Understanding Spatial Skills and Encoding Strategies in Student Problem Solving Activities

Jack Parkinson
jack.parkinson@glasgow.ac.uk
University of Glasgow
United Kingdom

Quintin Cutts
quintin.cutts@glasgow.ac.uk
University of Glasgow
United Kingdom

ABSTRACT

Background and Context. Margulieux's Spatial Encoding Strategy theory (SpES) provides a possible reason for the relationship between spatial skills and success in STEM fields, including CS. While there is indirect evidence to suggest that the theory holds, there is little work which explicitly explores the core theory in practice. Furthermore, current work in spatial skills has largely focused on introductory courses, and it is unclear whether advanced students (and then experts) use spatial skills in computing.

Objectives. We wish to determine whether we can see senior students in CS with high spatial skills utilising non-verbal encoding strategies when solving CS programming problems.

Method. Transcripts from a think-aloud exercise with experienced students (final year of undergraduate), whose spatial skills were measured, were analysed to identify utterances which indicated spatial encoding strategies being employed, such as the construction and alteration of mental models on the fly, and to determine differences according to spatial skills level.

Findings. Students with higher spatial skills were more likely to exhibit evidence of the construction of flexible, comprehensive mental models to solve the programming problems, demonstrating advanced encoding and chunking strategies. Students with lower spatial skills were more likely to struggle with the construction and alteration of mental models, indicating that they typically lack the capability to effectively chunk and save working memory space.

Implications. This work confirms the predictions of SpES more precisely than prior work by showing that skilled problem solving involves the mental model creation and manipulation that underlies SpES. It demonstrates that students with better spatial skills are more likely to succeed in programming problem solving, even in the later stages of study, due to their ability to encode non-verbal information.

CCS CONCEPTS

• **Social and professional topics → Computing education**.

KEYWORDS

spatial skills, non-verbal encoding, spatial encoding strategy theory, theoretical model, qualitative

1 INTRODUCTION

Margulieux's Spatial Encoding Strategy theory (SpES) [28] provides a possible reason for the relationship between spatial skills and success in Computing Science (CS), a relationship which has been increasingly explored over the past few years [4, 5, 17, 21, 22, 25, 26, 38, 40–44, 48]. The theory states that spatial skills are valuable for STEM learning because they give people better encoding and orientation strategies for non-verbal information. This allows people to chunk more effectively and store more information in working memory, which in turn frees up capacity for tracking complex information, making alterations to existing information or constructing mental models [28].

There is some recent evidence which suggests that SpES theory holds true [25, 26, 42, 44], however the core of the theory – related to cognitive mechanisms – is largely untested. There is also some conflating evidence on whether experts and advanced students require spatial skills in advanced STEM practice [25, 28, 63, 68]. To address both these gaps in existing theory, our research questions are:

- **RQ1** Are non-verbal representations predicted by SpES evident in advanced problem solving practice?
- **RQ2** Is there an association between measured spatial ability and the use of non-verbal representations during advanced problem solving practice?

In answering these questions, this paper exposes cognitive mechanisms related to SpES by utilising a think-aloud protocol to observe the practices of students as they solve programming problems, seeking evidence of the kinds of mental model building and related strategies which Margulieux describes. It also contributes to recent evidence which has shown that spatial skills continue to correlate with academic results further along in a CS degree [25, 43], not just introductory outcomes, which is at odds with one of Margulieux's facets of the theory stating that spatial skills are of less use to experts [28].

The study reported here found that senior students with better spatial skills were more likely to succeed at building robust mental models, manipulating them on the fly and tracking several pieces of information at once – all hallmarks of strong non-verbal encoding skills – than their peers with lower spatial skills. Students with lower spatial skills were more likely to struggle to adapt their models or reached limits where they could develop them no further, as well as relying on strategies like pure recall to try to solve problems.

Given the involvement of final year undergraduate students, this work not only provides more precise evidence for the cognitive mechanisms underpinning SpES than has been discovered to date, but it also has implications about the role of spatial skills in post-introductory study and practice.

2 BACKGROUND

2.1 Spatial Skills

Spatial skills are cognitive skills associated with understanding and perceiving space and spatial concepts. A challenge in spatial skills research espoused by Carroll in a review of about a century of spatial skills research is that there have been many overlapping and incomplete definitions of spatial skills put forward over the years [6]. He opens his 1993 review of various investigations of spatial factors with a quote by Elliot & Smith: "Spatial ability has been defined in such a variety of different ways that it is often difficult to be precise about the meanings which we ascribe to the term" [14]. For the purpose of discussion, Carroll provides his own broad definition of spatial skills in an effort to capture all their constituent parts:

> Spatial and other visual perceptual abilities have to do with individuals' abilities in searching the visual field, apprehending the forms, shapes, and positions of objects as visually perceived, forming mental representations of those forms, shapes, and positions, and manipulating such representations "mentally" [6]

Carroll identifies six distinct factors in his 1993 review [6]:

- Spatial visualisation
- Spatial orientation
- Perceptual speed
- Closure speed
- Closure flexibility
- Visual imagery

Parkinson & Cutts explore all of these factors in their 2018 ICER paper [41]; we will only explore spatial visualisation and spatial orientation here for brevity, selecting these two because they are the two factors most studied in relation to CS in prior research.

The spatial skills factor studied most in STEM research is spatial visualisation. McGee states that spatial visualisation involves, "the ability to mentally rotate, manipulate, and twist two- and three-dimensional stimulus objects" [30]. There are several ways to measure spatial visualisation: 2-D rotation [68], 3-D rotation [18, 66, 70], object cross-sectioning [7], object construction [3] and paper folding [13] are a few.

Spatial orientation is a factor of spatial skills involving orienting or positioning with respect to perspective and the location of objects. Kozhevnikov & Hegarty have explored this factor in various contexts and have developed a test to measure spatial orientation [20, 24]. The test involves identifying the direction and locality of objects from a given object while oriented to face another. Spatial orientation often involves map reading, drawing or navigation tasks [45, 46].

2.2 Spatial Skills in STEM and Computing Science

Spatial skills, most frequently spatial visualisation, have been connected with success in STEM for some time. Super & Bachrach provide possibly the first published observation of the relationship, identifying that spatial skills were valuable for people in scientific careers [57]. Another notable study conducted by Wai *et al.* identified that university-level students in the US who completed STEM degrees were likely to have scored well on a spatial skills test several years before in high school [68]. Beyond just observing a correlation, Sorby has been training the spatial skills of engineers and has observed positive outcomes, such as improved grades and retention, for decades [53–55]. Veurink & Sorby also observed that module grades in other STEM subjects, including CS, improved when they were taken as electives by engineering majors who completed spatial skills training [67].

In CS specifically, Mayer discovered that paper folding prediction questions (which have been used as a component in spatial visualisation tests [68]) in a battery of logical tests correlated with success in a BASIC exam for new learners [29]. Cox *et al.* theorised about the relationship between spatial skills and navigation of a codebase [11], and Jones & Burnett later found that better spatial skills led to more effective source code navigation in a CS master's conversion cohort (that is, students with an undergraduate degree in a field different to CS) [21]. Parker *et al.* also observed that, of a range of factors combined through structural equation modelling, spatial skills were a more powerful intervening variable contributing to CS success than a combined measure of access to computing, which was comprised of "formal exposure to computing, informal exposure to technology, perceptions of computing, and encouragement to pursue computing" [38].

Fincher led the BRACE (Building Research in Australasian Computing Education) project which involved examining several factors related to success in introductory computing at eleven institutions [17]. They found a correlation between paper folding tests (spatial visualisation) and CS assessment scores [48]. They also found that students who were capable of drawing more complex maps with more spatial and navigation information – in the hierarchy of landmark, route and survey [46, 69] – scored higher in CS assessment than students who drew more basic maps or were not capable of drawing maps at all [61].

Spatial skills as measured by mental rotation have also been associated with several measures of CS success in introductory contexts: module assessment [26, 42], standardised college entry tests [10] and a reduced set of the SCS1 [37], a CS1 concept inventory [4, 5, 38]. Additionally, Sorby's spatial skills training programme [56] has been used in introductory CS, with gains observed in multiple contexts [5, 10, 42].

2.3 Theories for the Relationship

Frameworks for the relationship between spatial skills and CS – and their wider relationship with STEM – were proposed first by

Parkinson & Cutts in 2018 [41]. Parkinson & Cutts proposed that there is a cognitive skill, separate from spatial skills and STEM domains, which connects them both. Their reasoning for this "underlying cognitive ability" was based around findings that spatial skills have a relationship with a range of STEM domains, including ones – like CS, arguably – which do not have many explicitly *spatial* activities. Rather, they proposed that spatial skills tests exposed a more abstract skillset related to constructing mental models which could be applied in many domains.

