

## תכנות מכוון עצמים ושפת C++ – תרגיל 1

התרגיל מתחלק ל-4 חלקים, וזהו פרויקט מתמשך המתפתח לאורך הסמסטר. בפרויקט נממש מערכת למשחקי לוח ל-2 שחקנים אשר משוחקים על גבי לוח משבצות (דמקה, שחמט, וכו').

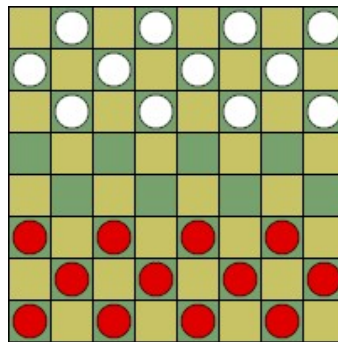
בחלק הראשון נממש משחק **דמקה** ל-2 שחקנים.

את חוקי המשחק אפשר לקרוא כאן:

<https://he.wikipedia.org/wiki/%D7%93%D7%9E%D7%A7%D7%94>. אנו נממש דמקה

בינלאומית/ברזילאית, ללא חובה לבצע אכילה (בלי "שרופים") וללא דילוגים רצופים – כלומר, שחקן בתורו יכול לבצע תנועה אחת עם אבן משחק אחת, או לבצע דילוג אחד. מלך יכול לנוע בכיוון אחד באלכסון כרצונו (כל עוד המשבצות בדרך פנויות), ולבצע אכילה אחת בלבד באותו כיוון (ולעצור במשבצת שאחרי אבן המשחק שאכל). מלך אינו יכול לדלג על 2 אבני משחק רצופות.

המשחק מסתיים בניצחון אם ליריב אין יותר אבני משחק, או שאין לו מהלכים חוקיים.



סידור אבני הלוח בתחילת המשחק

לכל **משחק** יש את הנתונים הבאים:

- מנצח (אם יש), מצב המשחק (הסתיים או לא), תור נוכחי.
- **לוח המשחק**: רשימת המשבצות ואבני המשחק בכל משבצת.
- **שחקנים**: לכל שחקן יש את רשימת אבני המשחק שלו ומיקומם בלוח, האם תורו, האם ניצח/הפסיד (אם הסתיים המשחק), כמה אבנים נותרו לו במשחק.
- **אבני משחק**: לכל אבן משחק יש את המיקום שלה, האם מלך, מהלכים אפשריים/חוקיים, סיכונים (רשימת אבני משחק שיכולים לאכול אותה במהלך אחד), שחקן אליו האבן שייכת.

ראשית, עליכם לעצב את המחלקות למשחק. יש לכתוב הצהרות (בקבצי H) לכל המחלקות המתאימות למערכת, כולל כל השיטות הפומביות של כל מחלקה.

### שימו לב:

- רשימת הנתונים של משחק אינה בהכרח מהווה את כל המחלקות שיש להגדיר. יתכנו בהחלט מחלקות נוספות, ויש שלל עיצובים אפשריים שניתן לעשות עבור פתרון התרגיל.
- יתכן וידרשו מחלקות נוספות בהתאם לחלק השני של התרגיל (התפריט האינטראקטיבי).
- לפתרון יש להשתמש בנושאים שנלמדו עד בנאים (כולל `const-i enum`). אין צורך להגדיר בנאים ומפרקים.
- רצוי מאוד, אך לא חובה, להגדיר את לוח המשחק באופן כמה שיותר גנרי (מבחינת גודלו והערכים שמכילה כל משבצת), וגם את אבני המשחק. זה לא יבדק, אך יעזור בתרגילים בעתיד.

