

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

תרגיל 4

תאריך אחרון להגשה:

הנביאים – יום ב', 26/12/2021, בשעה 23:59

שטראוס גברים – יום א', 25/12/2021, בשעה 23:59

שטראוס נשים – מוצאי שבת, 24/12/2021, בשעה 23:59

מטרת התרגיל:

גם בתרגיל זה נמשיך לעסוק בתכנון ממשק ונושאים אחרים שכבר עסקנו בהם. בנוסף, בתרגיל זה נלמד להשתמש בספרייה הגרפית SFML ונוכל לתרגל שימוש בירושה.

הנחיות להתקנת הספרייה:

קיים באתר דף הנחיות להתקנת הספרייה ולהגדרת פרויקט ב-Visual Studio כך שישתמש בספרייה. 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 בקובצי הפרויקט של דוגמאות השימוש ב-SFML שנגלה למודל (הקובץ SFMLExamples.zip).

תיאור כללי:

בתרגיל זה נבנה צייר שלבים גרפי עבור המשחק Super-Pacman שיצרנו בתרגיל 2. המשתמש יוכל לבחור היכן למקם את הדמויות השונות במשחק ואיך הוא רוצה שיראה לוח המשחק שלו. לאחר שסיים לצייר את הלוח, תהיה לו אפשרות לשמור אותו בקובץ (ובקובץ שנוצר - נוכל להשתמש בהמשך כקלט לפרויקט!).

פירוט הדרישות:

הצייר יהיה חלון בודד, ובו כל אפשרויות הציור. בפתיחה הראשונה של הצייר יהיו שתי אפשרויות:

1. אם קובץ הקלט "Board.txt" (עליו נרחיב בהמשך) עדיין לא קיים, מה שיופיע זה משטח עריכה נקי, ואז יוצג חלון נוסף (אפשר בעזרת הטרמינל, ראו בהמשך) שמבקש מהמשתמש להכניס את גודל הלוח (גובה ורוחב) אותו הוא רוצה לעצב.

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

הערה: כנזכר לעיל, כשמריצים את התוכנית – הקובץ לא בהכרח קיים. במידה והוא קיים, או לאחר ששמרנו את הלוח – מיקום הקובץ יהיה בתיקיה בה נמצאת התוכנית המקומפלת.

חלון התכנית יורכב מהחלקים הבאים:

1. שטח שימש כמשטח העריכה, בו "נצייר".

2. תפריט שימוקם **בצד ימין** של ה"דף".

התפריט, שכאמור ימוקם בצד ימין, יאפשר למשתמש לבחור את סוג האובייקט שהוא רוצה למקם על הלוח ואת הפעולות הרצויות.

התפריט יכיל את הכפתורים הבאים:

1. **כפתור עבור כל אחד מסוגי האובייקטים שהוגדרו במשחק** וכפי שהם מפורטים שוב להלן, לבחירת הצורה לציור. אם המשתמש לחץ על אחד מהאובייקטים בתפריט, הוא יכול עכשיו ללחוץ על משטח העריכה וכך למקם את האובייקט, אם זה אפשרי כמובן. כל לחיצה עם העכבר על משטח העריכה תניח עליו אובייקט חדש מהסוג שנבחר בתפריט הכלים, כלומר אין צורך ללחוץ על הכפתור לפני כל הוספת אובייקט, מספיק לבחור פעם אחת ואז אפשר להניח מספר אובייקטים חדשים מאותו הסוג. כשהעכבר עובר על המשטח במצב ציור, נדאג שיהיה ברור מה הולך לקרות. האפשרות האידיאלית היא לצייר כבר את האובייקט כפי שהוא הולך להיות מונח על הלוח (אם אפשר להניח אותו במקום הנוכחי) אבל "שקוף למחצה" והוא יזוז יחד עם העכבר (עד שהלחיצה תקבע אותו במקום מסוים ואז הוא לא יהיה שקוף למחצה יותר). אפשרות פשוטה יותר היא לצייר מראש את משטח הציור עם רשת של משבצות, כך שיהיה ברור כשלוחצים באיזו משבצת ימוקם האובייקט החדש ונדגיש את הריבוע שעליו העכבר מצביע כרגע (ושוב, נבדיל בין מצב שבו ניתן לצייר על הריבוע הזה או שלא ניתן).

