

תרגיל בית מס. 5 – שעון קוקיה

את התרגיל הזה אפשר להגיש עד ה-15 ביולי, 2021. זהו תרגיל קל, אבל אני חושב שרצוי שתרכשו נסיון מעשי בבניית יישום עם ממשק משתמש גרפי (GUI): זה חומר שנלמד בכתה ותחום ידע ראוי לשאלה בבחינה 😊.

בתרגיל זה נבנה שעון קוקיה. השעון לא מראה את השעה, אלא רק מתקתק ופעם ב-8 שניות יוצאת קוקיה ועושה "קוקו!". מצורף קובץ וידאו CuckooClock.mp4 המדגים את פעולת התכנית.

בתרגיל נשתמש בתמונות וקובצי שמע ונראה כיצד ניתן לעשות שימוש בקבצים מעין אלה ב-Java בקלות יחסית. אלא מכם שנמשכים למשחקי מחשב ימצאו בזה עניין, אני חושב.

את היישום נבנה בשלבים:

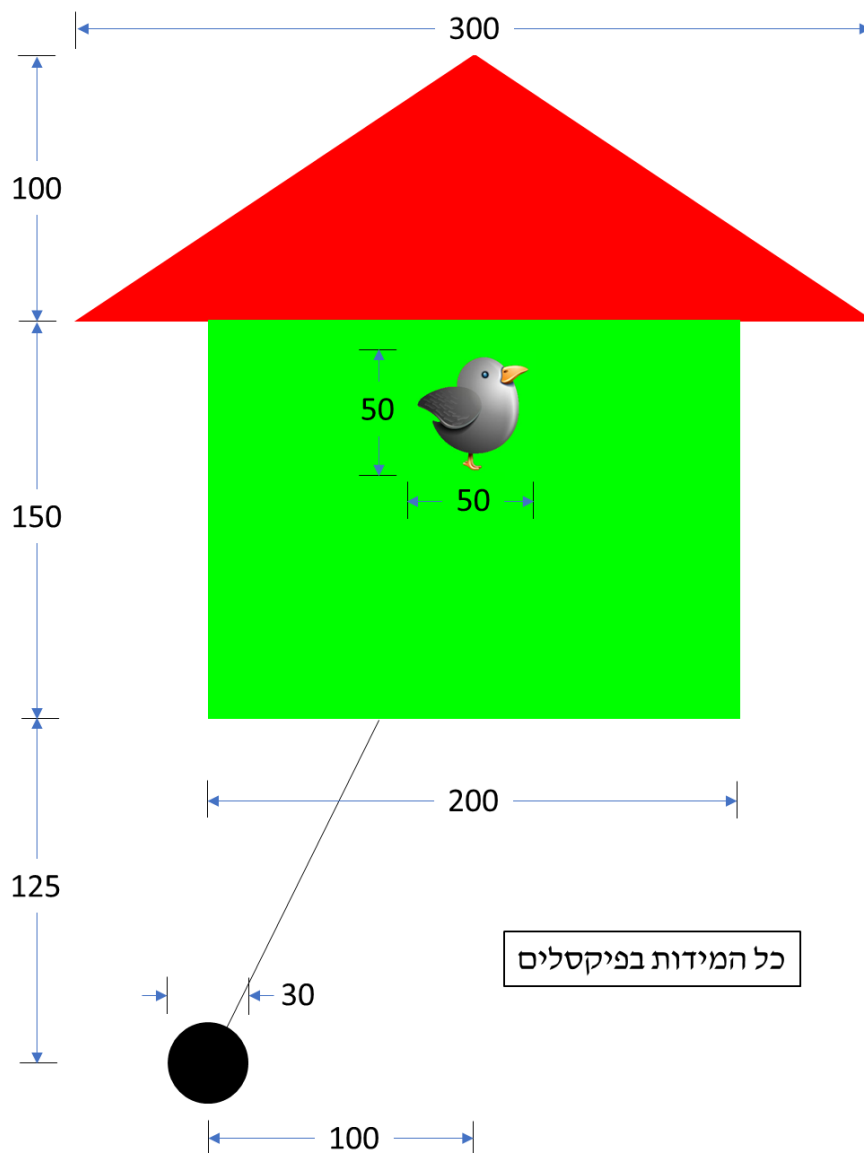
1. ציור החלק הסטטי: הגג האדום, הגוף הירוק, ומטוטלת בשני מצבים: בצד שמאל ובצד ימין.
2. ציור הקוקיה במקום והגודל הנכונים.
3. תכנות האנימציה
4. הוספת קריאת הקוקיה

ציור החלק הסטטי

כדאי להכין מתודות נפרדות לציור כל חלק בשעון: כך ניתן לבחור אילו חלקים לצייר בכל רגע. מתודות אלה מקבלות פרמטר מסוג Graphics שמשמש אותן לצורך הציור.

