

**מכללת הדסה, החוג למדעי המחשב**  
**מבוא לתוכנות מונחה עצמים והנדסת תוכנה**  
**סמסטר א', תשפ"א**

**תרגיל 4**

**תאריך אחרון להגשה:**

הנביאים – יום א', 20/12/2020, בשעה 23:59

**מטרת התרגיל:**

גם בתרגיל זה נמשיך לעסוק בתכנון ממשק ונוסאים אחרים שכבר עסקנו בהם. בנוסף, בתרגיל זה נלמד להשתמש בספריה הגրפית SFML ונוכל לתרגל שימוש בירושה.

**הנחיות להתקנת הספרייה:**

קיים באתר דף הנחיות להתקנת הספרייה ולהגדרת פרויקט ב-Visual Studio כך שיישתמש בספריה. עקבו אחריו בקפדנות. כדי להבטיח שהפרויקט שלכם יתקמפל גם אצל הבודק, אין מנוס מהתקנת הספרייה לפי ההוראות הנ"ל. כמו כן, לצורך הגדרת הפרויקט שלכם לתרגיל, מומלץ להיעזר בקובץ הפרויקט של דוגמאות השימוש ב-SFML שנעלה למודול (הקובץ SFMLEExamples.zip).

**תיאור כללי:**

בתרגיל זה נבנה צייר שלבים גרפי עבור המשחק Lode Runner שייצרנו בתרגיל 2. המשמש יכול לבחור היכן למקם את הדמויות השונות במשחק ואיך הוא רוצה Shiraea לוח המשחק שלו. לאחר שישים לצייר את הלוח, תהיה לו אפשרות לשמר אותו בקובץ, כמו זה שהתקבל כתובן למשחק בתרגיל 2.

**פירוט הדרישות:**

**הצייר** יהיה חלון בודד, שבו כל אפשרות הציור. בפתיחת הראשונה של הצייר יהיו שתי אפשרויות:

- אם קובץ הקלט "Board.txt" (עליו נרჩיב בהמשך) עדין לא קיים, מה שיופיע זה משטח עריכה נקי, ואז יוצג חלון נוסף (אפשר בעזרת הטרמינל, ראו בהמשך) שմבקש מהמשתמש להכנס את גודל הלוח (גובה ורוחב) אותו הוא רוצה לעצב.

2. אם קובץ הקלט "Board.txt" קיים, יוצג מסطח עריכה שמכיל את האובייקטים כפי שטענו מהקובץ. באופן כללי, איןכם צריכים לבדוק את תקינות הקובץ ואתם יכולים להניח שאם הקובץ קיים, הוא אכן קובץ תקין. בהמשך נרחיב מה הפורמט של אותו קובץ.

### חלון התכנית ירכיב מוחלקים הבאים:

1. שטח שיישמש כמשטח העריכה, בו "נצח".
2. תפריט שיופיע מצד **שמאל** של ה"דף".

התפריט, שכפי שנאמר ימוקם בשמאל החלון, מאפשר למשתמש לבחור את סוג האובייקט שהוא רוצה לשים על הלוח ואת הפעולות הרצויות.

התפריט יכול את הcptories הבאים:

1. **cptor עבר כל אחד מסוגי האובייקטים שהוגדרו במשחק** וכפי שהם מפורטים שוב להלן, לבחירת הצורה לציר. אם המשתמש לחץ על אחד מהאובייקטים בתפריט, הוא יוכל עכשו ללחוץ במשטח העריכה וקר למקם את האובייקט, אם זה אפשרי כמובן. כל לחיצה עם העכבר על משטח העריכה תשים עליו אובייקט חדש מהסוג שנבחר בתפריט הכללים, ככלمر אין צורך לפני הcptor לפני כל הוספה אובייקט, מספיק לבחור פעם אחת ועוד אפשר להניח מספר אובייקטים חדשים מאותו הסוג. כשהעכבר עובר על המשטח במצב ציר, נdag שיהיה ברור מה הולך לkrות. האפשרות האידיאלית היא לצייר כבר את האובייקט כפי שהוא הולך להיות מונח על הלוח (אם אפשר להניח אותו במקום הנוכחי) אבל "SKUOF למחצה" והוא יוזץ יחד עם העכבר (עד שהלחיצה תקבע אותו במקום מסוים וזה לא יהיה SKUOF למחצה יותר). אפשרות פשוטה יותר היא לצייר מראש את משטח הציר עם רשת של משבצות, כך שיהיה ברור CSLUCHIM באיזו משבצת ימוקם האובייקט החדש ונdag את הריבוע שלו הUBLIC מצביע כרגע (ושוב, נבדיל בין מצב שבו ניתן לצייר על הריבוע הזה או שלא ניתן).