2. **כפתור למחיקה** כדי למחוק אובייקט קיים. לאחר לחיצה על כפתור 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), נדגיש את הריבוע או האובייקט שהעכבר עומד מעליו, ונבדיל בין מצב שבו אנחנו מעל אובייקט או מעל ריבוע ריק (ואין מה לבחור). גם כאן, כמו בכפתורים מסעיף 1, אין צורך ללחוץ על כפתור המחיקה לפני כל מחיקת אובייקט, מספיק ללחוץ על כפתור המחיקה פעם אחת ואז אפשר למחוק מספר אובייקטים מהלוח.

3. **כפתור "דף חלק"** לניקוי משטח העריכה על ידי מחיקת כל האובייקטים שנוצרו לפני כן. לחיצה על הכפתור הזה גם תאפשר למשתמש לקבוע את גודל משטח העריכה (גודל השלב). עליכם להציג חלון שיבקש מהמשתמש להקליד: 1. את מספר השורות בשלב (כלומר גובה השלב), 2. את מספר העמודות בשלב (כלומר רוחב השלב). כמובן לאחר מכן עליכם להתאים את גודל השלב ומשטח העריכה בהתאם. כדי להקל, אפשר לקבל את הקלט הזה דרך חלון הטרמינל. אם בחרתם באפשרות הזו, הציגו הודעה על החלון הגרפי שמבקשת מהמשתמש להכניס את המידע בחלון הטרמינל, וגם בחלון הטרמינל הציגו בקשה מתאימה. כמו כן, תצטרכו להגדיר את התוכנית כ-Console Application ולא כ-Win32 Application כדי שבכלל יוצג חלון טרמינל. איך תעשו זאת ב-CMake המשלב SFML? כל שתצטרכו לעשות זה לשנות את השורה:

```
add_executable (${CMAKE_PROJECT_NAME} WIN32)
```

להיות:

```
add_executable (${CMAKE_PROJECT_NAME})
```

ולמעשה למחוק את המילה "WIN32".

4. **כפתור שמירה**, לשמירת השלב לקובץ.

האובייקטים האפשריים הם: פקמן, שדון, קיר, דלת, מפתח, עוגיה, מתנה.

לבחירתכם מה קורה אם המשתמש מנסה לשים את פקמן יותר מפעם אחת, תוך מגבלה אחת: פקמן יכול להופיע רק פעם אחת, כי אנחנו רוצים שקובץ התוצאה יוכל לשמש כ-input למשחק (כנראה שיש שתי אפשרויות אם הדמות של פקמן כבר מופיעה: 1. לא ניתן יהיה לשים את פקמן במקום אחר לפני שמוחקים את הדמות הקיימת. 2. הדמות הראשונה תימחק אוטומטית כששמים את פקמן במקום אחר).

באחריות המשתמש לסדר נכון את הלוח על מנת לאפשר מעבר שלב. למשל, שלא ייווצר מצב שיש עוגיה שלא ניתן להגיע אליה מכיון שהיא מוקפת בקירות.

הצגת האובייקטים על משטח העריכה יכולה להיות זהה להצגתם בתרגיל 2, כך:

a – פקמן.

& – שדון.

– קיר.

D – דלת.

% – מפתח.

* – עוגיה.

\$ – מתנה.

ניתן כמובן להציג אותם גם בכל דרך אחרת. (אתם משתמשים בספריה גרפית, תחשבו בגדול! הניקוד בהתאם)

