

ניתוח מבוסס-לוגים של פרקטיקות חשיבה חישובית: המקרה של ניפוי שגיאות בתכנות מבוסס-בלוקים

ארנון הרשקוביץ
אוניברסיטת תל אביב
arnonhe@tauex.tau.ac.il

גורן דיין
אוניברסיטת תל אביב
gorendayan@mail.tau.ac.il

Log-Based Analysis of Computational Thinking Practices: The case of Debugging in Block-Based Programming

Goren Dayan
Tel Aviv University
gorendayan@mail.tau.ac.il

Arnon Hershkovitz
Tel Aviv University
arnonhe@tauex.tau.ac.il

Abstract

Computational thinking (CT) is a thinking skill that involves a set of problem-solving methods for expressing problems and their solutions in ways that a computer could also execute. The main challenge of educators and researchers remained the ability to assess and evaluate the development of CT in young learners. Current instruments are missing the practices and processes which reflect the actual cognitive and intellectual development while performing a learning activity. This difficulty emerges mainly due to the inherent difficulties in capturing detailed process data for large numbers of students. Our focus in this study is the CT Practices dimension that help us understand the process of learning. This is enabled as we take a learning analytics approach, using detailed digital traces of learners in an online programming learning environment. Using these traces, which are automatically and continuously stored in log files, we aim to operationalize well-defined CT practices by calculating log-driven research variables. The research context is Kodetu, a block based programming environment. The research was conducted over an experience study across 189 Israeli students middle school played a Kodetu game. Their programming activities were recorded in a system log files that will serve as the foundation for the research. Using an advanced learning analytics methods for analyzing the log files, the research aim to identify and assess CT practices for Debugging processes. Using machine learning (ML) models we identified different debugging behaviors and strategies. The findings help us provide a better real time support materials and strategies for learners.

Keywords: Computational thinking, Computational thinking practices, Learning analytics, Debugging, Block-based learning environment.

תקציר

חשיבה חישובית (ח"ח) היא מיומנות המתארת את תהליכי החשיבה הכרוכים בפתרון בעיות חישוביות. מחקרים מדגישים את החשיבות בפיתוח כלים להערכה ומדידה של מיומנויות ח"ח כבסיס עובדתי לתרומתם בפתרון בעיות. לשיטות המסורתיות טווח בדיקה מוגבל ורובן מתמקדות בתהליכי ההרכשה הבסיסיים של יכולות ח"ח. המחקר הנוכחי מבקש לעלות רמה נוספת ולהתמקד ביכולת לזהות ולמדוד פרקטיקות ח"ח בסביבת למידה מקוונת תוך שימוש בשיטות מחקר כמותיות חדשניות. המחקר מתבסס על כלים ושיטות של ניתוח קבצי יומן

המתעדים פעילויות גרנטוריות של התלמיד בסביבת הלמידה. המחקר מתמקד בפרקטיקת ניפוי שגיאות (Debugging) כאחת מהמיומנויות המרכזיות ביכולת לטפח כישורי פתרון בעיות. הנתונים שנתחו נאספו במסגרת ניסוי שנערך במהלך 2021 עבור אוכלוסייה של 189 תלמידי חט"ב ממרכז הארץ. במהלך המחקר הופקו קבצי יומן (Log files) על בסיס ביצועי התלמידים בפתרון שלבים בסביבת Kodetu, סביבת תכנות מבוססת בלוקים. המחקר בוצע בגישה כמותית תוך שימוש בכלים מתקדמים מעולם ה-Learning analytics הכוללים ניתוח קבצי יומן ושימוש במודלים של Machine learning. תוך שימוש באופרציונליזציה של פרקטיקות ניפוי שגיאות ואלגוריתמים של אשכול היררכי, עלה בידנו לזהות ולאפיין אסטרטגיות ודפוסי התנהגות המלמדים על הדרך שבה תלמידים מנפים שגיאות בסביבת למידה מקוונת. היכולת לנתח בזמן אמת התנהגויות מורכבות בסביבת למידה מקוונת מאפשרת לנו להבין את דפוסי הפעולה והחשיבה המניעים את התלמיד במהלך פתרון בעיות ולספק בזמן אמת משוב ותמיכה פרסונלית לתלמיד.

מילות מפתח: חשיבה חישובית, פרקטיקות חשיבה חישובית, ניתוחי למידה, ניפוי שגיאות, סביבת למידה מבוססת משחק.

מבוא

חשיבה חישובית (להלן ח"ח), Computational Thinking, היא מיומנות המתארת את תהליכי החשיבה הכרוכים בהגדרה ופתרון בעיות חישוביות. היא כוללת אוסף של אסטרטגיות מנטליות כמו חשיבה פרוצדורלית, פירוק לגורמים, זיהוי דפוסים והפשטה (Wing, 2006, 2010). אחת ההגדרות המקובלות לח"ח היא של ברנאן ורזניק (Brennan & Resnick, 2012) שבחנו את המושג ח"ח בסביבות תכנות מבוסס-בלוקים. הם כללו בהגדרתם את מכלול המרכיבים, התהליכים והפרקטיקות שניתן לזהות בעבודת התלמיד בסביבת הלמידה כדפוסי חשיבה ופעולה המאפיינים ח"ח. ברנאן ורזניק, כמו גם מחקרים רבים בתחום (e.g., Gonzalez, 2015; Korkmaz et al., 2017a, 2017b), מדגישים את החשיבות בפיתוח כלים להערכה ומדידה של מימדים אלו כבסיס להתפתחות של תהליכים קוגניטיביים וחשיבה אלגוריתמית.

