

עבודה 4 - עקרונות תכנות מונחה עצמים

Tower Defence



מתרגלים אחרים: גיא דניאל, ונתן קינצ'לינגרא

הסבר כללי

במשימה זו תצטרכו למשג גרסה של המשחק Tower Defence בערת הממשק משתמש גרפי של JAVA - ממשק ה SWING. התכנית נבדקת על פי נוכנות, ציות לעקרונות תוכנת מונחה עצמים, שימושיות משתמש וארקטיביות. אתם יכולים להוסיף תוספות משלכם כל עוד עמדתם בכל הדרישות שיגדרו בתרגום. דרישות נוספות יכולות להעלות לכם את הציון ממופרט בסעיף הבונוס. סך הציון בתרגום יכול להגיע ל-115, כאשר 15 הם הנקודות של הבונוס. עלייכם להתחיל את המטלה בקריאה מסמך זה, ניתוח של הבעה ובבניה של תרשימים מחלקות (UML) המתאים, על פי הדרישות.

אם אתם לא מכירים את המשחק Tower Defence, ניתן לקרוא עליו ב[ויקיפדיה](#) כמו כן, תוכלו לשחק בגרסאות שונות של המשחק בקישור הבא:
<http://www.kongregate.com/tower-defense-games>

שים לב!

- הגראס בעבודה זו שונה מהגרסא המקורי.
- אין צורך להכיר את המשחק בגרסא המקורי בכך בלבד בצע את המטלה.
- **מומלץ מאוד (!!!) לקרוא את כל העבודה לפני תחילת המימוש.**
- מנסיון עבר, מלאיצים לכם לא לדלг על קריית החלק של הוראות ההגשה - הgesה לא לפ' המבנה הנכון תגרור הורדה משמעותית של נקודות.

פירוט המשחק

הסבר כללי (מוקייפדיה):

הגנת מגדלים (באנגלית: **Tower defense**) הוא ז'אנר משחק מחשב מהסוג אסטרטגי בזמן אמיתי אשר מורכב מלוח (עם שבילומפה). מטרת המשחק היא לנסות לעצור את מעברן של דמיונות עינויים מצד אחד של מפת המשחק לשנהו על ידי בניית מגדלים שיתקפו את הדמיות. לבנים שונים יש יכולות שונות, גם בהתאם לאיוביים השונים. הבחירה בסוג המגדלים ומיקומם במפה היא חלק מהאסטרטגיה החיונית המשחק. כמו שצווין לעללה, הגרסה אותה אתם תמצאו היא גרסה מצומצמת של המשחק המקורי בוצאות רבות ברוחן הרשת.

שביל : נקודת כניסה : נקודת יציאה

על המפה מסומן שביל עליי עוביים. לשביל נקודת כניסה שנמצאת הצד השמאלי במפה ונקודת יציאה שנמצאת הצד ימני של המפה. האויבים נכנסים לשביל, אחד אחרי השני, דרך נקודת הכניסה, ויוצאים דרך נקודת היציאה. האויבים נעים אך ורק בתוך השביל, בקצב קבוע ובכיוון אחיד (פרט למקרים מיוחדים; במקרים מסוימים לפעמים מגדלים גורמים לאויבים להתבלבל או לחטוף שוק, במקרים אלה הם עלולים לחזור אחורה בשביל או לעמוד במקום לאחר זמן כלשהו. אתם תמצאו סוג מסוים של חטיפת שוק, עליה מפורט בהמשך).

אויבים (Towers) : מגדלים (Creeps)

קיימים במסpiel מספר סוגי אויבים ומגדלים. כל אויב שנכנס למשחק מתחילה עם 100 חיים.¹ בזמן המשחק מגדלים תוקפים את האויבים, ויכולים להוריד להם חיים. אויב שהגיע ל-0 או פחות חיים מת ונעלם מהигра. מגדלים יכולים לתקוף אויבים שנמצאים למרחק מסוים מהם. מרחק זה מוגבל ומוגדר לכל סוג מגדל. השטח מסביב למגדל עד למרחק זה יקרא (לטובות ההסביר) **שטח האיום של המגדל**.

במשחק שלנו ישנו שני סוגי מגדלים: מגדלים שתוקפים אויבים יחידים ("מגדלים רגילים") ומגדלים שתוקפים טחחים ("כשchnim" - מהשורש C.O.C, מגדל שמכסח ולא פושח). **מגדלים רגילים** בעת תקיפה, בוחרים אויב יחיד שנמצא בשטח האיום שלהם ותוקפים אותו. לעומת זאת, **מגדלים כשchnים**, בכל תקיפה פוגעים בכל האויבים שנמצאים בשטח האיום שלהם. האסטרטגיה שלפיה מגדלים רגילים בוחרים אויב לתקוף בהתאם לבחירתכם. אסטרטגיה סטנדרטית היא לתקוף את האויב שנמצא מლפנים.

בעת פגיעה של מגדל באויב, כמות החיים שהאויב מאבד (אם בכלל) מוגדרת על פי סוג המגדל וסוג האויב. חלק מהמגדלים יכולים להשאיר אפקט על האויבים שהם תוקפים (במשחק שלם: הרעה/האטה, יפורט בהמשך).

