Learning Environments*, 33, 6, (3rd July 2025), 4135–4158. doi:10.1080/10494820.2025.2479176.

[84] Baichang Zhong, Siyu Su, Xiaofan Liu and Zehui Zhan. 2023. A literature review on the empirical studies of technology-based embodied learning. *Interactive Learning Environments*, 31, 8, (17th Nov. 2023), 5180–5199. doi:10.1080/10494820.2021.1999274.

A Hardware

A.1 Bill of Materials

Item	SKU	Count
Arduino Nano	A000005	7
RS Pro 3.5mm TRS socket	913-1055	35
RS Pro 3.5mm TRS cable (jack-to-jack)	286-2873	15
NeoPixel Diffused 5mm Through-Hole LED	1938	42
Diptronics Tactile Switch	DTS-61N-V	42

A.2 Schematics and Designs

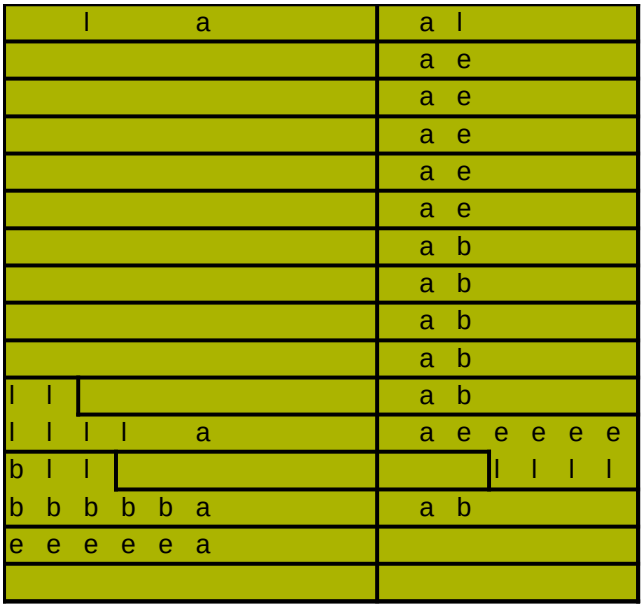


Figure 8: Stripboard design. Black lines indicate disconnected copper regions. a = Arduino, l = LED, e = TRS socket, b = button. Layout shown from reverse side.

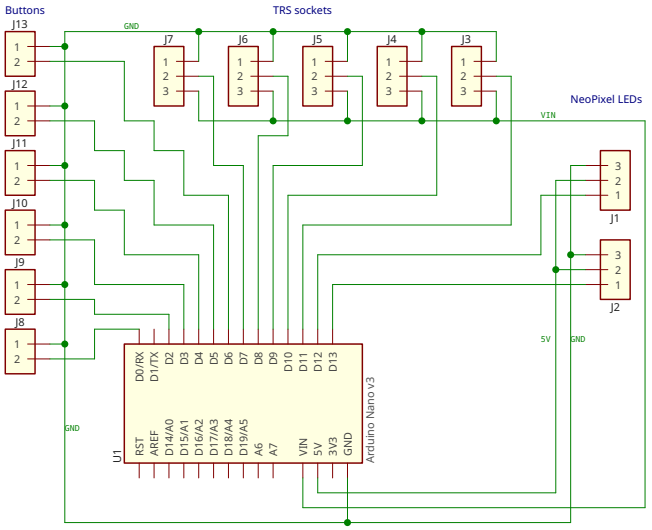


Figure 8: Schematic diagram

The CAD files are available alongside this PDF.

B Study

B.1 Latin square

Participant	First condition		Second condition	
	Condition	Algorithm	Condition	Algorithm
1	Buttons	BFS	Wires	DFS
2	Buttons	DFS	Wires	BFS
3	Wires	BFS	Buttons	DFS
4	Wires	DFS	Buttons	BFS
5	Buttons	BFS	Wires	DFS
6	Buttons	DFS	Wires	BFS
7	Wires	BFS	Buttons	DFS
8	Wires	DFS	Buttons	BFS
9	Buttons	BFS	Wires	DFS

B.2 Documentation

B.2.1 Paper. The following documents were shown to participants on paper during the study.

PARTICIPANT INFORMATION SHEET

Construction in tactile learning aids

Name of Researcher: Penn Mackintosh
Contact details of Researcher: pm2022@bath.ac.uk

Name of Supervisor: Ollie Hanton
Contact details of Supervisor: oph33@bath.ac.uk

This information sheet forms part of the process of informed consent. It should tell you what the research is about and what taking part will involve. Please read this information sheet carefully and ask one of the researchers named above if you have any questions.

1. What is the purpose of the project?

The project aims to investigate whether there is a difference between building your own learning aid and using a pre-made one. This will focus on how each learning aid affects your understanding and confidence. This may allow future work to design confidence-boosting curricula.

2. Who can take part in this project?

To take part, you must:

- be aged 18 or over;
- be fluent in English; and
- not meet any of the exclusion criteria set out in section 5

3. Do I have to take part?

It is completely up to you to decide if you would like to take part or not. Before you decide, you should read the information in this sheet carefully and ask any questions you may have. If you agree to take part, you will then be asked to sign a consent form. However, if at any time you decide you no longer wish to take part in this project, you are free to withdraw your participation without giving a reason.

4. What would taking part involve?

First, you would be asked to provide your age and gender, which will be treated confidentially. Then you would learn two different processes using two different learning aids. Before each one, you would take a short quiz and estimate your score. After each one, you would be asked how much you valued it. Then, you would take a different short quiz and estimate your score. Finally, after learning both processes, you would be offered a short debrief to thank you for your time and provide information on next steps.

Using the learning aids will involve connecting everyday cables (similar to a mobile phone charger) and pushing small buttons. There will also be small coloured lights which will blink on and off.

5. Are there reasons why I should not take part?

You cannot take part if:

- you have significantly reduced upper body strength or fine motor skills; or
- you are already familiar with graph theory

6. What are the possible benefits of taking part?

There are no specific benefits of taking part in the project. However, the information that you and other participants provide in this project will help us to better understand the difference between building your own learning aid and using a premade one. Additionally, you may gain a better understanding of graph theory.

7. What are the possible disadvantages and risks of taking part?

You may experience mild frustration at some points during learning sessions, although this is not likely. We do not anticipate any other disadvantages as a result of you taking part in the project. If you do feel any discomfort or appear upset, you can withdraw at any point and the researcher can direct you to an appropriate support service.

8. Who will have access to the information that I provide?

Only the research team (and University governance staff where appropriate) will have access to information that you provide, and this information will be treated as confidential.

9. What will happen to the data collected and results of the project?

We will treat all information that you provide as confidential and store it securely on the University’s secure servers. Storage of data will be done in accordance with current UK data protection legislation. If published, recorded data will be stored securely and confidentially in the University archives for a minimum of 10 years.

Your name or any other identifying information will not be disclosed in any presentation or publication of the research.

10. Who has reviewed the project?

This project has been given a favourable ethics opinion by the University of Bath, Department of Computer Science [reference: 13029-17026].

11. How can I withdraw from the project?

If you wish to withdraw your participation before completing all parts of the project, you can tell one of the researchers. You can withdraw from the project at any time without needing to give a reason and without any penalty.

Once you have completed all parts of the project, you will not be able to withdraw your data. This is because it will be stored in an anonymised form.

12. University of Bath privacy notice

The University of Bath privacy notice can be found here: https://www.bath.ac.uk/corporate-information/university-of-bath-privacy-notice-for-research-participants/.

13. What happens if there is a problem?

If you have a concern about any aspect of the project, you should ask to speak to the researchers who will do their best to answer any questions. If they are unable to resolve your concern or you wish to make a complaint regarding the project, please contact the Research Governance and Compliance Team at research-ethics@bath.ac.uk.

14. Who should I contact for more information?

Thank you for your interest in this project. Please do not hesitate to contact the researchers if you would like more information.

Name of Researcher: Penn Mackintosh
Contact details of Researcher: pm2022@bath.ac.uk

Name of Supervisor: Ollie Hanton
Contact details of Supervisor: oph33@bath.ac.uk

CONSENT FORM

Construction in tactile learning aids

Name of Researcher: Penn Mackintosh
Contact details of Researcher: pm2022@bath.ac.uk

Name of Supervisor: Ollie Hanton
Contact details of Supervisor: oph33@bath.ac.uk

Please initial box if you agree with the statement

1. I have been provided with information explaining what participation in this project involves. ☐
2. I have had an opportunity to ask questions and discuss this project. ☐
3. I have received satisfactory answers to all questions I have asked. ☐
4. I have received enough information about the project to make a decision about my participation. ☐
5. I understand that I am free to withdraw my consent to participate in the project at any time without having to give a reason for withdrawing. ☐
6. I understand the nature and purpose of the procedures involved in this project. These have been communicated to me on the information sheet accompanying this form. ☐
7. I understand and acknowledge that the investigation is designed to promote scientific knowledge and that the University of Bath will use the data I provide only for the purpose(s) set out in the information sheet. ☐
8. I understand the data I provide will be treated as confidential, and that on completion of the project my name or other identifying information will not be disclosed in any presentation or publication of the research. ☐
9. I understand that my consent to use the data I provide is conditional upon the University complying with its duties and obligations under current data protection legislation. ☐
10. I consent to my data being shared with the research team (and University governance where appropriate). ☐
11. I hereby fully and freely consent to my participation in this project. ☐

Participant's signature: _____ Date: _____

Template: February 2025

Participant name in BLOCK Letters: _____

Researcher's signature: _____ Date: _____

Researcher name in BLOCK Letters: _____

If you have any concerns or complaints related to your participation in this project, please direct them to the Research Governance and Compliance Team at research-ethics@bath.ac.uk.

Graph Theory Basics Worksheet

You can write on this worksheet if you want to, but you don’t have to.

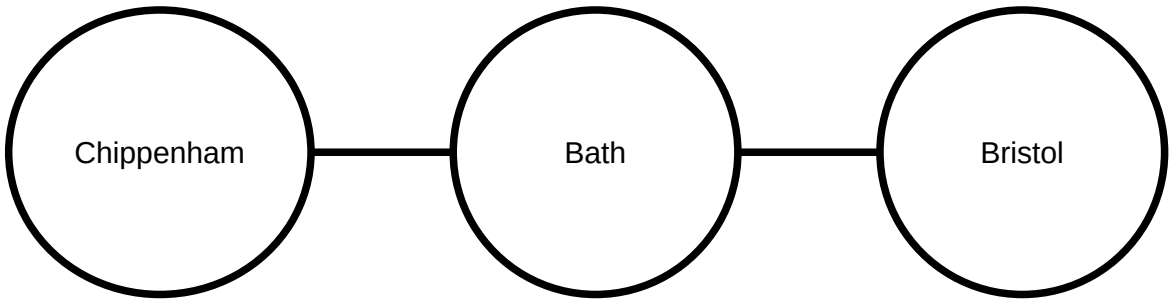
The purpose of this worksheet is to help you understand the basics of graph theory. There is no time limit.

If you have any questions the researcher will be happy to help you.

1. A graph is made up of objects and connections between pairs of those objects. For example, objects might be places, and there are connections between pairs of places if they have a direct non-stop train link.

   Here is a simple example. Chippenham has a direct train to Bath. Bath has a direct train to Bristol. But there is no direct train between Chippenham and Bristol without stopping in Bath.

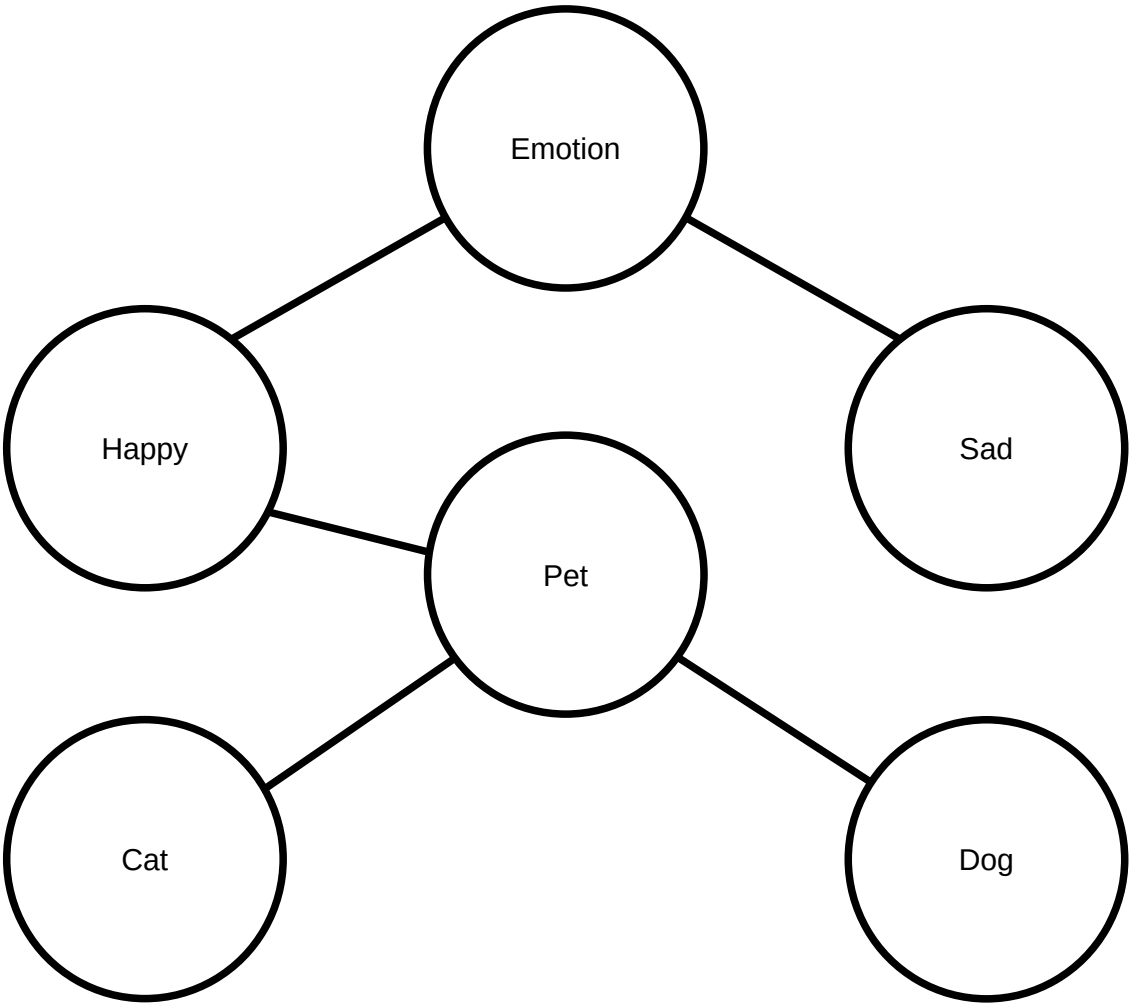
   Below is a graph showing these train connections:



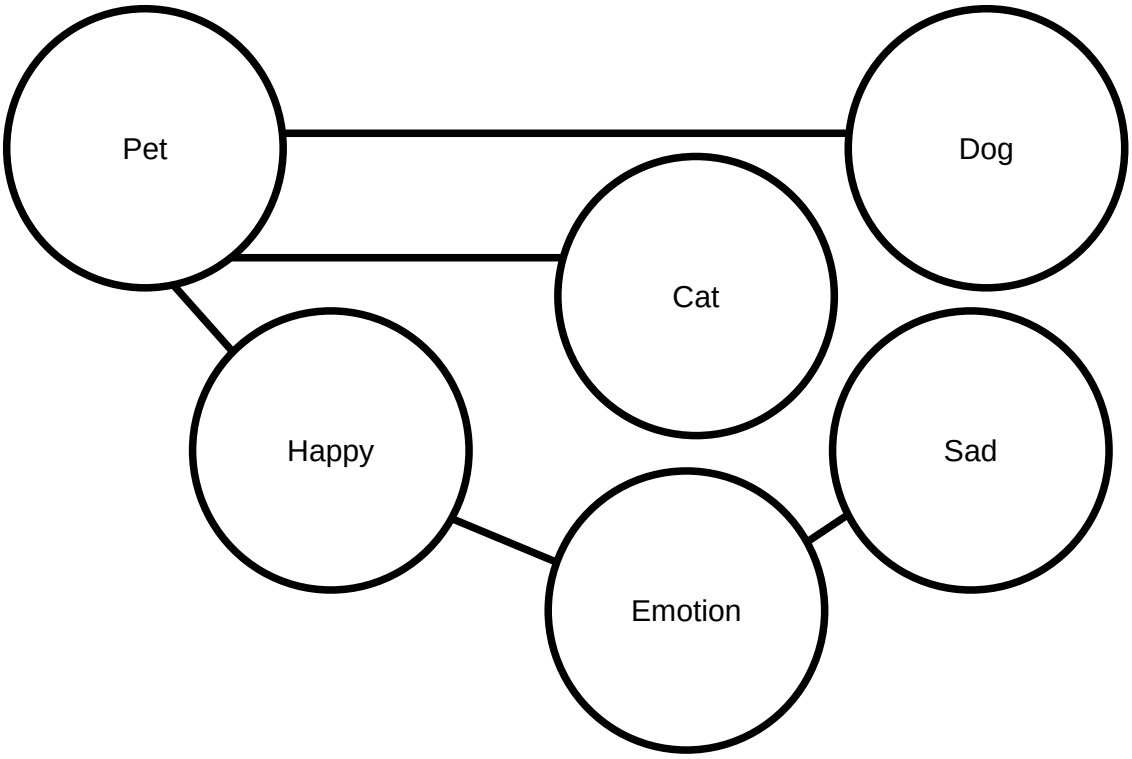
2. A graph can also represent abstract things. For example, it can represent the similarity or connection between words.

