

מכללת הדסה, החוג למדעי המחשב

מבוא לתכנות 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 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 שנעלה למודל (הקובץ SFMLExamples.zip).

תיאור כללי:

בתרגיל זה נבנה צייר שלבים גרפי עבור המשחק Lode Runner שיצרנו בתרגיל 2. המשתמש יוכל לבחור היכן למקם את הדמויות השונות במשחק ואיך הוא רוצה שיראה לוח המשחק שלו. לאחר שיסיים לצייר את הלוח, תהיה לו אפשרות לשמור אותו בקובץ, כמו זה שהתקבל כ־input למשחק בתרגיל 2.

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 שטענו מהקובץ. באופן כללי אינכם צריכים לבדוק את תקינות הקובץ ואתם יכולים להניח שאם הקובץ קיים, הוא אכן קובץ תקין. בהמשך נרחיב מה הפורמט של אותו קובץ.

חלון התכנית יורכב מהחלקים הבאים:

1. שטח שישמש כמשטח העריכה, בו "נצייר".

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

התפריט, שכפי שנאמר ימוקם בשמאל החלון, יאפשר למשתמש לבחור את סוג האובייקט שהוא רוצה לשים 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 במשטח העריכה וכך למקם את האובייקט, אם זה אפשרי כמובן. כל לחיצה עם העכבר על משטח העריכה תשים 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 (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 עליכם להתאים את גודל השלב ומשטח העריכה בהתאם. שימו לב! הפעם הלוח **לא** ריבועי. 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? למשל בפרויקט SMFLTest, כל שתצטרכו לעשות זה לשנות את השורה:

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

להיות:

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

ולמעשה למחוק את המילה "WIN32".

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

האובייקטים האפשריים הם: שחקן, אויב, מטבע כסף, קיר/רצפה, סולם, מוט.

לבחירתכם מה קורה אם המשתמש מנסה לשים יותר משחקן אחד, תוך מגבלה אחת: לא נאפשר לשים יותר משחקן אחד, כי אנחנו רוצים שקובץ התוצאה יוכל לשמש כ-input למשחק (כנראה שיש שתי אפשרויות אם יש כבר שחקן: 1. לא ניתן יהיה לשים שחקן נוסף במקום אחר לפני שמוחקים את הראשון. 2. השחקן הראשון יימחק אוטומטית כששמים שחקן חדש במקום אחר).

לא יתאפשר מצב בו שחקן נמצא על סולם (כלומר, בתא מסויים לא יכול להיות גם סולם וגם שחקן).

באחריות המשתמש לסדר נכון את הקירות ומטבעות הכסף על מנת לאפשר ניצחון במשחק, כלומר שלא ייווצר מצב שיש מטבע שלא ניתן לגשת אליו, מכיון שהוא חסום בקירות.

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

– קיר/רצפה.

@ – השחקן.

% – אויב.

- - מוט.

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 שומרת את המידע אליו. הצייר שלנו יתמוך לשמור לקובץ משחק המכיל שלב אחד בלבד, שהוא למעשה המסך שהמשתמש ערך. פורמט הקובץ הנתון צריך להיות צמוד לפורמט שקבענו בתרגיל 2, למעט הבדל אחד: שהפעם הלוח לאו דווקא ריבועי. כלומר 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 שהמשתמש הכניס בתחילת העריכה. השורות הבאות יתארו את מבנה השלב בדומה לתרגיל 2 ('#' – קיר, '@' – שחקן, '%' – אויב, '*' – מטבע, 'H' – סולם, '-' – מוט, ' ' (התרו רוח) – תא ריק), כך תוכלו לשחק בשלב שבניתם, בהנחה שבניתם שלב ריבועי, עם התאמות מינימליות בקובץ, וללא שום שינוי בקוד שלכם מתרגיל 2. שימו לב, כאמור לעיל, הקובץ הנשמר יכיל רק שלב אחד. אגב, למרות שהקובץ מכיל מספר המגדיר את רוחב השלב, תוכלו לגלות את זה בעוד דרך: פשוט על ידי קריאת השורות המתארות את מבנה השלב. וזאת מכיוון שאפשר להניח שהקובץ חוקי, ורוחב כל השורות בו זהות למספר הנתון.

