



הפקולטה למדעי ההנדסה

אוניברסיטת בן גוריון

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם



תרגיל בית מספר 4 - תאריך הגשה: 17.01.25 בשעה: 23:59

במטלה זה עליכם יהיה לכתוב משחק מחשב קטן דמוי `dungeon crawler`. למי שלא מכיר, במשחק זה, אתם הרפתקנים שנכנסים למבוך ותפקידכם הוא למצוא את היציאה ממנו תוך שאתם נלחמים במפלצות ומנסים לנוח כמה שניתן באזורי המנוחה המעטים שיש בתוך המבוך. מטלה זו מתמקדת בעיקר בכתיבה ב-`C++` תוך התנסות ביצירה וניהול של `classes` (במקום `structs`), ניהול זיכרון, ועבודה עם `operator overloading`.

המטלה מחולקת לשני חלקים: תחילה ניצור את המחלקות איתן נעבוד במהלך החלק השני כאשר יפורט מה צריך להיות בכל מחלקה וכיצד מומלץ לממשה. בחלק השני, נחבר את כל המחלקות הללו לכדי משחק שלם.

לפני תחילת פתרון המטלה, אנא קראו עד סוף העבודה כדי להימנע מאי-הבנות. בנוסף, בטרם הגשת המטלה, עברו על הצ'קליסט וודאו כי אתם מגישים כראוי וללא שגיאות.

לנוחיותכם, מצורפים קבצי קונפיגורציה וכן קבצי קלט\פלט. אולם יש להתנסות בעוד מקרי הרצה טרם הגשת המטלה.

שימו לב שבמקרה שבו יש הבדל בין מה שרשום במטלה לבין מה שרשום בקובץ, הקובץ תמיד הוא זה שקובע, אולם יש להתריע פער זה כמה שיותר מהר.

דגשים

- אין להשתמש במבני נתונים של `C++` כלל. המבנה המותר היחיד הוא מערך רגיל.
- כאשר רשום פלט במטלה, הוא מסומן באדום.
- ניתן להשתמש במחלקה `std::string` במטלה זו.



מחלקות המשחק

דמות (Entity)

מחלקה זו מייצגת דמות במשחק, בין אם זה השחקן ובין אם זו מפלצת. למחלקה זו יהיה שם, כמות חיים מקסימלית, וכמות נזק שהדמות תעשה בכל התקפה. שימו לב שכמות החיים של דמות לא תעלה על הערך המקסימלי שלה ולא תרד מתחת לאפס. על המחלקה לתמוך בפונקציונליות הבאה:

- אופרטור += שמקבל מספר שלם: אופרטור זה יעלה את כמות החיים הנוכחית במספר שהתקבל.
- אופרטור -= שמקבל דמות אחרת: אופרטור זה יוריד את כמות החיים הנוכחית בהתאם לערך ההתקפה של הדמות שהתקבלה.
- אופרטור -= שמקבל מספר שלם: אופרטור זה יוריד את כמות הנזק שהדמות עושה במספר שהתקבל.
- אופרטור הדפסה: האופרטור ידפיס את הדמות לפי התבנית הבאה כאשר ערך בסוגריים משולשים הוא ערך שנחליף בערך המתאים לדמות:
<damage> - (<currHealth>/<maxHealth>) <name>
- אופרטורי השוואה (>, ==) שמקבלים דמות אחרת: האופרטור ישווה דמויות לפי הנזק שלהן כפול כמות החיים הנוכחית שיש להן. למשל, דמות עם 40 חיים ו-60 נזק (2400) תהיה שווה לדמות עם 60 חיים ו-40 נזק (גם 2400).

חדר במבוך (Room)

החדר יכיל מספר ערכים שיצביעו על מה יש בו ולכן ניתן ללכת ממנו. החדר יקבל שלושה מספרים שלמים שיסמנו מה יש בחדר. הערך הראשון מסמן האם יש מדורה בחדר בה ניתן לנוח וכן כמה נקודות חיים היא תעלה לדמות של השחקן. שני הערכים האחרים מסמנים האם יש מפלצת בחדר זה, ומה ערך החיים והנזק שלה. בנוסף, כל חדר יחזיק מערך של החדרים אליהם ניתן להגיע מחדר זה. שימו לב שחדר שממנו אין חדרים נוספים הוא חדר שממנו ניתן לצאת מהמבוך ולנצח.