מדידת ח"ח בסביבות מקוונות ללימוד תכנות או ח"ח (כדוגמת Scratch, CodeMonkey ועוד) מבוצעת בשיטות וכלי מחקר שונים, החל מכלי מחקר איכותניים כדוגמת תצפיות וראיונות, שיטות מחקר כמותיות כדוגמת ניתוח קבצי יומן, כלים כדוגמת Scrape (Wolz et al., 2011) המבצעים ניתוח של התוצרים שהתלמיד פיתח ועוד. מעט כלים מאפשרים מדידה של ח"ח בסביבות מקוונות ומרביתם מתמקדים בתהליכי הרכשה בסיסיים של יכולות ומיומנויות ח"ח. נדרשת יכולת לזהות ח"ח ברמות שונות, בעיקר כזו שתספק לנו הבנה לגבי התפתחות תהליכי למידה וחשיבה אצל התלמיד. מטרת המחקר היא לזהות ולאפיין פרקטיקות ח"ח – התנהגויות מורכבות המבוססות על מדידה של משתנים מרובים וניתוחים מורכבים. ניתוח קבצי יומן המכילים תיעוד מפורט של פעילויות גרנטוריות המבוצעות על ידי התלמיד בסביבה מקוונת מאפשר זאת, ובכך נתמקד במחקר הנוכחי.

המחקר מתבסס על המסגרת התיאורטית של ברנאן ורזניק ומתמקד בזיהוי, מדידה והערכה של פרקטיקת ניפוי שגיאות (Debugging) בקרב משתמשים בסביבת למידה תכנותית משחקית לח"ח. ברנאן ורזניק מדגישים את החשיבות של ניפוי שגיאות כחלק מפיתוח אסטרטגיות להתמודדות וחיזוי של בעיות. פאפרט (Papert, 1980) מדגיש את החשיבות בתהליך ניפוי שגיאות דרכו מתוודעים תלמידים לדרכי הלמידה של עצמם ויש סיכוי שיכירו טוב יותר את אופן החשיבה שלהם. המחקר שלפנינו מתמקד אם כן ביכולת שלנו להבין כיצד תלמידים מנפים שגיאות, הבנה של דפוסי פעולה ודפוסי חשיבה המאירים, במובן הרחב, על הדרך שבה תלמידים לומדים ומתמודדים עם פתרון בעיות. לצורך כך, ינסה המחקר לענות על השאלות הבאות:

1. אילו דפוסים של ניפוי שגיאות באים לידי ביטוי בסביבת למידה לתכנות מבוסס-בלוקים?
2. כיצד משתנים דפוסים אלו לאורך המשחק?
3. האם דפוסים אלו מוסברים בצורה טובה יותר כתלויי הקשר (State) או כתלויי תלמיד (Trait)?

מתודולוגיה

ניתוח הנתונים במחקר זה הינו שניוני (Secondary analysis), קרי, של נתונים שנאספו לצורך מחקר אחר (Fishelson & Hershkovitz, 2022). הגישה המחקרית במחקר שלפנינו התבססה על שיטות של ניתוח למידה

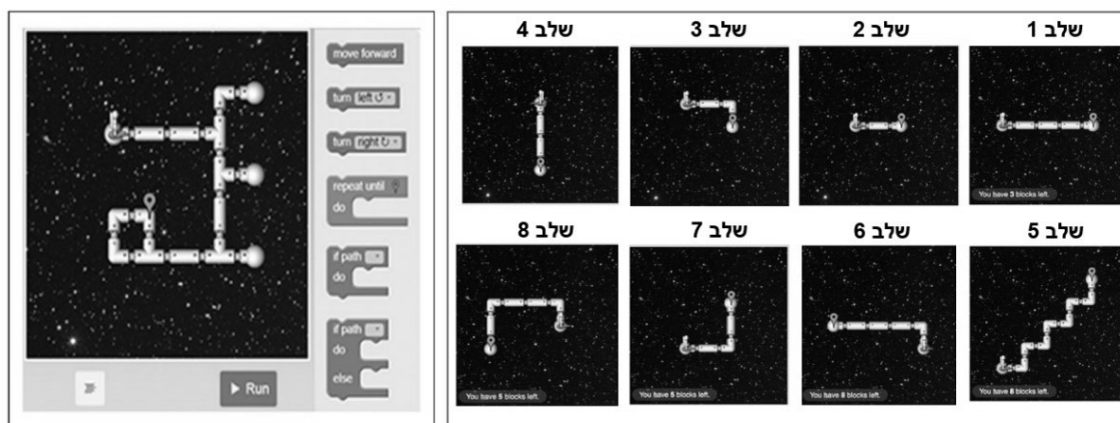
(Learning analytics) וכללה ניתוח של קבצי יומן (Log files) שנגזרו ממשימות תכנות שניתנו לאוכלוסיית המחקר בסביבת הלמידה Kodetu, עיבודים סטטיסטיים ושימוש במודלים של Machine learning.

