

מטלה 1 – תכנות מתקדם בשפת ++C

הנחיות: קראו היטב את ההנחיות, אי עמידה בהנחיות אלו תגרור הורדת ציון.

- העבודה וההגשה ביחידים בלבד!
- יש להגיש למערכת ה moodlearn עד מועד ההגשה, אם יש סיבה מוצדקת לאיחור בהגשה יש לדווח עד יומיים לפני מועד ההגשה, לאחר מכן לא תתקבל שום סיבה לאיחור.
- יש להגיש את כל הקבצים בתיקייה המכילה את קבצי הפרויקט .cpp וקבצי .h ובנוסף יש להגיש קובץ cmake ואת קובץ ה makefile שנוצר, את התיקייה יש לכווץ לקובץ ששמו הוא ת"ז של הסטודנט.
- חובה לכתוב בקובץ CmakeLists.txt עבור כל פקודה תיעוד מפורט מה הפקודה עושה.
- שימו לב שחובה שהפרויקט יעבור קומפילציה בקומפיילר g++. ניתן להשתמש גם ב ++C11
- שימו לב לתכנות נכון ולפי כל הכללים של "תכנות מונחה עצמים" שנלמדו בכיתה. בהחלט מותר להוסיף פונקציות או מחלקות נוספות מעבר למה שאתם נדרשים בתרגיל ולהגדיר את הקשרים ביניהם לצורך תכנות נכון בפרט יש לשים לב:
 - מתי יש להשתמש בירושה ובפולימורפיזם.
 - אלו פונקציות יהיו וירטואליות.
 - מתי להשתמש במחלקות אבסטרקטיות.
 - לשמור על עיקרון הכימוס.
 - לשים const במקומות שצריך.
 - מתי להשתמש במצביע, מתי ברפרנס ומתי בערך רגיל.
 - חובה לתעד כל פונקציה שאתם כותבים.
 - יש להימנע ממספרי קסם.
 - חובה עליכם לבצע בדיקות שאין לקוד שלכם דליפת זיכרון.
- לפני שאתם מתחילים לעבוד על התרגיל מומלץ לעבור ולהבין את כל המטלה כדי שתוכלו לכתוב ולעצב את הקוד בצורה הנכונה ביותר.

המטלה – משחק תפקידים

בתרגיל זה עליכם לכתוב משחק תפקידים – משחק שבו יש שחקנים: הגיבורים, אויבים, נשקים וכו' הנלחמים בתוך שדה קרב מדומה.

לכל סוג שחקן יש מאפיינים משלו, כפי שיפורט בהמשך.

המשחק מתחיל בכל פעם בתצורה (קונפיגורציה) התחלתית, אותה יש לקרוא מתוך קובץ. מצורף לתרגיל קובץ לדוגמא. בקובץ זה נגדיר את גודל ה"עולם", את כמות השחקנים, אילו נשקים יש להם, כמה כוח התחלתי יש להם וכו'.

הנתונים הללו ייקראו בתחילת המשחק ועל פיהם יאותחלו כל האובייקטים השונים.

זירת המשחק תוצג על המסך כמטריצה ובה יהיו כל השחקנים והפריטים השונים. לאחר כל שינוי, התוכנית תדפיס את כל המטריצה שוב במצבה החדש.

המשחק בעצם מתבצע ללא התערבותכם הישירה. מקובץ הקונפיגורציה יש לקרוא את המיקום הראשוני של כל שחקן ואת המיקום הסופי אליו הוא צריך להגיע. זה יעשה על ידי חישוב גיאומטרי של ההתקדמות על הקו בין שתי הנקודות הללו.

בדרך כלל שחקן יאסוף פריטים הנמצאים ברדיוס המוגדר לו וכן ילחם באויב במידה ויתקל בו במרחק קרוב, כפי שיוגדר.

בסיום המשחק תדפיסו את הקונפיגורציה הסופית לתוך קובץ, וזה הפלט של המשחק.

מימוש התרגיל

עליכם לכתוב את מחלקות המשחק, הקפידו לכתוב את שמות המחלקות כפי שמצוין בתרגיל, אי עמידה בדרישות אלו עלולה לגרור הורדת ציון.