הקובץ בו נשמור את הנתונים יהיה קובץ קבוע בשם Board.txt, שאותו נטען אוטומטית (אם קיים) בתחילת ריצת התוכנית וכל לחיצה על כפתור השמירה תשמור את המידע אליו. הצייר שלנו יתמוך בשמירה לקובץ משחק המכיל שלב אחד בלבד, שהוא למעשה המסך שהמשתמש ערך. פורמט הקובץ נתון לבחירתכם (ויכול להיות זהה לפורמט שאותו בחרתם בתרגיל 2. בכל מקרה, **עליכם לתעד בקובץ ה-Readme את הפורמט שבחרתם**, ולא רק להפנות ל-Readme של תרגיל 2). האובייקטים במשחק יתוארו בקובץ באמצעות אותם התווים בהם השתמשנו בתרגיל 2 ('a' – פקמן, '&' – שדון, '#' – קיר, 'D' – דלת, '%' – מפתח, '*' – עוגיה, '\$' – מתנה, ' ' (התו רווח) – אריח ריק), כך תוכלו לשחק בשלב שבניתם, עם התאמות מינימליות בקובץ, וללא שום שינוי בקוד שלכם מתרגיל 2. שימו לב, כאמור לעיל, הקובץ הנשמר יכיל רק שלב אחד.

הערות:

- גם אם החלטתם לשנות את הפורמט שבו השתמשתם בתרגיל 2, אין צורך לעדכן את תרגיל 2.
- כדי לגרום לכך שהמחלקות שניצור תהיינה משמעותיות, השימוש ב-SFML צריך להיות מצומצם רק לפעולת הציור עצמה, כלומר האובייקטים לא יירשו ממחלקות של SFML ולא יכילו אובייקט כזה. רק בפונקציית הציור של האובייקט ניצור את האובייקט המתאים של SFML לפי הפרמטרים השמורים באובייקט שלנו, ונצייר אותו (נבהיר כאן שהכוונה היא רק לאובייקטים שמייצגים את המחלקה, כמו sf::Text, sf::Sprite או שאר מיני ה-sf::Shape. מותר ואף רצוי להחזיק members מטיפוס צבע, Font, Vector2f, או Texture (או רפרנס/מצביע לכאלה). נשים לב שאובייקטים מסוג Font או Texture נקפיד לטעון במינימום הנדרש, ובלי העתקות, לכן לא נעשה אותם members של האובייקטים המונחים על הלוח, אלא של אובייקט אחראי כלשהו).
- ניתן לכם פה כר נרחב להחלטות אישיות על עיצוב המחלקות והממשק, בפרט בדברים שקשורים לדרך הנכונה לניהול התקשורת עם הספרייה הגרפית והאירועים השונים.
- ניתן להשתמש בירושה כדי להגדיר את מחלקות האובייקטים השונות (שימו לב שהתיכון במקרה הזה צריך להתאים לעורך ולא למשחק).
- אם השתמשנו בירושה – נקפיד על ממשק אחיד בין מחלקת הבסיס והמחלקות היורשות.
- שימו לב מה צריך להיות מחלקות שונות ומה יכול להיות רק הבדל במאפיינים הנשמרים.

הערות כלליות:

הרבה פרטים הוגדרו לעיל במדויק, אולם תמיד, בכל פרויקט מספיק מורכב, יש התנהגויות שלא מוגדרות במדויק על ידי הדרישות. אפשר להתייעץ במידת הצורך, אבל ההנחייה הכללית היא שעל דברים שלא נאמרו בהגדרות אתם מוזמנים להחליט בעצמכם מה הדרך הטובה יותר לביצוע הפעולה או לפתרון הבעיה (כלומר, טובה יותר מבחינת משחקיות, או סבירה מבחינת משחקיות וקלה מבחינה תכנותית:). בכל החלטה כזו שלכם, אתם מוזמנים לתעד את הדברים בקובץ README שלכם, ולכתוב שם מדוע בחרתם בהחלטה הזו.

תיעוד המחלקות:

אין צורך לתעד קוד שהוא מובן מאליה (למשל, בגלל שמות משמעותיים או אופרטור שמשמעותו ברורה), זה מיותר גם בהצהרות וגם במימושים. יש להוסיף תיעוד בקוד במקרים הבאים:

- תיעוד כללי (וקצר) על ממשק המחלקה (מטרת המחלקה עצמה ופונקציות ציבוריות למיניהן)
- בפונקציות פחות טריוויאליות (לא `setToDefault`, נניח)
- כשיש לוגיקה לא טריוויאלית (במימוש)
- במקרים של קוד לא ברור מאליה ו"מפתיע" (למשל, העברת ערך לא חוקי ככוונה כדי לקבל כתוצאה את ברירת המחדל)

קובץ ה-README

יש לכלול קובץ README שיקרא README.doc, README.docx או README.txt (ולא בשם אחר). הקובץ יכול להיכתב בעברית ובלבד שיכיל את הסעיפים הנדרשים.