Margulieux's Spatial Encoding Strategy theory presents the relationship between spatial skills and STEM success as depending on neurostructures in the hippocampus called grid and place cells [35]. These cells were originally developed for navigation, however they can also encode non-verbal information, even if it is not spatial in nature [8]. The discovery of these cells is accredited to O'Keefe [34, 36] and Moser & Moser [19], who won a Nobel Prize in Psychology or Medicine in 2014 in recognition of their work in the area [1]. Since non-verbal information is important in STEM fields [68], being able to utilise these cells more effectively helps in STEM learning. Margulieux theorises that spatial skills training can improve the strategies these cells utilise for encoding, thus making non-verbal representations easier and faster to generate.

This has implications for how we can store non-verbal information in practice. Working memory capacity is fixed at birth [2], but people can function as if their working memory is much larger by *chunking*; that is, storing larger amounts of information in a single chunk [1]. People can develop highly effective strategies for chunking information, which leads to high-capacity memory operations such as memorising the order of an entire deck of playing cards [15, 60]. Chunking frequently supersedes working memory capacity in any meaningful practice: any variation in peoples' fixed working memory capacity is dwarfed by their ability to chunk [59].

Margulieux theorises that better spatial skills lead to better strategies for storing (chunking) all kinds of non-verbal information. The theory states as follows (emphasis Margulieux's) [28]:

> **"Developing spatial skills (i.e., visualization, relations, and orientation) helps people to develop generalizable *strategies* for 1) encoding mental representations of non-verbal information, including 2) identifying useful landmarks to orient the representation.** Having strategies for rapidly encoding non-verbal mental representations and identifying landmarks would increase the amount of new information processed initially. In turn, encoding of larger chunks of information would afford learners more capacity in their WM [working memory], especially in the visuospatial sketchpad, for reasoning tasks (e.g., running mental models) or for building more complex representations (e.g., building a robust notional machine)."

In summary, spatial skills may work to assist in STEM subjects because non-verbal representations of ideas and knowledge are important in STEM. This is perhaps particularly true of computing science, where abstract ideas rarely have physical representations. We have brain structures which are used for working out spatial relationships which in turn are used in navigation tasks, and have been used so for evolutionarily significant periods of time. In studying STEM subjects, these brain structures are recruited to produce non-verbal representations (not necessarily spatial ones) which are necessary or at least valuable for STEM. According to SpES, this is what underlies the association of spatial skills with ability in STEM subjects.

[1]https://www.nobelprize.org/prizes/medicine/2014/summary/

2.4 Recent "Post Hoc" Evidence for SpES

In explaining SpES, Margulieux also describes several ways in which SpES aligns with prior research on spatial skills and STEM. These "post hoc explanations" provide possible ways that SpES can explain some relationships observed in existing research. They are: spatial skills training improves STEM attainment in many domains; spatial skills predict initial STEM performance more accurately than later performance; strategy and spatial training eliminate gender differences; and transfer of problem-solving skill between fields is limited [28]. These explanations can be considered the possible practical outcomes of SpES, demonstrating the relationship's existence in ways which align with Margulieux's theory without necessarily explicitly demonstrating the theory's cognitive mechanisms in action. More evidence – and some work which raises some contention – have been presented since SpES was published for these explanations, and are described in this section.

2.4.1 Spatial training improves achievement in many STEM domains. The evidence for spatial skills training being valuable in many STEM domains is already well documented and has been growing for several years [64], however its value in CS specifically is relatively new. Cooper *et al.* demonstrated the benefits of spatial skills training with a small cohort of pre-college CS students over a short period of time in 2015 [10], but since the publication of SpES, spatial skills training has been adopted at multiple institutions for introductory CS students and research has involved hundreds of participants, with positive results observed consistently [5, 26, 40, 42].

2.4.2 Spatial skill predicts initial STEM performance more accurately than later performance. Margulieux suggests that spatial skills are more valuable for early STEM learning and are of less value for later performance, since learners gradually develop domain specific strategies which are more efficient than more abstract non-verbal encoding strategies. In some cases, this appears to be true: it has been found that students with less CS experience show a stronger correlation with spatial skills when it comes to exam performance [42] and a more precise test of core computing skills [44].

However, there is other work which shows that spatial skills correlate even more strongly with grades in later study than they do in introductory years [43] and correlations have been discovered between spatial skills and individual modules taken in later years of CS study [25]. This indicates that there is still some association between spatial skills and success beyond just introductory CS. This does not invalidate the theory, but rather suggests that generalisable strategies are still relied upon when students move to new areas of study even if they could already be considered experts in some areas of CS. This indicates that domain-specific problem solving strategies don't transfer (which is another of Margulieux's items of post-hoc explanation).

However, it does also indicate that the relationship between spatial skills and STEM outcomes in experts is perhaps not as simple as SpES originally states. There is also very little work involving spatial skills and CS outcomes which does not involve introductory CS students, so study of students further along in their programmes is necessary to provide a full picture of the relationship.

2.4.3 Strategy and spatial training eliminate gender differences. A study by Ly *et al.* sheds some light on the proposal that gender differences can be eliminated through spatial skills training. They found that women with low spatial skills when beginning a CS1 programme are at the highest risk of dropping the course, though spatial skills training can close the gap and bring low scorers up to a par with the rest of the cohort [26].

2.5 Indirect Evidence and Theories for SpES Cognition

In addition to more evidence being found for these post hoc explanations, the examples of cognition provided by Margulieux also appear to be indirectly evident in more recent work. Spatial skills have been shown to be connected to a core, cognitive skill in CS through expression evaluation [44]. A test of expression evaluation used by Parkinson *et al.* had questions of high complexity in terms of the number of operations which must be tracked and the amount of data which needs to be followed. The authors explicitly instructed students not to use written aides in solving the questions, and those with higher spatial skills scored higher in the expression evaluation test. This suggests that these students were more able to chunk the complex non-verbal information effectively and hold more information in their head at once, which supports the theory.

Additionally, Parkinson & Cutts observed a higher correlation between spatial skills and long-form coding questions and long tracing questions in an exam than with the execution of single-line expressions or short code snippets [42]. This suggests that questions which involve more information tracking and open problem solving are more likely to require spatial skills to solve. In order to avoid confusion, by "problem solving" in this context we are referring to the ability to address a problem by recalling and utilising known information (learned code constructs and syntax) when there are many possible solutions. Liu *et al.* also explicitly state that they expect that spatial skills benefit problem solving through more effective chunking strategies [25].

While many aspects of SpES have been demonstrated, little work has yet been done to get at the root of the theory: demonstrating the theory's *cognitive mechanisms* in action. The work above represents some of the closest connections to seeing the cognition underpinning SpES being used, but the connections are indirect. There is limited evidence for students actually applying the strategies that Margulieux describes, such as explicit reference to mental model building and running, or the construction of a robust notional machine. Although SpES holds in many respects, and evidence has only grown since it was published, we have not yet explicitly observed the theory's cognitive mechanisms in action in actual student problem solving in CS.

3 RESEARCH QUESTIONS

The background research shows that several pieces of recent work have contributed to supporting SpES, but there are still some gaps. While there is evidence to suggest that novices require spatial skills to solve computing problems more than experts do, this has mostly only been examined in an introductory context and any work going beyond the first year of study demonstrates that the relationship between spatial skills and CS outcomes still exists, and even grows. Additionally, while there is some evidence that the cognitive mechanisms described by Margulieux are being used as predicted, this evidence is indirect and doesn't deliberately demonstrate how these mechanisms are being applied.

Bearing in mind that measures of spatial skills are indicative of non-verbal representation and encoding strategies at the core of SpES, our research questions are:

RQ1 Are non-verbal representations predicted by SpES evident in advanced problem solving practice?

RQ2 Is there an association between measured spatial ability and the use of non-verbal representations during advanced problem solving practice?

Spatial Encoding Strategy theory is a significant theory in our field and beyond. It is recent and non-trivial, linking performance, psychology and educational effects not only in CS, but also in wider STEM. It is worth testing and exploring; if we can answer these research questions, this will be the first major contribution to demonstrating the cognitive mechanisms in Margulieux's SpES in action, thus taking steps towards to the confirmation of the theory.

4 METHOD

In order to find out what is going on in a student's head, we need have the students make their thoughts external. A tried-and-tested method for examining human cognition is a *think-aloud protocol*, which uses introspection to expose internal processes and mental models [33, 62]. Think-aloud protocols have a long and well-regarded history of use: they were formalised by Ericsson & Simon [16], the former of whom went on to be awarded the ACM's A. M. Turing Award in 1975 along with Allen Newell for their contributions to AI and human cognition research involving early think-aloud protocols [32].