בתרגיל זה תקבלו חלק מהמחלקות שעליכם לכתוב עם המאפיינים – יש לעצב את התוכנית באופן הנכון ולהוסיף פונקציות או מחלקות במידת הצורך. בהחלט יש צורך להוסיף `getters/setters` במידת הצורך. חובה להוסיף לכל מחלקה פונקציית `toString` לתיאור האובייקט.

➤ המחלקות במשחק

1. מיקום – Point2d

מחלקה זו מייצגת נקודה במישור ובה נשתמש לייצג את מיקום השחקנים/הפריטים.

2. שחקן – Hero

התכונות של השחקן:

1. שם השחקן
2. מין השחקן (זכר או נקבה)
3. כמות החיים של השחקן – מספר ממשי!
4. עוצמת השחקן – מספר ממשי!
5. הנשק של השחקן.
6. ההגנות של השחקן.
7. מיקום השחקן.

ישנם 3 סוגים של שחקנים:

לוחם (Warrior), קשת (Archer) ומכשף (Wizard).

לכל שחקן יש רדיוס מקסימאלי בו הוא יכול ללכת, להתקיף ולאסוף פריטים (אותו ערך עבור שלושת הרדיוסים):

- **Warrior** – לוחם – רדיוס $\sqrt{2}$
- **Archer** – קשת – רדיוס 5
- **Wizard** – מכשף – רדיוס 3

הפעולות המוגדרות לכל שחקן:

- הליכה – **move** המקבלת נקודת יעד.
- התקפה – **attack** המקבלת אויב שאותו השחקן מתקיף.
- איסוף פריט – **use** המקבל את הפריט.
- ספיגת התקפה – **damage** המקבלת ערך ממשי.

3. אויב – Enemy

התכונות של האויבים:

1. כמות חיים.
2. מיקום האויב.
3. עוצמת ההתקפה.

ישנם 2 סוגי אויבים:

- אויב רגיל – **Enemy**
- אויב חזק – **Elite**

הפעולות המוגדרות לכל אויב:

- התקפה – **attack** המקבלת שחקן שאותו האויב מתקיף.
- ספיגת התקפה – **damage** המקבלת ערך ממשי.

4. נשקים – Weapon

לכל נשק במשחק יש את המאפיין של **עוצמת הנשק ומיקום הנשק**. הנשקים במשחק מתחלקים ל 2 סוגים: נשק המצריך **שימוש ב 2 ידיים** ונשק המצריך **שימוש ביד אחת**.

עוצמת הנשק הכוללת תחושב כעוצמת הנשק כפול התוספת של OneHaded/TwoHanded כאשר התוספת למכפלה לכל סוג היא:

1.2 – **OneHaded**

1.6 – **TwoHanded**

לכל דמות שחקן יש את הנשקים שלו – ללוחם, לקשת ולמכשף.

הנשקים במשחק זה יהיו:

הנשקים של הלוחם:

- **Hammer** – TwoHanded weapon
- **Sword** – OneHaded weapon

הנשקים של הקשת:

- TwoHanded weapon - Bow
- OneHanded weapon - CrossBow

הנשקים של המכשף:

- TwoHanded weapon - Staff
- OneHanded weapon - Wand

5. הגנות – Armor

לכל הגנה יש את המאפיינים הבאים: רמת הגנה ומיקום ההגנה.

ישנן 2 סוגי הגנות:

- הגנת גוף – BodyArmor
- מגן – ShieldArmor

השחקן יכול להשתמש בשתייהן בו זמנית. הערכים שרמת ההגנה יכולה לקבל הם בין 0.65 ל 1.0 כאשר הפגיעה בשחקן תחושב כמכפלת עוצמת הפגיעה בערך ההגנה ולכן 1.0 – לא מגן כלל ו 0.65 זו ההגנה הטובה ביותר.

6. שיקויים – Potion

לכל שיקוי יש את המאפיינים הבאים:

- סוג השיקוי
- הכמות שנמצאת בשיקוי
- מיקום השיקוי

ישנם 2 סוגי שיקויים במשחק:

- שיקוי חיים – health – מוסיף חיים לשחקן ששותה אותו לפי הכמות שבשיקוי.
- שיקוי עוצמה – mana – רק עבור המכשף: בנוסף לשיקוי החיים, המכשף משתמש בשיקוי זה להוספת כמות החיים.

