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

פרויקט סוף סמסטר

תאריכי הגשה: ראו לוח זמנים בהמשך.

מטרת התרגיל:

בתרגיל זה נממש שוב את המשחק Bomberman שמימשנו בתרגיל 2. ההבדלים המרכזיים הם שהפעם נשתמש בירושה ופולימורפיזם וכן נשתמש בספרייה הגרפית SFML.

מאז תרגיל 2 למדנו נושאים חדשים והתקדמנו באופן כללי בתכנות מונחה עצמים, כעת נוכל לעצב תיכון טוב יותר, יפה יותר ויעיל יותר של האפליקציה בהתאמה לעקרונות תכנות מונחה עצמים.

בנוסף, נוסיף תכונות נוספות לעומת מה שעשינו בתרגיל 2.

שינויים והוספות:

רוב פרטי המשחק נשארים בדיוק כפי שהיו בתרגיל 2, למעט מה שכבר הוזכר לעיל, שימוש בירושה, פולימורפיזם וממשק גרפי, וכן השינויים וההוספות המפורטים להלן.

משחק מונחה שעון:

המשחק כעת יהיה "בזמן אמת", מונחה שעון, ולא מונחה תורים. ראו פירוט בהמשך.

התנגשות בקירות או בין דמויות:

הגדרת התנגשות היא כאשר יש נקודה חופפת בין הריבועים החוסמים הרלוונטיים, כלומר הריבוע החוסם את המכשול והריבוע החוסם את הדמות או הריבועים החוסמים של שתי הדמויות המתנגשות (שחקן ושומר).

הגבלות ו"בונוסים" לשחקן:

בכל שלב יכולה להיות מוגדרת הגבלת זמן והגבלת מספר הפצצות הזמין לשימוש. אין חובה שבכל שלב תהיינה שתי ההגבלות האלה, אפשר ליצור שלבים שבכלל לא יהיו מוגבלים כך (למשל, אם רוצים שלבים ראשונים קלים ל"תרגול"), אבל יש חובה לתמוך באפשרויות האלה.

אם עבר הזמן המוגדר, זה נחשב לפסילה של השחקן (יורדים "חיים") והשלב מתחיל מחדש (באופן דומה להגבלת הצעדים שהייתה בתרגיל 2, כלומר כל השלב מתחיל מחדש, כולל הסלעים שהתפוצצו ושומרים שנהרגו).

כצפוי, אם השחקן כבר השתמש בכל הפצצות הזמינות, הוא לא יכול להניח פצצה נוספת. מה שגורר לפעמים הפסד בשלב. למשל, אם יש סלעים שהוא חייב לפוצץ כדי להגיע לדלת וכעת הוא נשאר

חסום, הרי שהוא נשאר תקוע עד שייתפס על ידי שומר או שיסתיים הזמן המוגדר. מכיוון שלפעמים ייתכן שאין הגבלת זמן ושאין שומרים (או שכולם כבר נהרגו), מומלץ לאפשר דרך לנטוש את השלב הנוכחי (בעזרת מקש מקלדת, כפתור בממשק או אפשרות בתפריט, מן הסתם) כדי שהשחקן לא יישאר תקוע כך. גם נטישה נחשבת כפסילה והשלב יתחיל מחדש אם עוד נשארו לשחקן "חיים".

תוספת נוספת היא "מתנות" המפוזרות בשלב לתוספת פצצות, כלומר כשהשחקן אוסף אותן (על ידי "דריכה" עליהן) נוספות לו פצצות למספר הפצצות הזמין (בהנחה שיש בשלב הזה מגבלה על מספר הפצצות). שומרים יכולים לעבור במקום שיש בו מתנה, אבל הם לא יכולים לקחת אותה. מתנה יכולה להיות גלויה או מוסתרת מתחת לסלע. לבחירתכם האם מתנה (גלויה) מושפעת מפיצוץ או לא.

שלבי המשחק:

עליכם לספק לפחות ארבעה שלבים (קל יותר עכשיו שיש לנו עורך גרפי :)).

תזכורת ועדכונים לפורמט הקובץ:

בשורה הראשונה מופיעים ארבעה מספרים, מספר השורות בשלב ומספר העמודות, כפי שהיה בתרגיל 4, וכעת גם הגבלת הזמן לשלב (בשניות) והגבלת מספר הפצצות. בשתי ההגבלות ניתן להשתמש במספר 1- כדי לסמן שההגבלה לא פעילה.