שנית, יש לממש את המחלקות שעוצבו ולכתוב תפריט אינטראקטיבי החוזר על עצמו, ומאפשר למשתמש לנהל ולשחק במספר משחקי דמקה במקביל. ננהל רשימה של משחקי דמקה, כאשר אחד מהם יהיה המשחק הנוכחי (עליו מתבצעות הפעולות). התפריט יציג אפשרויות ממוספרות למשתמש, ויאפשר למשתמש לבצע את הפעולה הנבחרת ע"י הקלדת המספר המתאים:

1. הוספת משחק – מוסיף משחק דמקה חדש לסוף רשימת המשחקים, ומגדיר אותו בתור הנוכחי.
2. מעבר משחק – מגדיר את המשחק הבא ברשימה בתור המשחק הנוכחי (אם עברנו את המשחק האחרון ברשימה, נחזור חזרה לראשון).
3. מחיקת משחק – מוחק את המשחק הנוכחי מרשימת המשחקים (המשחק הנוכחי יתעדכן להיות הבא אחריו).
4. הצגת פרטי משחק – מדפיס את לוח המשחק הנוכחי, מצב המשחק (הסתיים או לא), את השחקן שתורו אם המשחק לא הסתיים, או השחקן המנצח במידה והמשחק הסתיים.
5. הצגת מהלכים – מקבל כקלט  $x, y$  של אבן משחק (של שחקן כלשהו) ומציג רשימה  $x, y$  של כל המהלכים האפשריים עבורה (תנועה או דילוג).
6. הצגת סיכונים – מקבל כקלט  $x, y$  של אבן משחק (של שחקן כלשהו) ומציג רשימה  $x, y$  של כל אבני המשחק של השחקן השני המסכנים אותה.
7. ביצוע מהלך – מקבל כקלט  $x, y$  של אבן משחק ו- $x, y$  של מיקום פנוי בלוח ומבצע את המהלך (תנועה או דילוג) במידה והקלט מייצג מהלך חוקי והמשחק לא הסתיים.
8. יציאה מהתפריט – משחרר את כל המשאבים ויוצא מהתוכנית.

אם הוזן קלט לא חוקי (מחוץ לגבולות הלוח, אינו מכיל אבן משחק, אינו מכיל משבצת ריקה, מהלך לא חוקי, וכו') - יש להדפיס הודעת שגיאה ולא לבצע את הפעולה. אין לצאת במקרה של קלט לא תקין.

כאשר התוכנית עולה לראשונה, וגם כאשר נמחק המשחק האחרון ברשימה, רשימת המשחקים ריקה (ובהתאם לכך גם המשחק הנוכחי). במצב זה, כל האפשרויות בתפריט מלבד 1 ו-8 יודיעו על שגיאה.

באפשרות 4 יש להדפיס את הלוח כאשר התו  $s$  מייצג את אבני המשחק של השחקן הראשון, והתו  $x$  מייצג את אבני המשחק של השחקן השני. מלך יש לייצג באמצעות אות גדולה  $O$  או  $X$ . את הלוח כולו ניתן להדפיס כך (אין חובה להיצמד לפורמט המדויק, אך הלוח חייב להיות ברור):

```

1 2 3 4 5 6 7 8
-----
1 | | o | | o | | o | | o |
2 | o | | o | | o | | o | |
3 | | o | | o | | o | | o |
4 | | | | | | | | |
5 | | | | | | | | |
6 | x | | x | | x | | x | |
7 | | x | | x | | x | | x |
8 | x | | x | | x | | x | |
-----
```

## תכנות מכוון עצמים ושפת C++ – תרגיל 1

### הנחיות הגשה

1. העבודה על התרגילים היא בזוגות **בלבד**. ניתן להגיש עם שותף מקבוצה אחרת.
2. את התרגילים יש להגיש במאמא, על-ידי אחד השותפים **בלבד**.
3. פתרון התרגיל צריך להיות מורכב ממחלקות בלבד (קבצי h+cpp). הקובץ היחיד שאינו מחלקה הוא **main.cpp**, אשר מכיל את פונק' ה-main (ופונקציות עזר קטנות נוספות, אם יש).
4. יש להגיש קובץ zip עם קבצי הקוד שלכם (h, cpp), ובנוסף קובץ **readme** עם פרטי המגישים (שם + ת"ז). אין להגיש קבצים נוספים.
5. יש להשתמש אך ורק בנושאים שנלמדו בכיתה (הרצאות 1-2), ללא בנאים ומפרקים.