   “Happy” and “sad” are both emotions that people can feel. And a pet often makes a person happy. “Cat” and “dog” are both types of pet.

   Here is a graph that shows how these words connect:

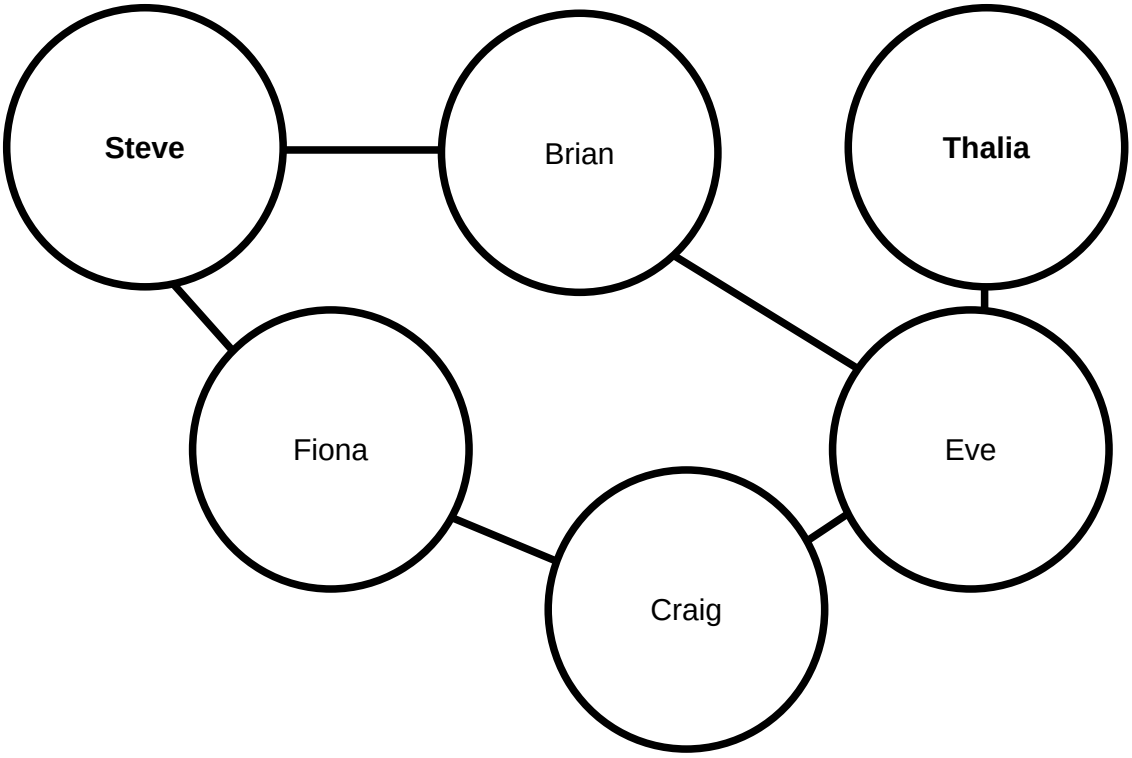


3. Here is another way of drawing the same graph:



These are the same because all the words connect to the same words as before.

4. Here is a graph showing some people. People who know each other are connected.

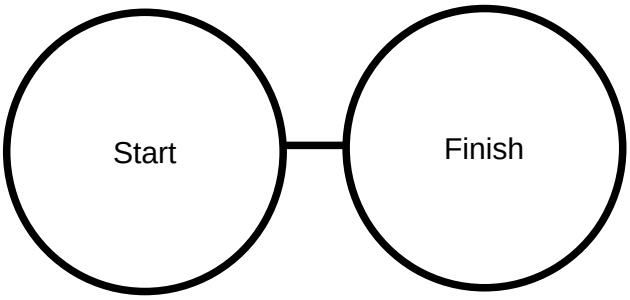


What is the fastest way for Steve to get a message to Thalia by word of mouth? Is there another way?

This is the type of problem you will be trying to solve today. You will be trying to find routes between two objects of the graph.

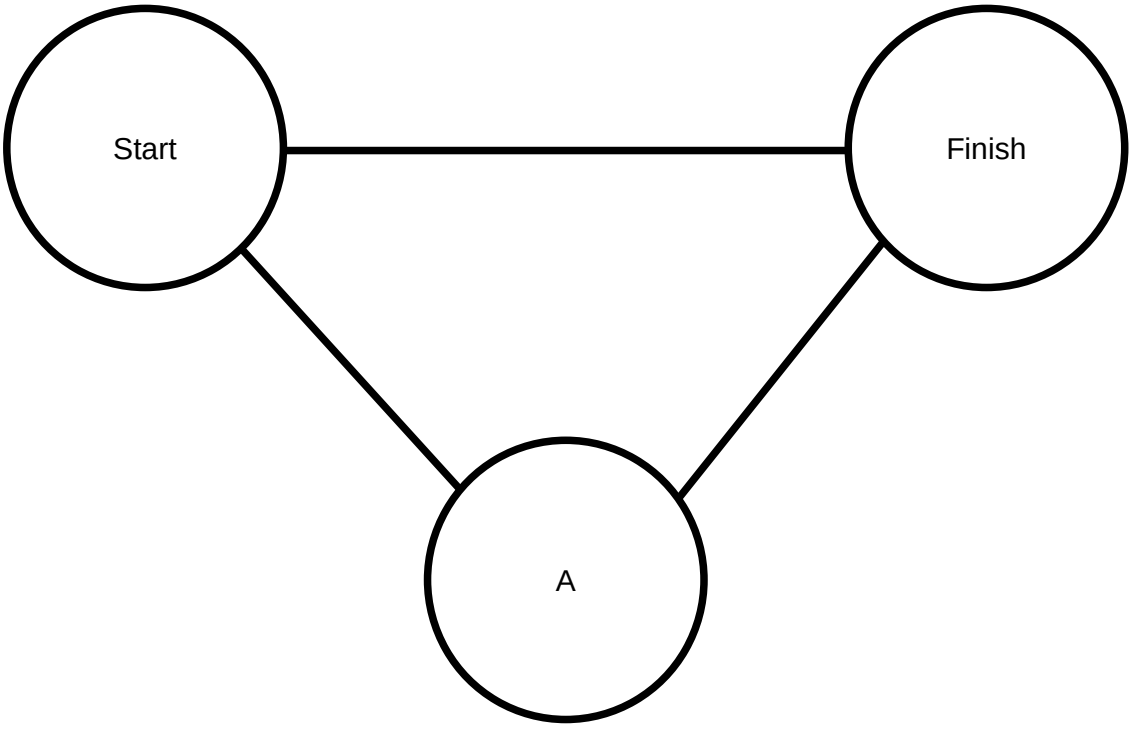
Quiz Guidance

1. What is the length of the path from start to finish in this graph?



Answer: 2

2. List all possible paths from start to finish in this graph.



Answer: Path 1: (Start, Finish); Path 2: (Start, A, Finish)

Bidding

Please enter the amount of money you would be willing to pay to purchase this learning aid in its current state.

£_____._____

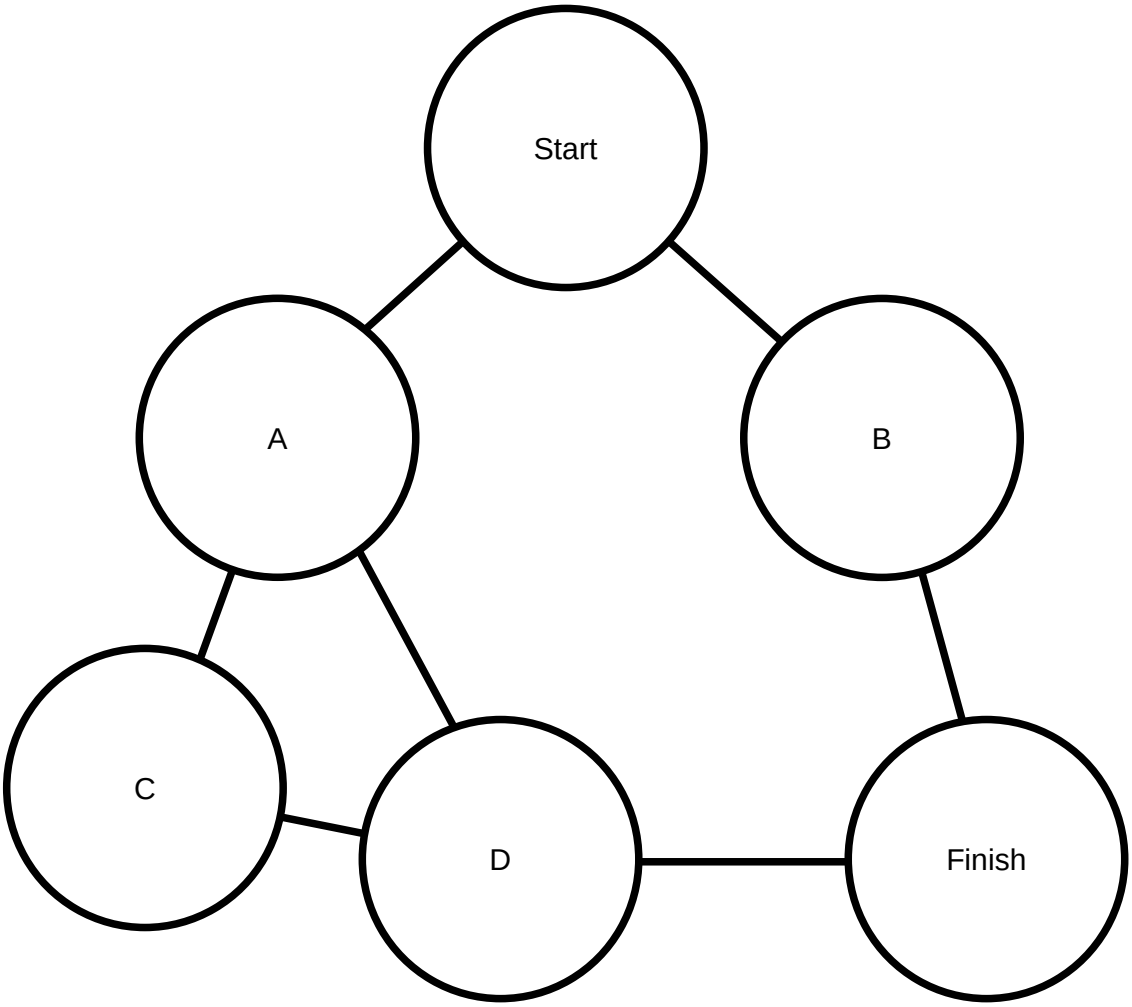
After you have chosen an amount we will pick a random amount of money. If you bid more than that amount, you can receive a digital copy of a photo of the product (as currently assembled), while if you bid less you will not receive anything.

You will not actually have to pay any money, but please ensure your bid is realistic.

Participant #: _____ Condition: ____/_____

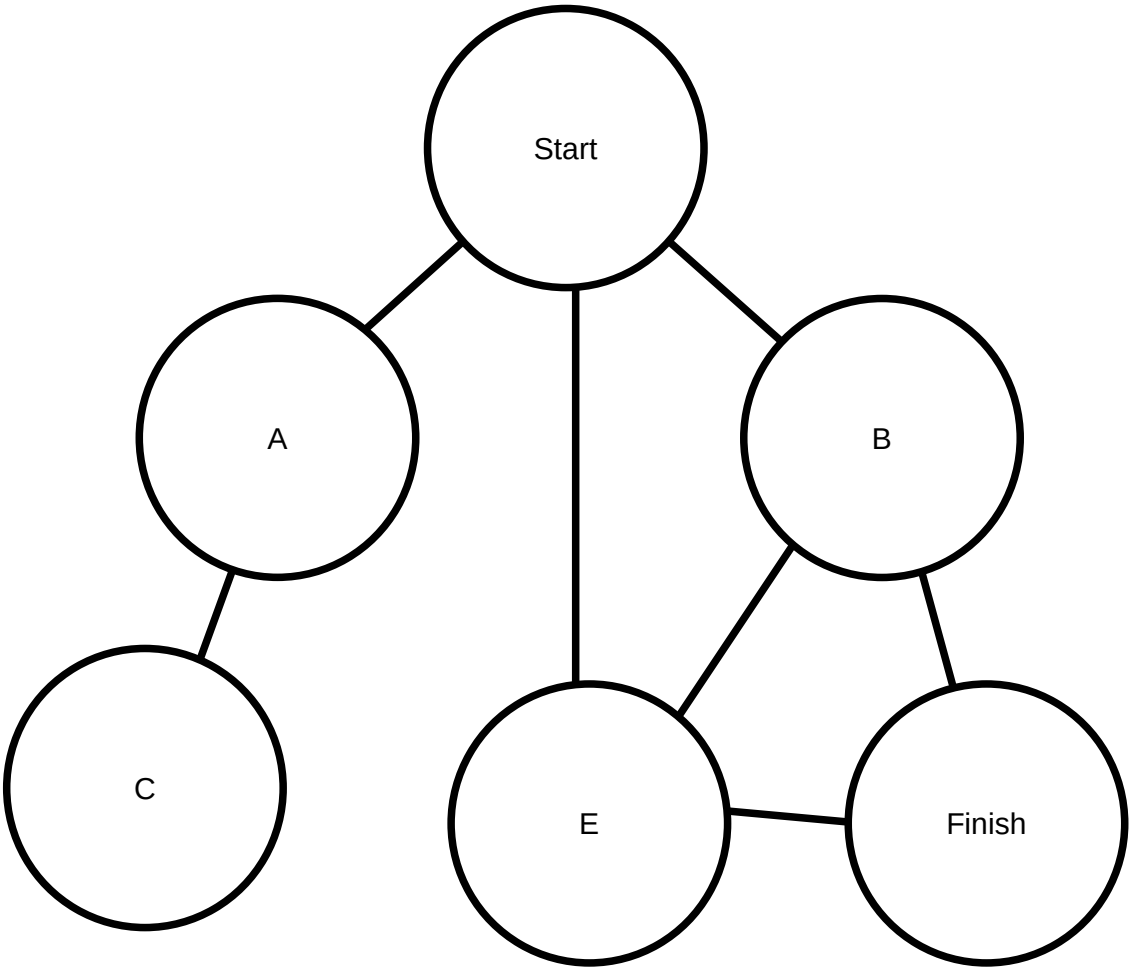
Quiz

1. What is the length of the longest possible search order (from start to finish) that Breadth First Search could take in this graph?



2. Thinking about the longest possible search that Breadth First Search could take in the graph from question 1, in what order are the objects visited by Breadth First Search?
3. What is the length of the shortest possible search that Breadth First Search could take in the graph from question 1?
4. Thinking about the shortest possible search that Breadth First Search could take in the graph from question 1, in what order are the objects visited by Breadth First Search?

