

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

פרויקט סוף סמסטר

תאריכי הגשה: ראו לוח זמנים בהמשך.

מטרת התרגיל:

בתרגיל זה נນם שוב את המשחק Lode Runner שמיימנו בתרגיל 2. ההבדלים המרכזיים הם שהפעם השתמש בירושה ופולימורפיים וכן השתמש בספרייה הגרפית SFML.
מאז תרגיל 2 למדנו נושאים חדשים והתקדמנו באופן כללי בתוכנות מונחה עצמים, כעת נוכל לעצב תיקון טוב יותר,יפה יותר ויעיל יותר של האפליקציה בהתאם לעקרונות תוכנות מונחה עצמים.
כמו כן, נוסיף תכונות נוספות לעומת מה שעשינו בתרגיל 2.

שינויים והוספות:

רוב פרטיו המשחק נשארים בדיק כפי שהיו בתרגיל 2, למעט מה שכבר הזכיר לעיל, שימוש בירושה, פולימורפיים וממשק גרפי, וכן השינויים וההוספות המפורטים להלן.

משחק מונחה שעון:

המשחק כעת יהיה "זמן אמיתי", מונחה שעון, ולא מונחה תוררים. ראו פירוט בהמשך.

התנטשות בקיימות או בין דמיות:

הגדרת התנטשות היא כאשר יש נקודה חופפת בין המלבנים החסומים הרלוונטיים, כלומר המלבן החוסם את המכשול והמלבן החוסם את הדמות או המלבנים החסומים של שתי הדמיות המתנגשות (שחקן ואיבר).

הגבלות או "בונוסים" לשחקן:

בכל שלב יכולה להיות מוגדרת הגבלת זמן. אין חובה שbullet שלב תהיה ההגבלה הזאת, אפשר ליצור שלבים שבכל לא יהיו מוגבלים כך (למשל, אם רוצים שלבים ראשונים קלים ל"תרגול"), אבל יש חובה לתמוך באפשרות הזאת.

אם עבר הזמן המוגדר, זה נחשב לפסילה של השחקן (ירדים "חיים") והשלב מתחל מחדש (כלומר השלב יכול מתחילה מחדש, כולל המטבעות שהוא).
עד תוספת היא "מתנות" המפוזרות בשלב. השחקן אוסף את המתנה על ידי "דריכה" עליה. סוג המתנות שאותם

נדושים למשם הם:

1. הגדרת הזמן המוקצב (בהתבה שיש בשלב זהה מגבלה על הזמן).

2. הוספת ניקוד.
3. הוספת חיים.
4. הכנסת אויב נוסף למשחק ("מתנה" רעה).

לבחירהכם האם התוספות קבועות מראש או רנדומליות. אויבים יכולים לעבור למקום שיש בו מתנה, אבל הם כנובן לא יכולים לקחת אותה.

יכולת חפירה:

השחקן יכול לחפור בור ברצפה חשופה מימין או משמאלי (באמצעות המקשים: א' לבור ימני, ז' לשמאלי). הבור שנחפר הוא לכמה שניות בלבד. אחרי שהן חולפות הבור נסגר ומי שהיה בתוכו משנה את מיקומו לתא אחד מעלה הבור. ההחלטה אם ניתן ללכט או לא על מי שנמצא בבור נתונה לשיקולכם. מומלץ למשוך את יכולת החפירה לאחר שהיכולות הבסיסיות יותר כבר עבדות היטב.

שלבי המשחק:

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

תזכורת ועדכונים לפורמט הקובץ:

בשורה הראשונה מופיעים שלושה מספרים: מספר השורות בשלב ומספר העמודות, כפי שהיא בתרגיל 4, וכעת גם הגבלת הזמן לשלב (בשניות). ניתן להשתמש במספר 1- כדי לסמן שהגבלת הזמן לא פועלה.

אחריה מופיעות שורות כמספר השורות שהוגדר עם הסימונים: # (קייר/רצפה), @ (שחקן), % (אויב), - (מוט), H (סולם), * (מטבע) ורוחח למשבצות הריקות. בנוסף, יכול להופיע הסימון + ("מתנה").