### הנחיות כלליות

1. את המחלקות יש לעצב לפי הגדרת המשחק והמערכת, ולא לפי התפריט והאפשרויות בו. לדוגמא, אם ברשימת סיכונים התפריט מדפיס x,y, אין זה אומר שבמחלקות יש לשמור x,y – אלא רשימת אבני משחק (מהן ניתן להסיק את המיקום).
2. יש להעדיף להשתמש ברפרנסים על פני מצביעים היכן שניתן.
3. אין לבצע קלט במחלקות הפנימיות. המודול הראשי (main) הוא זה שמבצע קלט מהמשתמש, מדפיס פלט למשתמש, שולח ערכים למחלקות ושיטות, וכו'.
4. עליכם להחליט כיצד התוכנית תיראה – חלוקה למחלקות, תפריט, תת-תפריטים, וכו'. הגיוני כי כל תרגיל יהיה שונה. אתם נבדקים על איכות ועיצוב הקוד, ולא על פרשנות מינורית כזו או אחרת של איך בדיוק הדברים צריכים להיראות.
5. ניתן להניח שהקלט המתקבל הוא מהטיפוס המבוקש (כלומר, אם הקלט הצפוי הוא מספר אז ניתן להניח שאכן הוקלד מספר). אם יש הגבלות לטווח ערכים אפשרי, **לא** ניתן להניח שהערך שנקלט הוא בטווח ויש לוודא את זה ולהודיע על שגיאה בהתאם.
6. אין צורך לבדוק הצלחה של הקצאות דינמיות וניתן להניח שכולן מצליחות.
7. אין להמציא הגבלות על הקלט מעבר למה שנתון.

## בהצלחה!

התרגיל מתחלק ל-4 חלקים, וזהו פרויקט מתמשך המתפתח לאורך הסמסטר. בפרויקט נממש מערכת למשחקי לוח ל-2 שחקנים אשר משוחקים על גבי לוח משבצות (דמקה, שחמט, וכו').

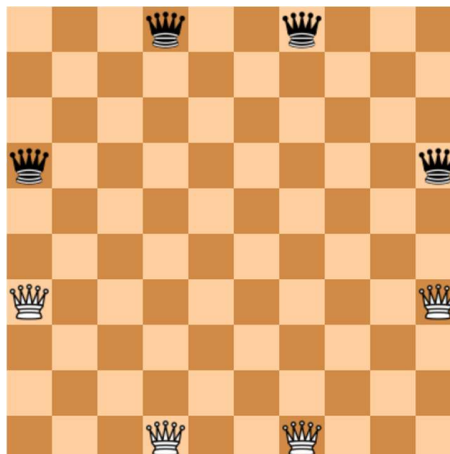
בחלק השני נממש את **משחק האמזונות** ל-2 שחקנים.

את משחק האמזונות משחקים על לוח משבצות שחור-לבן בגודל **10x10**. לכל שחקן ארבע אמזונות בצבע שלו, ובנוסף יש ערימת סמנים השייכת במשותף לשני השחקנים.

השחקן בכלים הלבנים הוא הראשון לשחק. כל תור מורכב משני מהלכים: תחילה, מזיז השחקן את אחת האמזונות בצבע שלו בקו ישר (אופקי, אנכי, או אחד משני האלכסונים, בדומה למלכה בשח) למשבצת פנויה בלוח. האמזונה אינה יכולה לקפוץ מעל אמזונה אחרת או משבצת מסומנת. לאחר מכן, יורה האמזונה חץ מהמקום החדש אליו הגיעה אל משבצת ריקה, כאשר החץ גם הוא נע בקו ישר (אפילו לאחר, לכיוון ממנו הגיעה האמזונה) ואינו יכול לקפוץ מעל משבצות מסומנות או אמזונות. המשבצת אליה נורה החץ מסומנת, ומעתה לא ניתן לדרוך עליה או לקפוץ מעליה.

המשחק מסתיים בניצחון אם ליריב אין מהלכים חוקיים לבצע.

