

תרגיל בית 4 – אבן, נייר ומספריים

יש להגיש את הפתרון (קבצי ה-Java בלבד - אינני זקוק לתמונות, למשל) עד סוף הסמסטר.

מטרת התרגיל היא לכתוב תכנית גרפית המשחקת עם המחשב את משחק הילדים "אבן, נייר ומספריים".

המשתמש	המחשב	התוצאה
1 אבן	אבן	תיק"ו
2 אבן	נייר	נצחון למחשב
3 אבן	מספריים	נצחון למשתמש
4 נייר	אבן	נצחון למשתמש
5 נייר	נייר	תיק"ו
6 נייר	מספריים	נצחון למחשב
7 מספריים	אבן	נצחון למחשב
8 מספריים	נייר	נצחון למשתמש
9 מספריים	מספריים	תיק"ו

התכנית פותחת חלון ובו שלושה איקונים: של אבן, של נייר ושל מספריים. המשתתף האנושי מסמן את בחירתו ע"י קליק בעכבר על האיקון הרצוי. התכנית מיד בוחרת גם היא באחד האיקונים באופן אקראי, משווה בין שתי הבחירות, ומדפיסה את תוצאת המשחק לפי הטבלה שמשמאל.

לנוחותכם, הקבצים הבאים נמצאים באתר ומכילים תמונות של העצמים:

- Rock.jpg
- Paper.jpg
- Scissors.jpg

למי שרוצה להרחיב, יש באתר תמונות נוספות: של אדם, של מחשב, וגם תמונה אחת ריקה למי שזקוק לכזאת על מנת לצייר שטח ריק (למשל, במקום בו מוצגות בדרך כלל הבחירות של האדם או של המחשב בטרם נעשתה בחירה כלשהיא):