לכל סוג אויב מוגדרת מהירותו שלו, ככלומר, כל כמה זמן האויב מתקדם. לכל מגדל מוגדרת מהירות תקיפה. ככלומר, כל כמה זמן המגדל יכול לתקוף את האויבים שנמצאים מסביבו.

ازהרה: אויבים עם מהירות גבוהה לא מורשים לעبور כמה משבצות בביטחון. ככלומר, אם, לצורך העניין, מימשتم את המשחק בצורה צו שאויב עם מהירות=1 עבר תא יחידה בשניה, אז אויב עם מהירות 2 עבר

¹: Health Points (a.k.a HP)

תא אחת כל חצי שנייה, ולא שתי משਬצות כל שנייה. באופן דומה, מגדלים עם מהירות תקיפה גבוהה לא תוקפים תקיפות מרובות בו-זמנית, הזמן בין תקיפה לתקיפה קטן. עבודות שיממשו תנוצה של אויבים מהירים בעזרת קפיצות גדולות של יותר מטה אחד, או מגדלים עם מהירות תקיפה גבוהה ע"י תקיפות מרובות בו-זמנית יאבדו נקודות.

ישנם ארבעה סוגי חובה של מגדלים (לבה, חז, רעל וקיסם) וארבעה סוגי חובה של אויבים (גול, מיק, אביר ונגי'). בנוסף, ישנם שלושה סוגי בונוס של מגדלים (סאם, גוקו ודינגו). כמות החיות וסוג האפקט שמגדל משאיר על כל סוג אויב מוגדרים בטבלה למטה. לצורך המימוש של פגיעת האויבים במגדלים אתם מתבקשים להשתמש ב-**Visitor Pattern** כפי שלמדתם בתרגול (דרישות שלד תוכלו למצוא בהמשך המטלה). פירוט על סוגי האויבים ומגדלים מופיע בטבלה בעמוד הבא.

מטריצת הפגיעה - השפעת מגדל על יצור

2	1	1	1	מהירות	סוג מגדל ← → מגדל	שטח האיים
נג'י (מהאגודה לתרבות הדיוו גברת ברכה) 	אביר 	מייק ווזובסקי 	גולן הגולגולת 	יצור ← → מגדל	סוג מגדל	שטח האיים
מוריד 15 חיים	מוריד 10 חיים	מוריד 15 חיים	מוריד 15 חיים	לבה 	כוחן	1
מוריד 30 חיים	אין השפעה	מוריד 30 חיים	מוריד 15 חיים	חץ 	רגיל	2
= מורעל עם חזק הרעללה = 1.5 למשר 5 שניות. לא מוריד חיים. מפורט מטה	מורעל עם חזק הרעללה = 2 = למשר 5 שניות. לא מוריד חיים. מפורט מטה	אין השפעה	מוריד 20 חיים	רעל 	רגיל	1
מוריד 10 חיים	מוריד 30 חיים	מוריד 10 חיים	מוריד 25 חיים	קופם 	רגיל	1
האטה למשר 3 שניות	האטה למשר 6 שניות	האטה למשר 3 שניות. מוריד 10 חיים	האטה למשר 3 שניות.	סאם (בונוס) 	כוחן	2
הורדת חיים מתגברת k=5 (מפורט מטה)	הורדת חיים מתגברת k=7 (מפורט מטה)	הורדת חיים מתגברת k=5 (מפורט מטה)	הורדת חיים מתגברת k=10 (מפורט מטה)	גוקו (בונוס) 	רגיל	2
מוריד 11 חיים	מוריד 8 חיים	מוריד 10 חיים	מוריד 15 חיים	דין המעופף (בונוס) 	רגיל	2

הבהרה: "אין השפעה" ממשועותה לא שהמגדל לא יכול לתקוף את האויב, אלא שכשהמגדל יתקוף את האויב לא תהיה פגעה שום ממשועות.

אפקטים מיוחדים של מגדלים על אויבים

מומלץ מאוד שהמשחק יומש בעדרת קוצב-זמן (ט'ימר) ראש. אנו מתייחסים לטיקים של הטימר בהגדרת משך ההשפעה של אפקטים מסוימים. הסבר מפורט יותר מסופק בהמשך (ראה "חובת מימוש").

כאשר אויב מושפע מאפקט כלשהו, האפקט נעלם רק לאחר משך זמן מסוים. משך זמן זה מוגדר לפי מספר הטיקים של הטימר שעוברים עד שהאפקט מבוטל באופן אוטומטי. כאשר אותו האויב נפגע כמה פעמים ע"י אותו סוג האפקט, משך האפקט לא מctrיבר אלא מתחזק - מתחילה מחדש (בין אם הסיבה אליו היא פגעה ע"י אותו המגדל כמה פעמים או פגעה מכמה מגדלים שונים). כמו כן, גם חזק האפקט **לא מctrיבר**.