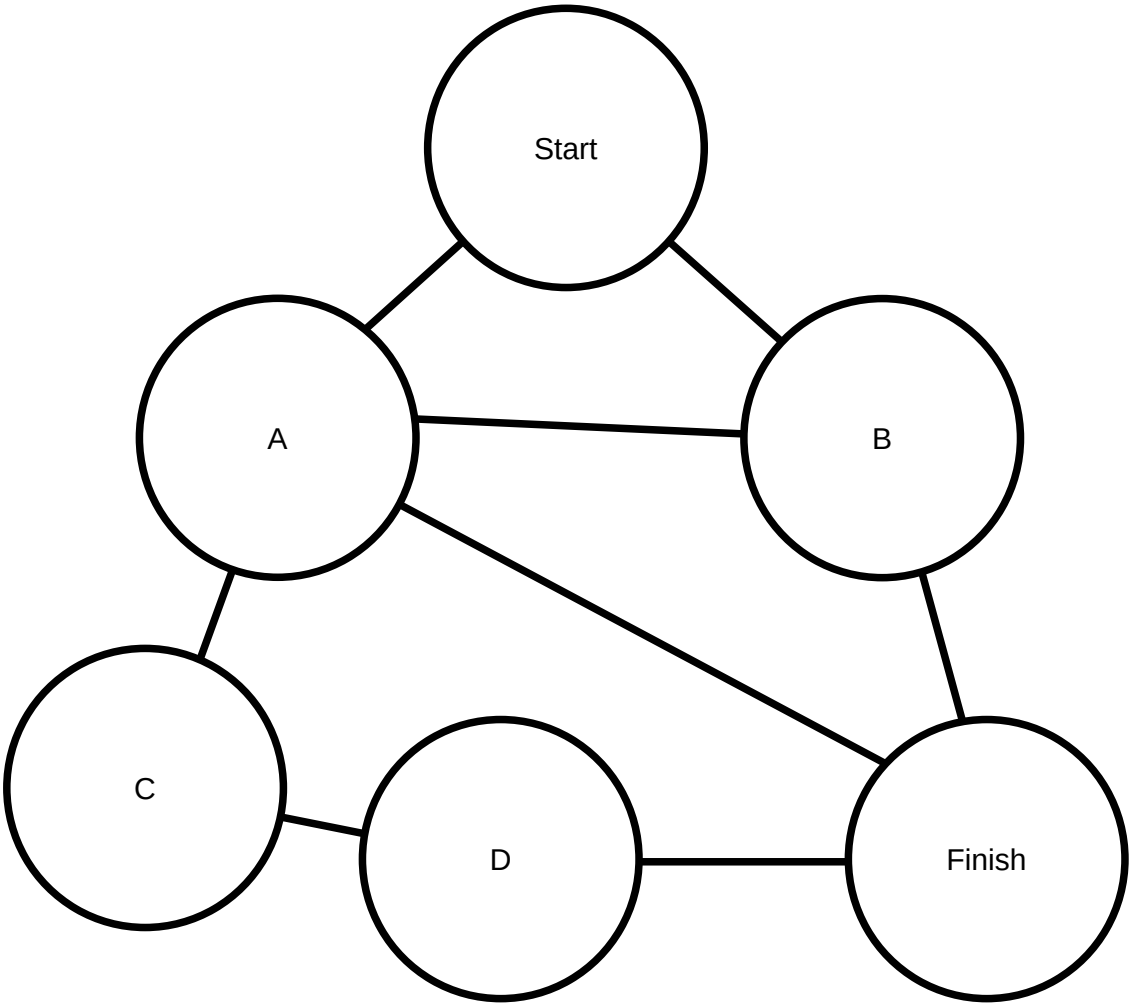
5. What is the length of the longest possible search order (from start to finish) that Breadth First Search could take in this graph?



6. Thinking about the longest possible search that Breadth First Search could take in the graph from question 5, in what order are the objects visited by Breadth First Search?
7. What is the length of the shortest possible search that Breadth First Search could take in the graph from question 5?
8. Thinking about the shortest possible search that Breadth First Search could take in the graph from question 5, in what order are the objects visited by Breadth First Search?
9. Find a way of performing a Breadth First Search that tries visiting object “C”. Write down every node that it visits, in order.

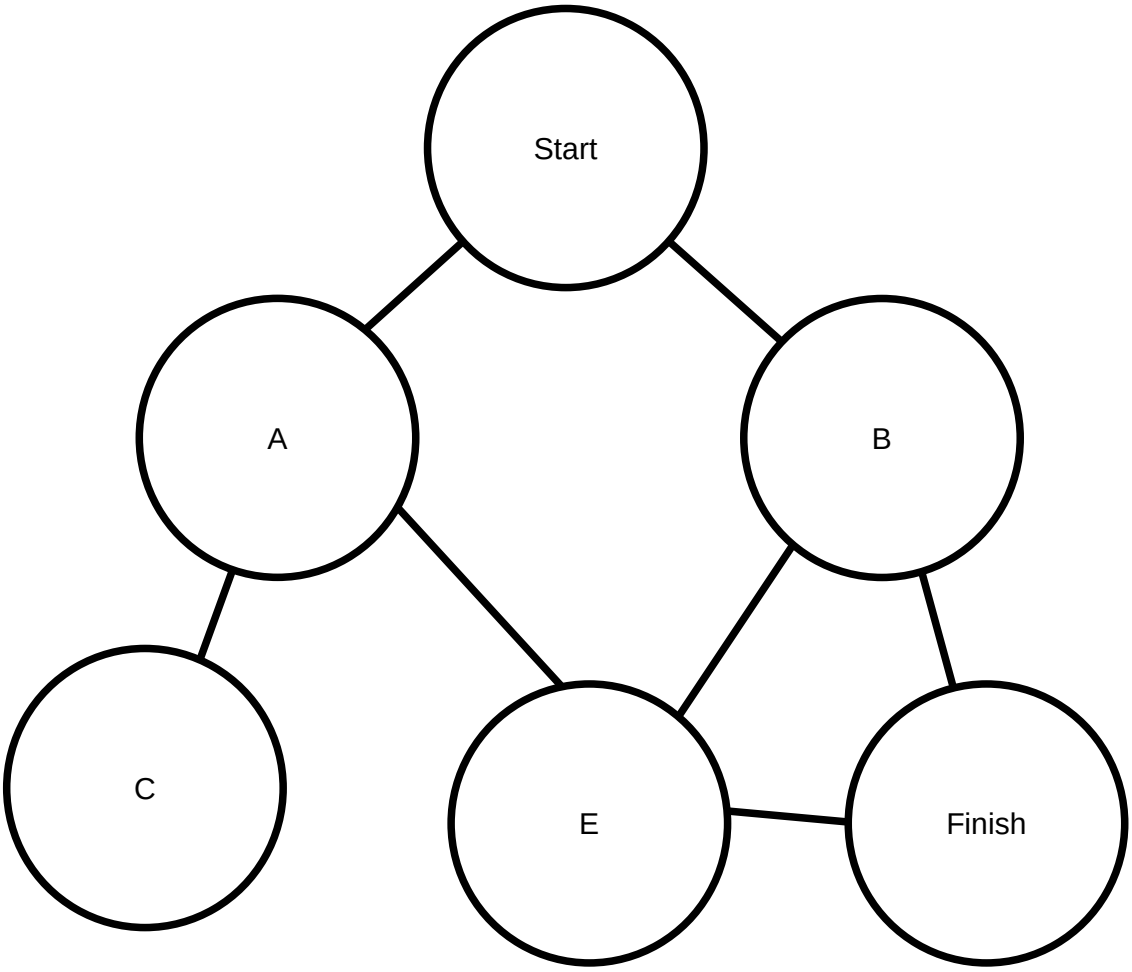
Quiz

1. What is the length of the longest possible search order (from start to finish) that Breadth First Search could take in this graph?



2. Thinking about the longest possible search that Breadth First Search could take in the graph from question 1, in what order are the objects visited by Breadth First Search?
3. What is the length of the shortest possible search that Breadth First Search could take in the graph from question 1?
4. Thinking about the shortest possible search that Breadth First Search could take in the graph from question 1, in what order are the objects visited by Breadth First Search?

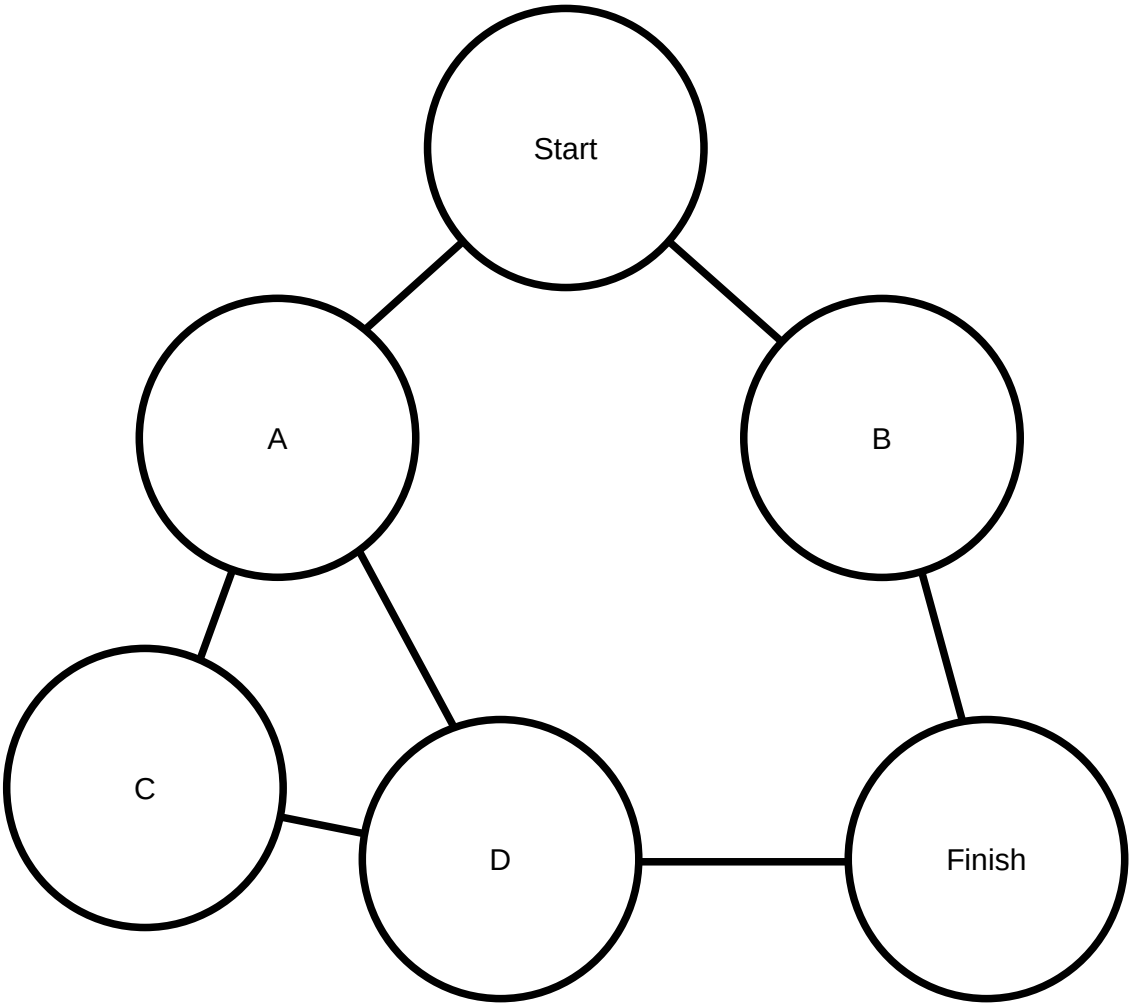
5. What is the length of the longest possible search order (from start to finish) that Breadth First Search could take in this graph?



6. Thinking about the longest possible search that Breadth First Search could take in the graph from question 5, in what order are the objects visited by Breadth First Search?
7. What is the length of the shortest possible search that Breadth First Search could take in the graph from question 5?
8. Thinking about the shortest possible search that Breadth First Search could take in the graph from question 5, in what order are the objects visited by Breadth First Search?
9. Find a way of performing a Breadth First Search that tries visiting object “C”. Write down every node that it visits, in order.

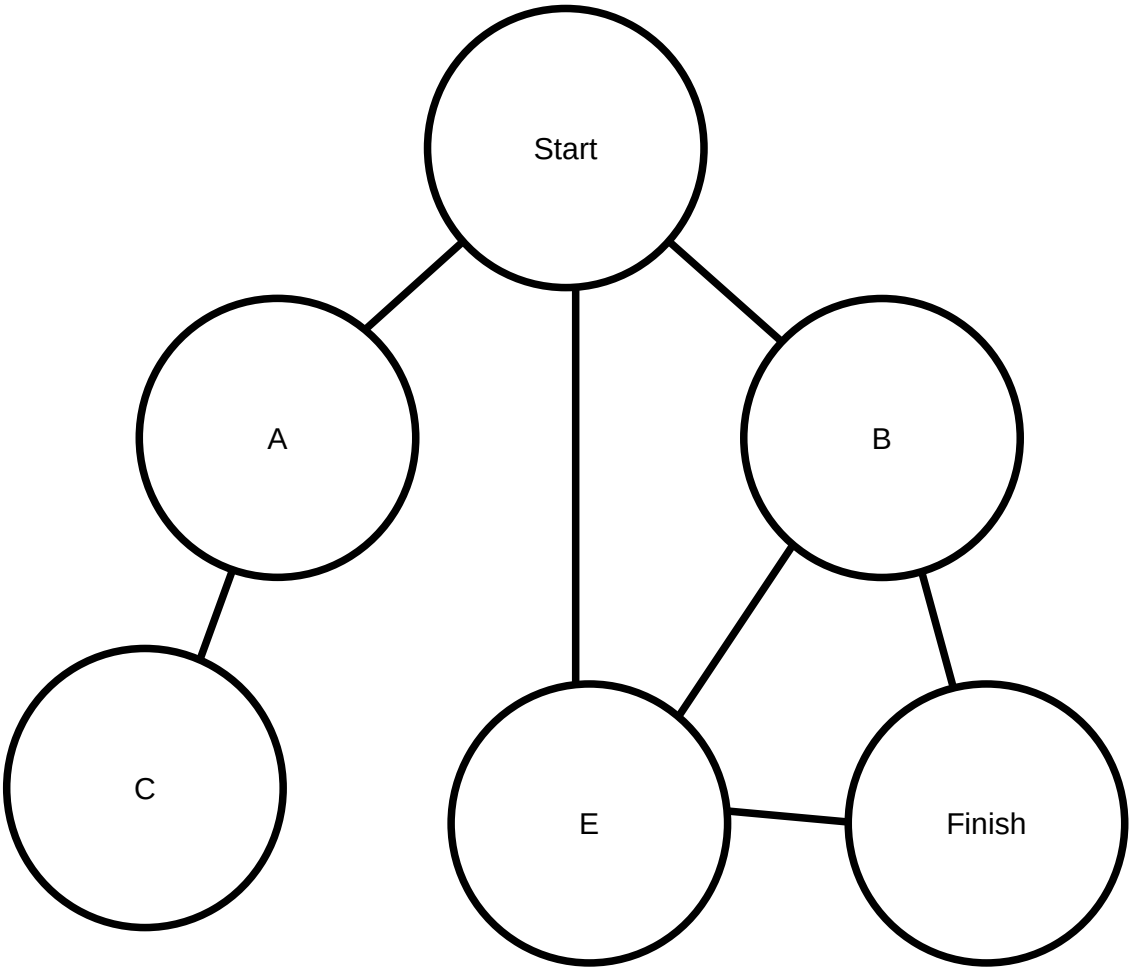
Quiz

1. What is the length of the longest possible search order (from start to finish) that Depth First Search could take in this graph?



2. Thinking about the longest possible search that Depth First Search could take in the graph from question 1, in what order are the objects visited by Depth First Search?
3. What is the length of the shortest possible search that Depth First Search could take in the graph from question 1?
4. Thinking about the shortest possible search that Depth First Search could take in the graph from question 1, in what order are the objects visited by Depth First Search?

