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

תרגיל 2

יומן שינויים:

הפחתת דרישות:

- התוכנית שלכם יכולה להניח שבכל שלב מספר המפתחות **זהה** למספר הדלתות.
- התוכנית שלכם יכולה להניח שיש **לפחות** עוגיה אחת בכל שלב.

הוספת דרישות:

- בלוח התצוגה יופיע גם מספר החיים שנותרו.
- (דרישות למימוש) קובץ main.cpp חייב להיות קטן.

תאריך אחרון להגשה:

הנביאים – יום א', 27/11/2022, בשעה 23:59

שטראוס גברים – יום א', 27/11/2022, בשעה 23:59

שטראוס נשים – מוצאי שבת, 26/11/2022, בשעה 23:59

מטרת התרגיל

בתרגיל זה נתרגל תכנון ממשק (Interface designing), שימוש במחלקות מורכבות (מחלקות המכילות אובייקטים של מחלקות אחרות), העברת מידע בין אובייקטים, וכתלות בהתקדמות ההרצאות גם נושאים של שימוש נכון בהפניות (References) וב־const.

תיאור כללי

בתרגיל זה נבנה תוכנית המשתמשת במספר אובייקטים. לשם ביצוע המשימה האובייקטים יצטרכו להעביר מידע ביניהם.

התוכנית אותה עליכם לממש היא סופר פקמן (Super-Pacman) בעל שלושה שלבים לפחות.
קישור לסרטון של המשחק: https://www.youtube.com/watch?v=KYpuKc86gT8
בהמשך יפורטו הכללים המדויקים של המשחק כפי שתממשו אותו.
באופן כללי, למי שאינו מכיר: המשחק בנוי ממבוך אשר בו מפוזרות עוגיות ומתנות שאותן על השחקן, פקמן, לאכול. בזמן שהשחקן אוכל להנאתו, מסתובבים במבוך מספר שדונים שרוצים לאכול את השחקן.

אתם צריכים לחשוב על התיכון (Design) המתאים מבחינת אלו מחלקות צריכות להיות, חלוקת האחריות ביניהן, אלו פונקציות יהיו בכל אחת וכו'. **שימו לב** שאתם נדרשים לתאר את התיכון ולהסביר אותו בקובץ ה-README (בנוסף ל-README "הרגיל" מהתרגיל הקודם. ראו בסוף הקובץ, בחלק המתאר את קובץ ה-README).

פירוט הדרישות

מטרת המשחק וסיומו

מטרתו של השחקן (-הפקמן) היא לגמור את העוגיות בכל שלב, מבלי שהשדונים יתפסו אותו. השחקן מסיים בהצלחה את השלב כאשר נגמרות כל העוגיות על המסך, ומופיע מיד המסך הבא, כלומר השלב הבא. המשחק ממשיך כל עוד יש שלבים נוספים והשחקן לא נפסל. לשחקן יש שלושה "חיים". המשחק יסתיים בסיום השלב האחרון, או לאחר שהשחקן נאכל שלוש פעמים ע״י שדון.

האובייקטים במשחק

הדמויות:

- **הפקמן בצורתו הרגילה**. כאמור, מטרת המשחק היא לאכול את כל העוגיות שבשלב. בכל התנגשות עם שדון הוא ימות. הפקמן לא יכול לעבור דלתות, וכמובן לא יכול לעבור קירות.
  הוא יסומן בתו a.

- **הפקמן בצורת "הסופר פקמן"**. אוכל עוגיות כמו בצורתו הרגילה, אבל הוא יכול לשבור דלתות (כלומר לסלק אותן), והוא לא מת בהתנגשות עם שדון (כלומר לא קורה לו כלום). הוא לא יכול לעבור קירות.
  הוא יסומן בתו @.

- **השדון**. רודף אחרי הפקמן, והורג אותו בהתנגשות (-רק אם הפקמן הוא בצורתו הרגילה).
  הוא יסומן בתו &.

סוגי העצמים שבמשחק

בלוח המשחק יכולים להופיע העצמים הבאים:

- **קיר**. אף דמות לא יכולה לעבור קירות.
  יסומן בתו #.