Using a think-aloud protocol, we wished to examine senior students' problem solving processes to identify if SpES was being applied, and whether there was a difference in the way mental representations were generated and used by students with different spatial skills. We wanted to see if we could witness students with high spatial skills utilising the strategies outlined by SpES as they solved programming problems, and of these were distinct from students with lower spatial skills.

4.1 Participants

The participants selected for this study were fourth year students at a large, research-focused institution. These students were in their final year of study in an undergraduate programme and were enrolled in Computer Science or a closely related programme (such as Software Engineering – at the authors' institution these programmes are very similar).

The reason for choosing fourth year students was because, by this stage in their degree, such senior students are required to enact some expert-level behaviours: pick up new languages, systems and computational domains, work in new problem domains and tackle complex real-world problems. This made these students at least close to "experts", permitting the examination of SpES's cognitive mechanisms in action in advanced students.

Using senior students also gives the best chance of eliminating minimum knowledge and experience effects: we were mostly interested in the students' ability to solve the problems in relation to their cognition and wanted their prior procedural and declarative knowledge to have minimal effect. All CS students learn Python for a full year in their first year, and will use Python in compulsory modules in their second and third year of study. This permitted the use of Python in the planned exercise. One of the exercises (see section 4.2) required reading and describing a program, and by choosing fourth year students we could assume at least a baseline of Python knowledge and experience suitable for completing the task.

Students were invited to take part in a mass email to all fourth year students which included an information sheet describing the experiment in broad terms and an indication that participants who completed the study would be compensated with an Amazon voucher. To express interest, the students were required to fill out a form requiring their student IDs and explicitly indicating that they provided their consent for their data to be used as described in the information sheet. The student IDs were used to check the student records to determine that they were suitable for the study (that is, actually in their fourth year of study and enrolled in CS).

Of a cohort of roughly 250 fourth year students, 17 responded to the survey. 2 students were removed because they were not fourth year students. 2 students ultimately did not attend to take part in the study, resulting in 13 students contributing to the overall dataset. Students gave consent for their university records to be examined for demographic data, specifically their age and gender: all 13 students were in the age range of 18–23, making them "traditionally university aged" in the UK; 5 participants had a recorded gender of "female" and 8 had a recorded gender of "male".[2]

The entire study methodology was approved by the university ethics committee prior to the research starting. It was made clear that students were permitted to withdraw from the study at any point and that their responses would be made anonymous before any form of publication or presentation.

4.2 Instruments

Once they had expressed interest and had signed the consent form, students were invited to take the Revised PSVT:R test of spatial skills [70]. The test consists of 30 multiple choice items of increasing difficulty. The test is a test of rotations: each question requires the participant to identify an orientation of an object from a selection of five orientations which matches a sequence of rotations shown applied to a different object. The test was issued on the institution's VLE. It was timed at 20 minutes, with given answers automatically submitted once the time ran out.

For the think-aloud study, participants indicated a 30 minute time slot to come – in person – to a quiet room on campus to complete a complex problem-solving programming exercise. The purpose of the exercise was to have the students think aloud as they completed it and have them expose their strategies so that they could be compared to SpES (see section 4.4 for more details on the analysis rubric) so the exercise needed to expose the students to a challenge of reasonable complexity and novelty. Three ten-minute exercises were designed:

(1) An open coding exercise, where a problem specification was provided and the student had to come up with a programmatic solution
(2) A sample solution to the first exercise was provided which the student had to explain to the interviewer
(3) An altered version of the first exercise which required a new approach to solve, again requiring the student to come up with a programmatic solution

Each problem was printed on a separate piece of paper and handed to the students as required. The exercises also came with sample inputs and expected outputs. The full exercises used can be seen in the appendix.

The students were expected to complete their solutions on blank paper, one sheet per exercise, in a language or pseudocode format of their choice. It was made clear that correct syntax and "real code" were not as important as demonstrating a clear understanding of the solution and any programmatic representations they generated. The students did not have access to any devices which could run code during the session.

The motivation for selecting these three exercises was to maximise the chance of observing some form of a mental representation of either the problem domain or the students' solutions. By posing a fairly complex coding question in the first exercise, we expected that the students would need to generate a model of the problem as they constructed a model of their solution, maintaining both at once. As they moved to the second part of the exercise, they would need to maintain their problem model and reconcile it with a new solution model, which may or may not be similar to their own from the first exercise. As they moved to the third exercise, they would need to modify their problem model or entirely rebuild it and come up with a new solution model, which may or may not be based on their original solution model or the one they constructed from the solution given to them in the second exercise. We could not guarantee that students would tackle the exercises in exactly this way, but they were designed specifically to try and prompt the generation and alteration of internal representations during and between exercises to allow us to determine how effective these representations were.

4.3 Data Collection

The practical procedures for think-aloud protocols detailed by Someren, Bernard & Sandberg were used as the primary guiding principles for delivery of the protocol [52]. The three exercises were completed over 30 minutes – 10 minutes maximum per exercise – in a quiet room with only the participant and the interviewer present.

[2] The university system permits non-binary gender identities which can be self-described by students, however the students involved in this study happened to only have binary genders recorded.

The interviewer made the think-aloud protocol clear to the student prior to starting: talk about your thought processes and make it clear what you are thinking as much as possible. The interviewer also made it clear that if the student had any questions or needed to make clarifications about the exercise, they were welcome to do so.

The students' thoughts were captured using a dictaphone, which was later transcribed in full. If a student fell silent for a period of more than a few seconds, the interviewer prompted them to externalise their thought process. Once a student completed their solution, or the 10 minutes ran out, the interviewer asked a few questions about their solutions.

4.4 Data Analysis

The qualitative methodology applied to analyse the participants' transcripts was a form of deductive coding. The transcripts were read by researchers with the text of Margulieux's theory close to hand (specifically, the paper was printed as it appears in the ACM Digital Library and the quote listed above in section 2.3 was highlighted as a main point of reference). The researchers were to highlight any utterance in the transcript which they felt exhibited either:

- SpES in action, such as evidence of the construction of a mental model or chunking large amounts of information. Such examples may be those Margulieux explicitly states, particularly "running mental models" or demonstrations of students constructing complex, multiple and overlapping mental models [39]. Any other utterances which may indicate encoding mental representations and similar or related activities were highlighted.
- Alternative strategies, particularly strategies at odds with SpES and the development or utilisation of mental representations. These might manifest in the form of clues that students are, for example, *not* running or building mental models, or do *not* have robust internal representations of the problem domain or their solutions.

These utterances and highlights will be referred to throughout the rest of this paper as *events*.

The first pass of analyses was conducted on three transcripts by two researchers individually (both researchers analysed all three transcripts). After completing their analysis, the researchers came together and compared the events they had isolated. The reason for the identification of each event had to be explained, with the researcher indicating why they believed the event was an example of either of the two factors being searched for. This process yielded 91% agreement on coded events and a Cohen's Kappa value of 0.83, indicating fairly consistent agreement across the coding process. In each instance where there was disagreement, or one researcher had marked an event which the other had not, the event was discussed until both researchers agreed. Using these corroborated events as a rubric, one researcher completed the analysis of all the remaining transcripts.

All the analysis of the transcripts was conducted without knowledge of the students' spatial skills. This was to avoid bias influencing the coding. After the analysis was completed, the highlights were drawn out as events and these were then attributed the spatial skills score of the student. In the results section, the students' spatial skills score is shown in parentheses at the end of each quote or description of an event.

5 RESULTS

5.1 Spatial Skills Test

Out of a maximum 30 points, the mean spatial skills score was 24.5, the median was 26, the minimum was 16 and the maximum was 30. This is fairly high compared to other students of a similar level tested, such as the combined third and fourth year students tested in Parkinson & Cutts 2018 work, where the mean was 22.9 points [41]. However, the fact that these are higher than we might expect to see with a cohort of fourth year students does not seriously affect the research outcomes, as discussed in section 6.1.

5.2 Transcript Event Coding

This section groups the events observed by the strategies that they expose. The intention is to demonstrate how students did, or did not, apply SpES-related strategies while solving the programming problems. The spatial skills of the participant exhibiting an event is shown in parentheses, with a letter used to differentiate students with the same scores.

Students with higher spatial skills (ranging from 26 to 30) demonstrated that they constructed mental models and were able to rapidly alter them as they understood more about the problem. A few examples of these events are:

- Realising an else clause is unnecessary in the first task: *"if this is equal to the target you append this coin and– I guess you don't want an else, actually [scratches out else clause]" (26b)*
- Addressing a coin duplication problem in the first task: *"Oh wait, I can't just iterate over the coins 'cos I won't know if they're duplicates. I'll use…I'll use enumerate and an index. The index will be unique per coin." (28a)*

One student (28b) demonstrated the ability to completely scrap their solution model while maintaining their problem model and start again from scratch: *"No this isn't…hmm…no, this isn't going to do it I don't think. Let me…let's try this again"*. Despite building a non-starting first solution, the student was still able to complete the task with a new, different solution. Another student (27) also almost completely rewrote their solution when they realised that the time complexity was going to be high: *"This is very inefficient! The time complexity is going to be…is going to be wild. That doesn't matter though, right? I don't need to…well I– well I should be more efficient I think, it should be more dynamic. I'll…let me just change it"*. The student identified a limitation of the program – which was not a hindrance to solving the problem, necessarily – and was able to rewrite their solution. both these events demonstrate an ability to hold a model of the problem in their head while actively constructing, deconstructing and reconstructing a separate model of the solution, exhibiting the ability to chunk effectively and dynamically manipulate their mental models.

