האוניברסיטה העברית המכינה לתלמידי חו"ל

מבוא למחשב - מבחן מועד א' תשנ"ט

זמן 3 שעות כל חומר עזר מותר בשימוש. מורה אורי קרן תאריך 28/6/99

לפניך applet של המשחק MANCHALA זהו משחק בן מאות שנים בעל וריציות שונות שמקורו באפרי'קה, המשחק משוחק גם בדרום מזרח אסיה בווריציות שונות.

המשחק בנוי מלוח שבו 12 גומות (בורות) המסודרים ב 2 שורות האחת מול השניה , שני השחקנים יושבים האחד מול השני כאשר הלוח נמצא בינהם. בתחילת המשחק בתוך כל גומה ישנם 4 אבנים, בכל תור לוקח השחקן את כל האבנים מגומה מסויימת הנמצאת בצד שלו (צד ה"בית") ומתחיל לפזר אבן אבן בכל אחת מהגומות שמימין לגומה שממנה נלקחו האבנים, פיזור האבנים נעשה בניגוד לכיוון השעון בצורה מעגלית עד שנגמרות כל האבנים שהוצאו המגומה המקורית.
במידה והאבן האחרונה (מאלו שנלקחו מהגומה) מושם (נכנס) לגומה בצד של השחקן השני (אחת מ 6 הגומות שמול השחקן שמשחק בתור זה) , כאשר הגומה מכילה אבן אחת או שתי אבנים , מורדות האבנים מהגומה האחרונה יחד עם האבן האחרונה לרשותו של השחקן האבנים מהוות נקודות לזכות השחקן ששיחק כעת.
בכך גם בא לסיום התור , התור מועבר לשחקן השני במידה ובצד שלו ישנה לפחות גומה אחת עם אבן אחת ויותר.
במידה וכל הגומות בצד של השחקן רייקות , נשאר התור בידי השחקן שסיים את תורו עכשיו.

ניצחון מושג כאשר אחד השחקנים צובר מעל ל 24 אבנים.

ה applet של המשחק משתמש במחלקה game על שיטותיה השונות . המחלקה שומרת את כל האינפורמציה הרלוונטית למשחק

int[] holes; - מערך הגומות
int current_player; - השחקן הנוכחי
int[] score; - מערך לשמירת התוצאה (כמה אבנים כל שחקן הצליח לאסוף)
int left; - כמה אבנים נשארו עדיין על הלוח

למחלקה השיטות הבאות

public void print_table(Graphics g, int x, int y) שיטה אשר מציירת את המצב הנוכחי של לוח המשחק.
public int play(int hole) שיטה לחישוב השלב הבא במשחק , הפרמטר המועבר לשיטה הוא מס' הגומה שהשחקן הנוכחי בחר לשחק.
Public int get_score1() מחזירה את מס' האבנים ששחקן 1 צבר.
Public int get_score2() מחזירה את מס' האבנים ששחקן 2 צבר.
Public int get_player() מחזירה את מספרו של השחקן הנוכחי (1 או 2)
בנוסף קיימת השיטה הבונה (contructor) - game() המאתחלת את האובייקט של המשחק.

ענה על 3 השאלות הבאות :

1. כתוב את 3 השיטות get_score1, get_score2, get_player כך שיחזירו את הערכים המצויים במערך הניקוד (score[]) , ואת הערך שב current_player . (10 נקודות)

2. כתבו את השיטה play המקבלת את הגומה שממנה נלקחים האבנים בתור הזה , השיטה יכולה להחזיר את הערכים הבאים -
   1- כאשר השחקן בוחר גומה שאינה בצד שלו (הגומות של שחקן 1 הן 0 עד 5 ושל שחקן 2 הן 6 עד 11)
   2 - כאשר השחקן בחר גומה שאין בה אבנים.
   0 או מספר חיובי כאשר המהלך בוצע , אם המספר חיובי זהו מס' האבנים שהשחקן הצליח לאסוף בתור הזה (יכול להיות 2 או 3).
   השיטה צריכה לבדוק את תקינות הקלט, במידה והקלט לא תקין להחזיר את הערך 1- או 2- (10 נק') לאחר מכן השיטה צריכה לבצע את פיזור האבנים (במידה והגומה שנבחרה תקינה) , הפיזור יעשה בניגוד לכיוון השעון (15 נק'), לאחר הפיזור של האבנים היא צריכה לבדוק האם השחקן זוכה באבנים נוספות (2 או 3) , במידה וכן היא צריכה לעדכן את הניקוד של השחקן וכן לדאוג להוריד את האבנים מהגומה ולעדכן את המשתנה left (10 נק').
   לבסוף על השיטה להעביר את התור לשחקן האחר (ע"י עדכון המשתנה current_player) זאת אך ורק במידה והוא באמת יכול לשחק בתור הבא (אם יש לו לפחות אבן אחד בצד שלו) (15 נק') . (סה"כ 50 נקודות)