לדוגמא (מומלץ לקרוא לאחר מעבר על סוג האפקט): אביר שמורעל בטיק הנוכחי, ולאחר שלוש שניות הוא מורעל שוב, יבריא בעוד 5 שניות ולא 7 (אלא אם יריעלו אותו שוב לאחר מכן, מה שיופיע שוב את הזמן עד ההבראה שלו). חזק ההרעלת נשאר 2 במשך כל הזמן הזה ואני משתנה.

הבראה מהרעלת לא משפיעה על האטה או ולהפר.

הרעלת

כאשר נגי או האביר נפגעים מגדל הרעל הם מורעלים. אויב שמורעל נהיה פגיע יותר למתקפות של מגדלים אחרים. ככלומר, כאשר אויב כלשהו מורעל, בעת פגעה הוא יאבד פי א' חיים מכפי שהוגדר לו במקור, כאשר א' הוא חזק ההרעלת כפי שצוין בטבלה. לדוגמא: אם נגי אינו מורעל, כאשר הוא נפגע מגדל הקסם הוא מאבד 10 חיים. אם הוא כן מורעל (עם חזק הרעל 1.5, כפי שצוין עבורי בטבלה הנ"ל) ונפגע מאותו המגדל, יאבד הוא 15 חיים.

האטה (בונוס)

כאשר אויב מועט, המהירות שלו יורדת לחצי מהמהירות המקורי שהוגדרה עבורי.

הורדת חיים מתגברת (בונוס)

למגדל גוקו יש יכולת להתרחב עם הזמן. תחילתו, מגדל הגוקו מאותחל עם פרמטר פגעה $H = 1$. כמות החיים שיורדת לאויב בעת פגעה של מגדל גוקו, היא $K \cdot H$, כאשר K הינו שונה עבור כל אויב והוא מוגדר בטבלה לעיל.
בכל 10 פגיעות שמגדל הגוקו מבצע, פרמטר הפגעה H מוכפל פי 2.

דינו המעופף (בונוס)

דינו המעופף הוא מגדל מיוחד אשר מעופף ברוחבי הלוח ויכול לפגוע בכל אויב הנמצא בשטח האיים שלו. מסלול התעופה של דינו נתון לשיקולכם (אפשר שיעוף במעגל, במסלול רנדומלי וכו').

אפשר לראות דוגמא ממשחק אמיתי בסרטון הבא (שים לב למטרו הירוק מצד שמאל):

<https://youtu.be/A74EInRvQsg?t=1m54s>

משתמש

המשתמש מתחילה את המשחק עם 20 חיים. כל פעם שהוא עבר את נקודת היציאה, השחקן מאבד נקודה אחת מהחיים (זהה לכל סוג האויבים). אם וכאשר נגמרו למשתמש החיים, הוא מפסיד במשחק. במשך כל ריצת המשחק, על המשתמש לראות כמה חיים נותרו לו.

גלים (Waves)

האויבים מגיעים בגלים. משחק אחד מורכב מחמישה גלים. בכל גל יש לחות קבוצה של אויבים למסלול (בזה אחר זה, מפורט בהמשך כיצד), ולאחר מכן שכל האויבים של הגל הנוכחי מתו או יצאו מהלוח, המשתמש יכול לקרוא לגל הבא. כאשר אין עוד גלים, אם למשתמש לא נגמר החיים, הוא ניצח.

הליך המשחק

כאשר המשחק נפתח, מוצגים למשתמש מספר מסלולים ביןיהם הוא יכול לבחור. לאחר שהמשתמש בחר במסלול נפתח לו מסך עם המסלול הנ"ל. תחילת המשחק בוחר הינו מקום את מגדליים שלו. לאחר מכן המשמש קורא לגל הראשון לצאת ע"י לחיצה על כפתור מתאים. לאחר שנגמר גל, המשתמש יכול להוסיף מגדליים על הלוח ולקורא לגל הבא לצאת, שוב פעם ע"י אותו הכפתור. המשחק נגמר כאשר נגמר חמשת הגלים או כאשר המשתמש הפסיד, כמוין לעילו. לאחר שנגמר המשחק המשתמש חוזר למסך בחירת המסלול.

חוובת מימוש - דרישות וUO, שלד ועיצוב המחלקות

הסבר כללי

בכדי למשוך את המשחק נדרש מכמם להתייחס ללוח כמטריצה דו-מימונית (כלומר שהמודל יהיה מיוצג ע"י מטריצה דו-מימונית - מפורט מטה הסבר עבור המימוש). כל נקודה בלוח יכולה להיות דשא או שביל. מגדים יכולים למקם על הדשא בלבד. אובייקטים נעים בשביל בלבד ובפרט מנוקודת הכניסה לכיוון נקודת היציאה. על כל מקום בלוח אשר מיוצג במודל ע"י שביל, יכולים לעמוד מספר אובייקטים. שני מגדים שונים לא יכולים לעמוד על אותו התא.

המלצת: לוח המשחק יהיה בגודל 800X800 פיקסלים וכל תא במטריצה יוצג ע"י 25 פיקסלים מרובעים, ובכך אנו מייצרים לוח (מטריצה) בגודל 32X32 תאים. מבנה הנתונים של הלוח ידוע למפות מיקומי ע,א של המטריצה לע,א במסך (ברמת הפיקסלים) ולהפוך.