- **דלת**. שדונים ופקמן בצורתו הרגילה לא עוברים דלתות, אבל כאמור, פקמן בצורת סופר-פקמן שובר את הדלת, כלומר הדלת נעלמת אחרי שהוא עובר.
  תסומן בתו D.

- **מפתח**. כשהפקמן אוסף מפתח, המפתח נעלם, ויחד איתו נעלמת דלת אחת (או דלת קבועה למפתח הזה או דלת רנדומלית, תלוי במימוש שבחרתם).
  יסומן בתו %.

- **עוגיה**. פקמן אוכל אותן. כששדון עולה על עוגיה לא קורה לה כלום והיא נשארת במקומה. את מי יראו? במקום שיש גם עוגיה וגם שדון, יראו רק את התו של השדון, ולא של העוגיה (למרות שהעוגיה עדיין שם).
תסומן בתו *.

- **מתנה**. פקמן אוכל אותן והופך לסופר-פקמן למשך 20 צעדים (כפי שיוסבר בהמשך, זהו משחק צעדים/תורות, ולא משחק "בזמן אמת"). כששדון דורך על מתנה, לא קורה לו כלום, והמתנה נשארת במקומה. את מי יראו? במקום שיש גם מתנה וגם שדון, יראו רק את התו של השדון, ולא של המתנה (למרות שהמתנה עדיין שם).
תסומן בתו $.

מהלך המשחק ותנועת הדמויות

מהלך המשחק

לשם פישוט, המשחק לא מתנהל "בזמן אמת" לפי שעון, כמו המשחק הרגיל, אלא לפי תור. המשתמש יעשה את התור של הפקמן, ואז המחשב (כלומר אתם תצטרכו לממש את זה...) יעשה את התורות של כל השדונים. בין שדון לשדון ניתן להוסיף **(אין זה חובה!)** המתנה קצרה, כדי שיהיה ברור למשתמש מי זז ולהיכן. איך? על ידי הוספת השורות הבאות בתחילת הקובץ:

```
#include <thread>
#include <chrono>
```

והוספת השורה הבאה במקום המתאים (תוכלו לשחק עם זמן ההשהיה, באופן שייתן בעיניכם משחקיות טובה):

```
std::this_thread::sleep_for(std::chrono::milliseconds(200));
```

תזוזת הדמויות

בכל תור דמות יכולה לזוז לאחד מארבעת הכיוונים (אך לא באלכסון). דמות זזה בכל תור רק משבצת אחת.

נבחין בין המקרים הבאים לגבי התא המבוקש:

- אם התא המבוקש חסום (יש בו מכשול, # או D), התנועה לא תתבצע. מה כן יקרה? נבדיל בין השחקן לבין השדונים, אם השחקן ניסה לזוז בכיוון המכשול ונחסם, התנועה לא תתבצע אך גם התור נשאר שלו, עד שהוא יזוז למקום חוקי או עד שהוא יוותר במפורש על התור שלו (ראו בהמשך). לעומת זאת, במקרה שאלגוריתם התזוזה של השדון ניסה צעד כזה ונחסם, הוא מפסיד את התור, והתור עובר לדמות הבאה, לפי הסדר שתואר.

- התנהגות גבולות המסך נתונה לבחירתכם. אפשרות אחת היא לממש את גבולות המסך כ"קירות", כלומר, גם אם ניתן להגיע למשל לקצה השמאלי של המסך בשורה מסוימת,

לחיצה שמאלה נחשבת כמו ניסיון כניסה לתא חסום (עם הכללים דלעיל). אפשרות אחרת היא לממש את גבולות המסך כמו במשחק המקורי, כך שהקצה הימני מחובר לקצה השמאלי, והקצה העליון מחובר לקצה התחתון. למשל הליכה לקצה המסך השמאלי, תגרום לדמות להופיע בצד הימני של המסך (אם אין שם קיר חוסם). את מה שבחרתם אתם צריכים לתעד בקובץ README בסעיף 9.