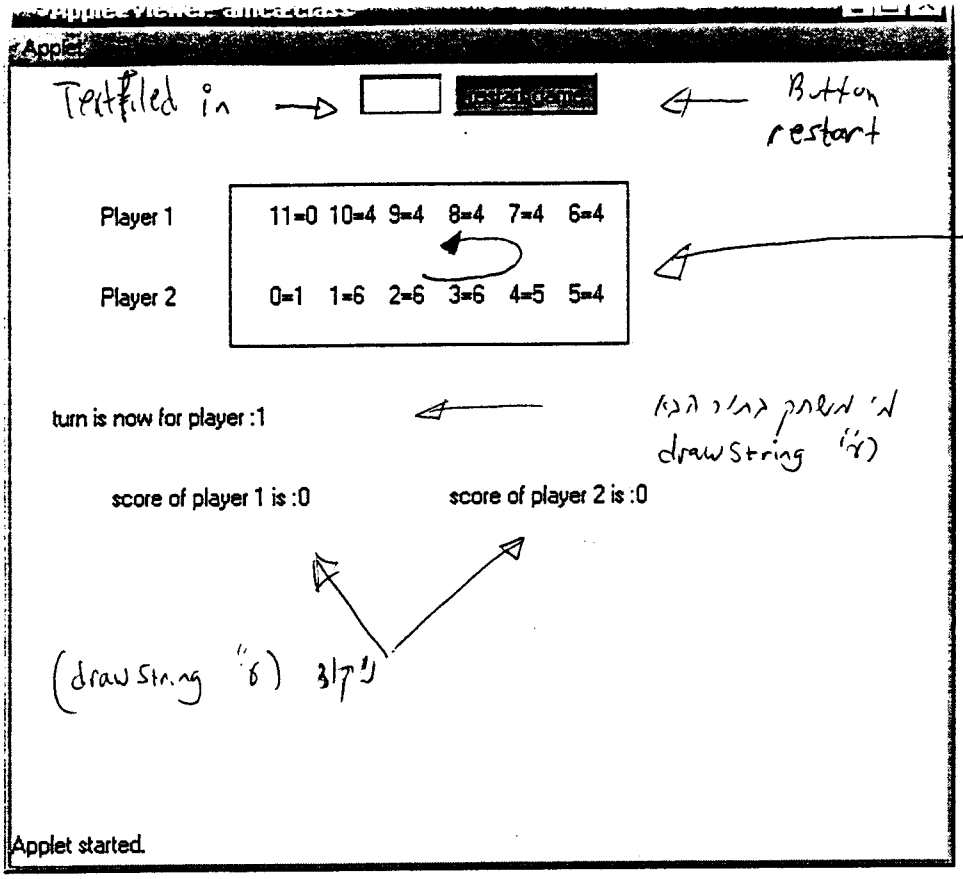
3. כתבו את השיטה public void actionPerformed אשר תעשה את הדברים הבאים
   1 במידה והשחקנים לחצו על כפתור ~~reset~~ restart אובייק המשחק יבנה מחדש , שדה הקלט in ינוקה והמסך ירוענן (repaint) (8 נק')
   2 במידה והקלט בא מהשדה in מס' הגומה שנבחרה יועבר לאובייקט המשחק על מנת לבצע את הסיבוב הנוכחי (ע"י הפעלת השיטה play) , בהתאם לתוצאה שחוזרת מ play תעודכן התווית (label) l בטקסט רלוונטי. ולבסוף המסך ירוענן (repaint). (12 נק')
   (סה"כ 20 נקודות)