2. **cptor למחיקה** כדי למחוק אובייקט קיים. לאחר לחיצה על cptor המחיקה בתפריט, לחיצה על אובייקט שנמצא על משטח העריכה תמחק אותו. כדי שיהיה ברור איזה אובייקט עומד להימחק, כשהעכבר רק עובר (עוד לפני שיש לחיצה), נdag את הריבוע או האובייקט שהעכבר עומד מעליו, ובבדיל בין מצב שבו אנחנו מעל אובייקט או מעל ריבוע ריק (ואין מה לבחור). גם כאן, כמו בcptories מסעיף 1, אין צורך ללחוץ על cptor המחיקה לפני כל מחיקת אובייקט, מספיק ללחוץ על cptor המחיקה פעם אחת ועוד אפשר למחוק מספר אובייקטים מהלota.

3. **כפטור "דף חלון"** לניקוי משטח העריכה על ידי מחייבת כל האובייקטים שנוצרו לפני כן. לחיצה על הכפטור זהה גם מאפשר למשתמש לקבוע את גודל משטח העריכה (גודל השלב). עלייכם להציג חלון (אותו הזכרנו בתיאור הכללי) שיבקש מהמשתמש להקליד: 1. את מספר השורות בשלב (כלומר גובה השלב), 2. את מספר העמודות בשלב (כלומר רוחב השלב). כמובן לאחר מכן עלייכם להתאים את גודל השלב ומשטח העריכה בהתאם. שימוש לבן הפעם הלוח **לא** ריבועי. כדי להקל, אפשר לקבל את הקטלט הזה דרך חלון הטרמינל. אם בחרתם באפשרות ההז, הציגו הודעה על החילון הגרפי שմבוקשת מהמשתמש להכניס את המידע בחילון הטרמינל, וגם בחילון הטרמינל הציגו בקשה מתאימה. כמו כן, תצטרכו להגדיר את התוכנית Console Application ולא C-application Win32 Application כדי שבכל יוצג חילון טרמינל. איך תעשו זאת ב-

CMake המשלב SFML? למשל בפרויקט SFMLTest, כל שתצטרכו לעשות זה לשנות את השורה:

```
add_executable (SFMLTest WIN32 "")
```

להיות:

```
add_executable (SFMLTest "")
```

ולמעשה למחוק את המילה "WIN32".

4. **כפטור שמירה**, לשמרות השלב לקובץ.

#### האובייקטים האפשריים הם: שחזור, אובייב, מטבח סוף, קיר/רצפה, סולם, מוט.

לבחירתכם מה קורה אם המשמש מסנה לשים יותר משחקן אחד, תור מגבלה אחת: לא ניתן לשים יותר משחקן אחד, כי אנחנו רוצים שקובץ התוצאה יוכל לשמש כ-סקור למשחק (כנראה שיש שתי אפשרויות אם יש כבר שחזור: 1. לא ניתן יהיה לשים שחזור נוסף באותה מקום אחר לפני שМОחקים את הראשון. 2. השחקן הראשון ימחק אוטומטית כששים שחזור חדש באותה מקום אחר).

לא ניתן מצב בו שחזור נמצא על סולם (כלומר, באותה מסויים לא יכול להיות גם סולם וגם שחזור).

באחריות המשמש בסדר נקבע את הקירות ומטבחות הכסף על מנת לאפשר ניצחון במהלך המשחק, ככלומר שלא יוצר מצב שיש מטבח שלא ניתן לגשת אליו, מכיוון שהוא חסום בקירות.

הציג האובייקטים על משטח העריכה יכולה להיות זהה להציגתם בתרגיל 2, כך:

# – קיר/רצפה.

@ – השחזור.

% – אובייב.

- – מוט.

\* - מطبع סוף.

ניתן כמובן להציג אותם גם בכל דרך אחרת. (אתם משתמשים בספריה גרפית, תחשבו בגודל הnickod בהתאם)

**הקובץ בו נשמר את הנתונים** יהיה קובץ קבוע בשם Board.txt, שאותו נטען אוטומטית (אם קיים) בתחילת ריצת התוכנית וכל לחריצה על כפתור השמירה שומרת את המידע אליו. הczyir שלו יתמור לשימור לקובץ משחק המכיל שלב אחד בלבד, שהוא למעשה המשך שהמשמש ערך.