לא ניתן להעלות את כמות החיים מעל 100.

7. מחלקת Game

זוהי המחלקה הראשית, בה יש לקרוא את כל נתוני התרגילים, להריץ את התרגילים אחד אחרי השני ולדאוג לרשום את קובץ הפלט.

מחלקה זו תכיל את הנתונים הבאים:

- map של STL הממפה כל שחקן לנקודת היעד שלו.
- Vector של STL המכיל את הפריטים במשחק הניתנים לאסיפה (נשק/שיקוי/הגנה).
- מטריצה של המשחק. המטריצה תכיל תווים כפי המפורט בהמשך, וגודלה יקבע דינאמית על פי קובץ הקונפיגורציה של התרגיל.

הפעולות שיוגדרו במחלקה הזו:

- read – קריאה מקובץ תרגיל את כל הנתונים ושמירתם במבני הנתונים הנ"ל (ה-map וה-vector).
- update – עדכון של המטריצה בהתאם לצעדים שהתבצעו.
- play – הלולאה המרכזית. בלולאה זו נקרא את קבצי הקונפיגורציה (לפי הפרמטרים שנתתם לתוכנית, ראו בהמשך), כל פעם קובץ אחד, נריץ את התרגיל עד סופו ונרשום את הפלט, ואז נמשיך לתרגיל הבא.

כל פריט יוצג ע"י תו בודד באופן הבא:

Potion – 'P'
Weapon – 'W'
Armor – 'S'
Enemy – 'E'
Elite – 'L'
Warrior – 'A'
Archer – 'R'
Wizard – 'Z'
empty – '.'

לאחר קריאת הנתונים מהקובץ לולאת המשחק מתחילה להריץ צעד אחרי צעד לפי סדר הנתונים בקובץ.

צעדי השחקנים יהיו באופן הבא:

בהינתן נקודת יעד, יש לחשב את הכיוון שאליו השחקן צועד (במרחב רציף) ובכל שלב יש לעדכן את מיקום השחקן (צעידה לכיוון הנקודה) במטריצה באופן דיסקרטי – בכל פעם משבצת אחת למעלה/למטה/לצדדים/באלסכון.

יש לשים לב שהשחקן לא יכול לצאת מזירת המשחק ולכן כאשר השחקן הגיע לקצה, עליו לעצור את התקדמותו.

יש לכתוב מחלקה/מחלקות המטפלות בקריאה מהקובץ ובהרצת המשחק לפי הצעדים לעיל.

8. מחלקת Console

מחלקת console אחראית על מילוי מטריצת המשחק והדפסתה על המסך.

המחלקה תכיל את הפעולות הבאות:

- fillData – מילוי המטריצה בנתונים שנקראו מהקובץ ושמורים במחלקת game.
- print – הצגת המטריצה על המסך. פונקציה זו תיקרא לאחר כל עדכון שיתבצע.

הרצת המשחק

התוכנית צריכה לקבל ארגומנטים באופן הבא:

`./program_name -n [number] -files [file1 file2 ...]`

כאשר:

`program_name` = שם קובץ הריצה שיוצרתם.

`number` = מספר קבצי הקונפיגורציה שאתם מצרפים לתרגיל. כלומר: כמה משחקים אתם רוצים להריץ ברצף מההרצה הזו.

`file1 file2...` - קבצי הקונפיגורציה של התרגילים שאתם רוצים להריץ.

דוגמא הפעלה:

`./EX1 -n 2 -files g2.csv g3.csv`

התוכנית מקבלת את מספר קבצי הקונפיגורציה שיש להריץ (כל קובץ הוא משחק בפני עצמו) ואת רשימת הקבצים – קבצי CSV המכילים את כל הקונפיגורציה של המשחק.

פורמט הקובץ: ראו קובץ `input` מצורף.