המרה מפיקסל לנקודה על המטריצה:

$$x_{matrix} = [x_{pixel} * 32] / 800$$

$$y_{matrix} = [y_{pixel} * 32] / 800$$

[x] → עיגול של x לפני מטה

דוגמה 1:

נקודה על המסך (פיקסל) (475,530) ממופת לנקודה [19,21] על המטריצה (לוח).

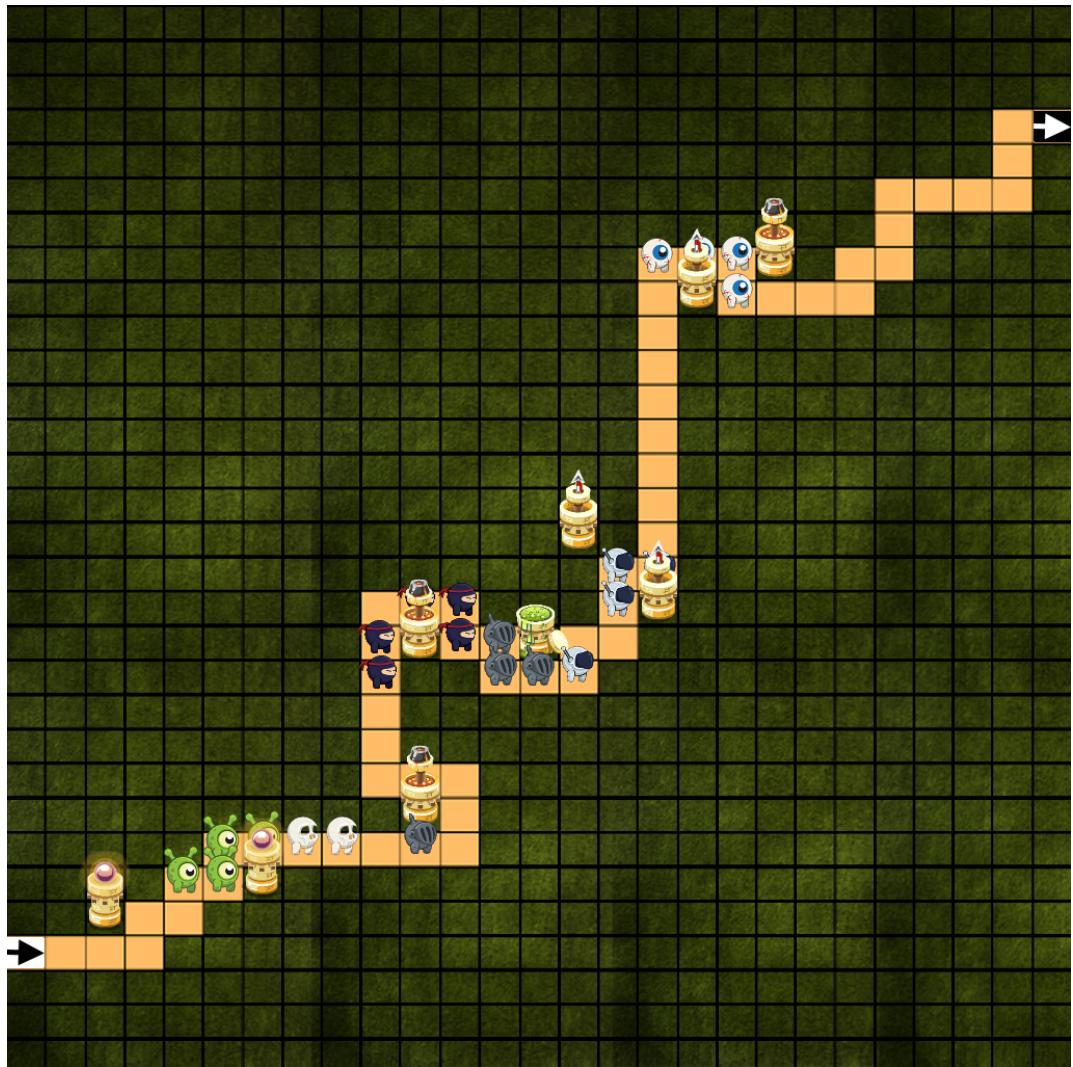
דוגמה 2:

הנקודה השמאלית העליונה ביותר בלוח, תהיה מיוצגת במודל שלו (במטריצה) ע"י הערכים [0,0] ונקודות

אפשריות על המסך (פיקסלים) אותן היא תתאר בView תהינה

$\{ \{0, 1, \dots, 25\}, \{0, 1, \dots, 25\} \} \in y, x | (y, x) \in \{0, 1, \dots, 25\}$ (כל האים בין 0 ל-25 וכל העים בין 0 ל-25,

.(800/32)



- * האסטרונואוט אינו חלק מהמשחק בפועל ומופיע לשם הדגמה בלבד
- * המשחק שלכם לא חייב להראות כמו בתמונה הנ"ל, יש לכם חופש לבחור איך לעצב אותו. make it pretty

מימוש ובנייה המסלול

מימוש המסלול

בכדי למש את המסלול עליו להתייחס ללוח שלכם כמטריצת כיוונים, כאשר מטריצת הכוונים תוגדר בצורה הבאה: כל תא במטריצה מכיל 2 מספרים (אפשר לומר שכל נקודה היא וקטור בעל שני ממדים). כל אחד מהמספרים נבחר מטווח הקבוצה $\{1, -1, 0\}$. כאשר אויב דורך על תא שהווקטור שלו שווה ל-(v_u) אויב יזוז לכיוון u על ציר x -ו ולכיוון v על ציר y -ו.