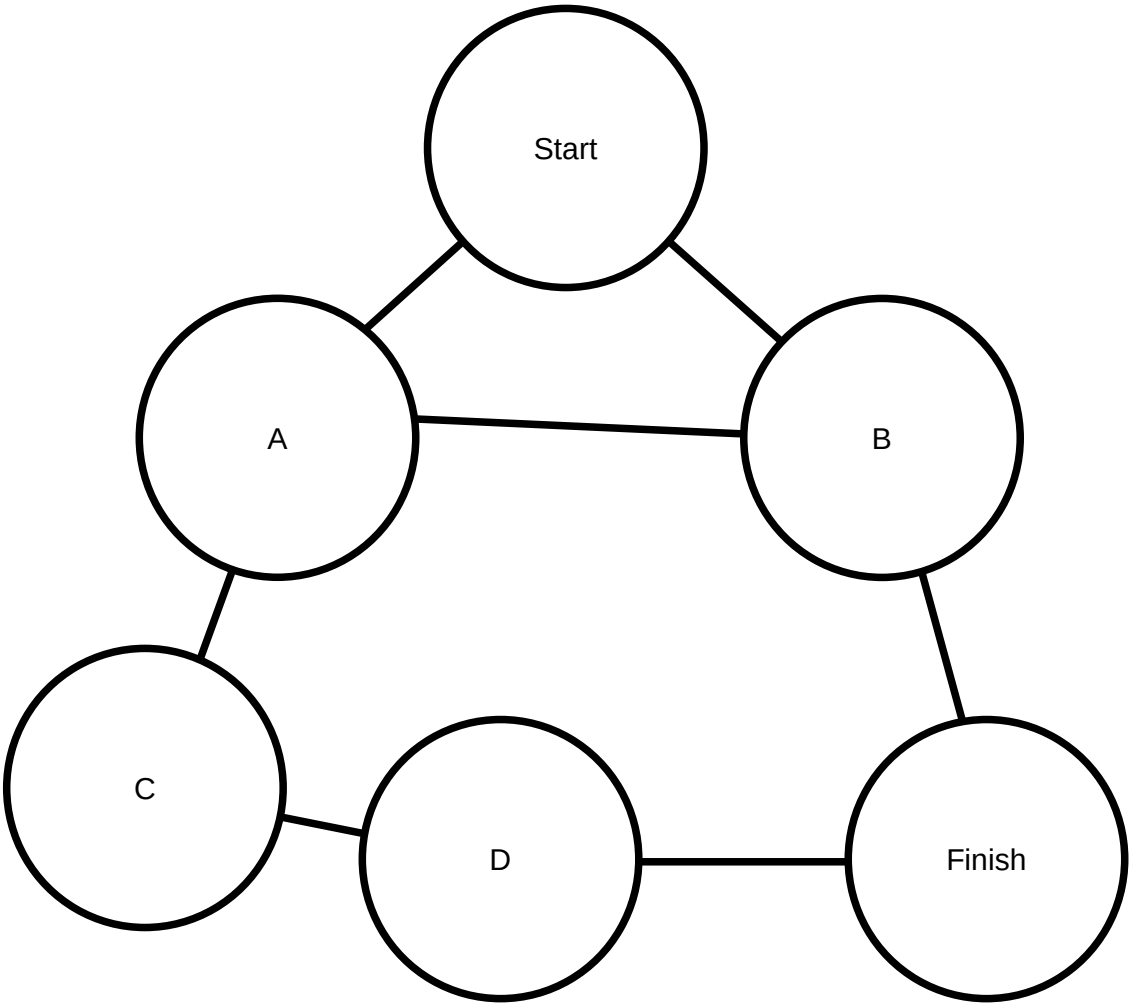
5. What is the length of the longest possible search order (from start to finish) that Depth First Search could take in this graph?



6. Thinking about the longest possible search that Depth First Search could take in the graph from question 5, in what order are the objects visited by Depth First Search?
7. What is the length of the shortest possible search that Depth First Search could take in the graph from question 5?
8. Thinking about the shortest possible search that Depth First Search could take in the graph from question 5, in what order are the objects visited by Depth First Search?
9. Find a way of performing a Depth First Search that tries visiting object “C”. Write down every node that it visits, in order.

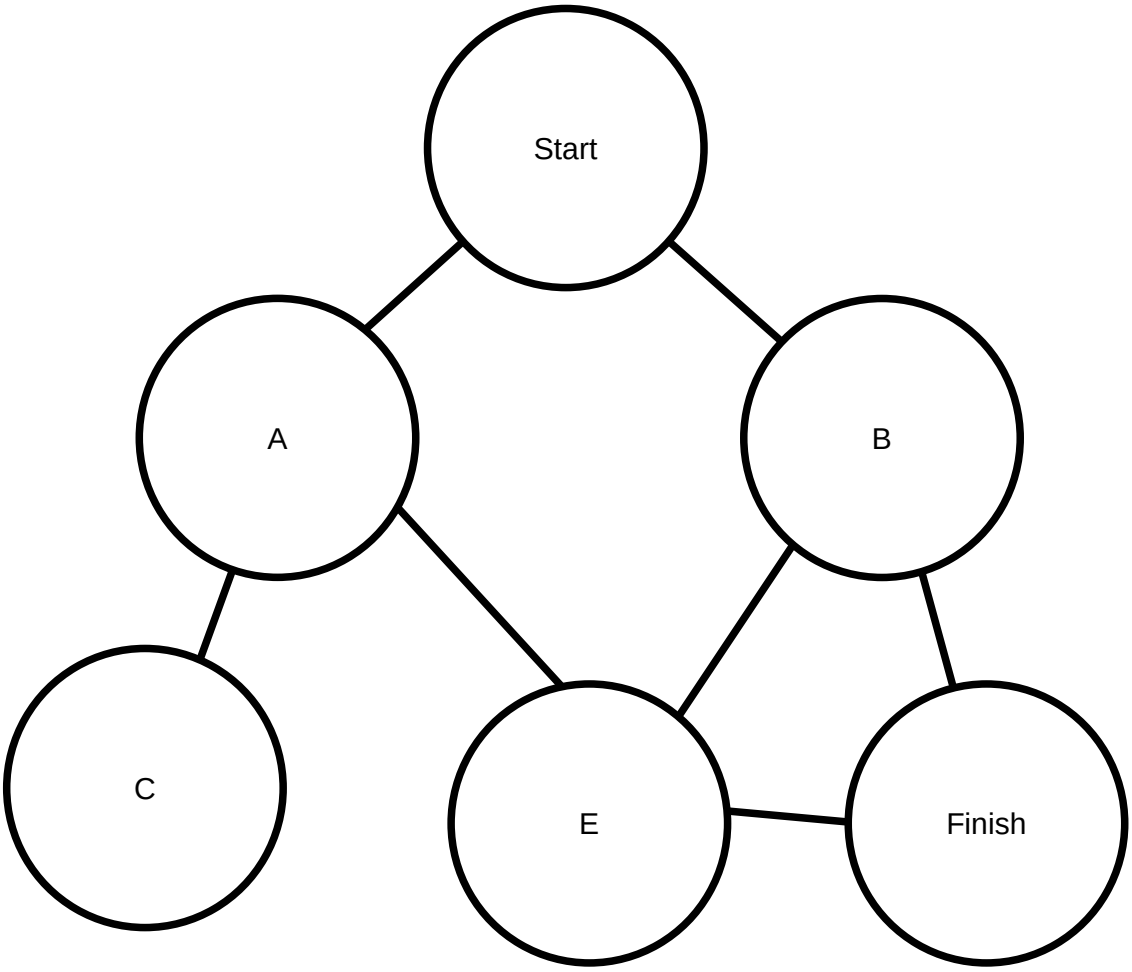
Quiz

1. What is the length of the longest possible search order (from start to finish) that Depth First Search could take in this graph?



2. Thinking about the longest possible search that Depth First Search could take in the graph from question 1, in what order are the objects visited by Depth First Search?
3. What is the length of the shortest possible search that Depth First Search could take in the graph from question 1?
4. Thinking about the shortest possible search that Depth First Search could take in the graph from question 1, in what order are the objects visited by Depth First Search?

5. What is the length of the longest possible search order (from start to finish) that Depth First Search could take in this graph?



6. Thinking about the longest possible search that Depth First Search could take in the graph from question 5, in what order are the objects visited by Depth First Search?
7. What is the length of the shortest possible search that Depth First Search could take in the graph from question 5?
8. Thinking about the shortest possible search that Depth First Search could take in the graph from question 5, in what order are the objects visited by Depth First Search?
9. Find a way of performing a Depth First Search that tries visiting object “C”. Write down every node that it visits, in order.

Questionnaire

This questionnaire is designed to help us understand your experience of the learning aid and how the different formats compare.

Please enter the number of marks you think you got correct in the quiz. Assume that you can get partial credit. There is one mark per question.

I think I received _____ marks in the quiz.

Now, for each of the following items, please indicate your agreement on a scale of 0 to 100 by entering the number that most accurately represents your agreement.

0 10 20 30 40 50 60 70 80 90 100

Do not agree at all Neutral Completely agree

Please think about your overall experience of completing the **quiz only**.

1. I put a lot of effort into this: _____/100
2. After working at this activity for a while, I felt pretty competent: _____/100
3. I didn't try very hard to do well at this activity: _____/100
4. I am satisfied with my performance at this task: _____/100
5. I tried very hard on this activity: _____/100
6. This was an activity I couldn't do very well: _____/100
7. It was important to me to do well at this task: _____/100
8. I didn't put much energy into this: _____/100
9. I think I did pretty well at this activity, compared to other participants: _____/100
10. I was pretty skilled at this activity: _____/100
11. I think I am pretty good at this activity: _____/100

Participant #: _____ Condition: _____/_____/_____

Hardware Worksheet

You can write on this worksheet if you want to, but you don’t have to.

The purpose of this worksheet is to help you understand how to use the learning aid. There is no time limit.

If you have any questions the researcher will be happy to help you.

This learning aid will help you to understand a specific process used in graph theory. It will do this by showing you what the process looks like.

The learning aid consists of 7 nodes. Each node represents one object from a graph.

To see the process, you need to enter the graph into the learning aid. To do this, enter each pair of connected objects. To enter a pair of connected objects, press one of the buttons at the edge of one of the nodes you want to connect, then press a button at the edge of the other node in the pair.

There are also small lights around the edge of each node, which tell you where the connection leads. If the colour at the edge of two nodes are the same, they are connected.

You can disconnect a pair of nodes by pressing the button next to the light.

Hardware Worksheet

You can write on this worksheet if you want to, but you don’t have to.

The purpose of this worksheet is to help you understand how to use the learning aid. There is no time limit.

If you have any questions the researcher will be happy to help you.

This learning aid will help you to understand a specific process used in graph theory. It will do this by showing you what the process looks like.

The learning aid consists of 7 nodes. Each node represents one object from a graph.

To see the process, you need to build the graph using the learning aid’s components. To do this, attach each pair of connected nodes using a cable. The nodes will light up in matching colours when the connection is made.

You can disconnect a pair of nodes by removing the cable.

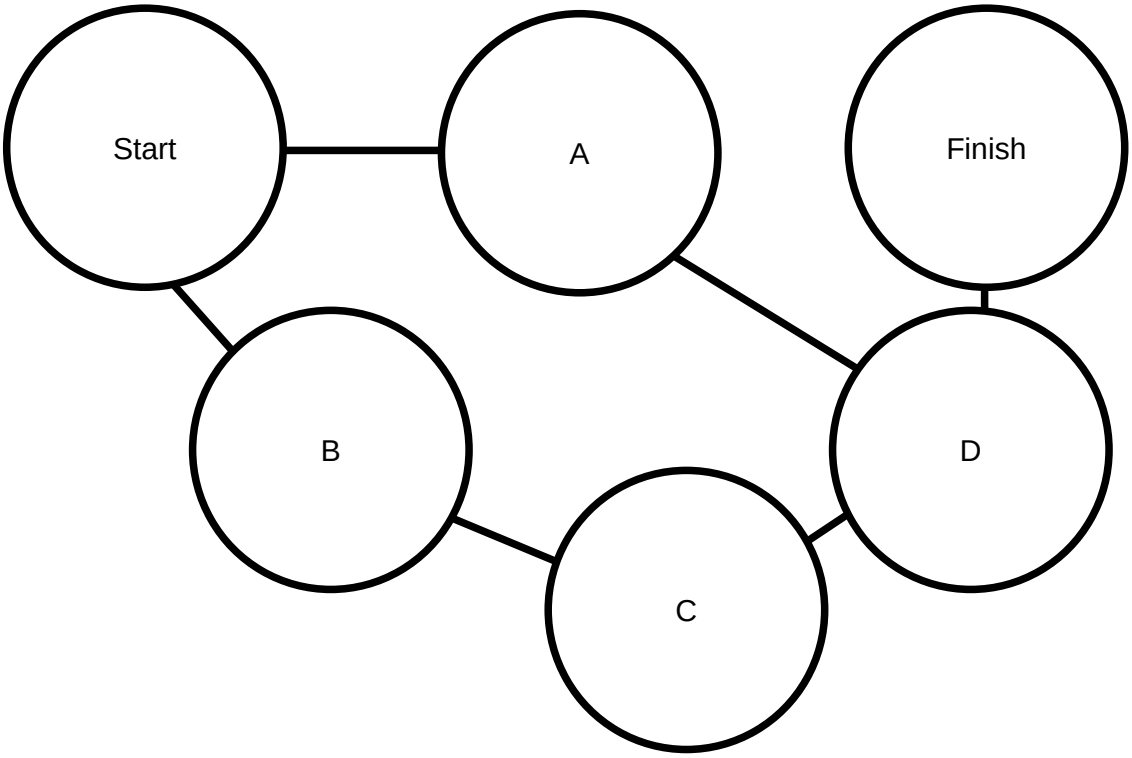
Breadth First Search Worksheet

You can write on this worksheet if you want to, but you don't have to.

The purpose of this worksheet is to help you understand a process called breadth first search. There is no time limit.

If you have any questions the researcher will be happy to help you.

1. Set up the graph shown below. Press the "next step" (centre) button repeatedly to see each step of the Breadth First Search.



   Where does it start? When does it stop?

2. Now try it a few more times. Does it always take the same route?

   The current object is blue. Objects turn white when they have been visited. The objects that are waiting to be selected turn yellow, and you can press the centre button on these to select them (if possible – they have to be visited in a valid order). If the search is successful, the finishing object turns red to indicate that the search is finished.

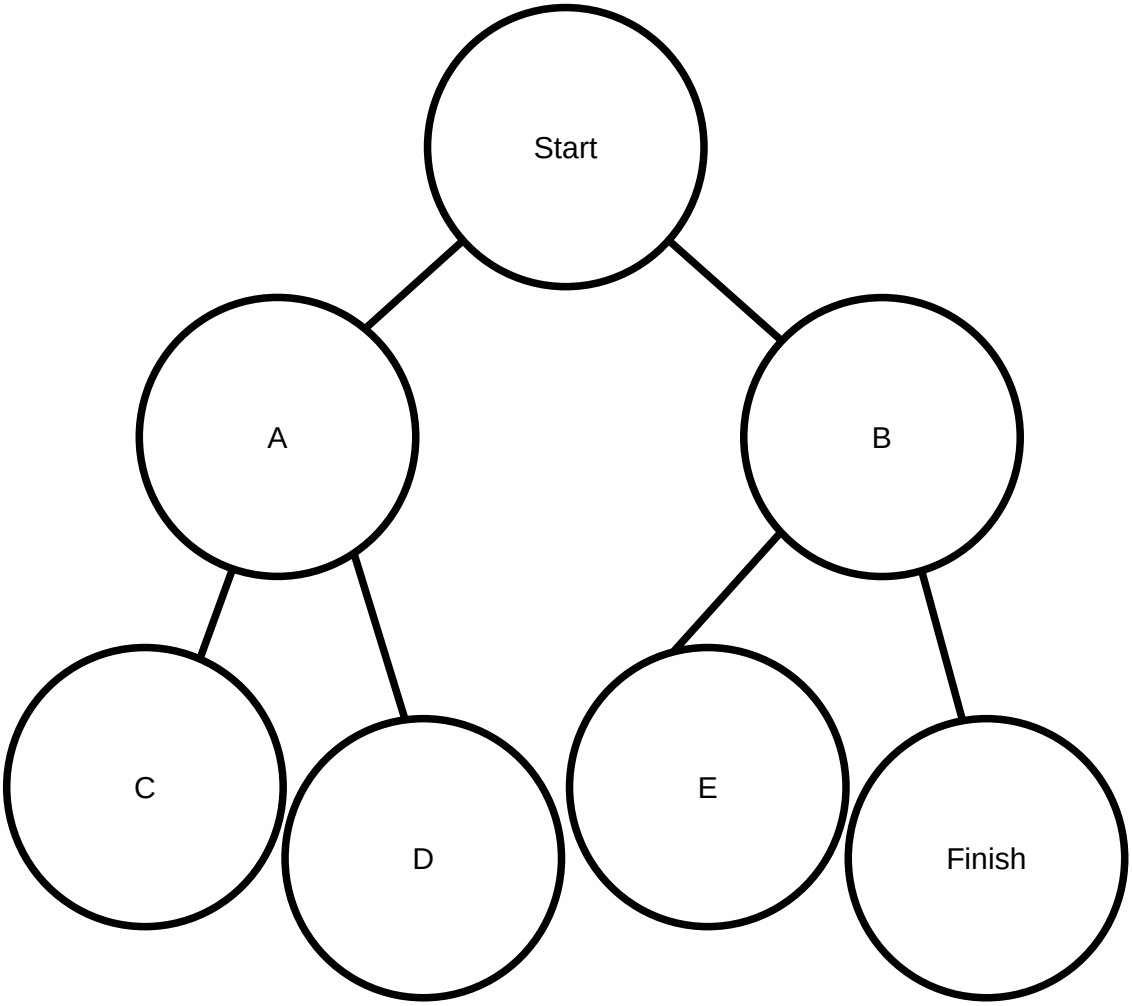
3. The Breadth First Search follows these rules:

   1. Create a "queue" containing the start object only.
   2. Take the object at the front of the queue. If you have been there before, ignore it and keep taking objects from the front of the queue. When you find one you have not been to before, remove it from the queue, go there, and continue to step 3.

Practically, this means to cross off objects from the start of the queue until you find a new one.

3. Check if the current object is the finish. If it is, stop.
4. Find all the objects connected to the current one. Add all of them to the back of the queue (in any order).
5. Repeat from step 2.