By contrast, a student with lower spatial skills (17) noticed at about the same time in their session that their initial solution wasn't going to work and in fact even noted that an alternative solution might have been more appropriate (*"hmm, perhaps…perhaps this should have been recursive. I think I will…I'm not sure how many…how many loops if it's not recursive."*) but did not change their approach and continued to try and make their solution work. Another student (16) noticed an error quite late in the process of building their solution, faltered a little, but then decided that they were not going to fix it and instead continued to try and finish the incorrect solution: *"Okay, well, I know it's not gonna work but…I think I won't figure that out so I'm just gonna- let me just do it like this and…and we'll see."* Both these events could be indicative of cognitive overload: they have identified that their solution is not going to work, but do not have the mental capacity to make alterations without losing important information. This could be addressed by having better chunking strategies.

Another student (20a) decided that the first problem was a *"dynamic programming problem"* and made repeated reference to this throughout their coding of the solution. It was unclear how this affected their solution, but they had made a decision early in reading the program specification about the nature of the problem and decided that – somehow – it must factor into their process. This demonstrates that they had established some form of mental model or internal representation of the problem which was rigid and they were not able to change. This may be indicative of limited working memory capacity, affecting their ability to incorporate or adjust their model with new information, suggesting that they have reached cognitive overload.

All the students who explicitly referred back to their own past solutions as they moved through the problems were higher spatial skills scorers of 26 and above. They appeared to be using their previous work to assist in the model construction of the next task. One student (27) noted that the recursive solution from the second task was gradually reducing the size of N until zero was reached while they had done the opposite, starting at zero and adding until they reached the target. The student referred to this as "basically the same" and moved on. Conversely, one of the students with lower spatial skills (16) had taken a similar incremental approach in their program and was unable to connect their process with the decremental process in the solution: they could not understand why you would want to alter the value of the target because then the program would lose track of what the final goal should be. These events indicate that the students with higher spatial skills were able to keep track of their previous solutions as they progressed in the problems, or at least were able to orient the new problems with models that they had previously encoded, while the student with lower spatial skills appeared to discard their solution or at least did not recognise its structure against a piece of code in the same pattern.

It was observed that two of those with lower spatial skills (16, 20a) tended to, when they were stuck, read the problematic text (either the task specification or their code) over and over again without necessarily gaining anything new. This was contrasted with students with higher spatial skills, who tended to take much more strategic approaches when they were stuck or unsure how to proceed, such as:

- Sketching out a diagram of their planned solution (27)
- Writing the procedures they planned to implement at a high-level in a list of bullet points (26d)
- Deliberately pushing the code they were writing away from themselves and looking to the ceiling before verbalising their plan again (28b)

By distancing themselves from the problem text, these students appeared to be strongly focusing on the development of a mental representation of the problem rather than the text surface.

Interestingly, students with higher spatial skills also had a tendency to remove themselves from the terminology of the task and create their own. Multiple students (26a, 26b, 27) began to frequently refer to N as "the target", which is an accurate conceptual description of what N represents even though N is never referred to as "the target" in the problem specification. This would suggest that these students have developed a conceptual idea of a target value which they are applying to the named variable N, which is a stark contrast to the students who continuously read the problem specification while trying to come up with their solutions.

Some high-scoring students also began to refer to some of their conceptual ideas in very abstract terminology. One student (29) kept referring to lists as "things" before correcting themselves (*"[pointing to list] we'll then add the coin to this thing- this list, sorry"*) which suggests that they have a conceptual understanding of the constructs they are working with, but its representation is abstract and non-verbal. This kind of behaviour was observed in some other high-scoring students who would catch themselves as they were explaining their programs or plans to insert clearer names, apparently for the interviewer's benefit.

These events indicate that students with higher spatial skills had a robust, internal representation of the problem with their own abstractions. This is indicative of a complex mental model being formed with pieces being converted to more fluid representations, demonstrating a robust non-verbal representation of the problem and solution spaces.

Many of the students with higher spatial skills, without prompting, would give brief, high-level summaries of both the problem as they understood it and their planned solution. In most cases these solution plans were still evolving, also representing that the students had a flexible mental model which they were updating as they verbalised their process. Here is an example from one student with high spatial skills (26a):

> "So I'm thinking of using a recursive function. So that would pass in parameters like what the remaining coins are in the collection, what the current value is that you've totalled up so far and what the target value would be…and then at each point it just picks one random one- or no, it'd loop through all the coins that could be pulled out that were less than adding up the total that you're looking for and then…it calls that recursive call back, um, with that coin being taken out that coin being added to the total so far."

This – and the other similar examples from other students – looks like an approach to chunking and encoding the information needed into a model which can be recalled later. It also demonstrates an ability to evaluate and adjust their models as they are constructing them.

Students with lower spatial skills were more likely to struggle with building a clear understanding of the problem or their planned solution and tended to either "get lost" or fail to fit all the pieces of their model together. One student (16) wondered, early in solving the first task, if the coins in the coin list would need to be sorted, and decided that they would need to be. The interviewer questioned why they might need to be sorted, and the student wasn't sure, so then decided not to sort it. This indicates that the student didn't have a very clear model of their solution or the problem. There *were* valid reasons to sort the list, for some solutions, but the student did not appear to have a clear idea of what they were.

The interviewer also asked any students who finished their written solutions if they knew any ways they would improve them. Those with higher spatial skills appeared to have developed a better model of the problem domain because they usually were able to indicate some ways that the solution could have been improved if they were to try again or had more time. For example, three students (26c, 26d, 28a) all indicated that the time complexity of the program could be reduced by tracking combinations of coins which added up to intermediate values to avoid duplication of additive procedures. This indicates that even as they were building their solution, with some additional thought they were able to expand their model to improve it.

Another event observed among lower spatial skills scoring students was an apparent over-reliance on memorisation and recall. Two students in particular (17, 20a) appeared to be trying to recall a potential solution that they had seen before rather than trying to build a solution from first principles: *"I'm trying to remember the permutations algorithm from [first year module]" (20a)*. This seems to indicate that they relied heavily on other known solutions or patterns – which were note well-developed enough in their memory to effectively recall – and were not capable of or comfortable with building solutions from scratch.

5.3 Additional Observations

This section describes some additional observations made during the exercise sessions or in relation to aspects of the study which were not part of the originally planned observations. These observations were not included in the original research methodology and were not explicitly being monitored, so should not be considered as being presented with the same precedence as the other results presented earlier in this section. However, these observations are still interesting to the authors and may provide further evidence of students with higher spatial skills demonstrating the strategies associated with SpES while the lower scoring students do not.

Since the recordings were audio only, the interviewer took note of physical cues and their time codes, particularly when students pointed to artefacts on paper referring to them as "this" or "that", so that during transcription an explicit note could be added about what the students were referencing. One of the events being explicitly recorded by the interviewer was when students started writing on the paper provided – as well as recording the kind of writing: code, pseudocode, doodles, diagrams, etc. – so as to differentiate between silences on the tape (i.e. to determine whether students busy writing or just not verbalising, for example). In doing so, we have accurate measures of when students started putting their solutions to paper. In analysing these timestamps, for both the first and third tasks, most high-scoring spatial skill students started writing substantially later than those with low spatial skills: high spatial skills scorers spent time reading the task, clarifying any queries and verbalising both the task and their planned solution. Those with lower spatial skills were fairly quick to start writing once they felt that they had a good grip on the task, but as is evidenced by other observations, this didn't necessarily mean that they would be able to come up with a solution more quickly (or at all).

Another observation was that some students made use of hand movements more frequently. The interviewer only noticed this after two interviews had already been completed, so we do not have a complete set of data, but the interviewer did begin to record all instances of expressive hand movements the students made. Of the students recorded, the use of gesture was much more frequent and expressive in those with higher spatial skills (one student (27) accidentally knocked the dictaphone off the table in describing their solution plan). Students with lower spatial skills were more likely to keep their hands still while describing their processes. While we cannot say anything firm about this relationship because we did not initially consider gesture and did not collect rich data on it, the research on this area suggests that gesture is both valuable for generating understanding of abstract concepts in CS [49–51] and that gesture is useful in both developing and recalling encodings in memory [9].