ניתן לקבל מידע על אופן הפעלת התוכנית ע"י הדגל `help`, לדוגמא:

`./ Ex1 -help`

שידפיס את השורה הבאה (בפורמט הזה):

`-n [number] -files [file1 file2 ...]`

כללי המשחק

כאשר ברדיוס המוגדר לשחקן נמצא פריט – השחקן אוסף את הפריט לפי התנאים הבאים:

- אם הפריט הוא **נשק** השייך לדמות השחקן, ויש לנשק **עוצמה** חזקה יותר – על השחקן לבצע החלפה של הנשק שבידו לנשק החדש.
- אם **הנשק** מצריך **2 ידיים**, על השחקן לזרוק את **המגן** שבידו במידה והוחלט לקחת את הנשק.
- אם הפריט הוא **הגנת גוף** – השחקן אוסף את הפריט אם **יחס ההגנה** טוב יותר.
- אם הפריט הוא **מגן** – השחקן יאסוף את המגן רק אם אין בידו נשק המצריך **2 ידיים** ויחס ההגנה טוב יותר.
- אם הפריט הוא **שיקוי** – אם השיקוי מתאים לדמות השחקן, השחקן ישתמש בו מייד.

- אם השיקוי הוא חיים – יש לוודא שאין להגיע למצב בו יש יותר מ 100 חיים.
- כאשר ברדיוס המוגדר לשחקן נמצא אויב - השחקן מפסיק לזוז ובכל שלב של המשחק יש חילופי התקפות עד לחיסול אחד מהצדדים (בכל תור התקפה אחת משני הצדדים). לאחר שהאויב חוסל, השחקן ממשיך לנוע לכיוון נקודת היעד.
- עוצמת הפגיעה באויב – תחושב כעוצמת השחקן כפול עוצמת הנשק, אם אין לשחקן נשק עוצמת הפגיעה תהיה שווה לעוצמת השחקן בלבד.
- עוצמת הפגיעה בשחקן – תחושב כמכפלת עוצמת האויב באחוזי ההגנה של השחקן.
- כאשר כמות החיים של השחקן יורדת ל 0 (או מתחת ל 0) – השחקן מת , לא מבצע שום פעולה נוספת ואינו מופיע יותר בתצוגה.

פלט המשחק

בסיום המשחק, יש להדפיס לתוך קבצי CSV (לפי כמות קבצי הקלט) את המטריצה בקונפיגורציה הסופית בכל משחק (מצב הזירה בסוף).

בנוסף, יש להדפיס בכל קובץ, לאחר הדפסת המטריצה את הפרטים הבאים:
 לכל שחקן/אויב (גם אם הוא מת) – יש לרשום את דמות השחקן
 (Warrior/Archer/Wizard/Enemy/Elite) , את המיקום ואת כמות החיים שנותרה לו בסיום המשחק,
 כרשימה – שורה נפרדת לכל אובייקט, לדוגמא:

Enemy : Point : (8,8),Health : 100, Power : 2

Warrior : Point : (17,14),Health : 100, Power : 3.11

שמות קבצי הקלט יהיו עם סיומת in_ ושמות קבצי הפלט יהיו זהים לקבצי הקלט עם סיומת out_ באופן הבא, לדוגמא קובץ הקלט הוא:

g1_in.csv

קובץ הפלט יהיה:

g1_out.csv

- יש לוודא שהקלט/הפלט יהיו בדיוק בפורמט המבוקש.
- יש להסתכל בקובץ CSV המצורף למטלה.

דוגמא – קונפיגורציה התחלתית:

	A	B	C	D	E	F	G	H	I	J	K	L
1	world											
2	type	length	height									
3	matrix	15	15									
4	items											
5	type	quality	x	y								
6	mana	10	3		3							
7	health	13	2		9							
8	actors											
9	type	power	x_start	y_start	x_end	y_end	gender					
10	archer	3.11	1	1	13	14	1					
11	enemy	2	8	8	8	8						
12	weapons											
13	type	power	x	y								
14	sword	5	5	5								
15												
16												
17												
18												
19												

Sheet 1 / 1 Default Sum=14 100%

Enemy : Point : (8,8),Health : 100, Power : 2
 Warrior : Point : (1,2),Health : 100, Power : 3.11

בהצלחה!