מכללת הדסה, החוג למדעי המחשב

תכנות מונחה עצמים ופיתוח משחקים

סמסטר ב', תשפ"ג

תרגיל 4

תאריך אחרון להגשה:

הנביאים - יום א', 21/05/23, א' סיוון ה'תשפ"ג, בשעה 23:59

שטראוס-גברים - יום א', 21/05/23, א' סיוון ה'תשפ"ג, בשעה 23:59

שטראוס-נשים - יום א', 21/05/23, א' סיוון ה'תשפ"ג, בשעה 23:59

מטרת התרגיל:

שימוש במבני נתונים, איטרטורים ואלגוריתמים בסגנון STL וריענון השימוש ב־SFML.

תיאור כללי:

בתרגיל זה נבנה את המשחק "שישה צבעים" (בגרסת המשושים) תוך שימוש בספריה הגרפית. המשחק כולל שני שחקנים – אנושי וממוחשב.

דוגמה לריצת המשחק:

כדי להבין את כללי המשחק, אפשר פשוט לשחק בו. 6colors.zip (קובץ ZIP שבתוכו המשחק עבור Windows; במקור הופיע באתר http://pyva.net/eng/pc/colors.html, שכבר איננו זמין). גרסה של המשחק עבור אנדרואיד נמצאת בקישור הבא: https://play.google.com/store/apps/details?id=com.wetpalm.colorflood

כללי המשחק:

מטרת המשחק היא לצבור שטח גדול ככל האפשר.

בתחילת המשחק, המשושים שעל הלוח נצבעים באופן רנדומלי, ואז השחקנים מקבלים אוטומטית את הצורה שבאחת הפינות: השחקן האנושי מתחיל בפינה השמאלית התחתונה (כלומר הוא שולט על הצורה שבפינה זו), והמחשב מתחיל מהפינה הימנית העליונה.

בכל תור, השחקן בוחר צבע כלשהו הגובל בשטח שכבר שייך אליו, ובכך מגדיל את השטח שלו אל האריחים השכנים בעלי הצבע הזה. בכל שלב, מותר לבחור אחד מהצבעים הפנויים, כלומר כל הצבעים למעט שני הצבעים שנבחרו בשני הצעדים האחרונים על ידי השחקנים (וכרגע הם "תפוסים" על ידם). המנצח הוא השחקן הראשון ששולט על יותר מ־50% משטח הלוח.

דרישות לממשק המשחק:

עליכם לממש את הדברים הבאים:

1. לוח המשחק. חשוב להקפיד שהקוד יהיה גמיש ויהיה אפשר לשנות את הגודל בקלות (למשל, בעזרת קבוע יחיד או מספר קבועים בודדים בקוד).
2. צבעים לבחירה (באמצעות העכבר).
3. חיווי המראה של מי התור כעת (השחקן האנושי או הממוחשב).
4. הצגת אחוז השטח שכוסה על ידי כל שחקן.
5. כפתור "New" שמתחיל משחק חדש (גם אם כרגע הלוח באמצע משחק).
6. כפתור יציאה (כלומר, שכפתור ה־X של החלון יאפשר סגירה כראוי).
7. שלושה סוגים של שחקן ממוחשב:
   a. שחקן רנדומלי, שבוחר רנדומלית את אחד הצבעים האפשריים
   b. שחקן שמשתמש באלגוריתם חמדן "מקומי", כלומר בוחר את הצבע שנותן את הכי הרבה אריחים מהשכנים המידיים של השטח הקיים (בלי לבדוק שכנים של שכנים וכו', כלומר ייתכן שהצעד בכללותו עדיין לא נותן את התוצאה המיטבית)
   c. שחקן שמשתמש באלגוריתם חמדן "גלובלי" כך שבכל צעד הוא בוחר את הצבע שיוסיף לו הכי הרבה אריחים (בחישוב השלב המלא, כלומר כולל שכנים של שכנים וכו')
8. בתחילת משחק חדש (בין אם בתחילת התוכנית או בלחיצה על כפתור New), יוצג תפריט שיאפשר לבחור את סוג השחקן הממוחשב מבין שלוש האפשרויות הנ"ל.
9. לוח המשחק לא יכלול את כל האפשרויות שבדוגמאות לעיל, אלא יהיה מורכב רק ממשושים.
10. אתם נדרשים לממש משחק שנעים לשחק בו, שהכל הולך חלק, ושדברים לא "נתקעים".
11. אין חובה לממש לוח שיאים וכדומה.
12. לוח המשחק צריך להיווצר אוטומטית. כלומר בתחילת כל משחק ייווצר לוח חדש על ידי הגרלת צבעי האריחים.