We can also make some observations about how well students tackled the tasks themselves. Students didn't have machines to test their solutions on so had to keep track of whether their solutions worked themselves, but the paper versions were analysed by the authors to determine if they would work without significant changes. Of the 13 students involved, only two – both lower spatial skills scorers (16, 20a) – were unable to come up with working solutions to the first problem. Four students were unable to fully explain the recursive solution in the second task: two were the same students from before, and the other two had spatial skills scores of 17 and 20. There were some details about the recursive solution that some of the students with higher spatial skills struggled to fully articulate, but generally they were clear on how the program worked and could explain how the results were found – in the case of the four students who could not explain it, they very plainly indicated that they couldn't understand the procedure (*"I'm not sure why we need to do n1 and n2 at the same time, at all. It will take a lot more thinking." (20b)*)

It should be noted, however, that these assumptions of completeness and correctness are not completely dependable. Most students did not write code in any formal sense or stick to consistent syntax. Instead, they used representations of constructs and concepts from programming languages which they were familiar with, sometimes using arrows or flow diagrams to demonstrate operations. Therefore, it is not really possible to determine whether the produced programs were *truly* correct. Rather, the authors can infer from the structure and the demonstration of the written operations whether the student had a complete and correct solution to the problems, though this may not have translated into a fully complete solution.

Even with the time spent planning, students with higher spatial skills scores also tended to complete the tasks more quickly. The students were aware that they had 10 minutes for each task so were not explicitly told that the task was timed, which may have affected the time that they took (i.e. the students with lower spatial skills *may* have been able to complete the tasks more quickly, but did not feel compelled to move quickly). To give clear distinctions, the fastest students to put working solutions to the first task to paper, with speeds of 4:17, 4:47 and 5:10 had spatial skills scores of 30, 26(b) and 29 respectively, whereas only two of the four students scoring below 21 were able to complete the task at all within the allotted 10 minutes, with times of just over 8 minutes and just under 10 minutes.

6 DISCUSSION

6.1 Participants' Spatial Skills

The spatial skills of the participants in this research were high. The average PSVT:R score was 24.4 out of a possible 30. This is to be expected, based on the work of Parkinson & Cutts [41] and Wai, Lubinski & Benbow [68] who showed that spatial skills on average tend to be higher in those further along in their academic STEM careers. However, this does mean that the results may be somewhat skewed towards the upper bound of spatial ability, meaning that we may not be able to draw as many distinct conclusions about those with weaker spatial skills as if the distribution were more spread across lower spatial skills scores. In the analysis of the think-aloud transcripts we tend to group the students as low and high spatial skills scorers with high scoring 21+ in the PSVT:R, leaving only 4 students in the lower spatial skills group and therefore fewer data points to examine.

Regardless, it is interesting that there are still clear differences between the higher scorers and the lower scorers across so many categories, even though the lower scorers have fairly high skills. Sorby's work has always split first-year engineering students into three categories by spatial skills: 18 and below require training, 19–21 inclusive are "marginally passing" students for whom spatial skills training will likely be beneficial but is not mandatory, and 22 and above indicates a solid pass where additional training should not be required [53]. But as we move beyond first year, we can see that these divides may not be appropriately scaled for later study: the majority of participants were in the "solid pass" threshold, but we still see fairly distinct divides within this group. The students scoring 26–30 exhibit different behaviours from those around the 21–25 mark, suggesting that while the categorical breakdowns may be suitable to identify which students have *enough* to succeed in their STEM study, they may still benefit from *higher* spatial skills. This, of course, with the caveat that we still expect that students with lower spatial skills *can* succeed in further STEM study – even below Sorby's lowest threshold – but will have a harder time than those who pass because they will struggle with their non-verbal encoding strategies.

6.2 Strategies for Programming Problem Solving Directly Related to SpES

Recall that our research questions related to whether non-verbal representations predicted by SpES – such as mental model building – were evident in programming problem solving and whether the use of these representations was associated with spatial skills. While there were some instances of students straying from the norms described above, generally the differences between high- and low-scoring spatial skills students were stark. Those with lower spatial skills were less likely to be able to even complete the assigned exercises while those with higher spatial skills were able to complete them well within the allotted time, which is only the beginning of the differences between the two skill groups. We observed many expected behaviours as per SpES: students with high spatial skills created well developed mental models of both the problem domain and their solutions which they were able to adjust and abstract from, while those with poorer spatial skills focused on the text surface, struggled to make alterations to their mental models and got stuck in unproductive loops. We also observed students with better spatial skills maintaining their models between questions, utilising them in the solving of the next problems, while students with lower spatial skills almost appeared to "flush" their memory between exercises to free up more working memory space.

There were several cognitive mechanisms identified from the events captured, such as evidence of chunking, cognitive overload and rapid, flexible development of mental models. The presence of all of these can be explained with SpES. Chunking is something Margulieux talks about explicitly as being valuable for STEM achievement, and also forms a basis for the other mechanisms observed. Chunking makes more effective use of working memory, allowing for greater intrinsic or extrinsic cognitive load to be taken onboard; conversely, if working memory is full due to inefficient chunking, students will experience cognitive overload and will not be able to process new information [58], and having free working memory space is necessary for updating mental models [65]. This ties into mental model construction as well: cognitive overload can inhibit the initial construction of mental models [23]. In each of these instances, SpES provides a reason for why better spatial skills – by way of non-verbal representations – contribute to more effective use of these cognitive mechanisms, which have been observed in action in these students.

Therefore, these results appear to indicate that the research questions posed in this paper can be answered affirmatively: non-verbal representations related to SpES are evident in advanced programming problem solving and their use appears to be connected with spatial skills.

6.3 Real-Time Versus Historic Representations

There is an unanswered question inherent in this work: do non-verbal encoding strategies support in-the-moment mental model building, or do students with better non-verbal encoding strategies construct more robust and complete models in *initial learning*, which they can store to long-term memory and effectively recall? We have framed our results and discussion around the former, though the latter could still be how the students solve problems in practice. Students with better long-term memory encodings (or schemata [12], or plans [47]), which they created possibly years prior, can more rapidly recall them and apply them in context whereas students who have only been able to develop incomplete or inefficient long-term memory representations will only find them of limited use in practice.

Both possibilities align with Margulieux's theory. Margulieux discusses mental model building with respect to problem solving explicitly, but also discusses how spatial skills may also lead to "stronger and faster connections which lead to more efficient learning" to be recalled later [28].

We theorise that either of these kinds of representations – transient, real-time generated representations or long-term, stored representations being recalled – or perhaps combinations of both, could be being applied by our students in this study and it would not necessarily affect the outcome. The result would be the same: students with better spatial skills will end up with more robust models which take up less working memory whether they are recalled or generated on the spot. While we do not consider the distinction to seriously affect this particular study, we do think that it has some interesting implications, which are discussed in section 6.7.

6.4 Contributing to Theory

Malmi *et al.* highlighted a concerning lack of effective use of theory in CS education [27]. While Nelson & Ko argue that there are cases where theory can hinder design [31], spatial skills researchers must be cautious about this. Even as more work emerges demonstrating that there is benefit in developing the spatial skills of students, without a good understanding of and and trust in the underlying theory, we cannot be clear about *why* spatial skills are of value. Spatial skills are "off the beaten path" in terms of skills one might think of as valuable in CS, which is all the more reason for understanding the relationship as best as we can.

Margulieux explicitly tries to address this with SpES. The theory clearly and deliberately explores the relationship through cognitive mechanisms and provides a reasonable, well-informed conclusion on possibilities about the relationship, which was well received by the ICER community (winning the John Henry award in 2019, a community voted award for the paper which best "pushes the upper limits of our pedagogy"[3]).

This paper provides a significant empirical study which has not been conducted before, is a test of the theory and leads to possible theory modifications in relation to advanced students. The work has been conducted in the spirit of continuing to strengthen theory to push our understanding of the *why* further and to ensure solid foundations on which to build future pedagogy.

6.5 Limitations and Threats to Validity

The greatest challenge (but also, the greatest potential value) in this work was exposing the hidden cognition that students were using. We argue that the events we identified in the results section of this work demonstrate the kind of cognition involved in SpES taking place, such as the construction and running of mental models. This could be debated. The events we selected were indicative of our perspectives and are, as such, subjective. This is a characteristic of conducting qualitative analysis in this form, which we have attempted to address through confirming our event coding strategy, as described in section 4.4.

However, it's important to note the relationship between the two authors here: PhD student and supervisor. This relationship carries "shared baggage" in the sense that both authors had many discussions about the concepts underpinning this study prior even to its inception. The is a good chance that the two authors are more likely to code events similarly than, say, a non-related third party, which is a threat to the replication potential of this kind of study. A more comprehensive rubric is required to crystallise and expose any hidden shared ideology about the cognition involved so that anyone could attempt to code the events in the same way and achieve similar results.