הערות:

- אין צורך בשלב זה לעדכן את תרגיל 2. כל מטרת התרגיל הוא להכין קובץ שיעבוד בפרויקט שתכינו בסוף הסמסטר.
- 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 sf::Shape). מותר ואף רצוי להחזיק members מטיפוס צבע, Vector2f, Texture או Font (או רפרנס/מצביע לכאלה). נשים לב שאובייקטים מסוג Texture או Font 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).
- במידה והשתמשנו 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 במקרה 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 מאליו (למשל בגלל שמות משמעותיים), זה מיותר גם בהצהרות וגם במימושים. אבל אם מדובר ברכיב ייחודי בלתי מובן או מדובר בפונקציות שהן פחות טריוויאליות יש להוסיף תיעוד בקוד.

קובץ ה־README:

יש לכלול קובץ README שיקרא README.docx, README.doc, 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. תיכון (design): הסבר קצר מהם האובייקטים השונים בתוכנית, מה התפקיד של כל אחד מהם וחלוקת האחריות ביניהם ואיך מתבצעת האינטראקציה בין האובייקטים השונים.
5. רשימה של הקבצים שנוצרו ע"י הסטודנט, עם הסבר קצר (לרוב לא יותר משורה או שתיים) לגבי תפקיד הקובץ.
6. מבני נתונים עיקריים ותפקידיהם.
7. אלגוריתמים הראויים לציון.

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 ומתחתיה פסקה ריקה. תכתבו ב-README כל דבר שרצוי שהבודק ידע כשהוא בודק את התרגיל.

אופן ההגשה:

הקובץ להגשה: יש לדחוס כל קובץ הקשור לתרגיל, למעט מה שיצוין להלן, לקובץ ששמו `exN_firstname_lastname.zip`, כאשר N הוא מספר התרגיל ו-`firstname_lastname` הוא השם המלא (לדוגמא אלברט איינשטיין יגיש כך את התרגיל הראשון: `ex1_albert_einstein.zip`). במקרה של הגשה בזוג, שם הקובץ יהיה לפי התבנית `exN_firstname1_lastname1_firstname2_lastname2.zip`, עם שמות המגישים בהתאמה (ללא רווחים; כלומר, גם בשמות עצמם יש להחליף רווחים בקו תחת, כפי המודגם לעיל). כמו כן, במקרה של הגשה בזוג, רק אחד מהמגישים יגיש את הקובץ ולא שניהם.

לפני דחיסת תיקיית הפרויקט שלכם יש למחוק את הפריטים הבאים:

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

שתי התיקיות האלה נמצאות בתיקייה הראשית (זו שאנחנו פותחים בעזרת VS). התיקייה `vs`. לפעמים מוסתרת, אבל אם תפתחו את קובץ ה-`zip` שיצרתם, בוודאי תוכלו למצוא אותה ולמחוק אותה. ככלל אצבע, אם קובץ ה-`zip` שוקל יותר ממ"ב אחד או שניים, כנראה שלא מחקתם חלק מהקבצים הבינאריים המוזכרים.

וודאו כי קובץ ה-`zip` מכיל תיקייה ראשית אחת, ורק בתוכה יהיו כל הקבצים ותתי התיקיות של הפרויקט.

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

הגשה חוזרת: אם מסיבה כלשהי סטודנט מחליט 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 לגרור הורדת נקודות בציון.

מספר הערות:

1. שימו לב לשם הקובץ שאכן יכול את שמות המגישים.

2. שימו לב שעליכם לשלוח את תיקיית הפרויקט כולה, לא רק את קובצי הקוד שיצרתם. תרגיל שלא יכול את כל הקבצים הנדרשים, לא יתקבל וידרוש הגשה חוזרת (עם כללי האיחור הרגילים).

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

בהצלחה!