

~~PMIC~~

JAVA

האוניברסיטה העברית - החוג לסטטיסטיקה
מבוא לשימושי מחשב א' (52299) - מבחן מועד ב' תשנ"ט

זמן - שעתיים כל חומר עזר מותר בשימוש. מורים: כנעני, פינשטיין, קרן.
תאריך המבחן: 28.3.99

לפניך **applet** של המשחק "צוללות" - במשחק זה נתונה טבלה בגודל 10 שורות ו 10 עמודות ("המפה") ובה מפורטים מספר כלי שייט, כלי השייט הינם באורך משתנה (גודל 1 או 2 בלבד), כאשר תא בטבלה תפוס במפה ע"י כלי שייט הוא מסומן ע"י 1, תא לא תפוס במפה (ים) הוא מסומן ב '0'. כלי שייט בעל גודל 2 תופס 2 תאים רציפים האחד ליד השני (לא באלכסון).
חשוב - כלי השייט מפורטים על גבי המפה כך שאף כלי שייט אינו נוגע בכלי שייט אחר, גם לא באלכסון (ראו דוגמא בהמשך) *כאן נרשם ונרשם שייט "גזא גזא" הגפה!!*

לרשות השחקן עומד 'תותח' עם 50 פגזים, מטרתו היא לנסות ולהטביע כמה שיותר כלי שייט. בכל סיבוב המשחק מעביר קואורדינטה לירי (מספר שורה ומספר עמודה). תוצאת הירי מעודכנת ע"ג ה'מפה' ובנוסף מקבל השחקן מסר מילולי על מהות הפגיעה - אין פגיעה, יש פגיעה בכלי שייט, יש פגיעה וכלי השייט טבע.

בתחילת המשחק השחקן רואה לפניו מפה ריקה. עם התקדמות המשחק נחשפים לפניו האזורים במפה אליהם הוא כבר 'ירה'.

המשחק מסתיים לאחר 50 סבבים (50 יריות), שחקן טוב יותר הוא זה אשר הצליח להטביע כמה שיותר כלי שייט.

ה **applet** עצמו מכיל את הממשק של השחקן (2 שדות קלט לקואורדינטות, כפתור 'אתחול' וכן 3 שדות טקסט למסרים ולהצגת מספר הסיבוב). בנוסף משתמש ה **applet** באובייקט בשם **map1** מהמחלקה **Map** אשר מכילה את כל השיטות הרלוונטיות למשחק.

המחלקה **Map** עצמה מכילה 2 פרטי מידע הרלוונטיים למשחק:
מערך דו מימדי של מספרים שלמים בשם **table** בעל גודל **10x10** המתאר את ה'מפה' עצמה וכן משתנה שלם בשם **sunk** המחזיק את מספר כלי השייט שהוטבעו עד עתה (כלי שייט שהוטבעה הינו כזה שכל חלקיו נפגעו!)
התאים של המערך הדו מימדי יכולים להכיל את הערכים הבאים בלבד -

- 0 - תא ריק (ים)
- 1 - תא מאויש ע"י כלי שייט (לא נפגע)
- 2 - תא בן היה חלק של כלי שייט שנפגע מפגז.
- 3 - תא ריק (ים) שנורה אליו פגז באחד הסיבובים הקודמים.

המחלקה מכילה את השיטות הבאות (חלקן כבר כתובות) -

- שיטה בנאית **constructor** - המאפסת את **sunk** ל 0 (כאשר מתחילים משחק חדש) (כבר קיימת).
- שיטה **public void print(Graphics g)** - אשר מציירת את המפה ע"ג ה **applet**. (כבר קיימת)
- שיטה **public int getsunk()** - המחזירה את מס' כלי השייט שהוטבעו.
- שיטה **public int total_hits()** - המחזירה את מס' הפגזים שפגעו בכלי השייט.
- שיטה **public int shot(int r,int c)** - אשר מקבלת את נתוני הירי מהשחקן ואשר מחזירה אחד מ 4 מספרים -

- (-2) - כאשר נתוני הירי אינם תקינים (מחוץ ל'מפה') או שהפגז נורה למקום שכבר הופגז לפני כן.
- (-1) - כאשר הפגז נורה לנקודה שבה אין כלי שייט.
- 1 - כאשר הפגז פגע בכלי שייט.
- 2 - כאשר הפגז גרם להטבעת כלי השייט.