- אם השחקן נע לתא שיש בו שדון, המהלך חוקי (הוא מתבצע), אולם הוא "ייאכל" מיד.
- כצפוי, גם אם שדון נע לתא שבו נמצא השחקן, השחקן "נאכל".
- אם שדון נע לתא שיש בו שדון, המהלך חוקי. כלומר, שני שדונים יכולים להיות בתא אחד.
- אם שדון נע לתא שיש בו עוגיה או מתנה, המהלך חוקי. נראה רק את השדון, אבל עדיין העוגיה או המתנה נמצאים שם.
- אם השחקן נע לתא שיש בו עוגייה, הוא "אוכל" את העוגייה, והניקוד שלו עולה. ראו להלן פירוט לגבי הניקוד.
- אם השחקן נע לתא שיש בו מתנה, הוא "אוכל" את המתנה, והופך לסופר פקמן והניקוד שלו עולה. ראו להלן פירוט לגבי הניקוד.
- אם השחקן נע לתא שיש בו מפתח, הוא "אוכל" את המפתח, הניקוד שלו עולה, ודלת אחת נפתחת, כלומר נעלמת. אתם צריכים לבחור בין אחת מהאפשרויות: האם זהו מפתח קבוע לדלת הזו, או שזו דלת רנדומלית (אחת מבין הדלתות שנשארו). ההבדל יבוא לידי ביטוי כשנריץ את המשחק פעמיים. מה יקרה: האם אותו מפתח יפתח את אותה דלת, או לאו דווקא. את מה שבחרתם אתם צריכים לתעד בקובץ README, בסעיף 6 (בסעיף האלגוריתמים: כלומר באיזה אלגוריתם בחרתם כדי "לשדך" מפתח לדלת).
- הבדל נוסף בין השחקן לשדונים, השחקן יכול לבחור לוותר על תורו (ראו להלן), אולם שדון לא יכול "לבחור" לוותר על תורו, אלא רק להפסיד את התור במקרה שניסה לזוז לתא חסום, כנ"ל.

מקשי השחקן

השחקן מזיז את הדמות שלו בעזרת מקשי החיצים (ראו הסבר בהמשך), או לחיצה על מקש הרווח כדי לוותר על התור.

תזוזת השדונים

נשאיר לכם להחליט על השיטה להזזת השדונים. מספר נקודות מציון התרגיל מוקדשות לנושא זה וככל שהשיטה מוצלחת יותר, הניקוד לסעיף זה יעלה. תזוזה אקראית היא האפשרות הפשוטה כאן,

אולם היא גם "מתומחרת" בהתאם. בכל מקרה, עליכם לציין בקובץ ה-README, בסעיף 6 (בסעיף האלגוריתמים), מה השיטה שהחלטתם לממש ולהסביר את הבחירה בה, איך היא עובדת וכדומה.

סיום השלב

אם השחקן אכל את כל העוגיות בשלב הנוכחי, הוא סיים בהצלחה את השלב, והתכנית צריכה לטעון את השלב הבא (אם יש). שימו לב שכעת שוב צריך להציב את האובייקטים ואת הדמויות מחדש לפי השלב המופיע בקובץ (כפי שיוסבר בהמשך), וכן גודל השלב יכול להיות שונה בין שלב לשלב.

ניקוד

כל עוגייה שהשחקן אוכל מזכה אותו ב-2 נקודות.

כל מתנה שהשחקן לוקח מזכה אותו ב-5 נקודות.

כל מפתח שהשחקן אוסף מזכה אותו ב-7 נקודות.

כל סיום שלב מזכה ב- 50 נקודות + 2 נקודות בונוס עבור כל שדון שהיה בשלב. כלומר סיום שלב עם 3 שדונים יזכה את השחקן ב-56 נקודות (ההנחה היא שככל שיש יותר שדונים יותר קשה לסיים את השלב, לכן הניקוד בהתאם).

הצגת נתוני המשחק בעת המשחק

בעת המשחק, יש להציג את נתוני המשחק לכל אורך המשחק. כלומר יש להציג את מצב הניקוד, **מצב החיים (=מספר החיים שנותרו)**, ומספר השלב. כמובן, כל שינוי יעדכן מיד את התצוגה. התצוגה תהיה בתחתית המסך.