אוכלוסיית המחקר

נתוני המחקר נאספו בתחילת 2021 מ-189 תלמידים בגילאים 15-14 שנה בבית ספר תיכון שש-שנתי במרכז הארץ, ביישוב השייך לאשכול חברתי-כלכלי 8. 41% הם בנים (78 מ-189) ו-59% הן בנות (111 מ-189).

כלי המחקר

סביבת הלמידה קודטו (Kodetu) – סביבת למידה מקוונת המבוססת על תכנות מבוסס בלוקים (Block based programming) ומתאימה לתלמידים צעירים ללא ניסיון מוקדם בתכנות. התוכנה כוללת מגוון רחב של משימות עם דרגת קושי עולה ומשמשת כפלטפורמה יעילה להרכשה של יכולות ומיומנויות ח"ח. המבנה הכללי של התוכנה מורכב ממספר משחקים, כל משחק מורכב ממספר שלבים, כשבכל שלב נדרש התלמיד להעביר דמות של אסטרונאוטית במסלול שונה מנקודת ההתחלה לנקודת יעד. התלמיד נדרש לתכנן ולהגדיר את המסלול תוך שימוש בתבניות תכנות מבוססות בלוקים. התוכנה פותחה לצרכי מחקר ומציעה גישה לקובצי הלוג של המשתמשים בתוכנה. במחקר נבנו ונבדקו 8 שלבים (איור 1). ההתקדמות בשלבים מלווה במעבר מפעולות סידרתיות פשוטות לשימוש בתבניות בלוקים מתקדמות יותר כדוגמת לולאות ומשפטי תנאי.



איור 1. הפעילויות שניתנו לתלמידים בכל אחד משלבי המשחק

קבצי יומן – כללו כ-146,000 שורות שכל אחת מהן מייצגת פעילות שבוצעה על ידי אוכלוסיית המחקר בסביבת Kodetu. כל שורה בקובץ כללה, בין היתר, מזהה של התלמיד, השלב שבו בוצעה הפעולה, תאריך ושעת ביצוע הפעולה וקוד המתאר את הפעולה. קבצי היומן נותחו בכלי Oracle PL/SQL.

משתני המחקר

לעתים רחוקות, אם בכלל, פתרון לבעיה מורכבת מסתבר כתיקין כבר בניסיון הראשון. לרוב, מתגלות שגיאות לאורך הדרך, ואז יש צורך לאתר ולנפות אותן. בחינה של ניפוי שגיאות נעשית מהרגע בו מבוצעת ההרצה השגויה הראשונה של התוכנית, ובמקטעים שבין כל שתי הרצות שגויות עוקבות. תהליך האופרציונליזציה של המושג ניפוי שגיאות הוביל אותנו להגדיר מספר משתנים מדידים:

תבנית שינויי זמן בין הרצות שגויות – מדידת קו המגמה בערכי פרקי הזמן בין כל שתי הרצות עוקבות שגויות. בתהליך "טיפוסי" של ניפוי שגיאות יעיל נצפה למגמה של התקצרות הזמן בין כל שתי הרצות עוקבות.

שיעור הזמן (%) שהוקדש לתיקון קוד שגוי – מדידת פרק הזמן שעבר במקטע שבין ההרצה השגויה הראשונה ועד להרצה המוצלחת מתוך מתוך סך הזמן שהתלמיד השקיע בשלב. ככל שערכו של משתנה זה גדול יותר יכולת ניפוי השגיאות של התלמיד נמוכה יותר.

שיעור שינויי הקוד – מדידת מספר השינויים שהתלמיד מבצע בקוד לאורך ביצוע השלב. כמות שינויים קטנה יותר תעיד על יכולת ניפוי שגיאות טובה יותר.

תבנית שינויי קוד בין הרצות שגויות – מדידת קו מגמת כמות השינויים בקוד לאורך ביצוע השלב. בתהליך יעיל של ניפוי שגיאות, נצפה לירידה במספר שינויי הקוד לאורך השלב. על ציר הזמן, התלמיד מצמצם את מספר השגיאות בתוכנית עד להרצה המוצלחת.

שיעור שינויי הקוד בהרצה האחרונה – מדידת שיעור שינויי הקוד בהרצה השגויה האחרונה שהתלמיד ביצע. בתהליך יעיל של ניפוי שגיאות נצפה למספר קטן של שינויי קוד במקטע ההרצה האחרון של התוכנית.