נשאר 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 מסויימת לשמות הקבצים, כך שהתוכנית יודעת אוטומטית מה שם הקובץ הבא שהוא צריך לנוסות, או שימוש בקובץ שמננו קוראים את רשיימת שמות קובצי השלבים).

באופן כללי, ניתן להניח שהקובץ תקין.

ניקוד:

הניקוד נותר כפי שהיא, על כל מطبع – 2 נקודות כפול מספר השלב, על סיום שלב – 50 נקודות כפול מספר השלב.

סוגי אויבים:

יהו שלושה סוגים אויבים שיבדלן זה מזה באופן התנועה שלהם: תנועה רנדומלית, תנועה קבועה (למשל ימינה עד הסוף ומשמאלת עד הסוף, הלוך ושוב) או תנועה לעבר השחקן. הבחירה של סוג האויב נעשית בצורה אקראית ביצירת האויב בתחילת השלב (כזכור, את המידע על מיקום האויבים ומספרם אנחנו מקבלים בקריאה מהקובץ).

שימוש בתפריט:

עליכם לאפשר תפריט בסיסי של בחירת משחק חדש או יציאה. התפריט יוצג בתחלת ריצת התוכנית או לאחר המשחק נגמר (כ噫 השחקן סיים את כל השלבים בהצלחה או כי נגמרו לו החימ). לבחירתם האם התפריט יופעל בעזרת העכבר או המקלדת (או שניהם). אם הוא מופעל בעזרת המקלדת רואו לתמוך בבחירה אפשרות מהຕפריט על ידי ללחיצה על אותן מתוך הכתוב המופיע על הcptor. במקרה זה, נדגש את אותן על ידי קו תחתי

(כמו בדוגמה כאן: ).

תצוגת מידע:

אתם נדרש להציג עבור השחקן את מספר הנקודות שצבר, את מספר הפסילות שנשארו לו ואת הזמן שנותר לו, וכן להציג את מספר השלב הנוכחי.

תיזוזה:

למרות שבשימוש במסך גרפי אנחנו יכולים להציג את הדמויות גם באלאסון, כמובן, בכל זאת נשאיר את התזוזות רק בקווים ישרים לאורך הצירם, וכן גם המקשים של השחקן נשארים כפי שהם (חיצים לתיזזה לאחד מרבעת הצדדים בהתאם לההתאמה).

הלחיצה על המקשים נשארת כשהיתה (רק שהפעם מזחים זאת בעזרת אירועים של SFML במקום באמצעות getch). ההבדל הוא שעכשו אנחנו לא מתנהלים לפי תור אלא לפי השעון. מילא, צריך לשים לב שקיוב התיזזה לא يستمر על קצב קבלת האירועים של key pressed, שאינו אחד, אלא על משך הלחיצה ועל השעון. הדריך לעשות זאת היא כך שהלחיצה רק קובעת האם השחקן יזוז ולאיזה כיוון הוא יזוז, כאשר נרצה להציג אותו. התיזזה בפועל מתבצעת לפי הזמן שחולף. השחקן נוצר (גם בלי להיתקע במכשול) ברגע שהשחקן הפסיק ללחוץ על המקש (אירוע של key released).

MRIORIOTOT TZOOZA:

עליכם להחליט כיצד לעדכן את מיקום הדמויות. מכיוון שהגדכנו שהמשחק מונחה שעון, הרי שתצטרכו לבצע זאת כל פרק זמן מסוים ותצטרכו לקבוע בכמה הדמות זהה בכל פרק זמן זהה, לפי הכיוון אליו היא זהה כרגע. האם עדכון כל עשרית שנייה יהיה מצליח? אולי יותר? האם תיזזה של פיקסל או שניים סבירה? האם, כדי לחתך דינטגר, האויבים צריכים להיות מהירים יותר (כלומר, יותר פיקסלים בכל איטרציה) מהשחקן? עד כמה מהירים? לא נקבע כללים בנושא, אלא נשאיר זאת עבורכם לנסוט ערכים שונים ולהחליט איך ליצור משחקים טובים. בכל מקרה, תעדו את החלטותיכם בקובץ README.