את חוקי המשחק אפשר לקרוא כאן: [https://en.wikipedia.org/wiki/Game\\_of\\_the\\_Amazons](https://en.wikipedia.org/wiki/Game_of_the_Amazons)



סידור הלוח בתחילת המשחק

ראשית, עליכם לעצב את המחלקות למשחק. יש לכתוב הצהרות (בקבצי H) לכל המחלקות המתאימות למערכת, כולל כל השיטות הפומביות של כל מחלקה.

### שימו לב:

- לוח המשחק בגודל 10x10, ולא 8x8 כמו משחק הדמקה מהתרגיל הראשון.
- במשחק האמזונות אין סיכונים ואכילות. האפשרות המתאימה בתפריט שונתה בהתאם.
- עליכם להחליט לבד על המחלקות והנתונים בכל מחלקה. נסו שהמשק הפומבי של המחלקות יהיה תואם ככל האפשר (ואף תואם לחלוטין) למשחק הדמקה מהתרגיל הראשון. זה יעזור בחלק השלישי, וגם מעיד על עיצוב מוצלח של המחלקות.
- לפתרון יש להשתמש בנושאים שנלמדו עד העמסת אופרטורים (כולל). יושם דגש על שימוש נכון ומלא בבנאים ומפרקים, כולל מימוש בנאי העתקה והזזה. יש לממש העמסת אופרטור הדפסה של הלוח ולפחות 2 אופרטורים נוספים (שונים) לבחירתכם.

## תכנות מכוון עצמים ושפת C++ – תרגיל 2

שנית, יש לממש את המחלקות שעוצבו ולכתוב תפריט אינטראקטיבי החוזר על עצמו ומנהל רשימת משחקים בדומה לתרגיל הראשון. התפריט יאפשר למשתמש לבצע את הפעולה הנבחרת ע"י הקלדת המספר המתאים, כאשר הפעולות זהות לתרגיל הראשון, מלבד:

5. **הצגת מהלכים** – מקבל כקלט  $x, y$  של אבן משחק (של שחקן כלשהו) ומציג רשימה  $x, y$  של כל המהלכים האפשריים עבורה (לתנועה בלבד, ללא המשבצת לסימון).
  6. **הצגת סיכונים** – מקבל כקלט  $x, y$  של **משבצת ריקה** בלוח ומציג רשימה  $x, y$  של כל המשבצות הריקות שניתן לסמן ממנה (על-ידי חץ, אם הייתה אמזונה במשבצת שנבחרה).
  7. **ביצוע מהלך** – מקבל כקלט  $x, y$  של אבן משחק ו**שני** מיקומים פנויים בלוח, ומבצע את המהלך (תנועה לראשון וסימון בשני) במידה והקלט מייצג מהלך חוקי והמשחק לא הסתיים.
- באפשרות 4 יש להדפיס את הלוח כאשר התו 0 מייצג את האמזונות של השחקן הראשון, והתו X מייצג את האמזונות של השחקן השני. את הלוח כולו ניתן להדפיס בצורה דומה לתרגיל הראשון.

### הנחיות

יש להקפיד על הנחיות ההגשה וההנחיות הכלליות מהתרגיל הראשון.

שינויים / הבהרות / תוספות:

1. יש להשתמש אך ורק בנושאים שנלמדו בכיתה (הרצאות 1-4).
2. את הלוגיקה של המשחק (כולל בדיקות תקינות הקשורות למבנה המשחק) יש לממש בתוך מחלקות המשחק בלבד, ולא בתפריט. התפריט אחראי אך ורק על קלט ופלט מול המשתמש, וקריאה לפעולות עם הפרמטרים המתאימים.

**בהצלחה!**

התרגיל מתחלק ל-4 חלקים, וזהו פרויקט מתמשך המתפתח לאורך הסמסטר. בפרויקט נממש מערכת למשחקי לוח ל-2 שחקנים אשר משוחקים על גבי לוח משבצות (דמקה, שחמט, וכו').