מהלך המחקר

קבצי היומן נגזרו מפעילות התלמידים בפתרון שלבים בסביבת Kodetu, ועל בסיסם בוצעו שלבי המחקר הבאים (איור 2):

- **טרום עיבוד (Pre-Processing)** – טיוב קבצי היומן המקוריים ויצירת קבצי עזר ונתונים סיכומיים.
- **עיבוד וחישוב משתני המחקר** – כתיבת שאילתות ותוכניות מחשב לחישוב משתני המחקר.
- **אשכול (Clustering)** – הרצת מודלים של אשכול היררכי במטרה לזהות תבניות הומוגניות של קבוצות תלמידים עם דפוסים דומים של ניפוי שגיאות. תהליכי האשכול בוצעו באמצעות תוכנת JASP.
- **ניתוח ביטויי השתנות ההתנהגויות** – על בסיס דפוסי ההתנהגות שזוהו ואופיינו, חישוב ואפיון שכיחות המעברים של תלמידים בין דפוסים אלו לאורך המשחק.
- **State or Trait** – בדיקת שאלת המחקר הנוגעת לסיווג ההתנהגויות הנצפות כ-SOT באמצעות הרצת מודלים של עצי החלטה תוך שימוש ב-RapidMiner Studio.



איור 2. שלבי המחקר

ממצאים

זיהוי התנהגויות נצפות בפרקטיקת ניפוי שגיאות

על מנת שנוכל לאפיין דפוסים של ניפוי שגיאות, הורץ אלגוריתם לניתוח אשכולות היררכי (Hierarchical Cluster Analysis) על משתני המחקר. האלגוריתם התבסס על ההיפר-פרמטרים הבאים: מדד לדמיון או שוני בין פריטים נקבע כמקדם המתאם של פירסון והמרחק בין האשכולות חושב לפי שיטת Ward. ערכי המשתנים תוקנו על פי ציון תקן Z. בקביעת הפרמטרים המשפיעים על מספר האשכולות נבחנו מספר אפשרויות. האפשרות הראשונה התבססה על אי קביעת מספר האשכולות מראש ואופטימיזציה של מספר האשכולות על בסיס מטריקות מוכרות כדוגמת BIC, AIC ו-Silhouette. נציין שמטריקות אלה הם למעשה מדדי התאמה המשמשים בניית אשכולות כדי לבחור את מספר האשכולות הטוב ביותר. במקרה שלפנינו, בכל המקרים בהם אפשרנו למודל לבצע אופטימיזציה של מספר האשכולות, נוצרו אשכולות עם מספר איברים קטן מדי ($N < 10$) או מקרים בהם זוהה מספר גדול של אשכולות אותם לא ניתן היה לאפיין. האפשרות שנבחרה התבססה על קיבוע אלגוריתם האשכול ל-3 אשכולות שהובילה ליצירה של אשכולות "משמעותיים ומעניינים" אותם יכולנו לאפיין על בסיס משתני המחקר שנקבעו.

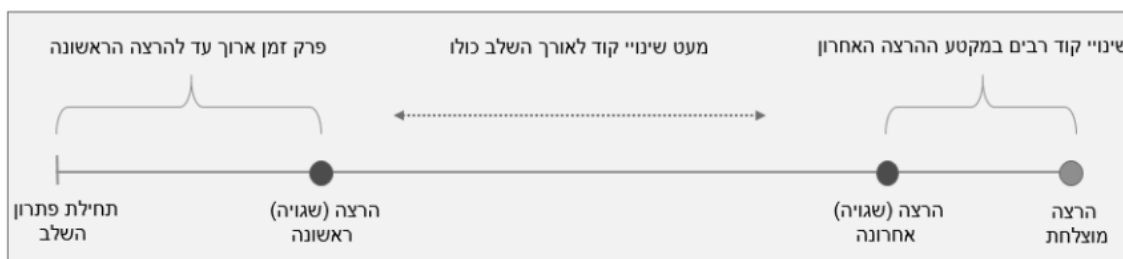
שלב הראשון אופיינו ההתנהגויות ברמת המשחק. טבלה 1 מציגה את תוצאות האשכול – ממוצעי ערכי משתני האשכול המנומרים עבור כל אחד מ-3 האשכולות שזוהו על ידי אלגוריתם האשכול.

טבלה 1. תוצאות האשכול ברמת המשחק

שיעור שינוי הקוד בהרצה האחרונה	תבנית שינויי קוד בין הרצות שגויות	שיעור שינויי הקוד	שיעור הזמן שהוקדש לתיקון קוד שגוי	תבנית שינויי זמן בין הרצות שגויות	
0.84	0.03	<i>-0.48</i>	<i>-0.46</i>	<i>-0.13</i>	אשכול 1 (N=81)
<i>-0.66</i>	<i>-0.23</i>	0.58	0.71	<i>-0.21</i>	אשכול 2 (N=79)
<i>-0.51</i>	0.51	<i>-0.22</i>	<i>-0.59</i>	0.91	אשכול 3 (N=31)

סימון ערכי הקיצון (Bold – ערך מקסימלי, Italic – ערך מינימאלי) וניתוח הממוצעים של ערכי משתני האשכול, סיעו לנו לאפיין את כל אחד מהאשכולות הבאים המייצגים דפוסי התנהגות שונים של ניפוי שגיאות:

אשכול 1 – הממצאים (איור 3) מצביעים על תלמידים הנוטים לבצע מעט שינויי קוד לאורך המשחק. עיקר תהליך ניפוי השגיאות מבוצע מנקודת ההרצה השגויה האחרונה ועד להרצה המוצלחת של התוכנית. נצפית מגמת התקצרות פרקי הזמן שבין כל שתי הרצות שגויות אולם מגמה זו אינה מובילה את התלמיד לתיקון של חלק מהשגיאות. במידה מסויימת אלו הרצות "סרק" שייטכן וניתן לייחס אותן לכך שהפתרון לשלב מתחיל להתגבש אצל התלמיד רק לקראת סוף השלב. מגמת ההתכנסות המאוחרת מקבלת חיזוק גם לאור העובדה שההרצה הראשונה מבוצעת בשלב מאוחר. על בסיס מאפיינים אלו כונה אשכול 1 כאשכול "**התכנסות מאוחרת**".