דרישות לפרטי מימוש

קובץ main.cpp קטן

עליכם להגיש קובץ main.cpp קטן. כמה קטן? שפונקציית ה-main שלו תכיל לכל היותר 2 שורות.
איך?
למשל: עליכם ליצור אובייקט שינהל את כל התוכנית שלכם בשורה אחת, ולקרוא לפונקציה שתריץ את המשחק בשורה השניה. דוגמא לקוד כזה ניתן למצוא במודל בדוגמא של ForwardDeclaration, בקובץ main.cpp שבתוכו.

לוח המשחק וטעינת השלבים

עליכם לממש את שלבי המשחק בעזרת טעינת קובץ טקסט בשם Board.txt, כאשר התכנית תקרא את כל שלבי המשחק מקובץ זה. כפי שנאמר לעיל, אתם נדרשים ליצור לפחות שלושה שלבים. כל השלבים נמצאים באותו הקובץ, כפי שיוסבר בהמשך. התוכנית שלכם יכולה להניח שהקובץ Board.txt הינו תקין וכתוב בפורמט שיוסבר בהמשך, והתוכנית שלכם לא צריכה להתמודד עם קבצי Board.txt לא תקינים.

בקובץ הזה "תציירו" את השלבים באופן הבא:

בשורה הראשונה **לפני** ציור **כל** שלב יהיו 2 מספרים שייצגו את גודל הלוח/השלב. המספר הראשון יהיה אורך השלב (כלומר גובה) והמספר השני יהיה רוחב השלב. כלומר אם המספר הראשון הינו M והשני הינו N אזי השלב יהיה בגודל של M שורות על N עמודות (כלומר, M שורות בנות N תווים כל אחת).

הערה למימוש: אם יותר פשוט וקל לכם, אתם יכולים להתעלם מהמספר השני, ולגלות את רוחב השורה פשוט על ידי קריאה עד סוף השורה, רק עליכם לוודא שאתם מגישים קובץ תקין שבו כל השורות ברוחב זהה למספר N.

הדמויות והעצמים יסומנו בקובץ על פי התווים שהוזכרו לעיל. תו 'רווח' משמעותו משבצת ריקה.

תהיה שורה ריקה בקובץ כדי להפריד בין שלב לשלב.

מיקום הדמויות

בעת כתיבת הקובץ Board.txt עליכם לוודא בכל שלב את הדברים הבאים:

- הפקמן מופיע רק פעם אחת ורק בצורתו הרגילה.
- **מופיעה לפחות עוגייה אחת.**
- **מספר הדלתות זהה למספר המפתחות.**

כלומר, בקוד שלכם, אינכם צריכים לתמוך בקובץ Board.txt שלא מקיים את התנאים הללו.

אם השחקן "נאכל" ויש לו עוד "חיים", כל הדמויות הנעות (כלומר הפקמן והשדונים) תמוקמנה מחדש כפי המתואר בקובץ, כך נמנע מצב שבו השחקן מיד "ייאכל" שוב על ידי אותו שדון. לעומת זאת, במקרה הזה, מצב העוגיות, המתנות, המפתחות והדלתות יישאר כפי שהיה בשעת הפסילה, ולא יחזור להיות כפי המתואר בקובץ.

הערות (שיעזרו לכם) למימוש

כדי לאפשר הצגה ברורה של העדכונים במסך ה־Console, נצטרך לנקות את המסך בין הדפסה להדפסה. זאת נשיג על ידי הפקודה `system("cls");` (ונוסיף בתחילת הקובץ: `#include <cstdlib>`). כדאי לציין שפקודת ה־system משתמשת בפקודות המערכת, ולכן הפקודה המסוימת הזו תעבוד רק בסביבת Windows ולא במערכות הפעלה אחרות.
אפשרות אחרת (שגם היא ספציפית ל־Windows) היא להיעזר בקובץ `io.h` שאתם מקבלים מאיתנו וקיימות בו הפונקציות `Screen::resetLocation()` ו־`Screen::setLocation()`. הפונקציות האלה מאפשרות להזיז את הסמן לתחילת מסך ה־Console או למיקום הרצוי במסך (בהתאמה). לכן כל מה שנדפיס, לאחר הקריאה לאותן פונקציות, **ישכתב** את מה שכבר מופיע באותו מיקום. אם אתם בוחרים בדרך הזו, שימו לב שכדי למחוק תווים יש להדפיס רווחים. כמו כן, אם אתם