Due to the open call for participation in the experiment, it is possible that there was discussion or collusion between students who had completed the study and students who had yet to take part. It is possible that some of the students shared answers to the PSVT:R or discussed the problem with other students. We reduced the risk of this, however, by conducting the sessions chronologically close together, over just three days. This limited the amount of time permitted for discussion or sharing of ideas. There was also no extrinsic motivation to attempt to "cheat" in the study. The students were advised that they would be rewarded with an Amazon voucher for the completion of the study and there was no indication that correctness or completeness was a requirement. In fact, the study was described in very loose terms in the advertisement, meaning that students probably were not aware of the structure of the study enough to know that there was any benefit in preparing or finding out more about what was being done. Therefore, while possible, the chances of collusion were low.

One of the reasons to select fourth year students was to ensure a minimum level of programming language fluency in a shared language so that the language could be used in one of the exercises. However, this also introduces a threat to validity: fourth year students have a wide range of skills and experiences based on many factors: their time spent studying, the kinds of elective modules they have taken, whether they have done a work placement, etc. All these factors – and more – could contribute to their ability to strategise and solve the problems presented. This kind of challenge is inherent in any study where not every confounding factor can be controlled for. Generally, the events detected fall in line with predictions based on spatial skills, but we cannot rule out other possible factors affecting the way the students solved problems.

6.6 Implications

This paper demonstrates that students' ability to create and modify complex non-verbal representations as they solve problems is related to their spatial ability, as predicted by SpES. Furthermore, these students were final year undergraduates with four years of CS learning experience, which means that these skills are not just valuable in the introductory stages. It has implications on how students solve problems all the way through their programme and perhaps beyond as they transition to industry.

It also demonstrates that while "good" spatial skills (as measured on a scale applied to introductory students) might be enough to get by, students scoring top scores in the PSVT:R outperformed their peers who were still above the threshold typically used to declare students as having strong enough skills to succeed in their programme. Clearly, these students *did* succeed in their programme, or at least managed to get to the final semester of their four-year enrolment, but still had some difficulties in forming good non-verbal representations. This indicates that perhaps upfront spatial

[3] https://archive.icer.acm.org/general-info/paper-awards/

skills training would be of benefit to more students than have previously been involved in research studies, or perhaps that more advanced spatial skills training periodically conducted throughout the programme would help students to achieve higher spatial skills standards, which should help them in their non-verbal representation building as their programme content becomes more computationally challenging.

6.7 Future Work

What this work lacks is a clear, concise and easily transferable definition of the encoding and orientation skills Margulieux describes in SpES. We have attempted to highlight perceived instances of these skills, but a clearer, more concrete rubric of the kinds of strategies and how they manifest would strengthen these findings and make them more easily and reliably replicable in other contexts. This would not be a simple undertaking, especially if the goal is a comprehensive rubric of *all* possible practices.

As already mentioned, it is not possible to determine from this study whether students with good non-verbal representation strategies succeeded in the programming exercises because they were good at constructing transient mental models to solve the problems on the fly, or if they had constructed better initial representations of the problem area which they could recall and apply rapidly, or a combination of the two. Work should be done to examine this to better understand spatial skills in practice, so that we can determine *when* spatial skills are of value – in initial learning or in practice – so that we can better decide when to conduct spatial interventions: as early as possible to support learning in the earliest stages, or as skills which can be developed gradually as problem solving tasks become more challenging throughout a semester, year or programme.

This study also contributes to the mounting body of work indicating that while students with experience in CS appear to rely on spatial skills less than students with no experience, particularly in familiar problem domains, spatial skills are still of value for some tasks in new problem contexts. As this body of work grows, it raises questions about the nature of work in the software industry and whether spatial skills would be valuable for employees in software development roles. In particular, the "experts" used in this study were in fact advanced students, rather than professionals in industry. While an advanced student would be expected to exhibit some expert-level behaviours, particularly as they are very probably about to enter the professional workforce, it would also be valuable to explore the relationship between spatial skills and task performance for better-established industry professionals, particularly ones who are truly considered experts in their field.

7 CONCLUSION

There is still much to understand about the relationship between spatial skills and computing success, but Margulieux's SpES goes a good distance to clarify how the relationship manifests. This work lends more credibility to Margulieux's theory, providing a close look "under the hood" at student cognition through the lens of SpES. It demonstrates that we can see the strategies being employed in practice and being demonstrated by final year students, not just beginners. This has implications for the value of spatial skills at many levels of CS education and their application in practice: even for students on the cusp of graduating and starting their careers across the computing workforce, spatial skills appear to be useful in solving programming problems.

REFERENCES

[1] John R. Anderson. 2015. *Cognitive psychology and its implications* (eighth edition ed.). Worth Publishers, New York. OCLC: ocn899031525.

[2] Alan Baddeley. 1992. Working Memory. *Science* 255, 5044 (Jan. 1992), 556–559. https://doi.org/10.1126/science.1736359

[3] G. K. Bennett, H. G. Seashore, and A. G. Wesman. 1947. *Differential aptitude tests*. Psychological Corporation, San Antonio, TX, US.

[4] Ryan Bockmon, Stephen Cooper, Jonathan Gratch, Jian Zhang, and Mohsen Dorodchi. 2020. Can Students' Spatial Skills Predict Their Programming Abilities?. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*. ACM, Trondheim Norway, 446–451. https://doi.org/10.1145/3341525.3387380

[5] Ryan Bockmon, Stephen Cooper, William Koperski, Jonathan Gratch, Sheryl Sorby, and Mohsen Dorodchi. 2020. A CS1 Spatial Skills Intervention and the Impact on Introductory Programming Abilities. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. ACM, Portland OR USA, 766–772. https://doi.org/10.1145/3328778.3366829

[6] John B. Carroll. 1993. *Human Cognitive Abilities: A Survey of Factor-Analytic Studies* (1 ed.). Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511571312

[7] CEEB. 1939. *Special Aptitude Test in Spatial Relations (MCT)*. Technical Report. Developed by the College Entrance Examination Board.

[8] Alexandra O. Constantinescu, Jill X. O'Reilly, and Timothy E. J. Behrens. 2016. Organizing conceptual knowledge in humans with a gridlike code. *Science* 352, 6292 (June 2016), 1464–1468. https://doi.org/10.1126/science.aaf0941

[9] Susan Cook and Kimberly M. Fenn. 2017. Chapter 6. The function of gesture in learning and memory. In *Gesture Studies*, R. Breckinridge Church, Martha W. Alibali, and Spencer D. Kelly (Eds.). Vol. 7. John Benjamins Publishing Company, Amsterdam, 129–153. https://doi.org/10.1075/gs.7.07coo

[10] Stephen Cooper, Karen Wang, Maya Israni, and Sheryl Sorby. 2015. Spatial Skills Training in Introductory Computing. In *Proceedings of the eleventh annual International Conference on International Computing Education Research*. ACM, Omaha Nebraska USA, 13–20. https://doi.org/10.1145/2787622.2787728

[11] Anthony Cox, Maryanne Fisher, and Philip O'Brien. 2005. Theoretical Considerations on Navigating Codespace with Spatial Cognition. In *Proceedings of the 17th Annual Workshop of the Psychology of Programming Interest Group*. Psychology of Programming Interest Group, Brighton, UK, 9. https://ppig.org/papers/2005-ppig-17th-cox/

[12] Benedict Du Boulay. 1986. Some Difficulties of Learning to Program. *Journal of Educational Computing Research* 2, 1 (Feb. 1986), 57–73. https://doi.org/10.2190/3LFX-9RRF-67T8-UVK9

[13] R. B. Ekstrom, J. W. French, and H. H. Harman. 1976. *Manual for Kit of Factor-Referenced Cognitive Tests*. Educational Testing Service, Princeton, New Jersey.

[14] John Eliot, Ian Macfarlane Smith, and Ian Macfarlane Smith. 1983. *An international directory of spatial tests*. NFER-Nelson, Windsor, Berks.

[15] K. Anders Ericsson and Walter Kintsch. 1995. Long-term working memory. *Psychological Review* 102, 2 (1995), 211–245. https://doi.org/10.1037/0033-295X.102.2.211

[16] K. Anders Ericsson and Herbert A. Simon. 1980. Verbal reports as data. *Psychological Review* 87, 3 (May 1980), 215–251. https://doi.org/10.1037/0033-295X.87.3.215

[17] Sally Fincher, Bob Baker, Ilona Box, Quintin Cutts, Michael de Raadt, Patricia Haden, John Hamer, Margaret Hamilton, Raymond Lister, and Marian Petre. 2005. *Programmed to succeed?: A multi-national, multi-institutional study of introductory programming courses*. Technical report. University of Kent, Canterbury, United Kingdom. https://kar.kent.ac.uk/14335/

[18] Roland. Guay, Purdue Research Foundation., Educational Testing Service., and Test Collection. 1976. *Purdue spatial visualization test*. Purdue University, [West Layfette, Ind.].