אחריה מופיעות שורות כמספר השורות שהוגדר עם הסימונים # (קיר), @ (סלע), / (רובוט-שחקן), ! (שומר), D (דלת) ורווח למשבצות הריקות. בנוסף, יכולים להופיע הסימונים + ("מתנה" גלויה) או & (סלע שמתחתיו מסתתרת "מתנה", כשמפוצצים את הסלע מופיעה המתנה במקומו).

נשאיר לבחירתכם האם הפעם כל השלבים יופיעו בקובץ אחד (כמו בתרגיל 2) או שכל שלב מופיע בקובץ נפרד. בכל מקרה, נדרש שתהיה דרך להוסיף שלב חדש בלי צורך להוסיף את שם הקובץ החדש לקוד (למשל, שימוש בתבנית מסוימת לשמות הקבצים, כך שהתוכנית יודעת אוטומטית מה שם הקובץ הבא שהיא צריכה לנסות, או שימוש בקובץ שממנו קוראים את רשימת שמות קובצי השלבים).

באופן כללי, ניתן להניח שהקובץ תקין.

פצצות:

כעת שאין תורים, הזמן לפיצוץ פצצה יוגדר כ־4 שניות. מומלץ להציג את הספירה לאחור אך אין חובה כזו.

ניקוד:

הניקוד נותר כפי שהיה, על כל שומר – 5 נקודות כפול מספר השומרים בתחילת השלב, על סיום שלב – 20 נקודות כפול מספר השומרים.

שומרים אקראיים ושומרים חכמים:

עליכם לאפשר שני סוגי שומרים. שומר אקראי – כשמו כן הוא, זז בצורה אקראית (אם כי עדיף שלא להגריל כיוון מחדש בכל צעד אלא רק פעם במספר צעדים/שניות, כפי שדיברנו בעבר, או משהו דומה כדי למנוע תנועה "תזזיתית"). שומר חכם – זז לפי אלגוריתם תנועה חכם יותר כדי לרדוף אחרי השחקן. הבחירה אם שומר הוא חכם או אקראי נעשית בצורה אקראית ביצירת השומר בתחילת השלב הנוכחי (כזכור, את המידע על מיקום השומרים ומספרם אנחנו מקבלים בקריאה מהקובץ).

שימוש בתפריט:

עליכם לאפשר תפריט בסיסי של בחירת משחק חדש או יציאה. התפריט יוצג בתחילת ריצת התכנית או אחרי שהמשחק נגמר (כי נגמרו השלבים או כי לרובוט נגמרו הפסילות). לבחירתכם האם התפריט יופעל בעזרת העכבר או המקלדת (או שניהם). אם הוא מופעל בעזרת המקלדת ראוי לתמוך בבחירת אפשרות מהתפריט על ידי לחיצה על אות מתוך הכיתוב המופיע על הכפתור. במקרה כזה, נדגיש את האות על ידי קו תחתי (כמו בדוגמה כאן: File Edit View Window Help).

תצוגת מידע:

אתם נדרשים להציג עבור השחקן את מספר הנקודות שצבר, את מספר הפסילות שנשארו לו ואת מספר הפצצות שהוא הניח בשלב הנוכחי, וכן להציג את מספר השלב הנוכחי.

תזוזה:

למרות שבשימוש במסך גרפי אנחנו יכולים להזיז את הדמויות גם באלכסון, לכאורה, בכל זאת נשאיר את התזוזות רק בקווים ישרים לאורך הצירים, ולכן גם המקשים של השחקן נשארים כפי שהם (חיצים לתזוזה לאחד מארבעת הצדדים בהתאמה, b להנחת פצצה; ראו בפסקאות הבאות לגבי מקש הרווח).

הלחיצה על המקשים נשארת כשהייתה (רק שהפעם מזהים זאת בעזרת אירועים של SFML במקום בעזרת getch). ההבדל הוא שעכשיו אנחנו לא מתנהלים לפי תור אלא לפי השעון. ממילא, צריך לשים לב שקצב התזוזה לא יסתמך על קצב קבלת האירועים של key pressed, שאיננו אחיד, אלא על משך הלחיצה ועל השעון. קיימות מספר דרכים לעשות זאת, נציע כאן מספר אפשרויות לדוגמה:

1. הלחיצה רק קובעת האם השחקן יזוז ולאיזה כיוון הוא יזוז כאשר נרצה להזיז אותו. התזוזה בפועל מתבצעת לפי הזמן שחולף. מכיוון שכך, השחקן ממשיך לזוז לאותו הכיוון כל עוד הוא לא לחץ על מקש אחר ולא נתקע במכשול. לחיצה על מקש הרווח תשמש כעת לצורך עצירה במקום. וריאציה קרובה תהיה להשתמש במקש הנגדי כדי לעצור (ותידרש לחיצה נוספת כדי להתחיל לזוז לכיוון הנגדי).
2. אפשרות אחרת היא שהלחיצה נותנת תאוצה מסוימת כך שלחיצה קצרה (ואז המתנה) מתבטאת בתזוזה קצרה של השחקן ולחיצה ארוכה יותר גורמת לתזוזה מהירה יותר

ולמרחק גדול יותר. במקרה כזה, בדרך כלל נשתמש במקש הנגדי כדי לעצור ואין טעם לתמוך במקש הרווח.

3. אפשרות שלישית תהיה כן להשתמש בלחיצה ובעזיבה של המקש, אבל עדיין נסתמך על השעון כדי להזיז את השחקן בקצב מתאים, ונשתמש באירוע key released כדי לעצור את התזוזה.

מהירויות תזוזה:

עליכם להחליט כיצד לעדכן את מיקום הדמויות. מכיוון שהגדרנו שהמשחק מונחה שעון, הרי שתצטרכו לבצע זאת כל פרק זמן מסוים ותצטרכו לקבוע בכמה הדמות זזה בכל פרק זמן כזה, לפי הכיוון שאליה היא זזה כרגע. האם עדכון כל עשירית שנייה יהיה מוצלח? אולי פחות? אולי יותר? האם תזוזה של פיקסל או שניים סבירה? האם, כדי לתת די אתגר, השומרים צריכים להיות מהירים יותר (כלומר, יותר פיקסלים בכל איטרציה) מהשחקן? עד כמה מהירים? לא נקבע כללים בנושא, אלא נשאיר זאת עבורכם לנסות ערכים שונים ולהחליט איך ליצור משחקיות טובה. בכל מקרה, תעדו את החלטותיכם בקובץ ה־Readme.

עיצוב האובייקטים:

לגרפיקה מוצלחת כנראה שתרצו לצייר את האובייקטים בתוכנה חיצונית או למצוא תמונות מתאימות ולטעון את התמונות בעזרת sf::Texture. במצב כזה, כדאי שהתמונה תהיה עם רקע שקוף (ניתן לעשות זאת בתמונות ששמורות כ־png, למשל), כדי למנוע רקע לבן שיתווסף לדמות. זה חשוב בעיקר אם מדובר בדמות שאין לכם צורה פשוטה שתכיל אותה ואתם צריכים להכיל אותה בתוך sf::RectangleShape או sf::Sprite שחוסם אותה. כמו כן, כדאי לדאוג שהדמות תתפוס ככל האפשר כמה שיותר מהמלבן החוסם, כדי שיהיה ברור יותר בשעת המשחק מתי נחשב שדמויות התנגשו זו בזו או במכשול.

תוספות שהן בונוס (קל):

- תפריט מרשים.
- גרפיקה מרשימה.
- סאונד.
- אלגוריתמים מתוחכמים.
- שימוש בתבניות עיצוב ובכלי תכנות מתקדמים.
- רעיונות נוספים לשיפור המשחקיות, למשל "מתנות" מסוגים נוספים עם השפעות שונות על ההתנהגות.

הערות:

- בניגוד לתרגיל 4, הפעם אין איסור להחזיק במחלקות משתנים מסוג הצורות של SFML. למעשה, אולי אפילו עדיף להחזיק את הצורות מוכנות כבר, וכמו כן לחשוב אלו מאפיינים אין צורך להחזיק בנפרד כי הם כבר נמצאים במאפייני הצורה.
- למעקב אחרי הזמן החולף – המחלקה הרלוונטית היא sf::Clock. אפשרות אחרת היא להשתמש בספריית std::chrono שמשולבת כחלק מהספרייה הסטנדרטית מאז C++11. שימו לב שאתם צריכים לבדוק בעצמכם כמה זמן עבר (בעזרת הפונקציה getElapsedTime()), במקרה של sf::Clock, או באמצעים מתאימים במקרה של שימוש בספרייה הסטנדרטית). מן הסתם המקום המתאים הוא בלולאת האירועים או בסמוך לה. (ואגב לולאת האירועים, כמובן שכדי להתנהל לפי השעון חייבים להשתמש הפעם ב־pollEvent() ולא ב־waitEvent()).
- הקפידו על תיכון נכון, מונחה עצמים, תוך שימוש ראוי בירושה ובפולימורפיזם.
- כרגיל, אתם מתבקשים לתעד בקובץ ה־Readme תוספות שעשיתם או החלטות לגבי דברים שלא מוגדרים בפירוש.
- משום מה, יש לסטודנטים רבים נטייה לרוץ לעבר הדברים שמוזכרים כאפשרות לבונוס כשהם שוכחים שחשוב יותר לממש היטב את הדרישות המחייבות. אל תיפלו בפח הזה! למען הסר ספק, אין מקום לבונוס על תוספות אם הדרישות הבסיסיות לא מתמלאות.
- כפי שהוזכר כבר בתרגיל 2, ונכון עוד יותר עבור הפרויקט, 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 :)).