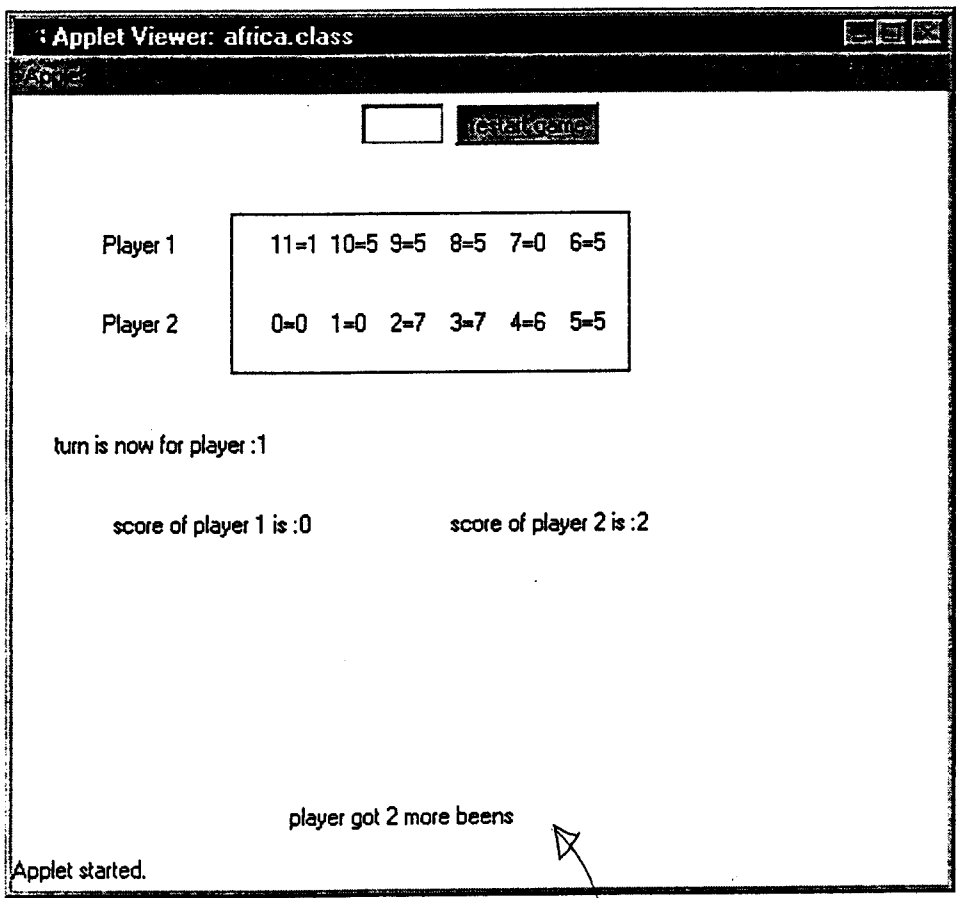
4. השלימו את החלק האחרון בשיטה paint אשר רושם את מס' הנקודות לכל אחד משני השחקנים , ובנוסף בודק האם אחד השחקנים ניצח , במידה וכן הוא רושם הודעה על המנצח , במידה ולא הוא רושם מיהו השחקן הבא שצריך לשחק כל הרישומים צריכים להעשות ע"י g.drawString תוך פניות לשיטות הרלוונטיות באובייקט ga.
   (20 נקודות)

בהצלחה

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class africa extends Applet implements ActionListener {

    game ga;
    TextField in ;
    Button restart;
    Label l;
    Panel p,p1;
    public void init() {
      resize(400,330);
      ga = new game();
      p = new Panel();
      p1 = new Panel();
      in=new TextField(3);
      p.add(in);
      restart = new Button("restart game");
      p.add(restart);
      l = new
Label("-------------------------------------------------------------"
);
      p1.add(l);
      this.setLayout(new BorderLayout());
      add(p,"North");
      add(p1,"South");
      in.addActionListener(this);
      restart.addActionListener(this);

    }

  public void paint(Graphics g){

   ga.print_table(g,100,100);

[place to add the rest of this method ]

}


public void actionPerformed(ActionEvent e){

[place to add content of this method ]

}


}
```

```
import java.awt.*;

public class game{

    int[] holes;
    int current_player;
    int[] score;
    int left;

 public game(){

     holes = new int[12];
     score = new int[3];
     score[1]=0;
     score[2]=0;
     current_player=1;
     int left = 48;
     for(int i=0;i<12;i++) holes[i]=4;

}

public void print_table(Graphics g , int x , int y){
    g.drawRect(x+10,y-40,200,80);
    g.drawString("Player 1",45,y-20);
    g.drawString("Player 2",45,y+20);
    for(int i=0;i<6;i++)
      g.drawString(""+i+"="+holes[i],x+30+i*30,y+20);
    for(int i=6;i<12;i++){
      g.drawString(""+i+"="+holes[i],x+30+(11-i)*30,y-20);
}

[place to add missing methods ]

}
```


Applet
Textfiled in
restart
Button restart
Player 1
11=0 10=4 9=4 8=4 7=4 6=4
Player 2
0=1 1=6 2=6 3=6 4=5 5=4
turn is now for player :1
score of player 1 is :0
score of player 2 is :0
Applet started.


: Applet Viewer: africa.class
restart
Player 1
11=1 10=5 9=5 8=5 7=0 6=5
Player 2
0=0 1=0 2=7 3=7 4=6 5=5
turn is now for player :1
score of player 1 is :0
score of player 2 is :2
player got 2 more beens
Applet started.