עיצוב האובייקטים:

לגרפיקה מוצלחת כנראה שתרצzo לצייר את האובייקטים בתוכנה חיצונית או למצאו תמונות מתאימות ולטעון את התמונות בעזרת Texture sf. במצב זה, כדי שהתמונה תהיה עם רקע שקוֹף (ניתן לעשות זאת בתמונות שמורות כ-*sf::Texture*, למשל), כדי למנוע רקע לבן שייתווסף לדמות. זה חשוב בעייר אם מדובר בדמות שאין לה צורה פשוטה שתכיל אותה ואתם צריכים להכיל אותה בתוך *sf::RectangleShape* או *sf::Sprite* שחווסם אותה. כמו כן, כדי לדאוג שההדרות תתופיע ככל האפשר כמה שיותר מהמלבן החום, כדי שהיא ברור יותר בשעת המשחק מתי נחשב שדמויות התנגשו זו בזו או במכשול.

אואנד:

אתם נדרשים להוסיף למשחק סאונד.

הערות:

- בוגר לתרגיל 4, הפעם אין איסור להחזיק במחלקות משתנים מסוג הצורות של SFML. למעשה, אין אפילו עדיף להחזיק את הצורות מוכנות כבר, וכך יכול לחשב אלו מאפיינים (למשל, מיקום) אין צורך להחזיק בנפרד כי הם כבר נמצאים במאפייני הצורה.
 - למשך אחורי הזמן החולף – המחלקה הרלוונטית היא `Clock`. אפשרות אחרת היא להשתמש בספרייה `std::chrono` שמשולבת כחלק מהספרייה הסטנדרטית מאז C++11. שימוש לב שאטם צרכים לבדוק בעצמכם כמה זמן עבר (בעזרת הפונקציה `getElapsedTime()`, במקרה של `Clock` או `sf::Clock`) ובדוק אם מתקיים מתחים מסוימים במהלך הפעולה (במקרה של שימוש בספרייה הסטנדרטית). מן הסתם המקום המתאים הוא בולולאת האירועים או בסמוך לה. (אגב לולאת האירועים, כדי להתנהל לפי השעון חיבים לשימוש הפעם ב-`pollEvent()` ולא ב-`waitEvent()`).
 - הקפידו על תיקון נכון, מונחה עצמים, תוך שימוש ראוי בירושא ובפולימורפיזם. שימוש לב, לא מספיק רק לשימוש בירושא. אתם נדרשים לשימוש בפולימורפיזם נכון, ולהקפיד על חלוקת אחריות נכונה בין המחלקות.
 - כרגע, אתם מתבקשים לטעוד בקובץ `Readme` תוספות שעשיתם או החלטות לגבי דברים שלא מוגדרים בפירוש.
 - כפי שהוזכר כבר בתרגיל 2, ונכון עוד יותר עבורuproject, בכל פרויקט מסוים מרכיב יש התנהוגות שלא מוגדרות במדדיק על ידי הדרישות. אפשר להתייעץ במקרה הצורך, אבל ההנחה הכללית היא של דברים שלא נאמרו בהגדרות אתם מוזמנים להחיליט בעצמכם מה הדרך הטובה יותר לביצוע הפעולה ולפתרון הבעיה (כלומר, טוביה יותר מבחינת משחקיות, או סבירה מבחינת משחקיות וקללה מבחינה תכונתית :)).
 - אתם רשאים כמובן להוסיף על הrama הבסיסית, הן מבחינה גרפית, והן תוספות לשיפור המשחקיות (סוגים נוספים של "מתנות", אויבים חכמים יותר וכו') וכן בשימוש באלגוריתמים מתחוכמים או בתבניות עיצוב וכל תכונות מתקדמות. בכל מקרה, אם הוסיףם - כדאי לציין זאת בקובץ `Readme`.