(אפשר למש את הלוח ע"י מס' מטריצות - כיוונים, מיקומים של אויבים וכדומה)

לדוגמא:

אויב שנמצא על תא u וקטור כיוונים $(1, 0)$ יצד ימינה, אויב שדורך על תא u וקטור כיוון $(0, -1)$ יצד לתא שמעליו וכדומה (כשמדוברים על משק גרפי של מחשבים, בדרך כלל מייחסים לראשי הציגים את הפינה השמאלית העליונה של המסך, כלומר ציר x גודל ימינה וציר y גודל לפני מטה. זה נכון גם לקומפוננטות הגרפיות של Swing).

לפניכם מערכ דו-מיידי המתאר את הלוח שנראה לעדלה, כאשר הפינה השמאלית של השביל היא נקודת ההתחלה והפינה הימנית היא נקודת הסיום.

בחירה נקודת הכניסה והיציאה, ובניית המסלול

נקודות הכניסה של המסלול ח'יבת להופע בעמודה השמאלית ביותר של הלוח, וכן' היציאה בעמודה הימנית ביותר.

קידוד יدني של המסלולים

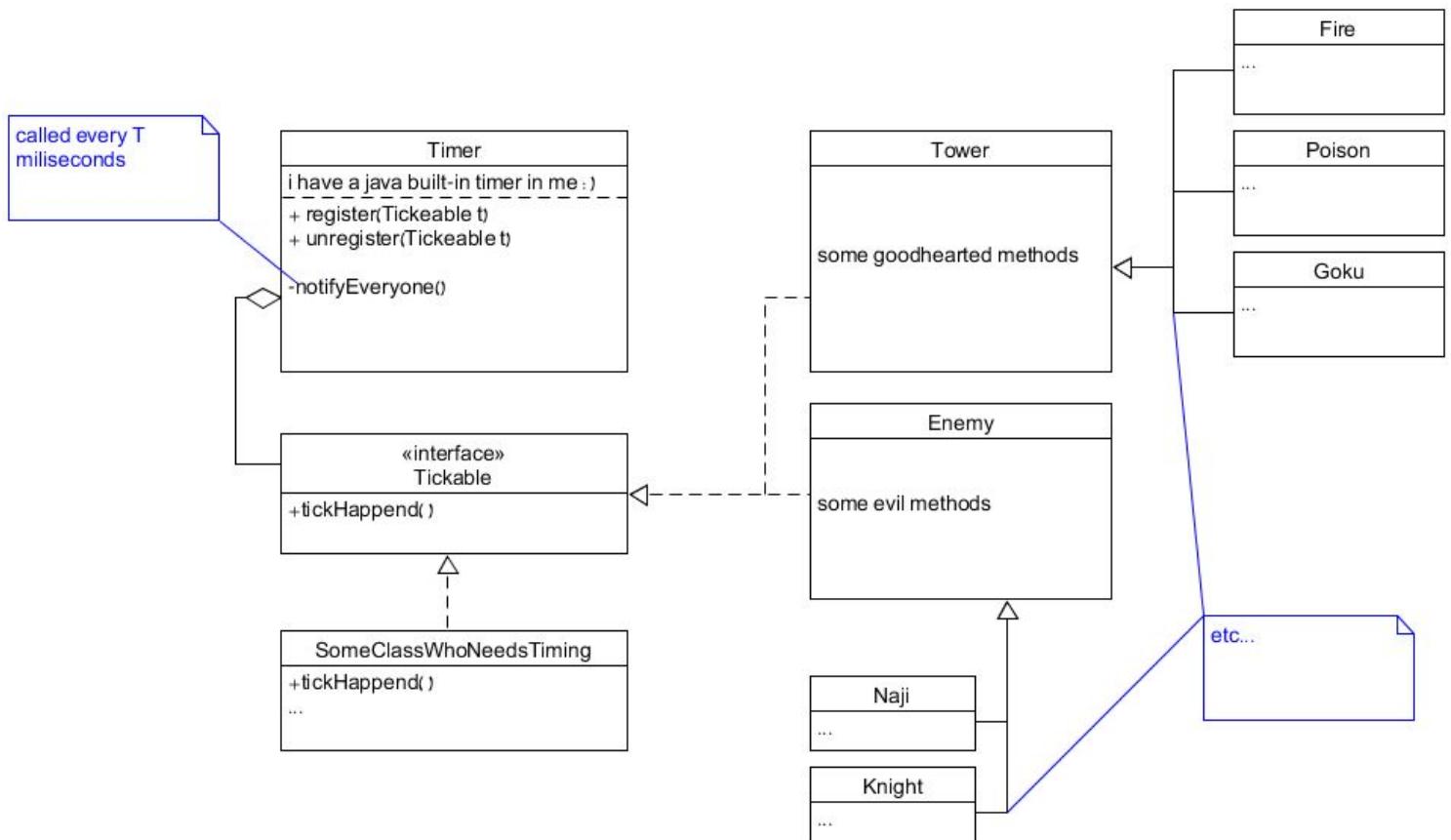
אפשר לקודד מראש את המסלולים. חיבים להיות במשחק לפחות שלושה מסלולים מוכנים מראש.