הגשת תיכון:

הגשת התיכון נעשית בהגשת דיאגרמת UML (מסוג class diagram) המציגה את המחלקות המתוכננות עם מאפיינים (שדות) עיקריים ופונקציות עיקריות שלהן ועם הקשרים בין המחלקות (לפחות ירושה, אבל עדיף גם הכלה). בנוסף, יוגש הסבר מילולי קצר שיסייע בהבנת הדיאגרמה וחלוקת האחריות בין המחלקות. ההסבר המילולי יכלול הסבר קצר על כל מחלקה ואם יש צורך – גם על הקשר בין מחלקות שונות.

חשוב להבין: בלוח הזמנים מוקדש זמן יחסית קצר עבור הגשת התיכון, מכיוון שמטרתו העיקרית היא לגרום לכם לתכנן לפני שתיגשו לממש את הפרויקט. ההערכה היא שלא תידרשנה לשם כך יותר ממספר שעות במקרה הגרוע ביותר. אתם לא צריכים לתכנן כאן את הקוד שלכם לפרטי פרטים, אלא רק לשבת ולחשוב איך נראה לכם כרגע שכדאי לממש את הדברים מבחינת חלוקת פונקציונליות בין המחלקות, מבחינת עצי ירושה וכדומה.

התיכון צריך להיות סביר והגיוני, אבל לא יקרה שום אסון אם במהלך כתיבת הפרויקט בפועל תגלו שאתם רוצים לשנות את חלוקת המחלקות שלכם וכדומה. זה לא אומר שזה בסדר אם התיכון יהיה מרושל מלכתחילה; זה כן אומר שלא צריך להילחץ אתו יותר מידי.

כדי לשרטט את הדיאגרמה ניתן לבחור אחת מהשיטות הבאות:

- שרטוט בכתב יד וסריקה של הדף (וודאו שהסריקה יוצאת ברורה דיה!).
- שימוש בתוכנה כמו Power Point או Visio כדי לשרטט זאת.
- שימוש ב־Visual Studio. ניתן ליצור דיאגרמת UML מתוך הגדרת מחלקות (מספיק להצהיר על המחלקות, כלומר לכתוב קובצי h בסיסיים, ובלחיצה ימנית על הפרויקט לבחור View ואז View Class Diagram). למידע נוסף ראו בקישור הבא: How to: Add class diagrams to projects

ההסבר המילולי יהיה בקובץ Word. זכרו, עדיף קובץ בעברית ברורה על פני קובץ באנגלית לא ברורה!

הגשת התיכון תתבצע כרגיל, כשהקבצים שאתם מגישים מכווצים לקובץ zip, ושם הקובץ, כרגיל, כולל את שמות המגישים אלא שהפעם במקום שם התרגיל תופיע המילה design, כלומר תבנית שם הקובץ היא Design_name1_name2.zip.

הצגת השלד:

כדי לגרום לכם להתקדם עם הפרויקט ולנצל את הזמן ולא לדחות הכול לסוף, כמו שעלול לקרות לרבים מאתנו, אתם נדרשים להציג התקדמות אחרי כשבועיים מהגשת התיכון. הצגת השלד תיעשה בזמן הסדנה. כל צוות ישב עם המתרגל/ת במשך 5-10 דקות ויציג את ההתקדמות הנוכחית והניקוד לשלב הזה ייקבע בהתאם.

אנחנו מודעים לכך שזה לא הרבה זמן אחרי תחילת הכתיבה, אבל כן נצפה לראות שלד. העדיפות היא אם ניתן להציג חלק מסוים שעובד כבר, אבל בכל מקרה נצפה לראות התקדמות מסוג כלשהו ולא רק את ההצהרות על המחלקות שבתכנון.

הצגת הפרויקט:

בזמן שייקבע, כל צוות בתורו יציג את הפרויקט בפני סגל הקורס (המרצה והמתרגל/ת הרלוונטיים). בשלב של הצגת הפרויקט, הוא אמור להיות גמור או לפחות כמעט גמור.