פורמט הקובץ הנתון צריך להיות צמוד לפורמט שקבענו בתרגיל 2, למעט הבדל אחד: שהפעם הלוח לאו דווקא ריבועי. כאמור השורה הראשונה בקובץ תכיל **שני** מספרים: הראשון יהיה מספר המגדיר את גובה השלב והשני יהיה מספר המגדיר את רוחב השלב. שני המספרים הללו ימולאו להיות המספרים שהמשמש הקיימים בתחילת העריכה. השורות הבאות יתארו את מבנה השלב בדומה לתרגיל 2 ('#' – קיר, '@' – שחקן, '%' – איבר, '\*' – מطبع, 'A' – סולם, '-' – מوط, '' (התו רוח) – תא ריק), כך תוכלן לשחק בשלב שבניהם, בהנחה שבניהם שלב ריבועי, עם התאמות מינימליות בקובץ, ולא שום שינוי בקוד שלכם מתרגיל 2. שימו לב, כאמור לעיל, הקובץ הנשמר יכול רק שלב אחד. אגב, למרות שהקובץ מכיל מספר המגדיר את רוחב השלב, תוכלן לאגלות את זה בעוד דרך: פשוט על ידי קריית השורות המתארות את מבנה השלב. וזאת מכיוון שאפשר להניח שהקובץ חוקי, ורוחב כל השורות בו זהות במספר הנתון.

## הערות:

- אין צורך בשלב זה לעדכן את תרגיל 2. כל מטרת התרגיל הוא להכין קובץ שייעבוד בפרויקט שתכינו בסוף הסמסטר.
- כדי לגרום לכך שהמחלקות שניצור תהינה משמעויות, השימוש ב-SFML צריך להיות מצומצם רק לפעולות הציור עצמה, **כלומר האובייקטים לא יירשו מחלקות של SFML ולא ייכלו אובייקט צזה**. רק בפונקציית הציור של האובייקט ניצור את האובייקט המתאים של SFML לפי הפרמטרים שהמורים באובייקט שלו, ונציג אותו (נבהיר כאן שהכוונה היא רק לאובייקטים שמייצגים את המחלקה – כמו sf::Text, sf::Sprite וכו'). שאר מיני ה-`sf::` מוטר **ואף רצוי** להזיז members לבעלי תפקידים מסוימים צבע, Vector2f, Font או Texture (או רפרנס/מצבייע לכאליה). נשים לב שאובייקטים מסוג Font או Texture נקייד לטעון במינימום הנדרש, ובלי העתקות, אך לא נעשה אותן members של האובייקטים המונחים על הלוח, אלא של אובייקט אחראי כלשהו).
- ניתן לכם פה כר נרחב להחלטות אישיות על עיצוב המחלקות והמשחק, בפרט בדברים הקשורים לדרכו הנכונה לניהול התקשרות עם הספריה הגרפיה והאירועים השונים.

- ניתן להשתמש בירושה כדי להגדיר את מחלקות האובייקטים השונות (שימוש לב שהתייכן במקרה זה צריך להתאים לעורך ולא למשחק).
- במידה והשתמשו בירושה - נקבע על ממשק אחד בין מחלוקת הבסיס והמחלקות היורשות.
- שימוש לב מה צריך להיות מחלקות שונות ומה יכול להיות רק הבדל במאפיינים הנשמרים.

## **הערות כלליות:**

הרבה פרטים הוגדרו לעיל במדדוק, אולם תמיד, בכל פרויקט מסוים מרכיב, יש התנהוגיות שלא מוגדרות במדדוק על ידי הדרישות. אפשר להתייעץ במקרה הצורך, אבל ההנחייה הכללית היא שלא לדברים שלא נאמרו בהגדרות אתם מוזמנים להחליט בעצמכם מה הדרך הטובה יותר לביצוע הפעולה או לפתרון הבעיה (כלומר, טובה יותר מבחינה משחקיות, או סבירה מבחינת משחקיות וקללה מבחינה תכניתית : ) . בכל החלטה כזו שלכם, אתם מוזמנים לטעד את הדברים בקובץ README שלכם, ולכתוב שם מודיע בחרתם בהחלטה זו.

## **תיעוד המחלקות:**

אין צורך לטעד קוד שהוא מובן מלאיו (למשל בגלל שמות שימושיים), זה מיותר גם בהצהרות וגם במימושים. אבל אם מדובר ברכיב ייחודי בלתי מובן או מדובר בפונקציות שהן פחות טריוויאליות יש להוסיף תיעוד בקוד.