[19] Torkel Hafting, Marianne Fyhn, Sturla Molden, May-Britt Moser, and Edvard I. Moser. 2005. Microstructure of a spatial map in the entorhinal cortex. *Nature* 436, 7052 (Aug. 2005), 801–806. https://doi.org/10.1038/nature03721

[20] Mary Hegarty and David Waller. 2004. A dissociation between mental rotation and perspective-taking spatial abilities. *Intelligence* 32, 2 (2004), 175–191. https://doi.org/10.1016/j.intell.2003.12.001

[21] Sue Jones and Gary Burnett. 2007. Spatial skills and navigation of source code. In *Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education - ITiCSE '07*. ACM Press, Dundee, Scotland, 231. https://doi.org/10.1145/1268784.1268852

[22] Sue Jones and Gary Burnett. 2008. Spatial Ability And Learning To Program. *Human Technology: An Interdisciplinary Journal on Humans in ICT Environments*

4, 1 (2008), 47–61. https://doi.org/10.17011/ht/urn.200804151352

[23] Paul A. Kirschner, John Sweller, and Richard E. Clark. 2006. Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching. *Educational Psychologist* 41, 2 (June 2006), 75–86. https://doi.org/10.1207/s15326985ep4102_1

[24] Maria Kozhevnikov and Mary Hegarty. 2001. A dissociation between object manipulation spatial ability and spatial orientation ability. *Memory & Cognition* 29, 5 (2001), 745–756. https://doi.org/10.3758/BF03200477

[25] Ken Liu, Burkhard C. Wünsche, and Andrew Luxton-Reilly. 2022. Relationship Between Spatial Skills and Performance in Introductory Computer Graphics. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1*. ACM, Dublin Ireland, 304–310. https://doi.org/10.1145/3502718.3524756

[26] Anna Ly, Jack Parkinson, Quintin Cutts, Michael Liut, and Andrew Petersen. 2021. Spatial Skills and Demographic Factors in CS1. In *Koli Calling*. ACM, Joensuu, Finland, 1–10. https://doi.org/10.1145/3488042.3488049

[27] Lauri Malmi, Judy Sheard, Päivi Kinnunen, Simon, and Jane Sinclair. 2019. Computing Education Theories: What Are They and How Are They Used?. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (2019-07-30). ACM, Toronto ON Canada, 187–197. https://doi.org/10.1145/3291279.3339409

[28] Lauren E. Margulieux. 2019. Spatial Encoding Strategy Theory: The Relationship between Spatial Skill and STEM Achievement. In *Proceedings of the 2019 ACM Conference on International Computing Education Research*. ACM, Toronto ON Canada, 81–90. https://doi.org/10.1145/3291279.3339414

[29] Richard E. Mayer, Jennifer L. Dyck, and William Vilberg. 1986. Learning to program and learning to think: what's the connection? *Commun. ACM* 29, 7 (1986), 605–610. https://doi.org/10.1145/6138.6142

[30] Mark G. McGee. 1979. Human spatial abilities: Psychometric studies and environmental, genetic, hormonal, and neurological influences. *Psychological Bulletin* 86, 5 (1979), 889–918. https://doi.org/10.1037/0033-2909.86.5.889

[31] Greg L. Nelson and Amy J. Ko. 2018. On Use of Theory in Computing Education Research. In *Proceedings of the 2018 ACM Conference on International Computing Education Research* (2018-08-08). ACM, Espoo Finland, 31–39. https://doi.org/10.1145/3230977.3230992

[32] Allen Newell and Herbert A. Simon. 1976. Computer science as empirical inquiry: symbols and search. *Commun. ACM* 19, 3 (March 1976), 113–126. https://doi.org/10.1145/360018.360022

[33] Donald A Norman. 2014. Some observations on mental models. In *Mental models*. Psychology Press, New York, 15–22.

[34] John O'Keefe. 1976. Place units in the hippocampus of the freely moving rat. *Experimental Neurology* 51, 1 (Jan. 1976), 78–109. https://doi.org/10.1016/0014-4886(76)90055-8

[35] John O'Keefe, Neil Burgess, James G. Donnett, Kathryn J. Jeffery, and Eleanor A. Maguire. 1998. Place cells, navigational accuracy, and the human hippocampus. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 353, 1373 (Aug. 1998), 1333–1340. https://doi.org/10.1098/rstb.1998.0287

[36] J. O'Keefe and J. Dostrovsky. 1971. The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Research* 34, 1 (Nov. 1971), 171–175. https://doi.org/10.1016/0006-8993(71)90358-1

[37] Miranda C. Parker, Mark Guzdial, and Shelly Engleman. 2016. Replication, Validation, and Use of a Language Independent CS1 Knowledge Assessment. In *Proceedings of the 2016 ACM Conference on International Computing Education Research*. ACM, Melbourne VIC Australia, 93–101. https://doi.org/10.1145/2960310.2960316

[38] Miranda C. Parker, Amber Solomon, Brianna Pritchett, David A. Illingworth, Lauren E. Marguilieux, and Mark Guzdial. 2018. Socioeconomic Status and Computer Science Achievement: Spatial Ability as a Mediating Variable in a Novel Model of Understanding. In *Proceedings of the 2018 ACM Conference on International Computing Education Research*. ACM, Espoo Finland, 97–105. https://doi.org/10.1145/3230977.3230987

[39] Jack Parkinson. 2022. What does Space look like in CS? Mapping out the Relationship between Spatial Skills and CS Aptitude. In *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 2*. ACM, Lugano and Virtual Event Switzerland, 46–47. https://doi.org/10.1145/3501709.3544284

[40] Jack Parkinson, Ryan Bockmon, Quintin Cutts, Michael Liut, Andrew Petersen, and Sheryl Sorby. 2021. Practice report: six studies of spatial skills training in introductory computer science. *ACM Inroads* 12, 4 (Nov. 2021), 18–29. https://doi.org/10.1145/3494574

[41] Jack Parkinson and Quintin Cutts. 2018. Investigating the Relationship Between Spatial Skills and Computer Science. In *Proceedings of the 2018 ACM Conference on International Computing Education Research*. ACM, Espoo Finland, 106–114. https://doi.org/10.1145/3230977.3230990

[42] Jack Parkinson and Quintin Cutts. 2020. The Effect of a Spatial Skills Training Course in Introductory Computing. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*. ACM, Trondheim Norway, 439–445. https://doi.org/10.1145/3341525.3387413

[43] Jack Parkinson and Quintin Cutts. 2022. Relationships between an Early-Stage Spatial Skills Test and Final CS Degree Outcomes. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education*. ACM, Providence RI USA, 293–299. https://doi.org/10.1145/3478431.3499332

[44] Jack Parkinson, Quintin Cutts, and Steve Draper. 2020. Relating Spatial Skills and Expression Evaluation. In *United Kingdom & Ireland Computing Education Research conference*. ACM, Glasgow United Kingdom, 17–23. https://doi.org/10.1145/3416465.3416473

[45] Herbert L. Pick and Jeffrey J. Lockman. 1983. Map Reading and Spatial Cognition: Discussion. In *Spatial Orientation*, Herbert L. Pick and Linda P. Acredolo (Eds.). Springer US, Boston, MA, 219–224. https://doi.org/10.1007/978-1-4615-9325-6_10

[46] Bruno Poucet. 1993. Spatial cognitive maps in animals: new hypotheses on their structure and neural mechanisms. *Psychological review* 100, 2 (1993), 163. Publisher: American Psychological Association.

[47] Anthony Robins, Janet Rountree, and Nathan Rountree. 2003. Learning and Teaching Programming: A Review and Discussion. *Computer Science Education* 13, 2 (June 2003), 137–172. https://doi.org/10.1076/csed.13.2.137.14200

[48] Simon, Sally Fincher, Anthony Robins, Bob Baker, IIona Box, Quintin Cutts, Michael De Raadt, Patricia Haden, John Hamer, Margaret Hamilton, and others. 2006. Predictors of success in a first programming course. In *Conferences in Research and Practice in Information Technology* (2006), Vol. 52. Australian Computer Society, Hobart, Australia, 189–196.

[49] Amber Solomon. 2019. The Role of Spatial Representations in CS Teaching and CS Learning. In *2019 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, Memphis, TN, USA, 237–238. https://doi.org/10.1109/VLHCC.2019.8818785

[50] Amber Solomon. 2021. *Embodiment in Computer Science Learning: How Space, Metaphor, Gesture, and Sketching Support Student Learning*. Ph. D. Dissertation. Georgia Institute of Technology.