איור 3. תלמידי אשכול 1 - "התכנסות מאוחרת"

אשכול 2 - כלל המשתנים באשכול זה מקבלים ערכי קיצון המאפיינים במקרה זה תהליך ניפוי שגיאות "יעיל/מיטבי". הממצאים (איור 4) מעידים על התקצרות פרקי הזמן ושל כמות שינויי הקוד שהתלמיד מבצע בין כל שתי הרצות שגויות עוקבות, קרי, מגמה ברורה של התכנסות. נצפית תבנית פעולה שמתחילה בהרצה קצרה, זיהוי השגיאות הדורשות תיקון ועבודת תכנון מתודית שמובילה לירידה בכמות השגיאות ככל שהתלמיד מתקדם לאורך השלב. על בסיס מאפיינים כונה אשכול 2 כאשכול "**ניפוי שגיאות מיטבי**".



איור 4. תלמידי אשכול 2 - "ניפוי שגיאות מיטבי"

אשכול 3 – הממצאים (איור 5) מלמדים על תהליך ניפוי שגיאות הנוטה לאי-התכנסות. נצפית הרצה ראשונה מאוחרת ומעט שינויי קוד, יחסית, שתלמידי אשכול 3 מבצעים לאורך המשחק. נראה שההתכנסות לפתרון נעשית בשלב מאוחר יותר של התהליך. הממצאים מעידים על מרכיב תכנון חלש, נצפה דפוס של ניסוי וטעיה,

התקלות בשגיאה- < תגובה -> ומציאת פתרון אד-הוק לבעיה כאשר היא מזוהה על ידי התלמיד בעת ביצוע ההרצה השגויה. תלמידים אשכול זה מאופיינים יותר כ-Tinkerer ופחות כ-Planners. על בסיס מאפיינים אלו כונה אשכול 3 כאשכול "התנהגות מתבדרת".

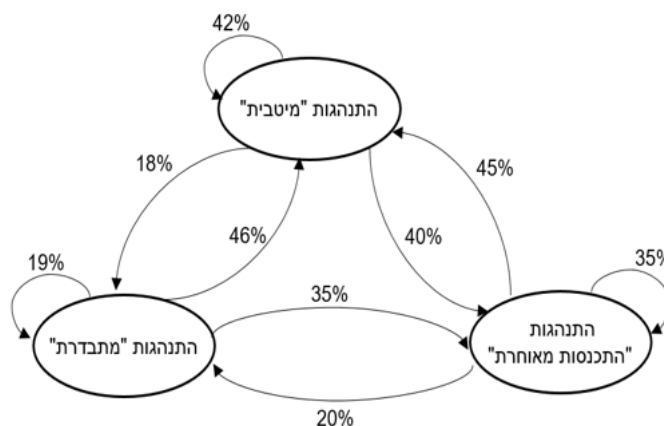


איור 5. תלמידי אשכול 3 - "התנהגות מתבדרת"

ניתוח הממצאים ברמת משחק הוליד אפיון ברור למדי של שלוש אסטרטגיות ניפוי שגיאות. לא מן הנמנע "שתיוג" התלמיד להתנהגות משוקללת זו (ברמת המשחק) אינה ההתנהגות שתאפיין אותו בכל אחד משלבי המשחק. הנחה זו הובילה אותנו לחזור על תהליך האשכול עבור כל אחד משלבי המשחק. הממצאים ברמת השלבים מלמדים על דפוסי התנהגות דומים מאוד לאלו שנצפו ברמת המשחק, קרי, בכל אחד מהשלבים זהו 3 אשכולות עם ערכי משתנים ומאפיינים הדומים לאלו שזיהינו ברמת המשחק.

השתנות ההתנהגויות הנצפות לאורך המשחק

בשלב זה ביקשנו לבחון את ביטויי השתנות ההתנהגויות הנצפות לאורך המשחק. עבור 13 תלמידים נצפתה רק התנהגות אחת שליוותה אותם לאורך שלבי המשחק, עבור 90 תלמידים נצפו 2 התנהגויות ועבור 88 תלמידים נצפו 3 התנהגויות. סך הכל נצפו 533 מעברים בין השלבים השונים. שרשרת מרקוב (איור 6), מודל הסתברותי המשמש לתיאור התפתחות של תהליכים כסדרה של מצבים, סייעה לנו לבחון את הסיכויים למעבר מהתנהגות נתונה לכל התנהגות אחרת. הממצאים שניתחו את שינויי ההתנהגויות במעברים לאורך שלבי המשחק מצביעים על כך שבמרבית המקרים הנטיה של התלמיד היא "לשמר" את ההתנהגות שלו או לעבור לדפוס התנהגות יעיל יותר.