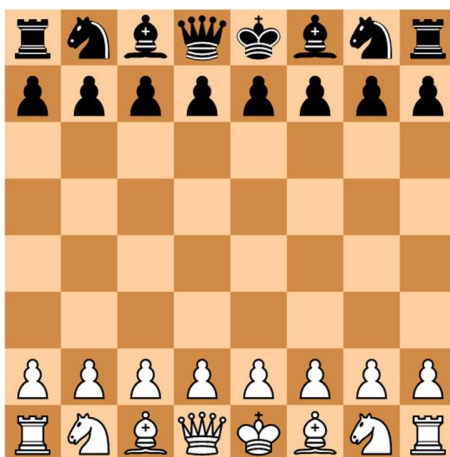
בחלק השלישי נממש משחק **שחמט** ל-2 שחקנים, ונשלב אותו לתוך מערכת גנרית למשחקי לוח.

את חוקי המשחק אפשר לקרוא כאן:

<https://he.wikipedia.org/wiki/%D7%A9%D7%97%D7%9E%D7%98>. אנו נממש שחמט ללא הצרחה

(גדולה/קטנה) וללא תיקו. כלומר, כל כלי יכול לנוע כפי שהוגדר, ואין מהלכים מעבר לכך. שימו לב במיוחד לחוקי התנועה המיוחדים של הרגלי, ולכך שהפרש יכול לדלג מעל כלים אחרים. השחקן בכלים הלבנים הוא הראשון לשחק.

המשחק מסתיים בניצחון כאשר מלך היריב מאוים בשח ואין מהלך שניתן לבצע כדי להימלט מהאיום.



סידור הלוח בתחילת המשחק

ראשית, עליכם לעצב מחלקות עבור **משחק** גנרי. מחלקות אלו יאפשרו לנו לנהל מערכת משחקים באופן גנרי, בצורה שאינה תלויה במימוש הספציפי של כל משחק. תפריט המשחק (בהמשך) יאפשר למשתמש לשחק במשחקים שונים על-ידי הזנת פקודות דומות, באופן שבו המשחק המשוחק בפועל שקוף לחלוטין למימוש של תפריט המשתמש.

מערכת זו תתמוך במשחקי לוח ל-2 שחקנים אשר משוחקים על גבי לוח משבצות ומתנהלים בתורות. לכל אבן משחק יש סט מהלכים שניתן לבצע, ולכל משבצת בלוח יכולה להיות (כתלות במשחק ובמשבצת) רשימת סיכונים. מהלכים וסיכונים אלו משתנים ותלויים במצב המשחק הנוכחי. בכל תור מתבצע מהלך הכולל 1 או יותר משבצות בלוח. למשחק יש מנצח (אם הסתיים), או תור נוכחי (אם לא הסתיים).

שנית, יש לממש מחלקות בהתאם לעיצוב ולהצהרות שביצעתם עבור המשחקים: **שחמט**, **דמקה**, ו**משחק האמזונות**. לכל המשחקים יהיה ממשק פומבי אחיד ומשותף, אך מימוש ספציפי שונה בהתאם לחוקי המשחק. ניתן להגדיר בנאים שונים לכל משחק.

יש לשים דגש רב על יצירת מערכת גנרית, שאינה תלויה במימוש של אף משחק ספציפי או שילוב תכונות של המשחקים בתרגיל. מערכת גנרית ברמה טובה תתמוך באופן שקוף בהוספה של כל משחק העומד בדרישות הנ"ל.

## תכנות מכוון עצמים ושפת C++ – תרגיל 3

לסיום, יש לכתוב תפריט אינטראקטיבי החוזר על עצמו ומנהל רשימת משחקים, בדומה לתרגיל הראשון והשני. התפריט ורשימת המשחקים יהיו גנריים לחלוטין, כך שבזמן ריצה ומלבד פקודה 1 (שתפורט בהמשך), התפריט אינו מודע למשחק הנוכחי שמשחק.

בתרגיל זה יש לשים דגש ולעצב היטב גם את מחלקות התפריט ורשימת המשחקים.

פעולות התפריט זהות לתרגיל הראשון והשני, כאשר כל פקודה תתבצע על המשחק המתאים ברשימה (אשר יכול להיות מכל סוג), מלבד:

1. **הוספת משחק** – מקבל כקלט סוג משחק (1- דמקה, 2- משחק האמזונות, 3- שחמט), מוסיף משחק מתאים לסוף רשימת המשחקים, ומגדיר אותו בתור הנוכחי.