[51] Amber Solomon, Mark Guzdial, Betsy DiSalvo, and Ben Rydal Shapiro. 2018. Applying a Gesture Taxonomy to Introductory Computing Concepts. In *Proceedings of the 2018 ACM Conference on International Computing Education Research*. ACM, Espoo Finland, 250–257. https://doi.org/10.1145/3230977.3231001

[52] Maarten W. van Someren, Yvonne F. Barnard, and Jacobijn A. Sandberg. 1994. *The think aloud method: a practical guide to modelling cognitive processes*. Number 13 in Knowledge based systems. Academic Press, London.

[53] Sheryl Sorby. 1999. Developing 3-D Spatial Visualisation Skills. *The Engineering Design Graphics Journal* 63, 2 (1999), 21–32.

[54] Sheryl Sorby, Norma Veurink, and Scott Streiner. 2018. Does spatial skills instruction improve STEM outcomes? The answer is 'yes'. *Learning and Individual Differences* 67 (2018), 209–222. https://doi.org/10.1016/j.lindif.2018.09.001

[55] Sheryl A. Sorby. 2009. Educational Research in Developing 3-D Spatial Skills for Engineering Students. *International Journal of Science Education* 31, 3 (2009), 459–480. https://doi.org/10.1080/09500690802595839

[56] Sheryl Ann Sorby, Anne Frances Wysocki, and Beverly Gimmestad Baartmans. 2003. *Introduction to 3-D spatial visualization: an active approach*. Thomson/Delmar Learning, Clifton Park, N.Y. OCLC: ocm51301756.

[57] Donald E. Super and Paul B. Bachrach. 1957. *Scientific careers and vocational development theory: A review, a critique and some recommendations.* Columbia Univer., Oxford, England. Pages: xii, 135.

[58] John Sweller. 2011. Cognitive Load Theory. In *Psychology of Learning and Motivation*. Vol. 55. Elsevier, Amsterdam, 37–76. https://doi.org/10.1016/B978-0-12-387691-1.00002-8

[59] Mirko Thalmann, Alessandra S. Souza, and Klaus Oberauer. 2019. How does chunking help working memory? *Journal of Experimental Psychology: Learning, Memory, and Cognition* 45, 1 (Jan. 2019), 37–55. https://doi.org/10.1037/xlm0000578

[60] Charles P. Thompson, Thaddeus M. Cowan, and Jerome Frieman. 1993. *Memory search by a memorist*. Erlbaum, Hillsdale, NJ.

[61] Denise Tolhurst, Bob Baker, John Hamer, Ilona Box, Raymond Lister, Quintin Cutts, Marian Petre, Michael De Raadt, Anthony Robins, Sally Fincher, and others. 2006. Do map drawing styles of novice programmers predict success in programming? A multi-national, multi-institutional study. *Research in Practice in Information Technology* 52 (2006), 213–222.

[62] Tom Trabasso and Joseph P. Magliano. 2023. Conscious understanding during comprehension. *Discourse Processes* 21, 3 (2023), 255–287. https://doi.org/10.1080/01638539609544959

[63] David H. Uttal and Cheryl A. Cohen. 2012. Spatial Thinking and STEM Education. In *Psychology of Learning and Motivation*. Vol. 57. Elsevier, Amsterdam, 147–181. https://doi.org/10.1016/B978-0-12-394293-7.00004-2

[64] David H. Uttal, Nathaniel G. Meadow, Elizabeth Tipton, Linda L. Hand, Alison R. Alden, Christopher Warren, and Nora S. Newcombe. 2013. The malleability of spatial skills: A meta-analysis of training studies. *Psychological Bulletin* 139, 2 (March 2013), 352–402. https://doi.org/10.1037/a0028446

[65] Derick F. Valadao, Britt Anderson, and James Danckert. 2015. Examining the influence of working memory on updating mental models. *Quarterly Journal of Experimental Psychology* 68, 7 (July 2015), 1442–1456. https://doi.org/10.1080/17470218.2014.989866

[66] Steven G. Vandenberg and Allan R. Kuse. 1978. Mental Rotations, a Group Test of Three-Dimensional Spatial Visualization. *Perceptual and Motor Skills* 47, 2 (1978), 599–604. https://doi.org/10.2466/pms.1978.47.2.599

[67] Norma Veurink and Sheryl Sorby. 2011. Raising the Bar? Longitudinal Study to Determine which Students Would Benefit Most from Spatial Training. In *2011 ASEE Annual Conference & Exposition Proceedings* (2011-06). ASEE Conferences, Vancouver, BC, 22.1210.1–22.1210.13. https://doi.org/10.18260/1-2--18592

[68] Jonathan Wai, David Lubinski, and Camilla P. Benbow. 2009. Spatial ability for STEM domains: Aligning over 50 years of cumulative psychological knowledge solidifies its importance. *Journal of Educational Psychology* 101, 4 (2009), 817–835. https://doi.org/10.1037/a0016127

[69] Steffen Werner, Bernd Krieg-Brückner, Hanspeter A. Mallot, Karin Schweizer, and Christian Freksa. 1997. Spatial Cognition: The Role of Landmark, Route, and Survey Knowledge in Human and Robot Navigation1. In *Informatik '97 Informatik als Innovationsmotor*, Matthias Jarke, Klaus Pasedach, and Klaus Pohl (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 41–50. https://doi.org/10.1007/978-3-642-60831-5_8 Series Title: Informatik aktuell.

[70] So Yoon Yoon. 2011. *Psychometric properties of the Revised Purdue Spatial Visualization Tests: Visualization of Rotations (the Revised PSVT:R).* Ph. D. Dissertation. Purdue University.

A PART 1: PROBLEM SPECIFICATION

Given a finite collection of coins C, and a positive value N, in what combinations of coins can we pay the value N precisely using only the coins in C?

Using a different coin of the same denomination counts as a different solution (e.g., see that there are two `[10]` solutions in Example 1 because there are two coins of value 10)

Your solution can assume that there exists a list of coins available and a number which must be paid.

Example 1

```
C = [10, 10, 2, 2, 2, 2, 2, 5, 5]
N = 10
Solution: 4 --> [10], [10], [2, 2, 2, 2, 2],
    [5, 5]
```

Example 2

```
C = [5, 3, 1, 3, 3, 2]
N = 10
Solution: 4 --> [5, 3, 2], [5, 3, 2], [5, 3,
    2], [3, 1, 3, 3]
```

B PART 2: POTENTIAL SOLUTION

```
def sum(mylist, start, finish):
    total = 0
    for i in range(start, finish):
        total += mylist[i]
    return total


def min(mylist, start, finish):
    mini = mylist[start]
    for i in range(start, finish):
        if mini > mylist[i]:
            mini = mylist[i]
    return mini
```

```
def solutions(mysolutions, mylist,
    number_index, N):
    if N == 0:
        print(mysolutions)
        return 1
    elif N < 0:
        return 0
    elif number_index >= len(mylist):
        return 0
    elif sum(mylist, number_index, len(
        mylist)) < N:
        return 0
    elif min(mylist, number_index, len(
        mylist)) > N:
        return 0
    else:
        mysolutions.append(mylist[
            number_index])
        n1 = solutions(mysolutions, mylist,
            number_index + 1, N - mylist[
            number_index])
        mysolutions.pop()
        n2 = solutions(mysolutions, mylist,
            number_index + 1, N)
        return n1 + n2
```

Questions asked by the interviewer to capture additional details about the student's understanding, which were not visible to the student)

(1) What does the function ultimately return?
(2) How is the `number_index` variable used?
(3) Can you describe the purpose of each `if...` `elif...` clause more explicitly?
(4) Why are `sum` and `min` used?
(5) What does `N` denote?
(6) Why are we modifying the value of `N` in this recursive call?
(7) Can you think of any specific improvements which could be made to the function?

C PART 3: ALTERED PROBLEM SPECIFICATION

Making changes to either your original solution or the recursive solution, make the following adjustment to the program:

Given a set of coins C, and a positive value N, in what combinations of coins can we pay the value N precisely using any number of each of the coins in C?

Assume that the set C now lists available denominations of coins, but any number of coins can be used to create the required value N.

There should be no duplicate solutions.

Example 1

```
C = {10, 2, 5}
N = 10
Solution: 3 --> [10], [5, 5], [2, 2, 2, 2,
    2]
```

Example 2

```
C = {5, 3, 2}
N = 15
```

```
Solution: 7 --> [5, 5, 5], [3, 3, 3, 3, 3],
    [5, 2, 2, 2, 2, 2], [5, 5, 3, 2], [3, 3,
    3, 2, 2, 2], [5, 3, 3, 2, 2], [3, 2, 2,
    2, 2, 2, 2]
```