קובץ זה יכיל לכל הפחות:

1. כותרת.
 2. פרטי הסטודנט: שם מלא כפי שהוא מופיע ברשימות המכללה, ת"ז.
 3. הסבר כללי של התרגיל.
 4. רשימה של הקבצים שיצרנו, עם הסבר קצר (לרוב לא יותר משורה או שתיים) לגבי תפקיד הקובץ.
 5. מבני נתונים עיקריים ותפקידיהם.
 6. אלגוריתמים הראויים לציון.
 7. תיכון (design): הסבר קצר מהם האובייקטים השונים בתוכנית, מה התפקיד של כל אחד מהם וחלוקת האחריות ביניהם ואיך מתבצעת האינטראקציה בין האובייקטים השונים.
 8. באגים ידועים.
 9. הערות אחרות.
- יש לתמצת ככל שניתן אך לא לוותר על אף חלק. אם אין מה להגיד בנושא מסוים יש להשאיר את הכותרת ומתחתיו פסקה ריקה. **נכתוב ב-README כל דבר שרצוי שהבודק ידע כשהוא בודק את התרגיל.**

אופן ההגשה

הקובץ להגשה: ניתן ליצור בקלות קובץ zip המותאם להגדרות ההגשה המפורטות להלן ישירות מתוך VS, כפי המוסבר תחת הכותרת "יצירת קובץ ZIP להגשה או לגיבוי" בקובץ "הנחיות לשימוש ב-VS 2022".
אנא השתמשו בדרך זו (אחרי שהגדרתם כראוי את שמות הצוות ב-MY_AUTHORS: **נשים את שמות המגישים בתוך המרכאות, נקייד להפריד בין השם הפרטי ושם המשפחה בעזרת קו תחתי ואם יש יותר ממגיש אחד, נפריד בין השמות השונים בעזרת מקף (מינוס '-')**) וכך תקבלו אוטומטית קובץ zip המותאם להוראות, בלי טעויות שיגררו אחר כך בעיות בבדיקה.

באופן כללי, הדרישה היא ליצור קובץ zip בשם oop1_exN-firstname_lastname.zip (או במקרה של הגשה בזוג – oop1_exN-firstname1_lastname1-firstname2_lastname2.zip), כשהקובץ כולל את כל קובצי הפרויקט, למעט תיקיות out ו-vs. כל הקבצים יהיו בתוך תיקייה ראשית אחת.

את הקובץ יש להעלות ל-Moodle של הקורס למשימה המתאימה. בכל מקרה, **רק אחד** מהמגישים יגיש את הקובץ ולא שניהם.

הגשה חוזרת: אם מסיבה כלשהי החלטתם להגיש הגשה חוזרת יש לוודא ששם הקובץ זהה לחלוטין לשם הקובץ המקורי. אחרת, אין הבודק אחראי לבדוק את הקובץ האחרון שיוגש.

כל שינוי ממה שמוגדר פה לגבי צורת ההגשה ומבנה ה-README עלול לגרום הורדת נקודות בציון.

מספר הערות:

1. נשים לב לשם הקובץ שאכן יכלול את שמות המגישים.
 2. נשים לב לשלוח את תיקיית הפרויקט כולה, לא רק את קובצי הקוד שהוספנו. תרגיל שלא יכלול את כל הקבצים הנדרשים, לא יתקבל וידרוש הגשה חוזרת (עם כללי האיחור הרגילים).
- המלצה כללית: אחרי הכנת הקובץ להגשה, נעתיק אותו לתיקייה חדשה, נחלץ את הקבצים שבתוכו ונבדוק אם ניתן לפתוח את התיקייה הזו ולקמפל את הקוד. הרבה טעויות של שכחת קבצים יכולות להימנע על ידי בדיקה כזו.

בהצלחה!