Try a Breadth First Search yourself on this graph:



You can check your answers by pressing the buttons on the learning aid, once you have set it up to match the diagram.

4. Find a way of doing a Breadth First Search on the previous graph, so that you visit every object. That is, you should go to every other node before reaching the finish.

   Hint: every time you add objects to the queue, try to put the ones furthest from the finish first.

5. Find a way of doing a Breadth First Search on the previous graph, so that you visit only four objects.
6. Design and set up a graph with six objects, so that every possible Breadth First Traversal takes the same number of steps.
7. Now find a different graph with six objects, so that every possible Breadth First Traversal takes the same number of steps.
8. Now find a graph with six objects, so that **not** every possible Breadth First Traversal takes the same number of steps.
9. Set up the graph you just designed. Try all possible Breadth First Traversals using the learning aid.

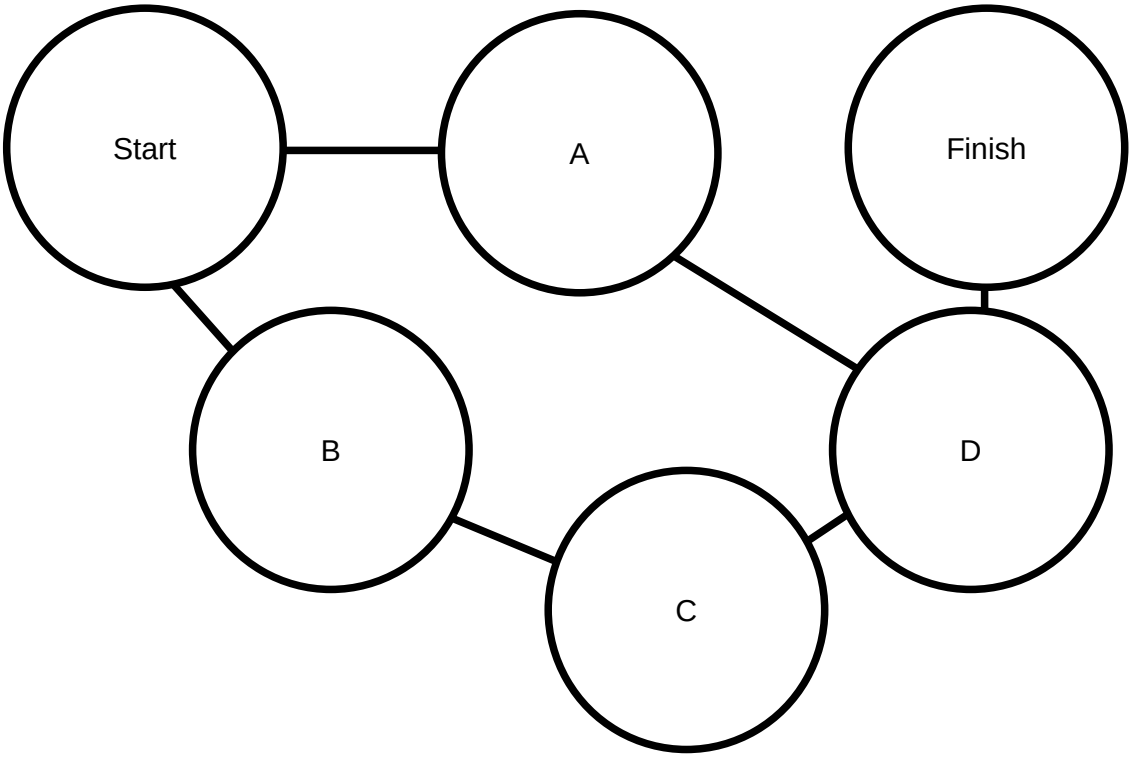
Depth First Search Worksheet

You can write on this worksheet if you want to, but you don't have to.

The purpose of this worksheet is to help you understand a process called depth first search. There is no time limit.

If you have any questions the researcher will be happy to help you.

1. Set up the graph shown below. Press the "next step" (centre) button on the start node repeatedly to see each step of the Depth First Search.



   Where does it start? When does it stop?

2. Now try it a few more times. Does it always take the same route?

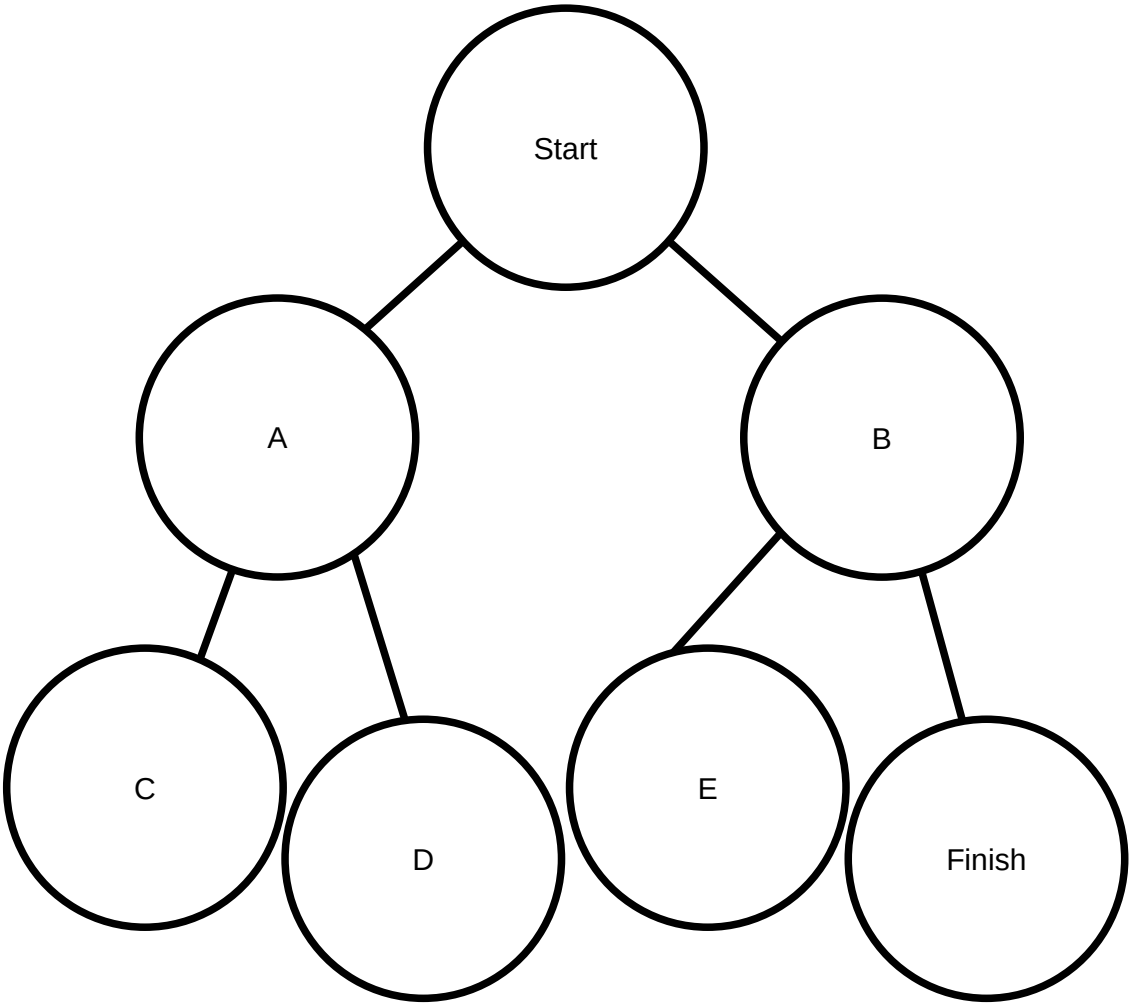
   The current object is blue. Objects turn white when they are on the route from the start to the current node. Objects that might be selected in the next step turn yellow, and you can press the centre button on these to select them. If the search is successful, the finishing object turns red to indicate that the search is finished.

3. The Depth First Search follows these rules:

   1. Start at the start
   2. Check if the current object is the finish. If it is, stop.

3. Go to a random object connected to this one, that you have not already visited on the path you took to get to it. If there are no more, go back the way you came.
4. Repeat from step 2.

Try a Depth First Search yourself on this graph:



You can check your answers by pressing the buttons on the learning aid, once you have set it up to match the diagram.

4. Find a way of doing a Depth First Search on the previous graph, so that you visit every object. That is, you should go to every other node before reaching the finish.

   Hint: every time you are making a random choice, try to go away from the finish.

5. Find a way of doing a Depth First Search on the previous graph, so that you visit only three objects.
6. Design and set up a graph with six objects, so that every possible Depth First Traversal takes the same number of steps.

7. Now find a different graph with six objects, so that every possible Depth First Traversal takes the same number of steps.
8. Set up the graph you just designed. Try all possible Depth First Traversals using the learning aid.

Debriefing Information

Thank you for taking part in this project which has been investigating how different types of learning aid influence learning and confidence.

If you would like to speak to us about the project please get in touch.

Name of Researcher: Penn Mackintosh
Contact details of Researcher: pm2022@bath.ac.uk

Or you can contact the project supervisor:

Name of Supervisor: Ollie Hanton
Contact details of Supervisor: oph33@bath.ac.uk
Address: Department of Computer Science, University of Bath, Claverton Down Bath, BA2 7AY

If you have any concerns about the ethics of this research study, please contact the Department Research Ethics Officer for Computer Science, Dr. Christopher Clarke (cjc234@bath.ac.uk).

B.2.2 Email. The following emails were sent to potential participants by email as part of the recruitment process.

We invite you to take part in a research study investigating how different types of learning aid influence learning and confidence.
This study aims to understand whether being involved in **building a learning aid** affects people's **motivation, understanding and confidence**.

Ethics code: 13029-17026
Title of study: The impact of the IKEA Effect on learning abstract skills

As a participant you will:
Learn about simple graph theory, a widely applicable branch of computer science and mathematics;
Try two different ways of using learning aids;
Complete **quizzes** designed to test your understanding; and
Complete **questionnaires**

Your participation will help advance research on education, especially ways of influencing learner confidence and motivation. You may also gain an improved understanding of basic graph theory.

Location: University of Bath (Claverton Campus)

Eligibility:
You must be:
- Aged 18 or over
- Fluent in English

You cannot participate if you:
- Have significantly reduced upper body strength or fine motor skills
- Are already familiar with graph theory

Duration: **~1 hour**
Booking link: Participant signup – Fill out form
Researcher: Penn Mackintosh – pm2022@bath.ac.uk
Supervisor: Ollie Hanton – oph33@bath.ac.uk

Thank you for your interest in taking part in a research study investigating how different types of learning aid influence learning and confidence.
This study aims to understand whether being involved in **building a learning aid** affects people's **motivation, understanding and confidence**.

Please read the details of the study below, and continue to book a slot if you would like to take part.

Ethics code: 13029-17026
Title of study: The impact of the IKEA Effect on learning abstract skills

As a participant you will:
Learn about simple graph theory, a widely applicable branch of computer science and mathematics;
Try two different ways of using learning aids;
Complete **quizzes** designed to test your understanding; and
Complete **questionnaires**

Your participation will help advance research on education, especially ways of influencing learner confidence and motivation. You may also gain an improved understanding of basic graph theory.

Location: University of Bath (Claverton Campus)

Eligibility:
You must be:
- Aged 18 or over
- Fluent in English

You cannot participate if you:
- Have significantly reduced upper body strength or fine motor skills
- Are already familiar with graph theory

Duration: **~1 hour**
Booking link: Participant signup – Fill out form
Researcher: Penn Mackintosh – pm2022@bath.ac.uk
Supervisor: Ollie Hanton – oph33@bath.ac.uk

B.3 Data

Raw data is available alongside this PDF, as well as the JASP files used for analysis showing statistical test results.