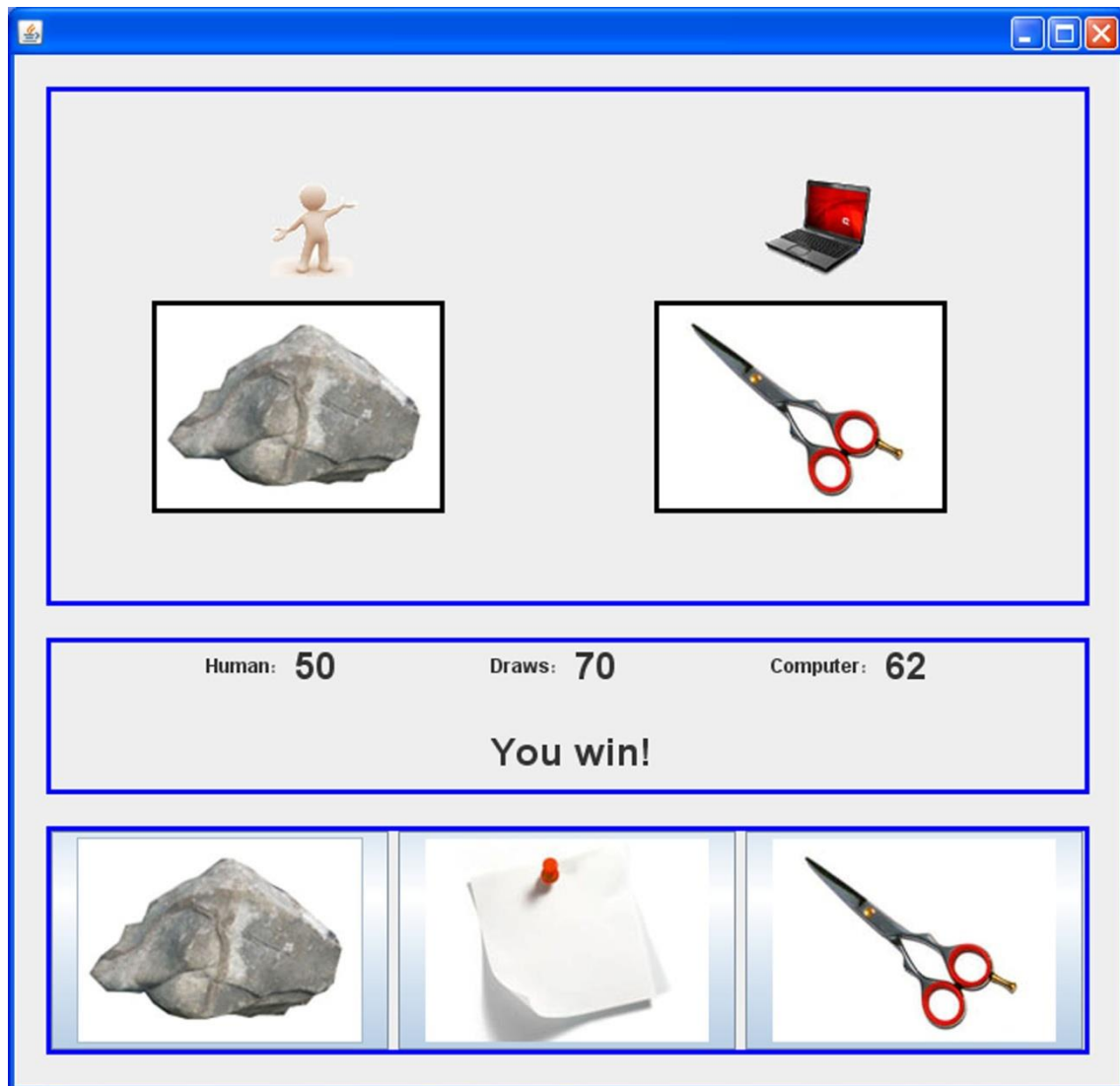
- Human.jpg
- Computer.jpg
- blank.jpg

התכנית גם מתחזקת לוח תוצאות, בו רשום כמה מפגשים היו, בכמה ניצח המשתמש האנושי ובכמה המחשב. דוגמא אפשרית לסידור החלון נראית בציור (ראו בעמוד הבא), אך זו הצעה בלבד, וכל ארגון אחר שנראה לכם יתקבל.

כמה עצות בנוגע לארגון עצמים בתוך GUI:

1. כדי לשמור שני Widgets ישארו ביחד ולא יתחילו לטייל בנפרד, כדאי להכניס את שניהם לתוך JPanel משותף. באופן רגיל, אין לעצם מסוג JPanel שום נוכחות על המסך, וכל ההשפעה שלו תסתכם בשמירת החלונות שבתוכו צמודים זה לזה. רצוי לקבוע את האמרגן של JPanel כזה להיות BoxLayout, המארגן את העצמים לאורך ציר יחיד (X או Y).
2. המערכת לא תמיד שמה לב לבקשות שלנו בנושא גודל של Widget. יש כמה מתודות שבאמצעותן אפשר להשפיע על האמרגן בעניין זה, ואני מנסה אותן בצירופים שונים עד שאני מקבל את התוצאה שאני מבקש.

```
setSize( width, height )
setPreferredSize( new Dimension( width, height ) )
setMinimumSize( new Dimension( width, height ) )
setMaximumSize( new Dimension( width, height ) )
```



3. הצגת טקסט לא אינטראקטיבי (כזה שהמשתמש אינו יכול לשנות) נעשית באמצעות JLabel. אפשר לקבוע את הגופן בו יוצג הטקסט באמצעות המתודה `setFont()` המקבלת עצם מטיפוס `Font`, כמו בדוגמא הבאה:

```
JLabel label = new JLabel( "text you wish to display" );
Font font = new Font( "Times new Roman", Font.BOLD, 36 );
label.setFont( font );
```

הקוד לעיל יגרום לכך שהטקסט המבוקש יוצג בגופן Times Roman, באות עבה, ובגודל 36 נקודות (חצי אינץ'), כך:

text you wish to display

גופנים אחרים הם Arial ו-Garamond, ובעברית David, Narkisim, Miriam.

4. כדי להפוך קובץ לתמונה שאפשר לשים בתוך JButton או JLabel משתמשים במחלקה ImageIcon, כמו בדוגמא הבאה:

```
JButton b = new JButton( new ImageIcon ( "Rock.jpg" ) );
```

5. כדי לסמן באופן גרפי את הגבול של Widget, אפשר להשתמש במתודה setBorder() של כל Widget. המתודה מקבלת עצם מסוג Border שניתן לקבל מאחת מתודות ה-factory הסטטיות במחלקה BorderFactory (ולא באמצעות new Border()), כפי שניתן בטעות (לחשוב):

```
BorderFactory.createLineBorder( Color color, int width )  
BorderFactory.createEmptyBorder( top, right, bottom, left )
```

אפשר גם לשים שני סוגי גבול על אותו Widget: אחד מסוג LineBorder כדי לסמן את הגבול בצבע על החלון, והשני מסוג EmptyBorder כדי ליצור שוליים סביב ה-Widget (ולחסוך את כל ה-Glue ו-Struts שלהם היינו אחרת זקוקים). הדבר נעשה באמצעות מחלקה מיוחדת, CompoundBorder, שתפקידה לחבר שני גבולות לאחד, כמו בדוגמא הבאה:

```
JPanel panel = new JPanel();  
Border outer = BorderFactory.createEmptyBorder( 0, 20, 0, 20 );  
Border inner = BorderFactory.createLineBorder( Color.blue, 3 );  
panel.setBorder( new CompoundBorder( outer, inner ) );
```

הערה: למרות שניתן להגדיר את הגבול של JButton באופן ישיר כמו לכל Widget אחר, מומלץ להמנע מכך, ולפעול באופן הבא במקום:

- a. להגדיר JPanel ולשים בתוכו את ה-JButton.
- b. להגדיר את הגבול הרצוי על ה-JPanel.

הערה זו תקפה לכל סוגי ה-Widget היותר מורכבים.

בהצלחה!