הגשת תיכון:

הגשת התיכון נעשית בהגשת דיאגרמת UML (סוג class diagram) המציג את המחלקות המתוכנות עם מאפיינים (שדות) עיקריים ופונקציות עיקריות שלhn ועם הקשרים בין המחלקות (לפחות ירשה, אבל עדיף גם הכלה). בנוסף, יוגש הסבר מילולי קצר שיסייע בהבנת הדיאגרמה וחלוקת האחריות בין המחלקות. ההסבר המילולי יכול לספר קצר על כל מחלקה ואם יש צורך – גם על הקשר בין מחלקות שונות.

הקפידו על שמות שימושיים גם למחלקות, גם לפונקציות המחלקה וגם למשתנים, כך שייהי ברור מקרית התיכון מה התפקיד שלו כל דבר.

חשוב להבין: בלוח הזמנים מוקדש זמן יחסית קצר עבור הגשת התיכון, מכיוון שמטרתו העיקרית היא לגרום לכם לתכנן לפני שתיגשו למשת הפROYיקט. הערכה היא שלא תידרשנה לשם כך יותר מספר שעות במקרה הגרוע ביותר. אתם לא צריכים לתכנן כאן את הקוד שלכם לפרטי פרטים, אלא רק לשבת ולהסביר איך נראה לכם כרגע שכך למשת הדברים מבחינת חלוקת פונקציונליות בין המחלקות, מבחינת עץ ירשה וכדומה.

התיכון צריך להיות סביר והגיוני, אבל לא יקרה שום אסון אם במהלך כתיבת הפROYיקט בפועל תגלו שאתם רוצים לשנות את חלוקת המחלקות שלכם וכדומה. זה לא אומר שה בסדר אם התיכון יהיה מושל מלכתחילה; זה כן אומר שלא צריך להילחץ אותו יותר מדי.

קיימות מספר שיטות מומלצות כדי לשרטט את הדיאגרמה. הסברים מפורטים נמצאים באתר בקובץ הנקראות **לייצרת Class Diagram (UML)**.

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

הגשת התיכון מתבצע כרגיל, כשהקבצים שאתה מגישים מכוחים לקובץ קז, ושם הקובץ, כרגיל, כולל את שמות המגישים אלא שהפעם במקום שם התרגיל תופיע המילה design, כלומר תבנית שם הקובץ היא lastname.zip, עבור מגיש בודד, כאשר firstname_lastname.zip הוא השם המלא (לדוגמא אלברט איינשטיין יגיש את התיכון כך: Design_albert_einstein.zip). במקרה של הגשה בזוג, שם הקובץ יהיה lastname_firstname.zip, עם שמות המגישים בהתאם לדוגמה lastname1_firstname1_lastname2_firstname2.zip. כמו כן, במקרה של הגשה בזוג, רק אחד מהמגישים יגיש את הקובץ ולא שניהם.

הציגת השלד:

כדי לגרום לכם להתקדם עם הפROYיקט ולנצל את הזמן ולא לדחות הכל לסופ, כמו שעלול לקרות לרבים מאננו, אתם נדרשים להציג התקדמות אחרת כשבועיים מהגשת התיכון. הצגת השלד תישא בזמן הסדנה. כל צוות ישב עם המתרגל/ת במשך 5-10 דקות ויציג את התקדמות הנווכחית והנិזוק לשלב זהה יקבע בהתאם.

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

הציגת הפROYיקט:

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

הגשת הפROYיקט:

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

לוח זמנים:

(בסוגרים מצין פרק הזמן מאז נקודת הציון הקודמת.)