C Software

C.1 Host program

Listing 3: Host control code

```
import asyncio
import collections
import itertools
import multiprocessing
import os
import random
import struct
import sys
import time
import queue
from typing import Optional
import logging
logging.basicConfig(level=logging.DEBUG)

import networkx
import matplotlib.pyplot as plt

transport_logger = logging.getLogger("transport")
transport_logger.setLevel(logging.INFO)
message_logger = logging.getLogger("message")
algorithm_logger = logging.getLogger("algorithm")
processing_logger = logging.getLogger("processing")


YIELD_TIMEOUT = 1000000 # ns == 1 ms
NEIGHBOUR_TIMEOUT = 3000000000 # ns = 2000ms
SENDER_OPTIONS = [0x9, 0xA, 0xB, 0xC, 0xD, 0xE, 0xF]
BUTTON_DEBOUNCE = 3000000000 # ns = 3000ms

BUTTON_MAP = { # (node, logical): phy
  (1, 0): 4, (1, 1): 0, (1, 2): 3, (1, 3): 2,
      (1, 4): 1, (1, 5): 5,
  (2, 0): 2, (2, 1): 3, (2, 2): 0, (2, 3): 1,
      (2, 4): 5, (2, 5): 4,
  (3, 0): 4, (3, 1): 0, (3, 2): 2, (3, 3): 1,
      (3, 4): 3, (3, 5): 5,
  (4, 0): 0, (4, 1): 5, (4, 2): 2, (4, 3): 4,
      (4, 4): 1, (4, 5): 3,
  (5, 0): 2, (5, 1): 3, (5, 2): 1, (5, 3): 0,
      (5, 4): 5, (5, 5): 4,
  (6, 0): 3, (6, 1): 5, (6, 2): 2, (6, 3): 1,
      (6, 4): 0, (6, 5): 4,
  (7, 0): 1, (7, 1): 4, (7, 2): 0, (7, 3): 3,
      (7, 4): 2, (7, 5): 5,
  (8, 0): 1, (8, 1): 0, (8, 2): 2, (8, 3): 3,
      (8, 4): 4, (8, 5): 5,
}
LED_MAP = { # (node, logical): phy
  (1, 0): 4, (1, 1): 3, (1, 2): 5, (1, 3): 1,
      (1, 4): 2, (1, 5): 0,
  (2, 0): 5, (2, 1): 3, (2, 2): 4, (2, 3): 2,
      (2, 4): 0, (2, 5): 1,
  (3, 0): 4, (3, 1): 3, (3, 2): 5, (3, 3): 1,
      (3, 4): 2, (3, 5): 0,
  (4, 0): 4, (4, 1): 2, (4, 2): 3, (4, 3): 0,
      (4, 4): 5, (4, 5): 1,
  (5, 0): 2, (5, 1): 4, (5, 2): 0, (5, 3): 5,
      (5, 4): 1, (5, 5): 3,
  (6, 0): 4, (6, 1): 0, (6, 2): 2, (6, 3): 5,
      (6, 4): 3, (6, 5): 1,
  (7, 0): 4, (7, 1): 0, (7, 2): 5, (7, 3): 2,
      (7, 4): 3, (7, 5): 1,
  (8, 0): 4, (8, 1): 5, (8, 2): 2, (8, 3): 0,
      (8, 4): 3, (8, 5): 1,
}

last_sender_by_led = {}
last_led_by_sender = {}

def prepare_packet(d):
return b"\xFF\xFF\xFF\xFF\x00\x00\x00\x00" +
    struct.pack("<I", d)


def _get_next_sender(message, sender_index):
if message & 0b1000 and not message & 0b100:
recipient = (message & 0x00F00000) >> 20
index = (message & 0x000F0000) >> 16
last_sender = last_sender_by_led.get((recipient,
     index), None)
if last_sender is not None and
    last_led_by_sender.get(last_sender, (None,
    None, False))[:2] == (recipient, index):
transport_logger.debug("reusing led sender %d
    for %d,%d", last_sender, recipient, index)
last_led_by_sender[last_sender] = (*
    last_led_by_sender[last_sender][:2], False)
return last_sender, sender_index

while not last_led_by_sender.get(SENDER_OPTIONS[
    sender_index], (0, 0, True))[2]:
last_led_by_sender[SENDER_OPTIONS[sender_index]]
     = (*last_led_by_sender[SENDER_OPTIONS[
    sender_index]][:2], True)
sender_index = (sender_index + 1) % len(
    SENDER_OPTIONS)

if message & 0b1000 and not message & 0b100:
transport_logger.debug("using new led sender %d
    for %d,%d", SENDER_OPTIONS[sender_index],
    recipient, index)
```

```
last_led_by_sender[SENDER_OPTIONS[sender_index]]
    = (recipient, index, False)
last_sender_by_led[recipient, index] =
    SENDER_OPTIONS[sender_index]
else:
last_led_by_sender.pop(SENDER_OPTIONS[
    sender_index], None)
return SENDER_OPTIONS[sender_index], (
    sender_index + 1) % len(SENDER_OPTIONS)

async def transceive_loop(raw_in_queue,
    raw_out_queue, data_in_queue, data_out_queue
    ):
last_inbound = time.monotonic_ns()
sender_index = 0
while True:
p = b""
while len(p) < 4:
c = await raw_in_queue.get()
p += c
n, = struct.unpack("<I", p)
transport_logger.debug("<%#010x", n)
log_message(n)
parity = n.bit_count() % 2
if parity:
# successfully yielded control or just
    reinitialised
last_inbound = time.monotonic_ns()
if parity:
await data_in_queue.put(n)
if n == 0 or (n & 2 and parity):
forwardable = True
try:
msg = data_out_queue.get_nowait()
except asyncio.QueueEmpty:
msg = 0x00000002
forwardable = False
if last_inbound < time.monotonic_ns() -
    YIELD_TIMEOUT // 2:
msg |= 0x00000002
msg &= ~0x0F000000
if forwardable:
sender, sender_index = _get_next_sender(msg,
    sender_index)
msg |= sender << 24
if msg.bit_count() % 2 == 0:
msg ^= 0b1
transport_logger.debug(">%#010x", msg)
log_message(msg)
raw_out_queue.put_nowait(prepare_packet(msg))

def log_message(n):
message_type = n & 0b1100
link_sender = n >> 28
remote_sender = (n >> 24) & 0xF
if not remote_sender:
return
if message_type == 0b0000:
# update neighbours
neighbours = ((n >> 4) & 0xF, (n >> 8) & 0xF, (n
    >> 12) & 0xF, (n >> 16) & 0xF, (n >> 20) &
    0xF)
message_logger.debug("%x: %x: neighbours: %x, %x
    , %x, %x, %x", link_sender, remote_sender, *
    neighbours)
elif message_type == 0b0100:
buttons = ((n >> 16) & 0b1, (n >> 17) & 0b1, (n
    >> 18) & 0b1, (n >> 19) & 0b1, (n >> 20) & 0
    b1, (n >> 21) & 0b1,
(n >> 22) & 0b1, (n >> 23) & 0b1)
message_logger.debug("%x: %x: buttons: %x, %x, %
    x, %x, %x, %x (%x, %x)", link_sender,
    remote_sender, *buttons)
else:
recipient = (n >> 20) & 0xF
led = (n >> 16) & 0xF
rgb = (n >> 4) & 0xFFF
reset = n & 0b0100
message_logger.debug("%x: %x -> %x: led %d :=
    %03X%s", link_sender, remote_sender,
    recipient, led, rgb, " and blank others" if
    reset else "")



neighbours_state = networkx.DiGraph()
connect_with_buttons = False
# edges contain a key of each node, which
    contains the logical pin number, as well as
    a key "colour"
buttons_graph = networkx.Graph()
EDGE_COLOURS = frozenset({0xF0F, 0xF00, 0x0F0, 0
    x00F, 0xFF0, 0x0FF, 0xFFF, 0x8FF})
pairing_node_pin: tuple[int, int, int, asyncio.
    Task] | None = None

async def update_neighbours(link_sender,
    remote_sender, neighbours):
current_time = time.monotonic_ns()
try:
neighbours_state.remove_edges_from(list(
    neighbours_state.out_edges(remote_sender)))
except networkx.NetworkXError:
pass
for i, neighbour in enumerate(neighbours):
if neighbour != 0 and neighbour != 0xF:
neighbours_state.add_edge(remote_sender,
    neighbour, time=current_time, pin=i)
neighbours_state.add_edge(0, link_sender, time=
    current_time)

neighbour_leds = {}
```

```
async def update_neighbour_leds_loop(
    data_out_queue):
while True:
if connect_with_buttons:
await asyncio.sleep(1)
continue
graph = get_graph()
possible_colours = set(EDGE_COLOURS)
for c, _ in neighbour_leds.values():
possible_colours.remove(c)
target = set()
current = set(neighbour_leds.keys())
for u, v, d in graph.edges(data=True):
k = (u, d[u]["pin"], v, d[v]["pin"])
target.add(k)
for k in current - target:
del neighbour_leds[k]
data_out_queue.put_nowait(set_led(k[0], k[1] +
    1, 0))
data_out_queue.put_nowait(set_led(k[2], k[3] +
    1, 0))
for k in target - current:
c = possible_colours.pop()
neighbour_leds[k] = c, time.monotonic_ns()
data_out_queue.put_nowait(set_led(k[0], k[1] +
    1, c))
data_out_queue.put_nowait(set_led(k[2], k[3] +
    1, c))
await asyncio.sleep(0.1)


async def refresh_neighbour_leds_loop(
    data_out_queue):
while True:
await asyncio.sleep(2)
if connect_with_buttons:
continue
if data_out_queue.qsize() > 3:
continue
if not neighbour_leds:
continue
# refresh something just for the sake of it
# could use a heap but this is easier
oldest = min(neighbour_leds.items(), key=lambda
    x: x[1][1])
processing_logger.debug("refreshing␣%d:%d␣from␣%
    d", oldest[0][0], oldest[0][1], oldest
    [1][1])
data_out_queue.put_nowait(set_led(oldest[0][2],
    oldest[0][3] + 1, oldest[1][0]))
neighbour_leds[oldest[0]] = oldest[1][0], time.
    monotonic_ns()


async def send_neighbours_mode(data_out_queue):
while True:
data_out_queue.put_nowait(0xFF060008 | (not
    connect_with_buttons) << 4)
await asyncio.sleep(10)


def get_active_neighbours(node):
try:
return get_graph().adj[node]
except KeyError:
return []

def set_led(recipient, led, colour):
return 0xFF000008 | recipient << 20 | colour <<
    4 | LED_MAP[recipient, led] << 16



async def stub_algorithm(start_node,
    data_out_queue):
neighbours = get_active_neighbours(start_node)
data_out_queue.put_nowait(set_led(start_node, 0,
     0x800))
for neighbour in neighbours:
data_out_queue.put_nowait(set_led(neighbour, 0,
    0x0F0))
await asyncio.sleep(3)
for neighbour in neighbours:
data_out_queue.put_nowait(set_led(neighbour, 0,
    0))
data_out_queue.put_nowait(set_led(start_node, 0,
     0))


def get_graph():
if connect_with_buttons:
return buttons_graph
current_time = time.monotonic_ns()
graph = networkx.Graph()
graph.add_nodes_from(neighbours_state)
try:
graph.remove_node(0)
except networkx.NetworkXError:
assert not graph.nodes
return graph
for u, v, d in neighbours_state.out_edges.data()
    :
if u != 0:
graph.add_edge(u, v)
graph.edges[u, v][u] = d
for u, v, d in list(graph.edges.data()):
# ignore if one of the nodes reports the link
    doesn't exist, or if neither node has
    checked in recently
if len(d.keys()) != 2 or (d[u]["time"] <
    current_time - NEIGHBOUR_TIMEOUT and d[v]["
    time"] < current_time - NEIGHBOUR_TIMEOUT):
```

```
graph.remove_edge(u, v)
return graph.subgraph(itertools.chain.
    from_iterable(networkx.
    node_connected_component(graph, n) for n, d
    in neighbours_state.adj[0].items() if d["
    time"] > current_time - NEIGHBOUR_TIMEOUT))

target_node = 8

def algorithm_control_filter(msg):
if msg & 0b1100 == 0b0100:
return True
return False


async def bfs_algorithm(start_node,
    data_out_queue):
graph = get_graph()
pending = collections.deque(({start_node},))
visited = {start_node}
guess = None

while pending and pending[0]:
if guess in pending[0]:
node = guess
else:
node = random.choice(tuple(pending[0]))
pending[0].remove(node)
if not pending[0]:
pending.popleft()
algorithm_logger.info("visit␣%d", node)
if node == target_node:
data_out_queue.put_nowait(set_led(node, 0, 0x0F0
    ))
break
successors = set(graph.adj[node])
successors.difference_update(visited)
pending.append(successors)
visited.update(successors)

data_out_queue.put_nowait(set_led(node, 0, 0x00F
    ))
for successor in successors:
data_out_queue.put_nowait(set_led(successor, 0,
    0xFF0))
guess = await step()

data_out_queue.put_nowait(set_led(node, 0, 0xFFF
    ))
await step()
for node in visited:
data_out_queue.put_nowait(set_led(node, 0, 0))


async def dfs_algorithm(start_node,
    data_out_queue):
graph = get_graph()
path = [(start_node, [])]

#data_out_queue.put_nowait(set_led(target_node,
    0, 0x800))
while path:
node, visited_successors = path[-1]
algorithm_logger.info("visit␣%d", node)
if node == target_node:
data_out_queue.put_nowait(set_led(node, 0, 0x0F0
    ))
break
if node in graph.adj:
successors = set(graph.adj[node])
else:
successors = set()
successors.difference_update(node for node, _ in
     path)
successors.difference_update(visited_successors)

data_out_queue.put_nowait(set_led(node, 0, 0x00F
    ))
for successor in successors:
data_out_queue.put_nowait(set_led(successor, 0,
    0xFF0))
guess = await step()

if not successors:
del path[-1]
data_out_queue.put_nowait(set_led(node, 0, 0))
else:
if guess in successors:
path.append((guess, []))
else:
path.append((random.choice(tuple(successors)),
    []))
visited_successors.append(path[-1][0])
for successor in successors:
if successor != path[-1][0]:
data_out_queue.put_nowait(set_led(successor, 0,
    0))

data_out_queue.put_nowait(set_led(node, 0, 0xFFF
    ))
await step()
for node, _ in path:
data_out_queue.put_nowait(set_led(node, 0, 0))



run_algorithm = dfs_algorithm
autostep = False
step_queue: Optional[asyncio.Queue] = None
running_algorithm: Optional[asyncio.Task] = None
button_times = collections.defaultdict(lambda:
    collections.defaultdict(lambda: 0))

async def step():
```

```
if autostep:
await asyncio.sleep(5)
return None
else:
return await step_queue.get()


async def blink_led(data_out_queue,
    remote_sender, led, colour):
while True:
data_out_queue.put_nowait(set_led(remote_sender,
     led, colour))
await asyncio.sleep(3)
data_out_queue.put_nowait(set_led(remote_sender,
     led, 0))
await asyncio.sleep(3)


async def button_pressed(data_in_queue,
    data_out_queue, link_sender, remote_sender,
    buttons):
global running_algorithm
global step_queue
global pairing_node_pin

current_time = time.monotonic_ns()
for button, state in enumerate(buttons):
if not state:
continue
if button_times[remote_sender][button] +
    BUTTON_DEBOUNCE > current_time:
processing_logger.info("button␣ignored:␣%d␣too␣
    close␣to␣%d", button_times[remote_sender][
    button], current_time)
continue
button_times[remote_sender][button] =
    current_time
global step_queue
if button == 0:
if running_algorithm is not None and not
    running_algorithm.done():
algorithm_logger.info("step␣from␣button")
step_queue.put_nowait(remote_sender)
else:
algorithm_logger.info("start␣from␣button")
step_queue = asyncio.Queue()
running_algorithm = asyncio.create_task(
    run_algorithm(remote_sender, data_out_queue)
    )
else:
if not connect_with_buttons:
return
pin = button - 1
if pairing_node_pin is None:
possible_colours = set(EDGE_COLOURS)
remove_edge = None
for u, v, d in buttons_graph.edges.data():
possible_colours.remove(d["colour"])
if d.get(remote_sender, {}).get("pin") == pin:
remove_edge = u, v, d
break
if remove_edge is not None:
buttons_graph.remove_edge(remove_edge[0],
    remove_edge[1])
processing_logger.info("edge␣removed␣between␣%d
    :%d␣and␣%d:%d", remove_edge[0], remove_edge
    [2][remove_edge[0]]["pin"], remove_edge[1],
    remove_edge[2][remove_edge[1]]["pin"])
data_out_queue.put_nowait(set_led(remove_edge
    [0], remove_edge[2][remove_edge[0]]["pin"] +
     1, 0))
data_out_queue.put_nowait(set_led(remove_edge
    [1], remove_edge[2][remove_edge[1]]["pin"] +
     1, 0))
continue
colour = random.choice(list(possible_colours))
pairing_node_pin = remote_sender, pin, colour,
    asyncio.create_task(blink_led(data_out_queue
    , remote_sender, button, colour))
processing_logger.info("started␣pairing␣%x:%d",
    remote_sender, pin)
else:
processing_logger.info("end␣pairing␣%x:%d",
    remote_sender, pin)
pairing_node_pin[3].cancel()
try:
await pairing_node_pin[3]
except asyncio.CancelledError:
pass

viable = True
if remote_sender == pairing_node_pin[0]:
processing_logger.debug("pairing␣cancelled␣due␣
    to␣self␣loop")
viable = False
if buttons_graph.has_edge(remote_sender,
    pairing_node_pin[0]):
processing_logger.warning("pairing␣cancelled␣due
    ␣to␣duplicate␣edge")
viable = False
if remote_sender in buttons_graph:
for _, _, d in buttons_graph.edges(remote_sender
    , data=True):
if d[remote_sender]["pin"] == pin:
processing_logger.warning("pairing␣cancelled␣due
    ␣to␣pin␣in␣use")
viable = False

if viable:
buttons_graph.add_edge(remote_sender,
    pairing_node_pin[0], colour=pairing_node_pin
    [2])
```

```
buttons_graph.edges[remote_sender,
    pairing_node_pin[0]][pairing_node_pin[0]] =
    {"pin": pairing_node_pin[1]}
buttons_graph.edges[remote_sender,
    pairing_node_pin[0]][remote_sender] = {"pin"
    : pin}
data_out_queue.put_nowait(set_led(remote_sender,
     button, pairing_node_pin[2]))
data_out_queue.put_nowait(set_led(
    pairing_node_pin[0], pairing_node_pin[1] +
    1, pairing_node_pin[2]))
else:
data_out_queue.put_nowait(set_led(
    pairing_node_pin[0], pairing_node_pin[1] +
    1, 0))
processing_logger.info("new␣graph:␣%r",
    buttons_graph)
pairing_node_pin = None


async def handle_message(n, data_in_queue,
    data_out_queue):
message_type = n & 0b1100
if message_type == 0b0000:
# update neighbours
link_sender = n >> 28
remote_sender = (n >> 24) & 0xF
neighbours = ((n >> 4) & 0xF, (n >> 8) & 0xF, (n
     >> 12) & 0xF, (n >> 16) & 0xF, (n >> 20) &
    0xF)
await update_neighbours(link_sender,
    remote_sender, neighbours)
elif message_type == 0b0100:
link_sender = n >> 28
remote_sender = (n >> 24) & 0xF
buttons = ((n >> 16) & 0b1, (n >> 17) & 0b1, (n
    >> 18) & 0b1, (n >> 19) & 0b1, (n >> 20) & 0
    b1, (n >> 21) & 0b1,
(n >> 22) & 0b1, (n >> 23) & 0b1)
mapped_buttons = tuple(buttons[BUTTON_MAP[
    remote_sender, i]] for i in range(6))
await button_pressed(data_in_queue,
    data_out_queue, link_sender, remote_sender,
    mapped_buttons)


async def message_handler(data_in_queue,
    data_out_queue):
while True:
n = await data_in_queue.get()
await handle_message(n, data_in_queue,
    data_out_queue)


def reader(f, raw_in_queue):
raw_in_queue.put_nowait(os.read(f, 1))


async def writer(f, raw_out_queue,
    writer_available):
while True:
await writer_available.wait()
data = await raw_out_queue.get()

writer_available.clear()
os.write(f, data)

raw_out_queue.task_done()

def notify_writer_available(writer_available):
writer_available.set()


def _approach(target_value, old, step):
if abs(target_value - old) < step:
return target_value
if target_value > old:
return old + step
return old - step

def _renderer(q):
plt.ion()
fig = plt.figure()
pos = {}
graph = None
min_x, max_x, min_y, max_y, = -1, 1, -1, 1

while True:
fig.canvas.flush_events()
try:
graph = q.get_nowait()
except queue.Empty:
pass
if graph is not None:
fig.clear()
pos = {n: pos[n] for n in graph.nodes() if n in
    pos}
pos = networkx.arf_layout(graph, pos=pos,
    max_iter=1)
step_ratio = 100
if pos:
target_min_x = min(x for (x, y) in pos.values())
     - .1
target_max_x = max(x for (x, y) in pos.values())
     + .1
target_min_y = min(y for (x, y) in pos.values())
     - .1
target_max_y = max(y for (x, y) in pos.values())
     + .1
x_step = (target_max_x - target_min_x) /
    step_ratio
y_step = (target_max_y - target_min_y) /
    step_ratio
min_x = _approach(target_min_x, min_x, x_step)
max_x = _approach(target_max_x, max_x, x_step)
```

```
min_y = _approach(target_min_y, min_y, y_step)
max_y = _approach(target_max_y, max_y, y_step)
ax = fig.add_subplot()
ax.set_xlim(min_x, max_x)
ax.set_ylim(min_y, max_y)
labels = {n: str(n) for n in graph.nodes()}
networkx.draw(graph, pos=pos, ax=ax, labels=
    labels)
plt.draw()
plt.show(block=False)
fig.canvas.draw_idle()

async def renderer():
q = multiprocessing.Queue(1)
p = multiprocessing.Process(target=_renderer,
    args=(q,))
p.start()
try:
while True:
await asyncio.to_thread(q.put, get_graph())
await asyncio.sleep(0.01)
finally:
p.terminate()


async def main(file):
raw_in_queue = asyncio.Queue()
raw_out_queue = asyncio.Queue()
data_in_queue = asyncio.Queue()
data_out_queue = asyncio.Queue()

tasks = []
if file == "":
f = None
else:
f = os.open(sys.argv[1], os.O_RDWR)
try:
if f is not None:
writer_available = asyncio.Event()
asyncio.get_running_loop().add_reader(f, reader,
     f, raw_in_queue)
asyncio.get_running_loop().add_writer(f,
    notify_writer_available, writer_available)
tasks.append(writer(f, raw_out_queue,
    writer_available))
else:
tasks.append(pump_fake_data(raw_in_queue,
    raw_out_queue))

tasks += [
transceive_loop(raw_in_queue, raw_out_queue,
    data_in_queue, data_out_queue),
message_handler(data_in_queue, data_out_queue),
renderer(),
update_neighbour_leds_loop(data_out_queue),
refresh_neighbour_leds_loop(data_out_queue),
send_neighbours_mode(data_out_queue),
]
await asyncio.wait((asyncio.create_task(task)
    for task in tasks))
finally:
if f is not None:
asyncio.get_running_loop().remove_reader(f)
asyncio.get_running_loop().remove_writer(f)
os.close(f)

async def pump_fake_data(raw_in_queue,
    raw_out_queue):
d = b''
raw_in_queue.put_nowait(b"\x00\x00\x00\x00")
while True:
m = await raw_out_queue.get()
await asyncio.sleep(0.01)
d += m
try:
di = d.index(b"\xFF\xFF\xFF\xFF\x00\x00\x00\x00"
    )
except ValueError:
pass
else:
if len(d) >= di + 12:
m = d[di+8:di+12]
d = d[di+12:]
n, = struct.unpack("<I", m)
if n.bit_count() % 2:
if n & 0b10:
if random.random() < 0.1:
raw_in_queue.put_nowait(b"\x05\x00\x01\x11")
raw_in_queue.put_nowait(b"\x00\x02\x00\x11")
raw_in_queue.put_nowait(b"\x03\x01\x00\x12")
else:
raw_in_queue.put_nowait(b"\x00\x00\x00\x00")
raw_out_queue.task_done()


if __name__ == "__main__":
asyncio.run(main(sys.argv[1]))
```