## **קובץ README:**

יש לכלול קובץ README שנקרא README.txt או README.doc, README.docx (ולא בשם אחר). הקובץ יכול להכתב בעברית בלבד שיכיל את הסעיפים הנדרשים.

קובץ זה יכול לכל הפחות:

1. כותרת.
2. פרטי הסטודנט: שם מלא כפי שהוא מופיע בראשימות המכלה, ת"ז.
3. הסבר כללי של התרגיל.
4. **תיקון (design):** הסבר קצר מהם האובייקטים השונים בתוכנית, מה התפקיד של כל אחד מהם וחולקת האחריות ביניהם ואיך מתבצעת האינטראקציה בין האובייקטים השונים.
5. רשיימה של הקבצים שנוצרו ע"י הסטודנט, עם הסבר קצר (לרוב לא יותר משורה או שתיים) לגבי תפקידן הקובץ.
6. מבני נתונים עיקריים ותפקידיהם.
7. אלגוריתמים הרואים לצוין.

8. באגים ידועים.

9. הערות אחרות.

יש לתמצת כל שניתן אך לא יותר על אף חלק. אם אין מה להגיד בנושא מסוים יש להשאיר את הכותרת ומתחתיה פסקה ריקה. כתובו README כל דבר שרצוי שהבודק ידע כשהוא בודק את התרגיל.

## אוף ההגשה:

הקובץ להגשה: יש לדוחו כל קובץ הקשור לתרגיל, כמעט מה שיצין להלן, לקובץ ששמו exN\_firstname\_lastname.zip, כאשר N הוא מספר התרגיל וfirstname lastname הוא השם המלא (לדוגמא אלברט איינשטיין יgive קר את התרגיל הראשון: ex1\_albert\_einstein.zip). במקרה של הגשה בזוג, שם הקובץ יהיה לפי התבנית zip.zip, exN\_firstname1\_lastname1\_firstname2\_lastname2.zip, עם שמות המציגים בהתאם (לא רוחים; לעומת זאת, גם במקרים שבהם יש להחילף רוחים בזווית, כפי המודגם לעיל). כמו כן, במקרה של הגשה בזוג, רק אחד מהציגים יgive את הקובץ ולא שנייהם.

לפני דחיסת תקיות הפROYיקט שלכם יש למחוק את הפריטים הבאים:

- תיקייה בשם out, אם קיימת
- תיקייה בשם .vs

שתי התקיות האלה נמצאות בתיקייה הראשית (זו שנחנו פותחים בעזרת SV). התקייה .vs. לעיתים מוסתרת, אבל אם תפתחו את קובץ הקובץ שיצרתם, בוודאי תוכלו למצוא אותה ולמחוק אותה. ככל אצבע, אם קובץ הקובץ שוקל יותר ממ"ב אחד או שניים, נראה שלא מחקתם חלק מהקבצים הבינאריים המוזכרים.

וודאו כי קובץ הקובץ מכיל **תיקייה ראשית אחת**, ורק בתוכה יהיו כל הקבצים והתיקיות של הפROYיקט.

את הקובץ יש להעלות ל-Moodle של הקורס למשימה המתאימה.

**הגשה חוזרת:** אם מסיבה כלשהי סטודנט מחליט להגיש הגשה חוזרת יש לוודא שם הקובץ זהה לchlotin לשם הקובץ המקורי. אחרת, אין הבודק אחראי לבדוק את הקובץ המקורי שיוגש.

כל שינוי מהה שוגדר פה לגבי צורת ההגשה ומבנה README עלול לגרום הורדת נקודות בציון.

מספר הערות:

1. שימוש לב שם הקובץ שאכן יכול את שמות התציגים.
2. שימוש לב שעיליכם לשולח את תקיות הפROYיקט כולה, לא רק את קובצי הקובץ שיצרתם. תרגיל שלא יכול את כל הקבצים הנדרשים, לא יוכל וידרש הגשה חוזרת (עם כללי האיחור הרגילים).

המלצת כללית: אחרי שהכנתם את הקובץ להגשה, העתיקו אותו לתיקייה חדשה, חלצו את הקבצים שבתוכו ובדקו אם אתם מצליחים לפתח את התיקייה זו ולקמפל את הקוד. הרבה טעויות של שכחת קבצים יכולות להימנע על ידי בדיקה כזו.

**בהצלחה!**