הגשת הפרויקט:

את הפרויקט מגישים באופן דומה לשאר התרגילים (רק עם השם Project במקום exN, כמפורט למטה). כמו תמיד, הנחיות הגשה מפורטות נמצאות בסוף הקובץ הזה.

לוח זמנים:

(בסוגריים מצוין פרק הזמן מאז נקודת הציון הקודמת.)

קמפוס הנביאים:

- תחילת הפרויקט – יום רביעי, 26/12/18
- הגשת תיכון (תוך שני "ימי עבודה") – יום חמישי, 27/12/18 (עד שעה 23:59)
- הצגת שלד (אחרי שבועיים) – יום חמישי, 10/1/19 בזמן (התרגול ו)הסדנה
- הצגת הפרויקט (אחרי שבוע וחצי) – יום שני, 21/1/19 (ייתכנו שינויים)
- הגשת הפרויקט (באותו היום) – יום שני, 21/1/19 (עד שעה 23:59)

קמפוס שטראוס-גברים:

- תחילת הפרויקט – יום רביעי, 26/12/18
- הגשת תיכון (תוך שני "ימי עבודה") – יום חמישי, 27/12/18 (עד שעה 23:59)
- הצגת שלד (אחרי שבוע וקצת) – יום ראשון, 6/1/19 בזמן הסדנה
- הצגת הפרויקט (אחרי שבועיים) – יום ראשון, 20/1/19 (ייתכנו שינויים)
- הגשת הפרויקט (יום למחרת) – יום שני, 21/1/19 (עד שעה 23:59)

קמפוס שטראוס-נשים:

- תחילת הפרויקט – יום חמישי, 20/12/18
- הגשת תיכון (תוך שלושה "ימי עבודה") – יום שני, 24/12/18 (עד שעה 23:59)
- הצגת שלד (אחרי כשבועיים) – יום חמישי, 3/1/19 בזמן הסדנה
- הצגת הפרויקט (אחרי שבועיים) – יום ראשון, 20/1/19 (ייתכנו שינויים)
- הגשת הפרויקט (יום למחרת) – יום שני, 21/1/19 (עד שעה 23:59)

ניקוד:

- הגשת התיכון – 5%
- הצגת השלד – 10%
- הצגת הפרויקט – 5%
- הגשת הפרויקט – 80%

קובץ ה־README:

יש לכלול קובץ README שיקרא README.docx, README.doc 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. **פורמט קובץ השלב, ופירוט אם יש צורך בצעדים נוספים להוספת שלב (למשל, הוספת שם הקובץ החדש לקובץ עם רשימת הקבצים).**
6. רשימה של הקבצים שנוצרו ע"י הסטודנט, עם הסבר קצר (לרוב לא יותר משורה או שתיים) לגבי תפקיד הקובץ.
7. מבני נתונים עיקריים ותפקידיהם.
8. אלגוריתמים הראויים לציון.
9. באגים ידועים.
10. 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. כתבו 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 Project_name.zip, כאשר name הוא השם המלא. במקרה של הגשה בזוג, שם הקובץ יהיה לפי התבנית Project_name1_name2.zip, עם שמות המגישים בהתאמה (ללא רווחים; גם בשמות עצמם יש להחליף רווחים בקו תחתי או להצמיד את שני חלקי השם).

לפני דחיסת תיקיית ה־solution שלכם יש למחוק את הפריטים הבאים:

- תיקיות בשם debug, release או x64 (לרוב יש יותר מתיקייה אחת בשם זה, אחת בתיקיית ה־solution ואחת בתיקיית כל פרויקט).
- תיקייה (לפעמים מוסתרת!) בשם vs. שאמורה להיות בתיקייה עם ה־solution.

ככלל אצבע, אם קובץ ה־zip שוקל יותר ממ"ב אחד או שניים, כנראה שלא מחקתם חלק מהקבצים הבינאריים המוזכרים.

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 יכלול את שמות המגישים.
2. שימו לב שעליכם לשלוח את תיקיית ה־solution כולה, לא רק את קובצי הקוד שיצרתם. עקבו אחרי ההסבר המפורט באתר, במקרה שאתם לא בטוחים איך למצוא את התיקייה. תרגיל שלא יכלול את ה־solution, 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 ה־solution שבתוכו ולקמפל את הקוד. הרבה טעויות של שכחת קבצים יכולות להימנע על ידי בדיקה כזו.

בהצלחה!