C.2 Peripheral program

Listing 4: Arduino control code

```
#define THROW_ERROR_IF_NOT_FAST 1

#include <digitalWriteFast.h>
#include <Adafruit_NeoPixel.h>
#include <EEPROM.h>


#define DEBUG 0
#define SINGLE_PIN 1


#define FLOATING_PIN A0
```

```
#if SINGLE_PIN
#define PIN_COUNT 1
#define COMPUTE_MICROS_GREEDY 500
#define COMPUTE_MICROS_OPTIMISTIC 500
#else
#define PIN_COUNT 5
#define COMPUTE_MICROS_GREEDY 250
#define COMPUTE_MICROS_OPTIMISTIC 125
#endif

#if DEBUG
#define CLOCK_MICROS 7000 // time between clock
    cycles on a given pin
#define CLOCK_TOLERANCE 700 // how early are we
    allowed to run
#define COMPUTE_MICROS_GREEDY 6000
#define COMPUTE_MICROS_OPTIMISTIC 6000
#define STARTUP_MICROS 2000
#define TIME_TYPE uint32_t
#define TIME_TYPE_MAX ((TIME_TYPE)0xFFFFFFFF)
#define TIME_JUST_OVERFLOWED ((TIME_TYPE)0
    x40000000)
#define TIME_MIGHT_OVERFLOW ((TIME_TYPE)0
    xC0000000)
#define CONNECTION_TIMEOUT 5000000
#define FORWARD_CUTOFF 1000000
#define BUTTON_CUTOFF 500000
#else
#define CLOCK_MICROS 550 // time between clock
    cycles on a given pin
#define CLOCK_TOLERANCE 150 // how early are we
    allowed to run
#define STARTUP_MICROS 2000
#define TIME_TYPE uint16_t
#define TIME_TYPE_MAX ((TIME_TYPE)0xFFFF)
#define TIME_JUST_OVERFLOWED ((TIME_TYPE)0x4000)
#define TIME_MIGHT_OVERFLOW ((TIME_TYPE)0xC000)
#define CONNECTION_TIMEOUT 600000
#define FORWARD_CUTOFF 1800000
#define BUTTON_CUTOFF 300000
#endif

#if DEBUG && !SINGLE_PIN
#define ERR(reason) { pinModeFast(LED_BUILTIN,
    OUTPUT); Serial.begin(115200); delay(10);
    Serial.println(); Serial.println(reason); }
#define DIE(reason) { ERR(reason); while (1) {
    digitalWriteFast(LED_BUILTIN, LOW);delay
    (100);digitalWriteFast(LED_BUILTIN, HIGH);
    delay(500); } }
#define OUT(msg) Serial.println(msg);
#else
#define ERR(reason) { pinModeFast(LED_BUILTIN,
    OUTPUT); Serial.begin(115200); delay(10);
    Serial.println(); Serial.println(reason); }
#define DIE(reason) { ERR(reason); while (1) {
    digitalWriteFast(LED_BUILTIN, LOW);delay
    (100);digitalWriteFast(LED_BUILTIN, HIGH);
    delay(500); } }
#define OUT(msg) {};
#endif

#define BRIGHTNESS_BITS 1

// 0b00------: Transmitting
// 0b0000----: tx hdr
// 0b0010----: tx 0
// 0b0011----: tx 1
// 0b0001----: tx complete, waiting for next
    window
// 0b10------: Receiving
// 0b10000000: prepare to receive
// 0b10000001: ready to receive
// 0b10------: receiving data
// 0b11------: Startup handshake
uint8_t status[PIN_COUNT] = {0};
uint32_t buffer[PIN_COUNT] = {0};
#define PACKET_SIZE 32 // bit count of tx buffer
uint8_t bitCount[PIN_COUNT] = {0};
TIME_TYPE next_wakeup[PIN_COUNT] = {0};

#define COMMS_PIN_SETUP(pin_index, pin) { \
  status[pin_index] = 0xc0; \
  next_wakeup[pin_index] = (pin_index * (
      CLOCK_MICROS - COMPUTE_MICROS_OPTIMISTIC *
       (PIN_COUNT / 3))) / PIN_COUNT +
      STARTUP_MICROS; \
  OUT(pin_index)\
  OUT(next_wakeup[pin_index])\
}

#define TX_NEXT_BIT(pin_index, bit_count) { \
  status[pin_index] = (((buffer[pin_index] >>
      bit_count) & 1) << 4) | 0b100000; \
}

#define RX_NEXT_BIT(pin_index, bit_count,
    next_status) { \
  if (bit_count == PACKET_SIZE-1) { \
    /* done on this packet */ \
    status[pin_index] = 0x80; \
  } else { \
    status[pin_index] = next_status; \
  } \
}

#define SWITCH_TO_RX(pin_index, pin) { \
  pinModeFast(pin, INPUT_PULLUP); \
  status[pin_index] = 0x80; \
  bitCount[pin_index] = 0; \
  buffer[pin_index] = 0; \
}
```

```
#if DEBUG
#define DELAY_MICROS(t) if (t >= 16383)
    delayMicroseconds(16383); else
    delayMicroseconds(t);
#else
#define DELAY_MICROS(t) delayMicroseconds(t);
#endif

#define COMMS_PIN_LOOP(pin_index, pin) {
                \
  \
  TIME_TYPE target_time = next_wakeup[pin_index
      ];     \
  {
      \
    TIME_TYPE clock_overflow_distance =
        TIME_TYPE_MAX - target_time; \
    if (clock_overflow_distance < CLOCK_MICROS)
        {     \
      /* time has overflowed */
                                  \
      target_time = ((TIME_TYPE)CLOCK_MICROS) -
          clock_overflow_distance - 1; \
    } else {
      \
      target_time += CLOCK_MICROS;
                                \
    }
      \
  };
    \
  next_wakeup[pin_index] = target_time;
                  \
  do {
    \
    TIME_TYPE current_micros = micros();
                  \
    TIME_TYPE time_to_wait;
                                  \
    if (current_micros < TIME_JUST_OVERFLOWED &&
        target_time > TIME_MIGHT_OVERFLOW) { \
      /* time overflowed while we were behind
          schedule, reset rx state */ \
      ERR(pin_index)\
      ERR(current_micros)\
      ERR(status[pin_index])\
      ERR(target_time)\
      DIE("time overflow behind schedule")
                                                \
    } else if (target_time <
        TIME_JUST_OVERFLOWED && current_micros >
         TIME_MIGHT_OVERFLOW) { \
      time_to_wait = TIME_TYPE_MAX -
          current_micros + 1 + target_time; \
    } else if (target_time < current_micros) {
            \
      ERR(pin_index)\
      ERR(current_micros)\
      ERR(status[pin_index])\
      ERR(target_time)\
      DIE("running behind schedule")
                          \
    } else {
      \
      time_to_wait = target_time -
          current_micros;     \
    }
      \
    if (time_to_wait >= COMPUTE_MICROS_GREEDY) {
            \
      compute_loop();
                                          \
    } else if (time_to_wait > CLOCK_TOLERANCE) {
            \
      DELAY_MICROS(time_to_wait -
          CLOCK_TOLERANCE)     \
      break;
        \
    } else {
      \
      break;
        \
    }
      \
  } while (1);
                                                \
  \
  uint8_t pinStatus = status[pin_index]++;
              \
  \
  switch (pinStatus) {
                                      \
    case 0x0c:
      \
    /* almost done sending hdr, move to bit
        state */  \
    TX_NEXT_BIT(pin_index, 0)
                                \
```

```
break;                                                  \
    \
case 0x20:                                              \
    \
case 0x30:                                              \
    \
digitalWriteFast(pin, HIGH);                            \
break;                                                  \
    \
case 0x00:                                              \
    \
ERR("invalid␣null␣state");                              \
break;                                                  \
    \
case 0x01: /* start sending header */                   \
case 0x23: /* start sending "0" */                      \
case 0x33: /* start sending "1" */                      \
digitalWriteFast(pin, LOW);                             \
break;                                                  \
    \
case 0x25:                                              \
    \
/* almost at end of "0" */                              \
/* fallthrough */                                       \
case 0x38: {                                            \
  /* almost at end of "1" */                            \
  /* careful of off-by-one (or two) errors              \
     */                                                 \
  if (bitCount[pin_index] < PACKET_SIZE-1) {            \
    TX_NEXT_BIT(pin_index, ++bitCount[                  \
        pin_index])                                     \
  }                                                     \
  break; }                                              \
    \
case 0x26:                                              \
    \
case 0x39:                                              \
    \
/* done on this packet */                               \
TX_DONE_CB(pin_index, pin);                             \
break;                                                  \
    \
case 0x0f:                                              \
    \
pinModeFast(pin, OUTPUT);                               \
status[pin_index] = 0x11;                               \
/* fallthrough */                                       \
case 0x10:                                              \
    \
/* pre-header delay */                                  \
digitalWriteFast(pin, HIGH);                            \
break;                                                  \
    \
case 0x1b:                                              \
    \
/* almost done waiting, check for data */               \
if (bitCount[pin_index] == 0) {                         \
  /* there is data, send header */                      \
  status[pin_index] = 0x01;                             \
} else {                                                \
  /* check again for data in one cycle time             \
     */                                                 \
  status[pin_index] = 0x1b;                             \
}                                                       \
    \
break;                                                  \
    \
```

```
case 0xc0:

    \
if (random(2)) {              \
  /* try rx first */
                                \
  pinModeFast(pin, INPUT_PULLUP);
                    \
} else {

    \
  /* try tx first */
                                \
  pinModeFast(pin, OUTPUT);
                             \
  digitalWriteFast(pin, LOW);
                           \
  status[pin_index] = 0xd1;
                             \
}

    \
break;

    \
case 0xc1:

    \
if (digitalReadFast(pin) == LOW) {
                    \
  /* if we are on the tail end of a pulse we
       can't tell if it is an ack or a ping
      */ \
  status[pin_index] = 0xc1;
                             \
}

    \
break;

    \
case 0xca:

    \
status[pin_index] = 0xc0;
                             \
/* fallthrough */
                                 \
case 0xc2:

    \
case 0xc3:

    \
case 0xc4:

    \
case 0xc5:

    \
case 0xc6:

    \
case 0xc7:

    \
case 0xc8:

    \
case 0xc9:

    \
if (digitalReadFast(pin) == LOW) {
                    \
  /* if there is someone there, we go low to
       ack */ \
  status[pin_index] = 0xe1;
                               \
  pinModeFast(pin, OUTPUT);
                             \
  digitalWriteFast(pin, LOW);
                           \
}

    \
break;

    \
case 0xd3:

    \
pinModeFast(pin, INPUT_PULLUP);
                      \
/* fallthrough */
                                 \
case 0xd4:

    \
case 0xd5:

    \
case 0xd6:

    \
case 0xd7:

    \
if (digitalReadFast(pin) == HIGH) {
                   \
  /* other side did not ack correctly */
                \
  status[pin_index] = 0xc0;
                               \
```

```
}                                                                    \
                                                                     \
break;                                                               \
                                                                     \
case 0xd8:                                                           \
                                                                     \
/* negotiated rx */                                                  \
SWITCH_TO_RX(pin_index, pin)                                         \
conn_init_cb(pin_index);                                             \
break;                                                               \
                                                                     \
case 0xe9:                                                           \
                                                                     \
/* negotiated tx */                                                  \
status[pin_index] = 0x0f;                                            \
bitCount[pin_index] = PACKET_SIZE;                                   \
conn_init_cb(pin_index);                                             \
break;                                                               \
                                                                     \
case 0x80:                                                           \
                                                                     \
case 0x81:                                                           \
                                                                     \
case 0x82:                                                           \
                                                                     \
case 0x83:                                                           \
                                                                     \
case 0x84:                                                           \
                                                                     \
case 0x85:                                                           \
                                                                     \
case 0x86:                                                           \
                                                                     \
case 0x87:                                                           \
                                                                     \
case 0x88:                                                           \
                                                                     \
case 0x89:                                                           \
                                                                     \
case 0x8a:                                                           \
                                                                     \
if (digitalReadFast(pin) == HIGH || bitCount[pin_index]) { \
  /* partial/no hdr or not ready to rx, reset */  \
  status[pin_index] = 0x80;                                          \
}                                                                    \
                                                                     \
break;                                                               \
                                                                     \
case 0x8d:                                                           \
                                                                     \
/* expect high */                                                    \
if (digitalReadFast(pin) == LOW) {                                   \
  /* too much hdr, reset */                                          \
  status[pin_index] = 0x80;                                          \
  bitCount[pin_index] = 0;                                           \
  buffer[pin_index] = 0;                                             \
}                                                                    \
                                                                     \
break;                                                               \
                                                                     \
case 0x90:                                                           \
                                                                     \
/* read pulse first half */                                          \
if (digitalReadFast(pin) == HIGH) {                                  \
  /* missing pulse, reset */                                         \
  status[pin_index] = 0x80;                                          \
  bitCount[pin_index] = 0;                                           \
  buffer[pin_index] = 0;                                             \
```

```
    }                                                                   \
                                                                        \
    break;                                                              \