הערות למבנה הנתונים:

אפשרות די טבעית למימוש המשחק היא בעזרת גרף ואלגוריתם מתאים לטיול בגרף, כמו BFS או DFS (כדי למצוא את מספר התאים הסמוכים מהצבע המבוקש וכדומה), אבל אולי ייתכנו אפשרויות טובות אחרות. נדגיש כי

בכל מקרה הציפייה היא שתבנו מבני נתונים מתאימים, כלומר מחלקה (או יותר מאחת, כתלות בצרכים) שתשמש בעצמה כמבנה נתונים גנרי (ועדיין לטובת המימוש שלה ניתן להשתמש במגוון הכלים של STL).

כאמור, מבנה נתונים כזה צריך להיות גנרי, לא רק מוכוון למטרה הספציפית שלנו כרגע, אבל צריך גם להיזהר לא לנסות לקחת את הדברים לכיוון גנרי מידי, לפי העיקרון שקיים בחלק מהשיטות של הנדסת תוכנה (למשל, XP, Extreme Programming) המכונה: YAGNI, ראשי תיבות של You Ain't Gonna Need It…

הערות לתיכון התוכנית:

שימו לב שיש בתרגיל למעשה שלושה חלקים:

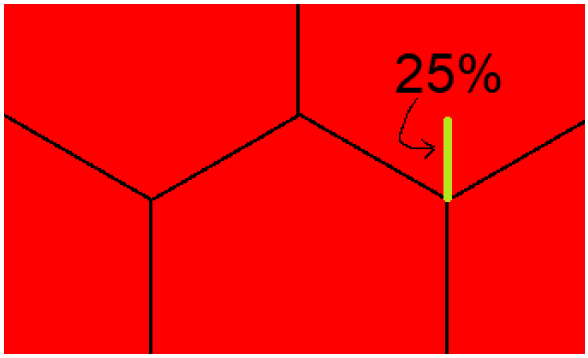
1. מבנה הנתונים שאנחנו נעזרים בו, האיטרטורים והאלגוריתמים.
2. המימוש של הלוגיקה של המשחק.
3. המימוש הגרפי.

כדאי **ומומלץ מאוד** להפריד בין מימוש מבני הנתונים, מימוש הלוגיקה והמימוש הגרפי, גם אם בסופו של דבר ברור שהכול מתחבר יחד לכדי תוכנה אחת שמפעילה את כל החלקים.

כרגיל, את ההחלטות שתחליטו תתעדו כראוי בקובץ ה־Readme. בפרט יש צורך לתעד את ההחלטות על בחירת מבני הנתונים ואופן השימוש בהם והשימוש בספרייה הסטנדרטית. תכתבו במה בחרתם להשתמש ולמה העדפתם מבנה נתונים או אלגוריתם מסוים על פני האחרים. תתעדו גם איך התרגיל שלכם מתחלק לשלושת החלקים שהוזכרו לעיל (למשל פירוט איזו מחלקה משוייכת לאיזה חלק).

הערות למימוש המשושים:

1. ל־sf::CircleShape ניתן להעביר פרמטר נוסף בבנאי שקובע את מספר הקודקודים של הצורה, כדי ליצור צורות משוכללות נוספות, לאו דווקא עיגולים.
2. ניתן להיעזר ב־setOutlineColor ו־setOutlineThickness כדי ליצור מסגרת בצבע שונה מסביב לאריחים ולשפר את הנראות שלהם.
3. כדי לחשב את המרחק ממרכז המשושה לאמצע אחת הצלעות, ניתן להכפיל את הרדיוס ב־(std::sqrt(3.f) / 2) (כלומר, $\frac{\sqrt{3}}{2}$).
4. כדי להתאים את המשושים בשורה הבאה אל בין המשושים שבשורה שמעליהם, המיקום הנכון מבחינת ציר ה־Y הוא להזיז אותם כלפי מטה רק ¾ מהגובה הכולל של המשושה (כלומר, הקטע המסומן בציור, הוא 25% מגובה המשושה)



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. 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 שיצרנו, 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 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 exN-firstname_lastname.zip (או במקרה של הגשה בזוג – 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 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. 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!