צריכים "לשכתב" שורה, חשוב לוודא שאתם מדפיסים את כל רוחב השורה (כולל רווחים) כדי למחוק את הטקסט שהיה שם קודם.

שימוש במקשי החיצים

עד היום הכרנו איך לקבל מהמשתמש קלט שמכיל תווי ASCII, כאלה שיש להם ייצוג מודפס על המסך. כמו כן, שיטות הקלט הרגילות (בין אם זו `cin, scanf` או אפילו `getc`) דורשות מאתנו להמתין עד שהמשתמש ילחץ על מקש ה־Enter לפני שהן מתחילות לקבל את הקלט. המגבלה הראשונה מונעת מאתנו להשתמש במקשי החיצים בכלל. המגבלה השנייה מונעת מאתנו לקבל את הקלט מהמשתמש (לא משנה באלו מקשים נבחר לשם כך) בלי לחייב אותו ללחוץ Enter בכל צעד, דבר שאיננו סביר במשחק כמו זה.
כדי להשתמש במקשי החיצים, וכדי לקבל את המקש שנלחץ גם בלי המתנה ל־Enter, השתמשו בפונקציה `_getch()` שנמצאת ב־`conio.h`. הפונקציה מחזירה int ובאמצעותו תוכלו להחליט מה לעשות עכשיו, לדוגמה לאן להתקדם וכו'. (כמו שהזכרנו קודם: אין להשתמש במספרים מפורשים, literals, ישירות בקוד hard-coded, אלא עליכם לעשות זאת בעזרת הגדרת קבועים). גם כאן, כדי לפשט, הוספנו כמה קבועים שימושיים, בקובץ `io.h` שאתם מקבלים מאיתנו.
מותר לכם לשנות את הקבצים שאתם מקבלים מאיתנו.

שיטת עבודה מומלצת

ההמלצה היא לעבוד בשלבים קטנים באופן כזה שבכל פעם מה שכתבנו מתקמפל ורץ, כך שאפשר מיד לראות האם זה אכן עובד.

למשל:
שלב ראשון: לקרוא את השלב מהקובץ ולהדפיס אותו.
שלב שני: ליצור את האובייקט של אחת הדמויות לפי המיקום שקראנו מהשלב.
שלב שלישי: להזיז את הדמות לפי לחיצות המקשים (זה בסדר אם הדמות כרגע תזוז בלי שום הגבלות, רק שנראה שהצלחנו להזיז אותה ולהדפיס אותה בצורה נכונה).
וכן הלאה.

כדאי לתכנן רגע לפני כן מה המחלקות הנדרשות, כמו שדיברנו בתרגול איך להסיק מהדרישות מה המחלקות שנצטרך. מצד שני, לפעמים לנסות לתכנן הכול מראש זה רק מסבך יותר, כי יש הרבה דברים לא מוכרים או לא וודאיים מה שמשאיר אותנו עם הרבה סימני שאלה. לכן, לפעמים כדאי להתחיל לכתוב את השלבים הראשונים אחרי התכנון הבסיסי, ואז דברים נהיים יותר ברורים ואפשר להמשיך הלאה.

הערה כללית

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 :)).

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־Visual Studio 2022". אנא השתמשו בדרך זו (אחרי שהגדרתם כראוי את שמות הצוות ב־MY_AUTHORS: **נשים את שמות המגישים בתוך המרכאות, נקפיד 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 לגרור הורדת נקודות בציון.

מספר הערות:

1. נשים לב לשם הקובץ שאכן יכלול את שמות המגישים.
2. נשים לב לשלוח את תיקיית הפרוייקט 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!