שימוש בטיימר

כדי למשם את העבודה צטרכו להשתמש בטי'רים. מומלץ מאוד שתשתמשו בטי'ר ייחד משלכם שעובדים בדונה לחומר observer שהוזכר בחלק מהתרגולים (לא חלק מהחומר הקורס, אל תdaggo אם אתם לא מכירם). יש מחלוקת טי'ר מובנת בשפת Java (הראנו אותה בתרגול על Animations), אך מומלץ לכתוב מחלוקת טי'ר משלכם לצורך המשחק (שתשתמש בטי'ר של Java בתוכה). הטי'ר הזה ישמש אתכם לצורך זמן תנועה ותקיפות של אויבים ומגדלים בהתאמה. המגדלים אויבים ירשמו אצל הטי'ר לקבלת Events של הטיקים שלהם. לצורך הסבר מפורט יותר של תבנית העיצוב חפשו Observer Design Pattern באינטרנט.

מגדל עם מהירות תקיפה = 1 צריך לתקוף כל שנייה; אויב עם מהירות תזוזה = 1 עובר משכט כל שנייה.

תרשים מחלקות חלקו שיעזר להבין את הרעיון הכללי של השימוש בטימר:



בחירה מסלול

אתם נדרשים למשמש את חלון בחירת המסלול, המסר צריך להכיל לפחות שלושה מסלולים (שאינם היכנסו מראש) בינויהם המשמש יכול לבחור.

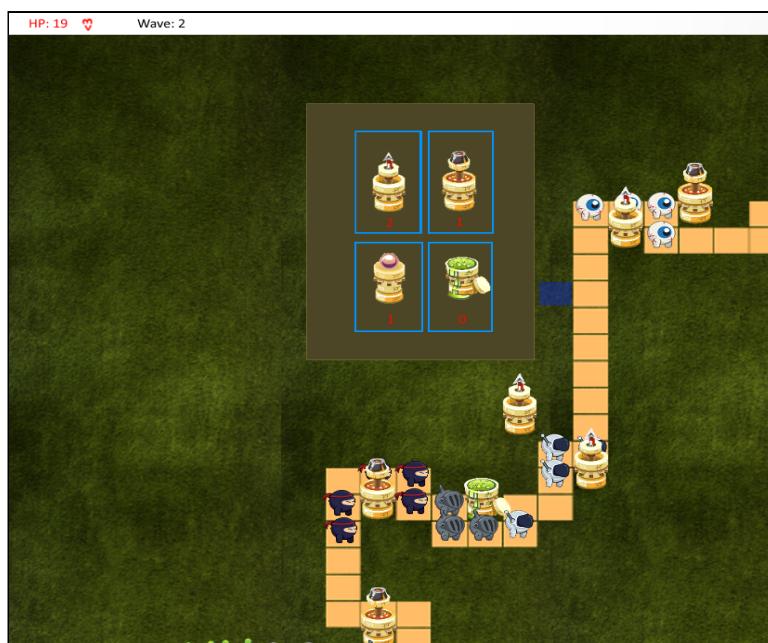
כניסת אויבים למסלול

לפני כל גל, המשחק יגריל את האויבים שיכנסו אחד אחריו השני למסלול לפי החוק הבא:
בגל הראשון - אויב אחד מכל סוג (סה"כ 4 אויבים).
בגל השני - שני אויבים מכל סוג (סה"כ 8 אויבים).
בגל השלישי - ארבעה אויבים מכל סוג (סה"כ 16 אויבים).
בגל הרביעי - שמונה אויבים מכל סוג (סה"כ 32 אויבים).
בגל החמישי - שישה עשר אויבים מכל סוג (סה"כ 64 אויבים).

האויבים נכנסים בסדר רנדומלי למסלול, ככלmor לא בקבוצות לפי הסוג שלהם.

מיקום המגדלים

יש לאפשר למשתמש למקם מגדלים על הלוח. כאשר המשתמש לוחץ על מיקום חדש יקפו צו לוחן עם אפשרות למגדלים שהוא יכול לבנות. המשתמש בוחר את המגדל הרצוי ומגדל נבנה בתא הנבחר. המשמש יוכל לבחור להניח על הלוח 3 מגדלים מכל סוג (חוץ מגדלי הבונוס שהמשמש יוכל לשים רק אחד מכל סוג).
הברורה: כאשר המשתמש בוחר מיקום לדינו המופיע, המיקום אותו בחר יהיה נקודת ההתחלה של דינו.
שיםו לב: עירכת הלוח תאפשר רק לפני תחילת כל גל ולא באמצעות הגל עצמו.
תמונה להדגמה:



פגיעה ממתקפה של מגדל באויב

כאשר מגדל תקף אויב מסוים, עליכם להציג זאת ב GUI שיהיה ברור איזה מגדל תקף איזה אויב (כדי להקל עליכם, אין צורך להראות איזה מגדל בדיקת תקף, מספיק שיהיה מובן סוג המגדל). בחירת אופן ההציגה נתונה לבחירתכם.