הפקולטה למדעי ההנדסה

אוניברסיטת בן גוריון

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם



set

על המחלקה לתמוך בשני מימושים של אופרטור [], אחד להשמה (צד שמאל של =) ואחד לגישה. האופרטור [] פועל על החדרים במערך, מקבל מספר שלם המייצג את מספר החדר מתוך מערך החדרים שאליהם ניתן להגיע, ומחזיר חדר זה.

get

משחק (Game)

כדי לשמור על דברים מסודרים, נרצה לשמור על כל האובייקטים שלנו ופונקציונליות ההרצה במקום אחד כך שיהיה קל לשחרר את הזיכרון ולהריץ כראוי את המשחק. שימו לב שבצורת מימוש זו, אין זה חובה שהקובץ ה-main יהיה ארוך (אם כי אין זו חובה אף פעם). למשחק יהיו שתי פעולות עיקריות שנרצה לתמוך בהן:

- אתחול משחק: אשר מאתחלת את המשחק מקובץ קונפיגורציה וערכים שניתנו על ידי המשתמש.
- הרצת המשחק: הפונקציה המרכזית שתפקידה להריץ את המשחק עצמו.

לפני שנמשיך לפרט על קובץ הקונפיגורציה. כל שורה בקובץ הקונפיגורציה מהווה חדר במשחק. שורה מורכבת מ-4 מספרים W X Y Z :

- המספר הראשון W מייצג את המספר המזהה של החדר. זהו גם האינדקס אותו צריך לקחת כדי להגיע לאותו חדר. כל ספרה במספר המזהה הוא המספר של החדר שאותו שאליו נצטרך להיכנס כדי להגיע לחדר המבוקש. למשל החדר 10 (כפי שניתן לראות באיור) אומר שצריך לעבור תחילה בחדר שממסופר 1, ולאחר מכן בחדר שממסופר 0 כדי להגיע לחדר 10. בדומה לכך, החדר 11 אומר שצריך תחילה לעבור בחדר שממסופר 1 ולאחר מכן בחדר שממסופר 1 פעם נוספת. שימו לב שאין הגבלה לגודל מספר זה.
- המספר השני X מהווה את כמות החיים שמדורה בחדר תרפא. מספר זה הוא לא-שלילי. אם הוא 0, אזי אין מדורה בחדר. אם הוא חיובי, יש מדורה בחדר שתרפא X נקודות חיים.
- שני המספרים Y Z מתארים את המפלצת בחדר. Y הוא נקודות החיים של המפלצת ו-Z הוא כמות הנזק. המספרים הם אי-שליליים. אם Y הוא אפס, אזי אין מפלצת בחדר ללא קשר ל-Z. אם הוא חיובי, יש למפלצת Y נקודות חיים, והיא תעשה בכל התקפה Z נזק לשחקן.



הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

אוניברסיטת בן גוריון



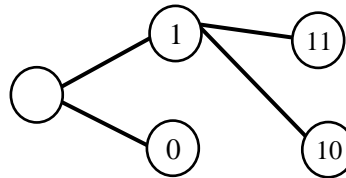
הנה קובץ קונפיגורציה לדוגמה:

1 0 0 0

0 10 0 0

10 0 20 10

11 1 100 100



החדר הראשון תמיד קיים, ריק ואין לו מספר מזהה (והוא גם לא מופיע בקובץ הקונפיגורציה).
כעת נפרט על כל חדר ונבין מה יש בו:

- חדר 1 – זהו חדר שבו אין מדורה ואין מפלצת.
- חדר 0 – חדר שבו יש מדורה שתרפא את השחקן ב-10 נקודות חיים. בחדר זה אין מפלצת.
- חדר 10 – חדר בו אין מדורה אך יש מפלצת עם 20 חיים שעושה 10 נזק.
- חדר 11 – חדר שבו יש מדורה שתרפא 1 חיים, ומפלצת עם 100 חיים שעושה 100 נזק.