ענה/על 4 השאלות הבאות :

1. כתבו את השיטה `public int getsunk()` במחלקה `Map` : `getsunk` צריכה להחזיר את הערך שנמצא במשתנה `sunk` . (10 נקודות)
2. כתבו את השיטה `public int total_hits()` במחלקה `Map` : `total_hits` סופרת סה"כ את מס' הפגמים שפגעו בכלי השייט (תאים המסומנים בערך 2 במערך הדו מימדי `table`) . (15 נקודות)
3. כתבו את השיטה `public int shot(int r, int c)` במחלקה `Map` , השיטה מקבלת את קואורדינטות הירי (מס' שורה ומס' עמודה) ומחזירה אחד מ 4 מספרים לתאור תוצאת הירי (ראו למעלה) , במידה והפגע פגע בכלי שייט עליכים לבדוק האם כלי השייט 'טבע', אם כן עליכם לקדם את המשתנה `sunk` (מונה ההטבעות) ב 1 . (40 נקודות)
 * לא יורא - אקצוק האם כלי השייט טבע איכס אקצוק איר האכסיה כ 4 טננין טא האג - שימו אז לא אהוה
4. כתבו את השיטה `public void actionPerformed(ActionEvent e)` ב `applet` - `applet` האקולר האשק !!
 א. השיטה צריכה להתייחס לאפשרות שהמשתמש לחץ על כפתור האתחול (`reset`) , במקרה זה עליה לבנות מחדש את האובייקט `map1` , לאפס את מונה הסיבובים `round` , לנקות את שדות הקלט והסקסט הרלוונטיים (13,14) ולרשום את מס' הסיבוב מחדש בשדה 15 .
 ב. במידה והארוע הגיעה מהשדה `col` - הרי השחקן 'יורה' פג , השיטה צריכה לפנות לשיטה הרלוונטית ב `map1` , להציג את תוצאות הירי בשדה 14 , לעדכן את מס' הסיבוב (שדה 15) , לנקות את שדות הקלט ולבסוף לגרום להדפסת המפה המעודכנת. במידה והיתה שגיאה בנתוני הירי - השיטה תציג הודעה רלוונטית בשדה 13 (שימו לב שלא לעדכן את מס' הסיבוב במקרה זה).
 ג. במידה ועברו 50 סיבובים , השיטה תדפיס בשדה 13 את מס' כלי השייט שהוטבעו , בשדה 14 את מס' הפגיעות ובשדה 15 הודעה שעל מנת לשחק משחק חדש יש ללחוץ על כפתור ה `reset` . (35 נקודות)

בהצלחה

<pre>import java.awt.*; public class Map{ int sunk; int[][] table = { {0,0,0,1,1,0,1,0,0,0}, {1,0,0,0,0,0,1,0,1,0}, {1,0,1,0,0,0,0,0,1,0}, {0,0,0,0,1,0,0,0,0,0}, {1,0,0,0,0,0,1,0,1,1}, {0,0,0,1,1,0,0,0,0,0}, {0,1,0,0,0,0,1,0,0,0}, {0,0,0,1,1,0,0,0,1,0}, {1,1,0,0,0,0,0,0,0,0}, {0,0,0,1,1,0,0,1,1,0} }; public Map(){ sunk =0; } public void print(Graphics g){ } }</pre>	<pre>int rounds; Map map1 ; Button reset ; Label l1 , l2 , l3,l4 ,l5; TextField row,col; public void init() { round =1; map1 = new Map(); resize(300,500); row = new TextField(2); col = new TextField(2); l1 = new Label("row "); l2 = new Label("col "); l3 = new Label(""); l4 = new Label(""); l5 = new Label("round: "+round); reset = new Button("restart game"); col.addActionListener(this); reset.addActionListener(this); } public void paint (Graphics g){ map1.print(g); }</pre>
<pre>import java.awt.*; import java.awt.event.*; import java.applet.*; public class battleShip extends Applet implements ActionListener;</pre>	

Applet Viewer: battleShip.class

Applet

restart game

row

col

	1	2	3	4	5	6	7	8	9	10
1	*	*	*	*	*	*	S	~	*	~
2	S	*	*	*	*	*	*	*	*	*
3	~	~	*	~	*	*	*	*	*	*
4	*	*	*	*	*	*	*	*	*	*
5	*	*	*	*	*	*	*	*	*	*
6	*	*	*	*	*	*	*	*	*	*
7	*	*	*	*	*	*	*	*	*	*
8	*	*	*	*	*	*	*	*	*	*
9	*	*	*	*	*	*	*	*	*	*
10	*	*	*	*	*	*	*	*	*	*

no hit, shoot again!

round: 8

Applet Viewer: battleShip.class

Applet

restart game ← reset

row ← row

col ← col

האמה

	1	2	3	4	5	6	7	8	9	10
1	*	*	*	*	*	*	S	~	*	*
2	S	*	*	*	*	*	*	*	*	*
3	~	~	*	~	*	*	*	*	*	*
4	*	*	*	*	*	*	*	*	*	*
5	*	*	*	*	*	*	*	*	*	*
6	*	*	*	*	*	*	*	*	*	*
7	*	*	*	*	*	*	*	*	*	*
8	*	*	*	*	*	*	*	*	*	*
9	*	*	*	*	*	*	*	*	*	*
10	*	*	*	*	*	*	*	*	*	*

a vessale was hit, shoot again!

round: 7

← 13 מאי
← 14 מאי
← 15 מאי

battleShip Applet 1 ואלו 2

הגדרת המספר -
* - לא יורד
~ - נ
S - ש