איור 6. שרשרת מרקוב לתיאור המעברים בין ההתנהגויות לאורך המשחק

דפוסי התנהגות התלמיד – תלויות הקשר (State) או תכונה של התלמיד (Trait)?

לבדיקת שאלה זו נבנו שני מודלים של עצי החלטה במטרה לחזות את ההתנהגות עבור מקבץ נתונים הכולל 268 זוגות של תלמיד-שלב. ההתאמה שלהם חושבה באמצעות אימות צולב (cross validation two-fold), טכניקה לאימון מודל ולמידת ביצועיו על קבוצת נתונים. על מנת לבדוק את מהימנות המודל נעשה שימוש במדד קאפה (Kappa). מדד זה מודד מהימנות של שני מעריכים שונים על אותה התופעה ומחשב את ההסכמה העודפת מהסכמה מקרית. בשני המודלים, המדד קיבל ערכים שליליים, משמע שקיימת הסכמה נמוכה, פחות מאופן מקרי, ולא ניתן להסביר את ההתנהגויות הנצפות לא כמאפיין של השלב במשחק ולא כמאפיין של התלמיד.

דיון

מטרתנו של המחקר הנוכחי היתה לזהות ולאפיין התנהגויות מורכבות בתהליכי ניפוי שגיאות. ניפוי שגיאות הוא אחת הדרכים היעילות ביותר לטפח כישורי פתרון בעיות (Yen et al., 2012) והבנת דפוסי התנהגויות לניפוי שגיאות משליכה על יכולת רחבה יותר של פתרון בעיות.

זוהו ואופיינו 3 דפוסי התנהגות לניפוי שגיאות אותן כינינו כהתנהגות מיטבית, התכנסות מאוחרת והתנהגות מתבדרת. הממצאים חיזקו דפוסי התנהגויות שזוהו במחקרים אחרים ובגישות מחקר אחרות. אחת ההתנהגויות שזיהינו במחקר נגזרה מאוסף של משתנים שתאמו את הבנתנו ככאלו התורמים לניפוי שגיאות יעיל ונכון. כינינו התנהגות זו **כהתנהגות ניפוי שגיאות מיטבית** והמרכיב הבולט ביותר בה היה מרכיב ההתכנסות. בכל נקודת הרצה התלמיד מבצע הערכה מחדש של הפער בין התוצר המצופה לתוצר אליו הגיע בהרצה הנוכחית. יין וחובירו (Yen et al., 2012) כיוונו במחקרם לשני שלבים המתקיימים בתהליך ניפוי שגיאות, שלב ההבנה (Comprehension) ושלב התיקון (Correction). שלב ההבנה כולל בתוך את מרכיב ההשערה (hypothesis) שאמור להצביע על מקור הבעיה. התנהגות מיטבית אינה יכולה "לדלג" על שלב ההבנה. היא מרכיב הכרחי בתהליך ההתכנסות. תובנה זו משליכה על תהליך ההוראה והלמידה של ניפוי שגיאות, תהליך שצריך להתעכב על שני השלבים גם יחד, שלב ההבנה ושלב התיקון.

דפוסי של **התנהגות מתבדרת** אופיינו במחקרים אחרים (Kim et al., 2018; Bers et al., 2014; Jadud 2006) כדפוסי שכיחים של ניפוי שגיאות המתבססים על Tinkering והסרה פשוטה, לעיתים בצורה ספורדית ורנדומלית, של חלקים שגויים מתוך התוכנית בשיטת ניסוי וטעיה. אימוץ דפוס התנהגות זה יכול להוביל לקוד תקין ועובד, אך במרבית המקרים התלמיד לא יידע לזהות את מקור הבעיה. במקרים אלו קשה מאוד להנחות את התלמיד לאיך נכון לפתור את הבעיה, זאת מכיוון שמקור הבעיה אינו ידוע. התנהגות זו מאפיינת תוכניתנים מתחילים אצלם קיימת נטיה להתמקד בפתרון עצמו ופחות באיך התוכנית עובדת (Vessey, 1985).

סוג נוסף של דפוס התנהגות שזיהינו כונה **התכנסות מאוחרת**. היכולת לייצר היפוטזה הנוגעת להבנת שורש הבעיה ובניית מודל סיבתי לשגיאות שנצפו (Causal model) הם תנאים הכרחיים לתהליך ניפוי שגיאות יעיל (Bers et al., 2014). ניתן להעריך שהתלמידים שזוהו עם דפוס התנהגות זה התקשו בחלקים רבים של התוכנית לייצר היפוטזה זו. Kim וחובירו (2018) מזהים שני דפוסי של תהליכי ניפוי שגיאות, האחד מבוסס על בחינה אקטיבית (Active examination) של התוכנית עם דפוסי של הרצות מרובות, מעברים דחופים ובדיקות של הקוד שנכתב. הדפוס השני וההפוך הוא הדפוס הפסיבי (Passive examination). במקרים מסויימים ניתן לצפות למעבר מדפוס פסיבי לדפוס אקטיבי. מעבר כזה מתקיים למשל כאשר התלמיד זוכה לעזרה ותמיכה (Scaffolding). הגם שבמחקר שלפנינו לא ניתנה תמיכה מסוג כלשהו, הרי שהמעבר מדפוסי פעולה פסיביים לאקטיביים מזכירים במידה רבה את מה שזיהינו בהתנהגות מאוחרת.