كمפואו הנביאים:

- תחילת הפROYיקט – יום שני, 21/12/2020
- הגשת תיקון (טור שלושה "ימי עבודה") – יום רביעי, 23/12/2020 (עד שעה 23:59)
- הציגת השלד (אחרי שבועיים) – יום חמישי, 7/1/2021, בזמן (התרגול) והסדנה 21/1/2021
- הציגת הפROYיקט (אחרי שבועיים) – יום רביעי, 20/1/2021, ויום חמישי, 21/1/2021 (עד שעה 23:59)
- הגשת הפROYיקט (באותה היום) – יום חמישי, 21/1/2021 (עד שעה 23:59)

ניקוד:

- הגשת תיקון – 5%
- הציגת השלד – 10%
- הציגת הפROYיקט – 5%
- הגשת הפROYיקט – 80%

: README

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

קובץ זה יכול לכל הפחות:

1. כוורת.
2. פרטי הסטודנט: שם מלא כפי שהוא מופיע ברשימות המכללה, ת"ז.

3. הסבר כללי של התרגיל.
4. תיקון (design): הסבר קצר ממה 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 בצדדים נוספים להוספה שלב (למשל, הוספה שם הקובץ החדש לקובץ עם רשימת הקבצים).
6. רשימה של הקבצים שנוצרו על ידכם, עם הסבר קצר (לרוב לא יותר משורה או שתיים) לגבי תפקיד הקובץ.
7. מבני נתונים עיקריים ותפקידיהם.
8. אלגוריתמים הרואים לציון.
9. באגים ידועים.
10. הערות אחרות.

יש לתמצת כל שנייתן 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.

אוף ההגשה:

הקובץ להגשה: יש לדוחו כל קובץ הקשור לפרויקט, למעט מה שיופיע להלן, לקובץ ששמוfirstname_lastname.zip כאשר firstname lastname הוא השם המלא (לדוגמא אלברט איינשטיין Project_albert_einstein.zip). במקרה של גישה בזוג, שם הקובץ יהיה לפי התבנית Project_firstname1_lastname1_firstname2_lastname2.zip, עם שמות המציגים בהתאם (לא רוחים); ככלומר, גם בשמות עצם יש להחליף רווחים בקו תחתוי, כפי המודגם לעיל). כמו כן, במקרה של גישה בזוג, רק אחד מהציגים יגיש את הקובץ ולא שנייהם.

לפני דחיסת תיקיית הפרויקט שלכם יש למחוק את הפריטים הבאים:

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

שתי התיקיות האלה נמצאות בתיקייה הראשית (זו שאנו חוננו פותחים בעזרת VS). התיקייה `av`. לעיתים מוסתרת, אבל אם תפתחו את קובץ `h-kpz` שיצרתם, בוודאי תוכלו למצוא אותה ולמחוק אותה.

כל אצבע, אם קובץ `h-kpz` שוגל יותר ממ"ב אחד או שניים, נראה שלא מחקתם חלק מהקבצים הבינאריים המוזכרים.

וודאו כי קובץ `h-kpz` מכיל תיקייה ראשית אחת, ורק בתוכה יהיו כל הקבצים והתיקיות של הפרויקט.
את הקובץ יש להעלות ל-[Moodle](#) של הקורס למשימה המתאימה.

הגשה חוזרת: אם מסיבה כלשהי סטודנט מחייב להגיש הגשה חוזרת יש לוודא שם הקובץ זהה לחולוטין לשם הקובץ המקורי. אחרת, אין הבודק אחראי לבדוק את הקובץ האחרון שיוגש.

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

מספר הערות:

1. שימוש לב לשם הקובץ שכן יכול את שמות המגיישים.
2. שימוש לב שעligeם לשלוח את תקיית הפרויקט כולה, לא רק את קובצי הקוד שיצרתם. תרגיל שלא יכול את כל הקבצים הנדרשים, לא יתקבל וידרשו הגשה חוזרת (עם כללי האיחור הרגילים).

המלצת כללית: אחרי שהכנתם את הקובץ להגשה, העתיקו אותו לתיקייה חדשה, חלצו את הקבצים שבתוכו ובדקו אם אתם מצליחים לפתח את התקייה הזו ולקיים את הקוד. הרבה טעויות של שכחת קבצים יכולות להימנע על ידי בדיקה צזו.

בהצלחה!