

Predictions - תרגיל 2:

מגישים: יאיר חנגלי 316590595, אופיר נקדאי 318382827

מיילים ליצירת קשר:

yairyehzekelhe@mta.ac.il

ofirna@mta.ac.il

***מימשנו את הבונוסים: אפשרות החלפת skin(מספר 1) , אנימציות (מספר 2)**

הסבר על המערכת:

מערכת זו הינה מערכת גנרית ליצירת הדמיות וסימולציות על ישויות שעליהן חלים חוקים מסוימים – כתלות ב"עולם" שרוצים להגדיר, ומטרתה לחזות מה יקרה לישויות לפי ההסתברויות שניתנות לה.

בהפעלת התוכנית יוצג החלון הראשי.

חלון זה מורכב מה header, אשר זמין תמיד לצפייה ובו ניתן לטעון קובץ סימולציות חדש למערכת ולראות את מצב הטרדים. במידה והקובץ שנטען נמצא תקין, יעודכנו פרטי הסימולציה במסכים השונים. אחרת - תופיע הודעת שגיאה המציגה את הסיבה לכך.

ב header ניתן בנוסף לסמן את אפשרות Animations וכך לראות בו חיווי על טעינת קובץ מוצלח והרצת סימולציה בדמות אנימציה וכמו כן ניתן לבחור בסקין הרצוי לאפליקציה.

בנוסף, בחלון הראשי ישנם 3 טאבים:

Details – בו ניתן לראות את פרטי הסימולציה הטעונה במערכת, במידה וישנה כזו. בצד שמאל מופיע עץ המתאר את רכיבי הסימולציה, כאשר לחיצה על אחד ה"עלים" בו, תציג פרטים מלאים עליו בצד ימין של החלון.

New Execution – בו ניתן להזין נתונים (כמות מכל ישות, ערכי משתני סביבה) ואיתם להריץ סימולציה חדשה. במידה והוכנסו נתונים שאינם תקינים – תוצג למשתמש הודעה המפרטת על כך בעת הכנסתם.

Results – בטאב זה ניתן לראות בצד שמאל את רשימת הסימולציות הקיימות במערכת. בעת בחירת סימולציה מהרשימה, יוצגו בצד ימין פרטים נוספים עליה: עבור כל סימולציה ניתן לראות את מצבה (יצירה, ריצה וסיום) ואת טבלת כמות הישויות כעת. בנוסף:

במידה ומדובר בסימולציה אשר רצה כעת: יוצגו ערכי משתני הסביבה שנבחרו, כפתורי הרצה (השהייה, הפעלה ועצירה), מספר ה tick הנוכחי, השניות שחלפו מתחילתה ומדי ההתקדמות שלה במידה ומוגדרת לסיום ע"י מספר שניות/tick-ים.

במידה ומדובר בסימולציה אשר הסתיימה: תוצג הסיבה לסיום - בין אם מדובר בשגיאה או בהגעה לאחד מתנאי הסיום, נתונים סטטיסטיים נוספים לבחירת המשתמש (ע"י בחירת ישות, תכונה והמידע הרצוי), גרף כמות הישויות לאורך הסימולציה ואפשרות להרצתה מחדש ע"י כפתור reRun.

עבור סימולציות שהסתיימו, בחרנו לממש את גרף כמות הישויות בצורה הבאה: עבור 100 tick-ים הראשונים, אנו מכניסים בכל tick נתון חדש.

לאחר מכן, בכל 100 tick-ים אנו מכניסים נתון חדש, כך עד סוף הסימולציה. לבסוף, למשתמש מוצגים בגרף נתונים לאורך כל הסימולציה בהתאם לכמות ה tick-ים הסופית, במידה שאינה מעמיסה על הרכיב.

הסבר על המחלקות החדשות העיקריות ותפקידן:

מודול SystemEngine:

המודול האחראי על הצד הלוגי של המערכת.
מכיל מחלקה עיקרית בשם SystemEngineImpl אשר מנהלת את הסימולציות השונות במערכת ומחזיקה את ה-poolThread האחראי על הרצתן.
במודול זה בעיקר עודכנו המחלקות השונות הקיימות על מנת לתמוך בפעולות החדשות במערכת ובלולאת הסימולציה החדשה הכוללת בין היתר ישות משנית.
מחלקות חדשות עיקריות בו:

GridManager – בתרגיל זה התווסף מימד המקום, לכן יצרנו רכיב שתכליתו לנהל את כל מה שקשור למיקום – אתחול, הקצאה ותזוזה.

מודול UI:

המודול האחראי על ניהול ממשק המשתמש החלונאי. מתקשר עם מודול SystemEngine בעזרת נתונים בצורת DTO, ובכך משתף נתונים שהוזנו על ידי המשתמש במסכים השונים עם המנוע ובנוסף מציג מידע רלוונטי למשתמש במודול זה נמצאת פונקציית main.
בנוסף ל-JAT שהוא thread היחיד שמשנה את רכיבי ה-UI, במודול זה רץ ברקע thread נוסף שתכליתו לדגום נתונים באופן קבוע (כל 300 מילי שניות) מתוך מודול SystemEngine, ולקרוא ל-JAT לעדכן את רכיבי ה-UI הרלוונטיים לשינוי.

מחלקות חדשות עיקריות בו:

DetailsController – אחראי על ניהול רכיב התצוגה שבטאב Details.

ExecutionController – אחראי על ניהול רכיב התצוגה שבטאב NewExecution.

ResultsController – אחראי על ניהול רכיב התצוגה שבטאב Results.

מודול DTO:

בתוכו מוגדרות מחלקות DTO אשר מטרתן להעביר מידע בין שכבת המנוע לשכבת ממשק המשתמש ולהיפך.

מודול OurExceptions:

בתוכו מומשו חריגות שונות אותן יצרנו בהתאם לצורכי המערכת.