ניתן להניח שהקלט תמיד חוקי, וללא מספרים שליליים. בנוסף, ניתן להניח שעבור כל חדר, מערך החדרים שניתן להגיע אליהם יהיה מלא בחדרים חוקיים בסוף קריאת קובץ הקונפיגורציה. ניתן לראות שבקובץ לדוגמה, אנו קוראים תחילה את השורה הראשונה- חדר מספר 1. הדבר אומר שלחדר הראשון יש במערך החדרים שניתן לגשת אליהם רק חדר אחד שמספר המזהה שלו הוא 1. לאחר קריאת השורה השניה נוסף למערך גם חדר מספר 0. מובטח שבסוף קריאת הקובץ, לכל חדר מערך החדרים שאליהם ניתן להמשיך יהיה מלא עד לחדר עם המספר המזהה הגבוה ביותר.

בנוסף, כל חדר יכול להוביל לכל היותר ל-10 חדרים שממוספרים מ-0 עד 9.



המשחק

המשחק מתחיל בקריאת קובץ הקונפיגורציה שניתן לשחקן ויצירת הדמות (נפרט אחר כך כיצד מקבלים את מידע זה), ולאחר מכן ממשיך ללולאת המשחק. כעת אנו נפרט כיצד לולאה זו עובדת ומה יש להדפיס בכל שלב:

תחילת המשחק

תחילה אנו נדפיס את המידע אודות דמות השחקן ששמו תמיד יהיה "Player". לאחר מכן, אנו נשאל את השחקן האם הוא רוצה לאתגר את עצמו ולהוריד את כמות הנזק שדמותו תעשה. אם כך המשחק ידפיס:

I see you like challenges, by how much do you want to reduce your damage?

לאחר מכן נקלוט מספר שלם אי-שלילי מהשתמש ונחסרו מכמות הנזק של הדמות.

חדר

כעת הדמות תתחיל לעבור בחדרים. בתחילת כל חדר, אנו נדפיס את דמות השחקן כדי לראות מה מצבה. לאחר מכן, כתלות במה יש בחדר, אנו נדפיס דברים שונים: אם בחדר יש מדורה, נדפיס את השורה הבאה כאשר `<health>` יוחלף בכמה שהמדורה מרפאת אותנו. שימו לב שכאן יש להדפיס את הערך המלא:

You sit by the campfire and heal `<health>` health

קרב

אם בחדר יש מפלצת, יחל קרב בינה לבין השחקן. תחילה אנו נדפיס אמירה לגבי הבדל הכוחות בין המפלצת לשחקן (אשר מוגדר לפי אופרטורי ההשוואה). אם השחקן חזק יותר, נגיד, שהמפלצת היא "**smaller**", אם המפלצת חזקה יותר, נגיד שהיא "**larger**" ואם היא שוות ערך, נגיד שהיא "**equally sized**". על כן נדפיס את השורה הבאה, כאשר `<desc>` יוחלף בתיאור המפלצת:

You encounter a `<desc>` monster

לאחר מכן נדפיס את המפלצת עצמה. כעת קרב יחל. תחילה השחקן תוקף ומוריד למפלצת כמות נקודות חיים ששווה להנזק שלו. אם כמות הנזק של השחקן גדולה מכמות נקודות החיים של המפלצת, הוא יוריד אותה ל-0 נקודות חיים. אם הוא הוריד את המפלצת ל-0 חיים, היא מובסת והוא ממשיך במשחק. אם לא, תורה של המפלצת לתקוף ולהוריד כמות חיים לשחקן ששווה לנזק שלה. אם כמות הנזק שלה גדולה מכמות החיים של השחקן, היא תוריד אותה ל-



הפקולטה למדעי ההנדסה

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם

אוניברסיטת בן גוריון



0 נקודות חיים. אם היא הורידה את השחקן ל-0 חיים, המשחק מסתיים בתבוסה ויש לצאת ממנו לאחר ניקוי מלא של הזיכרון.

הנה דוגמה מלאה לחדר ובו מתרחש קרב שבו השחקן מנצח:

Player (100/100) - 20

You encounter a smaller monster

Monster (30/30) - 20

You deal 20 damage to the monster and leave it with 10 health

The monster deals 20 damage to you and leaves you with 80 health

You deal 20 damage to the monster and leave it with 0 health

You defeat the monster and go on with your journey

ובמקרה שבו הוא מפסיד:

Player (100/100) - 20

You encounter a larger monster

Monster (100/100) - 100

You deal 20 damage to the monster and leave it with 80 health

The monster deals 100 damage to you and leaves you with 0 health

You lost to the dungeon

מעבר בין חדרים

כאשר השחקן סיים לטפל במדורה ובמפלצת, עליו לבחור בחדר הבא אליו הוא ממשיך.