השתנות ההתנהגויות לאורך המשחק

הממצאים בנקודה זו הצביעו על תלמידים הנוטים לשמר את דפוס ההתנהגות שלהם או לשפר אותו במעברים לאורך המשחק. ממצא זה מפתיע בעיקר על רקע העובדה ששום גורם אנושי או מחשבי לא היה מעורב בתמיכה או סיוע לתלמידים במהלך המשחק.

הסבר לממצא זה עשוי להימצא בתיאורית **מכוונות עצמית בלמידה** (SRL – Self regulated learning), תחום רחב המסביר את יכולת הלומד לכוון את מחשבותיו, רגשותיו והתנהגותו תוך כדי למידה כדי להשיג את המטרה הלימודית שאליה הוא שואף (Zimmerman, 1986). תיאוריה זו רואה בלמידה תהליך אקטיבי-קונסטרוקטיבי הכוללת שני מרכיבים שיכולים לסייע לנו להבין את ממצאי המחקר שלפנינו בשאלת שיפור דפוסי ההתנהגות לאורך המשחק: **מרכיב ההתנסות** ומרכיב **רכישת הידע והבנייתו**. במקרה שלפנינו, התלמיד חווה התנסות עצמית במהלך משחק רב שלבים, כולל מדרג קושי עולה התורם למרחב התנסות זו. המעבר בין השלבים תורם להרכשת ידע נוסף לא רק סביב דרישות המשימה אלא גם ידע הנוגע לסביבת הלמידה עצמה ודרך התפעול שלה.

הסבר נוסף לממצא זה מקורו גם **בסביבת הלמידה** עליה התבססנו במחקר – סביבת למידה מקוונת ומבוססת משחק. מחקרים מדגישים את ההשפעות המוטיבציוניות והקוגניטיביות על התלמידים הפועלים בסביבת למידה כזו (Huang, 2011), סביבה המעודדת אתגרים, חקירה, פתרון בעיות, פיתוח הסקרנות, יצירתיות ועוד. ניפוי שגיאות הוא תהליך שאבני הדרך שלו הם כישלונות איתם התלמיד מתמודד. Clark וחובירו (2015) טוענים שסביבת המשחק היא סביבה סלחנית המאפשרת לשחקן לחוות כישלון, אך לא תסכול, ומעבירה אותו תהליך של למידה מטעויות. באמצעות יצירת סביבה בטוחה שבה הכישלון הוא חלק ממהלך המשחק, מתגמלים המשחקים את ההתמדה ואת המאמץ של השחקנים ומעודדים אותם לקחת סיכונים ולנסות פתרונות חדשים ויצירתיים.

הסבר נוסף לשיפור בדפוסי ההתנהגות לאורך המשחק הוא **המשוב** שהתלמיד מקבל. על פי Hattie וחובריו (2007), משוב בלמידה נמצא בראש סולם גורמי ההצלחה של למידה בכלל ולמידה מקוונת בפרט. בסביבה לימודית משחקית המשוב שהתלמיד מקבל הוא משוב ויזואלי, ניתן בתדירות גבוהה ולעיתים אף מסופק בזמן אמת. בסביבת תכנות מבוססת בלוקים, המשוב ניתן לתלמיד בכל הרצה של התוכנית ומספק יכולת הכוונה וחישוב "מסלול מחדש". מחקרים מראים שבסביבות אלו המשוב הוא גורם הנעה מוטיבציוני ותורם רבות לביצועי התלמיד (Marwan et al., 2021). בתהליך ניפוי שגיאות נמצא מעגלים חוזרים של "הרצה- משוב- ניפוי שגיאות", בהם המשוב הוא מרכיב מרכזי באותו דפוס שיפור שראינו במהלך המשחק.

התלמיד נוטה אם כן לשפר את המיומנויות שלו לאורך זמן גם כאשר לא מופעלת התערבות כלשהי, אנושית או מיחשובית. לממצא זה חשיבות גדולה בהבנה כיצד מתפתחות מיומנויות למידה, מהי התרומה של התנסות עצמית בשיפור הביצועים של התלמיד בסביבת למידה, מהי התרומה של סביבת למידה משחקית לתהליך של חיזוק מיומנויות למידה ופתרון בעיות ועוד.