כמה אפשרויות לדוגמא:

1. הציג היריה של המגדל עד אשר פוגעת באויב
2. שינוי צבע לתא במסלול עליו עומד האויב שנפגע
3. שינוי צבע לאויב עצמו שנפגע
4. ועוד ועוד...

הילוך מהיר

אתם נדרשים למשמש כפטור הילוך מהיר במשחק. כאשר המשתמש ילחץ על הכפתור, כל האויבייקטים על המסך ינעו פי 2 מהר יותר. לחיצה חוזרת על הכפתור תחזיר את מהירותם הלווי למחרות הרגילה. ניתן למשש זאת בנסיבות ע"י מחלוקת הטימיר שלכם - חישבו כיצד.

שליחת הגל הבא ותחילת המשחק

אתם נדרשים למשמש כפטור שגורא לגל הבא. הכפתור חסר השפעה כל עוד הגל הנוכחי לא הסתיים. סך הכל במשחק ישנים חמישה גלים. בסוף הגל החמישי, במידה והמשתמש לא הפסיד לפני כן, המשחק נגמר, מופיעחלון סיכום משחק ולאחריו המשתמש חוזר למסך הראשי.

דרישות ממשק

בכל שלב, יוצג המידע הבא בתוך Toolbar:

- כמות החיים של השחקן.
- מספר הגל הנוכחי.
- זמן מתחילת המשחק (בשניות). שימו לב, כאשר מהירות המשחק היא פי 2, גם הזמן צריך להתעדכן פי 2 יותר מהר.

כאשר מסתiem המשחק (כלומר כל חמשת גלי האויבים נגמרים), על התכנית להציג למשתמש סיכום של המשחק:

1. כמות חיים שנותרה
2. כמות האויבים שעברו את נקודת הסיום
3. כמות האויבים שחוסלו
4. זמן (בשניות) מתחילת המשחק

במידה והשחקן מפסיד, יש להראות לו גם את מסך הסיכום.

בנוסף, כאשר משתמש לוחץ על מגדל, יש להציג לו על המסך את שטח האיים של מגדל זה. כמו בדוגמאות שלහן:



צד ימין: שטח האיום של מגדל ה'חץ'
צד שמאל: שטח האיום של מגדל הרעל.

אין דרישת לגבי הצבעתו יש לצבע בשטח האיום של כל מגדל, אפילו זה נתון לשיקוייכם.

שימוש ב Visitor Pattern

אתם נדרשים להשתמש בתבנית המארח (Visitor Pattern) שנלמדה בתרגול בכדי למש את סוג הפגיעה השונות של המגדלים באוביים, כפי שצוין לפני כן.

על התוכנית שלכם להכיל את המחלקות הבאות והיחסים ביניהם:

Visitor

ממשק המכיל חתימות לפונקציות בהתאם לתבנית המארח

Tower

מחלקה (רגילה, ממשק או מחלקה אבסטרקטית לבחירתכם), אשר מממשת (או מרחיבה) את הממשק **Visitor**

Creep

ממשק או מחלקה אבסטרקטית לבחירתכם, המכיל את הפונקציה:

`public void impact(Visitor v); /*symbolizes the method "accept" in Visitor pattern*/`

לוגיקת מפגש תקיפה של מגדל ואובי (תבנית המארח):

על הפונקציה `impact` לבצע את מימוש הלוגיקה של מפגש בין תקיפה של מגדל לאובי (על פי החוקים המוצגים מעלה). בשימוש זה **אסור להשתמש** ב casting או באופרטור `instanceOf`. כלומר עליהם להשתמש בתבנית המארח (pattern **Visitor**).

רמז: כל מחלקה הירושת מ **Tower** חייבת לדרוּס את `impact` באופן הבא:

`public void impact(Visitor v){ v.visit(this); }`

חוות דעתך זו, מאפיינת את חתימות הפונקציות במשק **Visitor**.

תמונה סופית של המשחק

(לצורך הדגמה בלבד!)



הגשה וניקוד

בונוס

- סך כל נקודות הבונוס בעבודה הוא 15. הבונוס מחולק כדלהלן:
- 4 נקודות עלימוש מגדל ההאטה (סאם) ומגדל גוקו.
 - 4 נקודות עלימוש דינו המעוופף.
 - 7 נקודות על שמירת טבלת ניצחונות.

