פרויקט – תרגיל 2

פרטי המגישים:

|  |  |
| --- | --- |
| שם: איתמר איילון  ת.ז: 206024796  מייל: [itamaraa@mta.ac.il](mailto:itamaraa@mta.ac.il) | שם:  ת.ז:  מייל: |

**בונוסים שמומשו:**

**בונוס 1: האנימציות הן: 1)... 2)...**

**בונוס 2: נמצא בצד ימין למעלה, שם ניתן לבחור את ה theme שרוצים.**

**בונוס 3: במרכז המסך, מתעדכן עם התקדמות האלגוריתם.**

**בונוס 4: ממומש**

**בונוס 5: באקורדיון בצד ימין. ניתן לטייל בדורות דרך combo box או עם חצים prev/next – לאחר שהאלגוריתם מפסיק לעבוד ניתן לבחור ב combo box את הפתרון בדור שרוצים לראות וממנו יפתח האופציה לכפתורים next/prev.**

הסבר (בסיסי) על המערכת:

תחילה יש לטעון קובץ חדש, בכפתור Load File בצד שמאל למעלה.

לאחר מכן בצד שמאל יהיה ניתן לראות באקורדיון את הנתונים שבמערכת כגון rules, mutations וכו'...

תחת System Information קיימים המידעים אודות המורים, כיתות, קורסים, וכן מס' הימים, שעות, אוכלוסיה ואליטיזם.

במרכז המסך מוצג הפתרון הטוב ביותר כמערכת שעות, או כרשימה של שיעורים. כמו כן קיים כפתור לקנפג את המנוע של האלגוריתם האבולוציוני.

בנוסף בחלק המרכזי התחתון קיימים שלושת תנאי העצירה שניתן לבחור להפעיל עם ההתקדמות כאשר האלגוריתם עובד.

בגרף יוצג ההתקדמות של הפתרון הטוב עם הדורות.

בצד ימין למטה ניתן להפעיל להפסיק ולהשהות את פעולת האלגוריתם (השהייה תהיה אפשרית רק כשהאלגוריתם עובד).

בצד ימין באקורדיון מופיע המידע אודות הפתרון הטוב ביותר שמוצג כרגע. שם גם ניתן להחליף את הפתרון הטוב ביותר שמוצג כרגע, לפתרון אחר שהיה הטוב ביותר לפניו בהיסטוריה לפי המרווח דורות שהמשתמש הזין.

הערות:

- ניתן לבצע קונפיגורציה לאלגוריתם – הערכים ישתנו באופן אוטומטי, אין פידבק מהUI לגבי מה הערכים צריכים להיות בכל פרמטר. כאשר ערך לא תקין מבחינת טיפוס, ה label יצבע באדום. אם הטיפוס תקין אך הערך לא, האלגוריתם ידע לקחת את הערך המקסימלי או המינימלי שניתן (למשל גודל אוכלוסיה 200, ואליטיזם 500, יתקבל בפועל אליטיזם 200).

- כדי להציג את הפתרון הטוב ביותר יש לסמן ב combobox מה בדיוק רוצים לראות בפתרון.

- ה progressbars פשוטים ולכן מתעדכנים בכל איטרציה של המנוע כדי לא ליצור מצב בו נראה כאילו המנוע לא עובד והאחוזים קופצים.

- השהיית האלגוריתם מפסיקה את פעולתו לחלוטין והורגת את ה thread. ו-resume מרימה מחדש thread ומפעילה אותו מחדש החל מה-state הקודם שהיה לו לפני ההשהייה.

הסבר על המחלקות העיקריות:

**במודול UI** ישנה package components ובו כל קומפוננטות ה fxml עם ה controllers שלה. בו גם הפעלת האפליקציה במחלקה MainApplication. בנוסף יצרנו UIAdapter שמחזיק את המנוע שהUI עובד איתו, ודרכו יהיה ניתן לתפעל את האלגוריתם.

יצרנו אובייקט EngineModel שכל הקומפוננטות UI מכירות, והוא זה שמכיר את ה Engine מהלוגיקה של התרגיל הקודם ויוצר את הקשר עם המנוע בעזרת properties ו-listeners.

קיימים 2 tasks שונים, אחד לטעינת הקובץ והשני להפעלת האלגוריתם, אותם מפעיל UIAdapter.

בנוסף קבצי CSS ב package styles לעיצוב המערכת.

**במודול EvolutionAlgorithm** הוצאנו את תנאי העצירה ככה שניתן יהיה להכניס איזה תנאי עצירה בנראה כל עוד הוא מממש את הממשק StopCondition.

**במודול Logic** – ללא שינויים קריטים.