שאלת ה- State or Trait (SOT)

מן הממצאים עולה שניפוי שגיאות אינו תלוי-הקשר וגם אינו תלוי-תכונה. מצבים כאלו, בהם שאלת ה-SOT אינה מוכרעת, אופיינית לסביבות למידה אינטראקטיביות (Baker, 2007). דוגמא לכך ניתן לראות בממצאים של הרשקוביץ ונחמייאס (Hershkovitz & Nachmias, 2011). במחקרם סביב רמת ההתמדה בשימוש באתרים מלווי-קורסים רמת ההתמדה הוסברה גם כתלויה הקשר (State) וגם כתלויה תלמיד (Trait). על מנת שנוכל להסביר את שאלת ה-SOT עבור פרקטיקת ניפוי שגיאות בצורה טובה יותר, כזו שתאפשר לנו לספק המלצות וכיווני פעולה הנוגעים לדרכי ההוראה של תהליכי ניפוי שגיאות, יש לבצע ניתוחים ובדיקות נוספות. ייתכן והגדלת סט הנתונים (dataset) יוביל לממצאים ברורים וחדים יותר בהקשר זה.

לסיכום: המחקר עושה שימוש בשיטות מחקר כמותיות חדשניות ובכך הוא תורם ליצירת כלים חדשים לבחינת ח"ח. המחקר מבסס את היכולת לזהות ולאפיין דפוסי התנהגויות בתהליכי ניפוי שגיאות על בסיס ניתוח כמותני של לוגי מערכת. יכולת זו מספקת תשתית אופרטיבית באמצעותה ניתן לספק למורה ולתלמיד, בזמן אמת, משוב על אסטרטגיית ניפוי השגיאות אותה התלמיד מאמץ במהלך כתיבת התוכנית, כמו גם היכולת לחזות האם האסטרטגיה אותה מאמץ התלמיד תוביל להצלחה או אי ההצלחה בביצוע המשימה.

מקורות

- Baker, R. S. (2007). Is gaming the system state-or-trait? Educational data mining through the multi-contextual application of a validated behavioral model. In Complete On-Line Proceedings of the Workshop on Data Mining for User Modeling at the 11th International Conference on User Modeling 2007 (Vol. 2007, pp. 76-80). Boston, MA: User Modeling Inc.
- Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145-157.
- Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada (Vol. 1, p. 25).
- Eguíluz, A. (2018). An Evaluation of Open Digital Gaming Platforms for Developing Computational Thinking Skills [Bookitem]. IntechOpen. <https://doi.org/10.5772/intechopen.71339>
- González, M. R. (2015). Computational thinking test: Design guidelines and content validation. In EDULEARN15 Proceedings (pp. 2436-2444). IATED.
- Hattie, J., & Timperley, H. (2007). The power of feedback. *Review of educational research*, 77(1), 81-112.
- Hershkovitz, A., & Nachmias, R. (2011). Online persistence in higher education web-supported courses. *The Internet and Higher Education*, 14(2), 98-106.
- Huang, W. H. (2011). Evaluating learners' motivational and cognitive processing in an online game-based learning environment. *Computers in Human Behavior*, 27(2), 694-704.
- Israel-Fishelson, R., & Hershkovitz, A. (2022). Cultivating creativity improves middle school students' computational thinking skills. *Interactive Learning Environments*, 1-16.

- Israel-Fishelson, R., & Hershkovitz, A. (2021). Micro-persistence and difficulty in a game-based learning environment for computational thinking acquisition. *Journal of Computer Assisted Learning*, 37(3), 839-850.
- Jadud, M. C. (2006, September). Methods and tools for exploring novice compilation behaviour. In *Proceedings of the second international workshop on Computing education research* (pp. 73-84).
- Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in human behavior*, 72, 558-569.
- Kim, C., Yuan, J., Vasconcelos, L., Shin, M., & Hill, R. B. (2018). Debugging during block-based programming. *Instructional Science*, 46, 767-787.
- Jadud, M. C. (2006, September). Methods and tools for exploring novice compilation behaviour. In *Proceedings of the second international workshop on Computing education research* (pp. 73-84).
- Marwan, S., Shabrina, P., Milliken, A., Menezes, I., Catete, V., Price, T. W., & Barnes, T. (2021, November). Promoting students' progress-monitoring behavior during block-based programming. In *Proceedings of the 21st Koli Calling International Conference on Computing Education Research* (pp. 1-10).
- Papert, S. (1980). *Mindstorms: Children, Computers and Powerful ideas*. Basic Books Inc.
- Vessey, I. (1985). Expertise in debugging computer programs: A process analysis. *International Journal of Man-Machine Studies*, 23(5), 459-494.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. M. (2010). Computational thinking: What and why? *The Link Magazine*, Carnegie Mellon University. <https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>
- Wolz, U., Stone, M., Pearson, K., Pulimood, S. M., & Switzer, M. (2011). Computational Thinking and Expository Writing in the Middle School [Article]. *ACM Transactions on Computing Education*, 11(2), 1-22. <https://doi.org/10.1145/1993069.1993073>
- Yen, C. Z., Wu, P. H., & Lin, C. F. (2012). Analysis of experts' and novices' thinking process in program debugging. In *Engaging Learners Through Emerging Technologies: International Conference on ICT in Teaching and Learning, ICT 2012, Hong Kong, China, July 4-6, 2012. Proceedings 7* (pp. 122-134). Springer Berlin Heidelberg.
- Zimmerman, B. J. (1986). Becoming a self-regulated learner: Which are the key subprocesses?. *Contemporary educational psychology*, 11(4), 307-313.