ו' כאשר משחק נגמר, המשמש יכול להחליט האם הוא מעוניין לשמר את המשחק שלו בטבלת הניצחונות, ע"י הכנסת שמו ולחיצה על כפתור אישור. בנוסף לכך, בתפריט הראשי של המשחק המשמש יוכל לראות את 5 השחקנים המובילים (דרוג השחקנים מחושב לפי כמות החיים שנשארה לו בסוף המשחק והזמן שלקח לו לסיים את המשחק. נוסחה פשוטה יכולה להיות $time * h(p - 20)$). שימוש לב - כאשר שחקן מסיד, אין אפשרות להציג לטבלת הניצחונות.

שימוש לב: טבלת הניצחונות צריכה להיות מוקדדת לתוך קובץ בפורמט שתבחרו. כך שבטעינת המשחק, יוכל לראות את השחקנים האחרונים ששיחקו ובפרט את אלו אשר שיחקו בטעינות קודמות של המשחק.

שימוש לב! חובה לכל מי שמשמש את החלק של הבונוס להוסיף קובץ `axt.suson` ולכתוב בתוכו את המידע על שימוש הבונוסים אותם מימשטים. (לא יותר מ 150 מילימטרים)

שאלות

שאלות על העבודה נא לשאול בפורום המועד לעבודה, ואני נשתדל לענות עליהם בהקדם האפשרי.

המלצות:

1. **משאבים:** לעובדה זו נדרש ארכיבון (קובץ ZIP) עם קבצי תמונות של האויבים והמגדלים. לכל אויב (וגם לדינו המעוופף) ישנו 2 תמונות, כדי שתוכלו לייצר מהם אнимציה כאשר הם צדדים על הלוח. אין זו חובה לעשות את האנימציה אך אנו ממליצים על כך בחום.
2. כדאי לכם להשתמש ב `double-buffer` (נלמד בתרגולים) כדי שלא תחו הבהובים של לוח המשחק שלכם אחריו כל `repaint`.
3. לוח המשחק עצמו - מומלץ להשתמש בקומפוננטה גרפית יחידה, ועילה לציר את כל האובייקטים (כולם, שמחלקה אחת עליונה - `frame` או `board` או `frame` ועוד), יכלו פעולה `g` (`paint(Graphics g)`, וישרשו את הקראיה לכל האויבים/מגדלים/יריות לא משנה מה עם אותו אובייקט `graphics`).
4. כדי שתוכלו להפעיל את המשחק עם קובץ `jar` ללא צורך שקבצי המדייה יהיו חיצוניים אליו, תוכלו להשתמש באחת האפשרויות המוצגות כאן:
<https://stackoverflow.com/questions/25635636/eclipse-exported-runnable-jar-not-showing-images>

הגשה:

החולן הראשי של התוכנית הגרפי צריך להרחיב את מחלקת JFrame (כפי שראינו בתרגומים).

- עליכם להגיש את הקבצים הבאים בתוך קובץ `zip.Y_X` (כאשר `Y,X` הם תעודות זהות של המגנים מופרדים ב`_`) ללא תיקיות:
1. [15 נקודות]. דיאגרמת מחלקות (UML) בתוך קובץ `pdf` בשם: "hw4uml.pdf". ניתן להויסף לקובץ מידע שלדעתכם יהיה חיוני שננדע על המימוש שלכם.
 2. [85 נקודות]. קוד המקור והקוד המkomפלי בתוך קובץ `jar` בשם `jar4hw`.
 - a. אנה לבדוק שאתם מצlichים להריץ את קובץ `jar` שלכם משורת הפקודות (cmd).
 3. [15 נקודות בונוס] קובץ `txt` bonus עם כל המידע על מימוש הבונוסים.
- הערה: לגבי קבצי מדיה (תמונות ועוד): עליכם לוודא שקבצי המדיה נטען כmo שציר בהרצה ה `jar`, אם אין אפשרותהם לטעון את הקבצים ישירות מתוך קובץ `jar` ניתן לשים את הקבצים הנ"ל מחוץ ל `jar` בתוך קובץ ה – `zip` .

תריצו את קובץ `jar` לראות שהכל עובד, ועל תסתמכו רק על זה שהמשחק עובד לכם בIDE** !**

חשוב מאד!!!

יש להגיש את העבודה **בדיקת** לפי הוראות הגשה! עבודה שתוגש לא לפ' ההוראות, תגרור הורדת נקודות משמעותית.

פתח ציון:

נקודות	דרישות
15 נק'	דיאגרמת מחלקות (UML) בתוך קובץ <code>pdf</code> בשם: "hw4uml.pdf".
25 נק'	שימוש של התנהגות הלוח (הצבת מגדים, תזוזת אויבים, תקיפה, פגעה)
20 נק'	שימוש בתבנית המארח וблיך התוכנית
15 נק'	שימוש של חוקי פגעה של מגדים באויבים
5 נק'	מצגה של מידע למשתמש בסיום של שלב ובמהלכו
15 נק'	שימוש של כל 5 הגלים והציגם לשחקן בתפריט הראשי
5 נק'	שימוש נכון בטימר
בונוס	
4 נק'	שימוש מגדל האטה ומגדל גוקו
4 נק'	שימוש דיננו המעוופף
7 נק'	הוספה של טבלאת ניצחונות אחרונים



בצלחה!