**שימו לב:** אין להסתמך רק על 3 המשחקים המפורטים בתרגיל זה בשום שלב בתרגיל מלבד פקודה 1.

### תבניות

במהלך התרגיל יש לממש לפחות 2 אלגוריתמי תבנית או מחלקות תבנית.

עליכם להחליט היכן נכון ומתאים להשתמש באלגוריתם או מחלקת תבנית, ולממש אותם בצורה גנרית ומתאימה. המימוש הנ"ל יכול להיות בכל חלק בתרגיל - מימוש המשחקים, התפריט וכו'.

### הנחיות

יש להקפיד על הנחיות ההגשה וההנחיות הכלליות מהתרגיל הראשון והשני.

שינויים / הבהרות / תוספות:

- יש להשתמש אך ורק בנושאים שנלמדו בכיתה (נושאים 1-7).
- המימוש הגנרי אינו מתייחס רק לעיצוב המחלקות והשימוש בהן מבלי להכיר את המשחק הספציפי, אלא כמו בתרגילים קודמים, גם מימוש המשחקים צריך להיות נפרד מהשימוש בהם בתפריט – כך שאת התפריט ניתן להחליף במשחק עם גרפיקה, משחק בתורות דרך הרשת, וכו', ללא כל שינוי במחלקות הגנריות ומחלקות המשחקים.

## בהצלחה!

## תכנות מכוון עצמים ושפת C++ – תרגיל 4

התרגיל מתחלק ל-4 חלקים, וזהו פרויקט מתמשך המתפתח לאורך הסמסטר. בפרויקט נממש מערכת למשחקי לוח ל-2 שחקנים אשר משוחקים על גבי לוח משבצות (דמקה, שחמט, וכו'). בחלק הרביעי נרחיב את מערכת המשחקים הגנרית עם שימוש בקבצים, טיפול בחריגות, ו-STL.

### קבצים

בחלק זה נממש **סריאליזציה** של משחק מ/אל קובץ.

עליכם להוסיף סריאליזציה של **משחק** כך שיהיה ניתן לשמור משחק ואת כל הנתונים הרלוונטיים לאותו משחק לקובץ ולטעון אותם מקובץ, כפי שנלמד.

הנתונים ישמרו באופן בינארי. יש לדאוג לשמירה נכונה של כל הערכים, מצביעים, הקצאות דינמיות וכו', ולטעינה נכונה של הנתונים. בנוסף, יש לוודא שפעולות השמירה והטעינה עובדות נכון עם פולימורפיזם. ניתן להוסיף מחלקות חדשות לשם כך.

יש להוסיף את 2 האפשרויות הבאות לתפריט:

9. **שמירת משחק** – מקבל כקלט שם קובץ ומבצע סריאליזציה של המשחק אל הקובץ הנתון. אם הקובץ לא קיים הוא נוצר, אחרת הוא משוכתב (הנתונים הקיימים נאבדים).
10. **טעינת משחק** – מקבל כקלט שם קובץ, טוען משחק מתוך הקובץ, מוסיף אותו לסוף רשימת המשחקים ומגדיר אותו בתור הנוכחי. אופציה זו זמינה גם כאשר רשימת המשחקים ריקה.

שימו לב: בכל קלט של שם קובץ, ניתן להניח ששם הקובץ אינו מכיל רווחים. המשתמש חופשי להכניס כל שם קובץ (ללא רווחים) ואין צורך לבדוק את סיומת הקובץ, ובפרט אין להניח סיומת כלשהי (כמו bin או dat), אלא לשמור ולטעון עם שם הקובץ בדיוק כפי שהתקבל. כמו-כן, אין להניח דבר לגבי תוכן הקובץ ונכונות הנתונים בו.

### חריגות

בחלק זה נעדכן את המערכת לזיהוי וטיפול בשגיאות באמצעות **חריגות**.