גוף השעון הוא Rect וצבעו ירוק (Color.green). הגג הוא Polygon וצבעו אדום (Color.red). המטוטלת צבעה שחור (Color.black) ומרכבת משני חלקים: המשקולת שהיא Oval, והציר המחבר אותה לשעון שהוא Line. הממדים של כל אחד מהמרכיבים הללו נתונים בציור שלהלן. בדקו את התייעוד של המתודות הרלוונטיות במחלקה Graphics כדי לצייר מרכיבים אלה.

יש להכין מתודה המציירת את המטוטלת בצד שמאל ומתודה נוספת המציירת אותה בצד ימין. בנוסף, יש להכין מתודה המוחקת את ציורי המטוטלות, כדי שיהיה אפשר לתת תחושה שהמטוטלת עוברת מצד לצד. (ללא מחיקה, המטוטלת תיראה בשני הצדדים בבת אחת.)



ציור הקוקיה

הקוקיה היא תמונה שנמצאת בקובץ Bird.jpg שבאתר. את הקובץ הזה יש להפוך למופץ של המנשק Image, באופן הבא:

```
Image image = ImageIO.read( new File( שם הקובץ ) );
```

כדאי לייצר את ה-Image מראש, לא לייצר אותו מחדש בכל פעם שנדרש לצייר אותו כיון שזה כרוך בקלט/פלט, תהליך יחסית ממושך, ואין מקום לעכב את המערכת החלונאית כדי ליצור אותו בכל פעם מחדש.

את ה-image הזה אפשר לתת למתודה של Graphics שמציירת תמונות – drawImage(). המתודה מקבלת גם פרמטרים לציון המיקום והגודל בהם מבקשים שהתמונה תוצג. יש לאפשר גם את מחיקת הקוקיה. כיון שהקוקיה מוצגת על רקע הגוף הירוק של השעון, מספיק לצייר את הגוף (המלבן הירוק) על מנת למחוק את תמונת הקוקיה.

הוספת האנימציה

דינאמית השעון מחליף את הצד בו מוצגת המטוטלת כל שניה, ופעם ב-8 שניות מופיעה הקוקיה לשניה אחת. תוצאה זו מושגת על ידי ציור ומחיקה של מרכיבים בתמונה בהתאם לזמן. את הזמן

אפשר לכוון על ידי משתנה מסוג int שמקודם ב-1 כל שניה. הביט הימני שלו קובע את מקום המטוטלת (נניח שמאל אם הוא 0 וימין אם הוא 1). שלושת הביטים הימניים קובעים מתי יש לצייר את הקוקיה. שימו לב שיש למחוק את ציורי המטוטלת לפני שמציירים אחת חדשה, וגם את ציור הקוקיה יש למחוק מקץ שניה אחת מהזמן בו היא הוצגה.

זכרו לקרוא ל-`repaint()` בכל איטרציה כדי לגרום למערכת החלונאים להפעיל את הציור של המסך שביקשתם.

השמעת צליל

כדי להשלים את האפקט, אפשר גם להשמיע ציוץ של קוקיה בכל פעם שהיא מופיעה. יש באתר קובץ נוסף, `CuckooCall.wav` שמכיל את הצלילים הדרושים. אפשר לגרום לקובץ הזה להיות מושמע באופן הבא:

```
Clip clip = AudioSystem.getClip();
clip.open( AudioSystem.getAudioInputStream( new File( שם הקובץ ) ) );
clip.start();
```

המחלקות `Clip` ו-`AudioSystem` נמצאות בחבילה הסטנדרטית `javax.sound.sampled` לה ניתן לעשות `import` (תנו ל-`eclipse` לעשות זאת עבורכם עם `ctrl-shift-O`). `Clip` היא מחלקה היורשת מ-`Thread` וההשמעה של קובץ ה-`wav` נעשית בחוט הזה. ועל כן ניתן לקרוא לה ישירות מ-`paintComponent()` – הקריאה ל-`start()` חוזרת מיד ולא מעכבת את המערכת החלונאית. שימו לב שהקוד הנ"ל זורק חריגה שמצריכה טיפול, אבל את זה אתם כבר יודעים לעשות.

לב השעון

הלב הפועם של השעון הוא לולאה אינסופית המקדמת את המונה, גורמת לציור השעון (על ידי יצירת אירוע של `repaint` באמצעות המתודה `repaint()`), והמתנה של שניה אחת (באמצעות `sleep()`). הכניסו כל זאת למתודה של המחלקה, ורק קראו לה מ-`main()`.

אתם מוזמנים להוסיף תכונות לשעון כדי הדמיון הטובה עליכם: להוסיף את השעה, להוסיף מחוגים, או אינטראקציה (אולי כפתור שסותם לקוקיה המעצבנת את הפה...), משקולות עולות ויורדות (כמו בשעון קוקיה מכני) או כל תכונה אחרת. התרגיל אמור להיות כיף – אם נתקלתם בקושי מתסכל, אל תהססו לחפש בגוגל, לשאול חברים, להעלות שאלה לפורום או דוא"ל אלי, כרגיל.

בהצלחה!