                                                                        \
    case 0x93: {                                                        \
                                                                        \
      /* read pulse second half */                                      \
      uint8_t bit_offset = bitCount[pin_index]++;                       \
      if (digitalReadFast(pin) == LOW) {                                \
        buffer[pin_index] |= ((uint32_t)1) << bit_offset; \
        RX_NEXT_BIT(pin_index, bit_offset, 0x8b)                        \
      } else {                                                          \
        /* short pulse, expect the next one */                          \
        RX_NEXT_BIT(pin_index, bit_offset, 0x8e)                        \
      }                                                                 \
      break; }                                                          \
  }                                                                     \
}

// packet format: update neighbours (00):
// 0xX------- link sender
// 0x-X------ original sender
// 0x--43210- neighbours (0-4)
// 0x-------X flags and parity bit
// flags:
// 0b00-X odd parity
// 0b00X- swap roles

// packet format: button pressed (01):
// 0xX------- link sender
// 0x-X------ original sender
// 0x--XX---- button states
// 0x-------X flags and parity bit
// flags:
// 0b01-X odd parity
// 0b01X- swap roles

// packet format: set LED colour (1X):
// 0xX------- link sender
// 0x-X------ original sender
// 0x--X----- recipient
// 0x---X---- LED index
// 0x----RGB- new LED colour (12-bit)
// 0x-------X flags and parity bit
// flags:
// 0b1--X odd parity
// 0b1-X- swap roles
// 0b1X-- reset all LEDs
// The physical LEDs have indices from 0 to 5. Setting index 6 will control the neighbour reporting feature.
// Recipient 0 will ignore the recipient.

uint8_t current_pin_index = 0;
uint32_t last_packets_by_remote[16] = {0};
uint32_t last_packet_times_by_remote[16] = {0};
uint32_t mask_link_remotes = 0;
uint32_t last_rx_times_by_pin[PIN_COUNT] = {0};
uint8_t last_remote_packet_forwarded[PIN_COUNT] = {0};
// circular buffer of most recent 5 pins where we received a message from each remote
// rppbr[r][0] is the pointer index
// pins are stored as pin_index+1 (0 is null)
uint8_t recent_packet_pins_by_remote[16][6] = {0};
uint32_t buttonPressTime = 0;

// populated in setup
uint32_t mask_link_sender;
uint32_t mask_myself_remote_sender;
uint8_t my_device_id;
uint32_t mask_recipient_me;


uint8_t buttonState = 0;
Adafruit_NeoPixel led_strips[2];
uint32_t pixels[6] = {0};
uint8_t led_strips_need_update = {0};
#if SINGLE_PIN
uint8_t serial_state = 0;
#endif
bool enable_neighbours = 1;



// http://www.graphics.stanford.edu/%7Eseander/bithacks.html#ParityParallel
inline bool compute_parity(uint32_t v) {
  v ^= v >> 16;
  v ^= v >> 8;
  v ^= v >> 4;
  v &= 0xf;
  return (0x6996 >> v) & 1;
}
```

```
inline void send_state(bool swap_roles) {
  uint32_t m;
  if (mask_link_remotes && enable_neighbours) {
    m = mask_myself_remote_sender |
        mask_link_remotes;
  } else {
    // no point sending a claim that we are not
        connected to any node as this is the
        default state
    // instead send an invalid empty message and
         yield (so we can discover who we are
        connected to)
    // this encourages early yielding at the
        start of a connection and so improved
        initialisation times
    // Also bail if sending neighbours is
        disabled
    m = 0;
    swap_roles = true;
  }
  send_msg(m, swap_roles);
}

inline void send_msg(uint32_t m, bool swap_roles
    ) {
  m &= ~0b10;
  m |= swap_roles << 1;
  m &= ~0xF0000000;
  m |= mask_link_sender;

  buffer[current_pin_index] = m ^ compute_parity
      (m) ^ 0b1;
  OUT("send")
  OUT(buffer[current_pin_index])
  bitCount[current_pin_index] = 0;
}

void dump_compute_state() {
  ERR(current_pin_index)
  ERR(bitCount[current_pin_index])
  ERR(buffer[current_pin_index])
  ERR(status[current_pin_index])
}

#define END_COMPUTE goto out;

void compute_loop() {
  uint32_t currentMicros = micros();
  if (status[0] > 0xC0) {
    digitalWriteFast(LED_BUILTIN, HIGH);
  } else {
    digitalWriteFast(LED_BUILTIN, LOW);
  }
  if (status[current_pin_index] > 0xC0) {
    mask_link_remotes &= ~((uint32_t)0xF0 << (
        current_pin_index * 4));
  }

  if (current_pin_index == 0) {
    OUT(((uint16_t)bitCount[current_pin_index]
        << 8) | status[current_pin_index])
    //OUT(status[current_pin_index])
  }

#if SINGLE_PIN
  // keep the serial buffer in a good state
  uint8_t available = Serial.available();
  if (available == 63) {
    // buffer full, clean slate
    for (; Serial.read() >= 0;);
    available = 0;
    serial_state = 0;
    Serial.write(0);
    Serial.write(0);
    Serial.write(0);
    Serial.write(0);
  }
  while ((serial_state < 8 && available > 0) ||
      available > 4) {
    uint8_t c = Serial.read();
    available--;
    if (serial_state < 4) {
      if (c == 0xFF) {
        serial_state++;
      } else {
        serial_state = 0;
      }
    } else /* if (serial_state < 8) */ {
      if (c == 0x00) {
        serial_state++;
      } else if (c == 0xFF) {
        serial_state = 4;
      } else {
        serial_state = 0;
      }
    }
  }
#endif


#if !SINGLE_PIN
  uint8_t new_button_state = ~(digitalReadFast
      (0) | digitalReadFast(2) << 1 |
      digitalReadFast(3) << 2 | digitalReadFast
      (4) << 3 | digitalReadFast(5) << 4 |
      digitalReadFast(6) << 5 | 0b11000000);
  if (new_button_state && new_button_state ^
      buttonState) {
    buttonState = new_button_state;
    buttonPressTime = currentMicros +
        BUTTON_CUTOFF;
  }
  if (buttonPressTime < currentMicros) {
    buttonState = 0;
```

```
}
#endif

// global operation:
// 1. send my id to each neighbour
// 2. send id of all discovered neighbours to
    all other neighbours
// 3. go to rx and prune disconnected
    neighbours

if (status[current_pin_index] >> 6 == 0b10 &&
    bitCount[current_pin_index] == PACKET_SIZE
    ) {
  // just finished receiving
  uint32_t msg = buffer[current_pin_index];
  buffer[current_pin_index] = 0;
  bitCount[current_pin_index] = 0;
  OUT(msg)
  if (!compute_parity(msg)) {
    // parity incorrect, discard
    goto out;
  }
  last_rx_times_by_pin[current_pin_index] =
      currentMicros;
  if (msg & 0b10) {
    // switch to tx
    status[current_pin_index] = 0x0f;
    bitCount[current_pin_index] = PACKET_SIZE;
  }
  #if !SINGLE_PIN
  mask_link_remotes = (mask_link_remotes & ~((
      uint32_t)0xF0 << (current_pin_index * 4)
      )) | ((msg & 0xF0000000) >> (24 - (
      current_pin_index * 4)));
  uint8_t origin = (msg >> 24) & 0xF;
  bool forward = true;
  if (msg & 0b1000) {
    bool handle = true;
    if ((msg & 0x00F00000) ==
        mask_recipient_me) {
      forward = false;
    } else if ((msg & 0x00F00000) != 0) {
      handle = false;
    }

    if (handle) {
      // update LEDs
      uint8_t led_index = (msg & 0x000F0000)
          >> 16;

      if (led_index == 6) {
        enable_neighbours = (msg & 0x0000FFF0)
            > 0;
      } else if (led_index < 6) {
        uint8_t strip_index = led_index & 1;
        OUT("led")
        if (msg & 0b0100) {
          pixels[strip_index] = pixels[2 +
              strip_index] = pixels[4 +
              strip_index] = 0;
        }
        pixels[led_index] = (msg & 0x0000F000)
            << (8 - BRIGHTNESS_BITS) | (msg &
            0x00000F00) << (3 -
            BRIGHTNESS_BITS) | (msg & 0
            x000000F0) >> BRIGHTNESS_BITS;
        led_strips[strip_index].setPixelColor
            (0, pixels[strip_index]);
        led_strips[strip_index].setPixelColor
            (1, pixels[2 + strip_index]);
        led_strips[strip_index].setPixelColor
            (2, pixels[4 + strip_index]);
        if (led_strips[strip_index].canShow())
            {
          led_strips[strip_index].show();
        } else {
          led_strips_need_update |= 1 <<
              strip_index;
        }
        if (msg & 0b0100) {
          strip_index = 1 - strip_index;
          pixels[strip_index] = pixels[2 +
              strip_index] = pixels[4 +
              strip_index] = 0;
          led_strips_need_update |= 1 <<
              strip_index;
        }
      }
    }
  }
  if (forward) {
    // forward packet, it wasn't for us
    uint8_t *recent_packet_pins =
        recent_packet_pins_by_remote[origin];
    recent_packet_pins[0] = (
        recent_packet_pins[0] % 5) + 1;
    recent_packet_pins[recent_packet_pins[0]]
        = current_pin_index + 1;
    last_packets_by_remote[origin] = msg;
    last_packet_times_by_remote[origin] =
        currentMicros;
  }
  #else
  if (Serial.availableForWrite() >= 4) {
    OUT("recv")
    OUT(buffer[0])
    Serial.write((char *)&msg, 4);
  }
  #endif
  goto out;
}
if (status[current_pin_index] >> 6 == 0b00 &&
    bitCount[current_pin_index] == PACKET_SIZE
    ) {
```

```
// send next
#if !SINGLE_PIN
// check remote is still here by yielding
    control
bool yield = last_rx_times_by_pin[
    current_pin_index] + (CONNECTION_TIMEOUT
     / 2) < currentMicros;
uint32_t forward_cutoff_time;
if (currentMicros < FORWARD_CUTOFF) {
  forward_cutoff_time = 0;
} else {
  forward_cutoff_time = currentMicros -
      FORWARD_CUTOFF;
}
if (buttonPressTime > currentMicros) {
  send_msg(mask_link_sender |
      mask_myself_remote_sender | ((uint32_t
      )buttonState) << 16 | 0b0100, yield);
  goto out;
}
uint8_t next_remote_for_packet_forwarding =
    last_remote_packet_forwarded[
    current_pin_index];
do {
  next_remote_for_packet_forwarding = (
      next_remote_for_packet_forwarding + 1)
       % 16;
  // missing or outdated packet or come full
       circle
} while (
(
// remote 0 doesn't exist
next_remote_for_packet_forwarding == 0
// packet too old to forward
|| last_packet_times_by_remote[
    next_remote_for_packet_forwarding] <
    forward_cutoff_time
// no packet stored from this remote
|| last_packets_by_remote[
    next_remote_for_packet_forwarding] == 0
// recently received a message from this
    remote on this pin (1-indexed)
|| recent_packet_pins_by_remote[
    next_remote_for_packet_forwarding][1] ==
     current_pin_index + 1
|| recent_packet_pins_by_remote[
    next_remote_for_packet_forwarding][2] ==
     current_pin_index + 1
|| recent_packet_pins_by_remote[
    next_remote_for_packet_forwarding][3] ==
     current_pin_index + 1
|| recent_packet_pins_by_remote[
    next_remote_for_packet_forwarding][4] ==
     current_pin_index + 1
|| recent_packet_pins_by_remote[
    next_remote_for_packet_forwarding][5] ==
     current_pin_index + 1
)
&& !(
// time to send our state
next_remote_for_packet_forwarding ==
    my_device_id
// tried all remotes, nothing is eligible
|| next_remote_for_packet_forwarding ==
    last_remote_packet_forwarded[
    current_pin_index]
)
);
if (next_remote_for_packet_forwarding ==
    last_remote_packet_forwarded[
    current_pin_index]) {
  // nothing we can forward, start listening
      instead
  yield = true;
}
if (
next_remote_for_packet_forwarding ==
    my_device_id
|| next_remote_for_packet_forwarding ==
    last_remote_packet_forwarded[
    current_pin_index]
) {
  OUT("nothing␣to␣forward")
  send_state(yield);
  last_remote_packet_forwarded[
      current_pin_index] = my_device_id;
  goto out;
}
last_remote_packet_forwarded[
    current_pin_index] =
    next_remote_for_packet_forwarding;
send_msg(last_packets_by_remote[
    next_remote_for_packet_forwarding],
    yield);
#else
if (serial_state == 8 && Serial.available()
    >= 4) {
  OUT("send")
  Serial.readBytes((char *)&buffer[0], 4);
  if (!compute_parity(buffer[0])) {
    OUT("invalid␣message␣to␣send")
  } else {
    bitCount[current_pin_index] = 0;
  }
  OUT(buffer[0])
  if (buffer[0] == 0xFFFFFFFF) {
    serial_state = 4;
  } else if ((buffer[0] & 0xFFFFFF) == 0
      xFFFFFF) {
    serial_state = 2;
  } else if ((buffer[0] & 0xFFFF) == 0xFFFF)
      {
    serial_state = 1;
  } else if ((buffer[0] & 0xFF) == 0xFF) {
```

```
      serial_state = 1;
    } else {
      serial_state = 0;
    }
    if (!(buffer[0] & 0b10)) {
      // signal ready for next packet
      Serial.write(0);
      Serial.write(0);
      Serial.write(0);
      Serial.write(0);
    }
  }
  #endif
  goto out;
}
#if !SINGLE_PIN
if (led_strips_need_update & 1 && led_strips
    [0].canShow()) {
  led_strips_need_update &= ~1;
  led_strips[0].setPixelColor(0, pixels[0]);
  led_strips[0].setPixelColor(1, pixels[2]);
  led_strips[0].setPixelColor(2, pixels[4]);
  led_strips[0].show();
  goto out;
}
if (led_strips_need_update & 2 && led_strips
    [1].canShow()) {
  led_strips_need_update &= ~2;
  led_strips[1].setPixelColor(0, pixels[1]);
  led_strips[1].setPixelColor(1, pixels[3]);
  led_strips[1].setPixelColor(2, pixels[5]);
  led_strips[1].show();
  goto out;
}
#endif
if (status[current_pin_index] >> 6 == 0b10 &&
    last_rx_times_by_pin[current_pin_index] +
    CONNECTION_TIMEOUT < currentMicros) {
  // they had their chance
  OUT("timeout")
  mask_link_remotes &= ~((uint32_t)0xF0 << (
      current_pin_index * 4));
  status[current_pin_index] = 0xc0;
  goto out;
}
// if tx timeout occurs we will not notice but
     eventually we will yield and it will sort
     itself out
// this is most likely to cause problems on a
    single-pin node
// if it happens there, it means the host
    controller was too slow

out:
#if !SINGLE_PIN
current_pin_index = (current_pin_index + 1) %
    PIN_COUNT;
#else
;
#endif
}

#define TX_DONE_CB(pin_index, pin) { \
  if (buffer[pin_index] & 0x00000002) { \
    /* swap roles was set */ \
    SWITCH_TO_RX(pin_index, pin) \
    last_rx_times_by_pin[pin_index] = micros();
        \
  } else { \
    digitalWriteFast(pin, HIGH); \
    status[pin_index] = 0x11; \
    bitCount[pin_index] = PACKET_SIZE; \
  } \
  OUT("tx␣done␣cb") \
}

inline void conn_init_cb(uint8_t pin_index) {
  last_rx_times_by_pin[pin_index] = micros();
  #if SINGLE_PIN
  if (!(status[pin_index] & 0x80)) {
    // signal ready for next packet
    Serial.write(0);
    Serial.write(0);
    Serial.write(0);
    Serial.write(0);
  }
  #endif
}

void setup() {
  randomSeed(analogRead(A0) | analogRead(A1) <<
      10UL | (unsigned long)analogRead(A2) << 20
      UL | (unsigned long)analogRead(A3) << 30);
  pinModeFast(LED_BUILTIN, OUTPUT);
  digitalWriteFast(LED_BUILTIN, LOW);

  #if DEBUG || SINGLE_PIN
  Serial.begin(115200);
  while (!Serial);
  #endif

  #if SINGLE_PIN
  while (Serial.available()) {
    Serial.read();
  }
  #endif

  my_device_id = EEPROM.read(0);
  mask_link_sender = ((uint32_t)my_device_id <<
      28);
  mask_myself_remote_sender = ((uint32_t)
      my_device_id << 24);
  mask_recipient_me = ((uint32_t)my_device_id <<
       20);
```

```
  led_strips[0] = Adafruit_NeoPixel(3, 12,
      NEO_GRB + NEO_KHZ800);
  led_strips[1] = Adafruit_NeoPixel(3, 13,
      NEO_GRB + NEO_KHZ800);
  for (int i = 0; i < 2; i++) {
    led_strips[i].begin();
  }

  #if !SINGLE_PIN
  pinModeFast(0, INPUT_PULLUP);
  pinModeFast(2, INPUT_PULLUP);
  pinModeFast(3, INPUT_PULLUP);
  pinModeFast(4, INPUT_PULLUP);
  pinModeFast(5, INPUT_PULLUP);
  pinModeFast(6, INPUT_PULLUP);
  #endif

  COMMS_PIN_SETUP(0,7)
  #if !SINGLE_PIN
    COMMS_PIN_SETUP(1,8)
    COMMS_PIN_SETUP(2,9)
    COMMS_PIN_SETUP(3,10)
    COMMS_PIN_SETUP(4,11)
    #endif

}

void loop() {
  while (1) {
    COMMS_PIN_LOOP(0,7)
    #if !SINGLE_PIN
    COMMS_PIN_LOOP(1,8)
    COMMS_PIN_LOOP(2,9)
    COMMS_PIN_LOOP(3,10)
    COMMS_PIN_LOOP(4,11)
    #endif
  }
}
```