עליכם להמיר את כל בדיקות השגיאות לשימוש בחריגות, כך שכל שגיאה אפשרית תיבדק במקום הנכון ותיתפס (והטיפול בה יתבצע) במקום הראוי. בכל הקוד כולו אין לעשות שימוש בערכי החזר, אלא בחריגות בלבד. כמו-כן יש להוסיף בדיקות תקינות לכל הבנאים וזריקת חריגות בהתאם, היכן שיש צורך ומתאפשרת שגיאה בבנאי או בארגומנטים המתקבלים בבנאי.

יש לעדכן בהתאם כל שיטה: בכל מקום בו תיתכן שגיאה יש לבצע בדיקה ולזרוק חריגה. יש לטפל בחריגות רק במקום המתאים, לדוגמה, קוד ביצוע פעולה בתפריט יתפוס חריגה וידפיס הודעת שגיאה מתאימה. אם יש כמה מקומות מתאימים, יש לטפל בחריגה על-ידי זריקה מחדש (לדוגמה, תפיסת חריגה לשחרור זיכרון, ואז זריקה שלה מחדש כדי לחלחל אותה הלאה).

יש לדאוג לשחרר זיכרון באופן נכון בכל מקרה של חריגה, ולבדוק ולטפל נכון גם בשגיאות של הקצאת זיכרון (חריגה של הפקודה new). יש לבדוק ולזרוק חריגות בהתאם גם בגישה לקבצים.

כל חריגה צריכה להיתפס באיזשהו שלב – כך שלא תתאפשר חריגה שתקריס את התוכנית.

עליכם לזהות ולעדכן את המערכת בהתאם לכל החריגות האפשריות. כמו בתרגילים הקודמים, אין להניח דבר לגבי קלט המשתמש מלבד הזנת קלט מהטיפוס הנכון. **אין להניח דבר** בשיטות של מחלקות המשחק למיניהן ואופן השימוש בהן, ויש לבדוק כל מקרה שגיאה או ארגומנט שגוי.



## תכנות מכוון עצמים ושפת C++ – תרגיל 4

STL

בחלק זה נעדכן את הקוד כך שיעשה שימוש במחרוזות ובמבני הנתונים והאלגוריתמים הסטנדרטיים. יש להמיר בקוד כל שימוש במחרוזות (`char*`) ומבני נתונים לטיפוסים המתאימים מתוך STL. כמו-כן יש לעדכן כל אלגוריתם כללי לאלגוריתם המתאימים מתוך STL, היכן שאפשר. שימו לב לכל קוד שהופך למיותר בעקבות שינויים אלו ועליכם להוריד.

### הנחיות

יש להקפיד על הנחיות ההגשה וההנחיות הכלליות מתרגילים 1-3.

שינויים / הבהרות / תוספות:

1. יש להשתמש אך ורק בנושאים שנלמדו בכיתה (כל הנושאים).
2. יש לבדוק שגיאות של הקצאות זיכרון ולהימנע משימוש בערכי החזר עבור שגיאות.
3. יש לבדוק שגיאות של גישה לקבצים ולהתריע על כך באמצעות חריגות.
4. יש לצאת מהתוכנית (באופן תקין, עם הודעת שגיאה ולאחר שחרור כל הזיכרון וסגירת כל הקבצים) בשגיאות חמורות בלבד, כמו שגיאה בהקצאת זיכרון. בשגיאות אחרות, כמו קלט לא תקין או גישה לקובץ שנכשלה, יש להודיע למשתמש בלבד ולהמשיך כרגיל.
5. כאשר רשימת המשחקים ריקה, כל האפשרויות בתפריט מלבד 1, 8, ו-10 יודיעו על שגיאה.
6. כרגיל, שימו לב לעיצוב נכון של המחלקות. עליכם לעצב את המחלקות באופן מלא וסגור כקופסה שחורה – לכל מחלקה תחום אחריות מוגדר משלה, ממשק פומבי ברור, ועיצוב המחלקות צריך לאפשר שימוש בהן באופן מלא גם במערכות אחרות, ללא תלות חזקה בתפריט או במערכת הספציפית (אלא רק לפי הגדרות המערכת והמשחקים שניתנו).

**בהצלחה!**