ישנם שלושה מקרים אפשריים:

כאשר אין חדרים נוספים, מדובר בסוף המבוך, כלומר הגענו לניצחון. לאחר ההדפסה אנו נצא

מהתוכנית לאחר ניקוי זיכרון:

The room continues and opens up to the outside. You won against the dungeon

אם ישנו חדר אחד:

You see a single corridor ahead of you labeled 0

ואם ישנם כמה חדרים (במקרה זה 3 חדרים):

You see corridors labeled from 0 to 2. Which one will you choose?

בכל אופציה בה יש לפחות חדר המשך אחד, נבקש מהמשתמש להכניס את מספר החדר אליו הוא רוצה ללכת ונתקדם אליו. ניתן להניח שהקלט חוקי ויסמן חדר קיים.



הפקולטה למדעי ההנדסה

אוניברסיטת בן גוריון

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם



כעת הלולאה תחזור על עצמה ונמשיך במשחק עד אשר הוא יסתיים.
כדי להריץ את המשחק אנו נקרא לקובץ ההרצה עם כמות החיים של דמות השחקן, כמות
הנזק שהיא עושה, וכתובת (חלקית או מלאה) לקובץ הקונפיגורציה, לדוגמה:
./Dungeon 100 20 configuration.txt

הנה דוגמת הרצה מלאה לשורה למעלה:

Player (100/100) - 20

I see you like challenges, by how much do you want to reduce your damage?

1

Player (100/100) - 19

You arrive to an empty room

You see corridors labeled from 0 to 1. Which one will you choose?

1

Player (100/100) - 19

You arrive to an empty room

You see corridors labeled from 0 to 1. Which one will you choose?

0

Player (100/100) - 19

You encounter a smaller monster

Monster (20/20) - 10

You deal 19 damage to the monster and leave it with 1 health

The monster deals 10 damage to you and leaves you with 90 health

You deal 19 damage to the monster and leave it with 0 health

You defeat the monster and go on with your journey

The room continues and opens up to the outside. You won against the
dungeon



הפקולטה למדעי ההנדסה

אוניברסיטת בן גוריון

המחלקה להנדסת מערכות מידע

372-1-2102 תכנות מתקדם



הנחיות הגשה

ההגשה ביחידים בלבד. נא לעבור על כל המדריכים במודל תחת "Assignments - General" כולל הציקליסט לפני ההגשה במודל, להבינם, ולסמן V על כל חלק וחלק בצ'קליסט. שימו לב שבהגשת מטלה זו אתם מתחייבים שקראתם והבנתם את המדריכים. יש להגיש 9 קבצים:

- Entity.cpp
- Entity.h
- Room.cpp
- Room.h
- Game.cpp
- Game.h
- main.cpp
- ReadMe.txt
- makefile - שימו לב: את העבודה אתם יכולים לכתוב בכל סביבת עבודה שנוחה לכם, אך יש להגיש קובץ makefile שכתבתם בעצמכם כפי שלמדנו במעבדה. שם התוכנית להרצה לאחר שימוש בקובץ הינו Dungeon .

התרגיל יעבור בדיקה אוטומטית. כדי שהתוכנית תעבור בשלום את הבדיקה האוטומטית, אין לשנות את השמות של הקבצים, ואין לשנות את הפורמט של ההדפסות שניתנו – הן מבחינת כמות upper/lower case , enters , ורווחים. אם ההדפסה אינה מדוייקת כנדרש, הבדיקה תיכשל. בנוסף, המטלה תעבור בדיקה מדגמית לשימוש נכון באופרטורים, והקפדה על ההוראות שנרשמו במטלה.

התרגיל יעבור בדיקות קפדניות למציאת העתקות, וסטודנטים שיימצא כי העתיקו יענשו בחומרה. אנא הימנעו מהעתקות. את המטלה יש להגיש דרך המודל במקום המיועד לכך. שאלות לגבי המטלה ניתן יהיה לשאול בפורום מיוחד שייפתח באתר הקורס.

Good luck in your adventure, and avoid taking an arrow to the knee