פיתוחים עתידיים

# שעונים

## LC

### האלגוריתם:

1. כל מעבד מכיל פרמטר שהוא השעון הלוגי שלו
2. כל הודעה מכילה פרמטר שהוא השעון הלוגי שלה
3. בשעת בניית ההודעה לוקחים את שעון הלוגי של המעבד
4. בשעת קבלת ההודעה מעדכנים את השעון הלוגי של המעבד (מקסימום בין השעון הלוגי של המעבד לזה של ההודעה + 1)

### שעון ווקטורי

1. השעון הווקטורי יוחזק במקום אחד – ב – RunningHandler
2. השעון הווקטורי יעלה בכל פעם ששולחים הודעה ומקבלים הודעה
3. בנוסף תהיה פונקציה להעלאת השעון הווקטורי של מעבד
4. השעון הווקטורי צריך להופיע ב :
   1. מסך ראשי
   2. מסך הודעות
   3. לוגים

### מימוש

1. רשימה של int ב – RunningHandler
2. כאשר מתקבלת הודעה הערך המתאים עולה + דווח ללוגר
3. כאשר נשלחת הודעה הערך המתאים עולה + דווח ללוגר
4. בנוסף תהיה פונקציה להעלאה של הערך עבור מעבד הפונקציה תקרא ReportInternalEvent + דווח ללוגר
5. הלוגר ידפיס את השעון בהודעות שלו
6. בחלון ההודעות השעון יופיע בתחילת כל הודעה

# דוקומנטציה בקוד

1. ב – Template להוסיף את הפונקציות לשליחת הודעות

# דוקומנטציה

## מבוא

## קונספט

1. מבנה הרשת
2. שלבים בריצה של התוכנית
3. שלבים בעבודה עם התוכנית

## מבני נתונים עיקריים

1. Attribute
2. AttributeDictionary
3. AttributeList
4. NetworkElement
5. התפקידים של המילונים
6. BaseNetwork
7. BaseProcess
8. BaseMessage
9. הקונצפט של ה - enums

## יצירת אלגוריתם

1. נתונים שצריך
2. יצירת הדוקומנטציה
3. יצירת הקוד הראשונית
4. יצירת קוד הריצה

## הכנת הרשת

1. יצירת /טעינת הרשת
2. הוספת אלמנטים
3. הורדת אלמנטים
4. עדכון אלמנטים
5. Reindex
6. בדיקה
7. בניה

## הרצת הרשת

1. Run to End
2. Single step all
3. Single step one
4. Terminate

## Debug

1. שמירת הקובץ בזמן הרצה
2. טעינה למצב debug

## תמיכה ב - Gui

1. Edittable
2. EndInputOperation
3. ElementWindowPrms

## Events

## מימוש אלגוריתמים

1. Permitions
2. MessageRouter
3. Config
4. יצירת אלגוריתמים
5. Events
6. נוהל בדיקת רשת
7. DeepCopy
8. ScanAndReport
9. Algorithm Termination
10. Running Step Termination Detection
11. CustomizedMessageBox
12. SyntaxHighlight
13. UpdateNetwork
14. LogicalClock
15. ניהול הפרמטר Round (כל ההגדרות נמצאות ב – base, ההשמה להודעות נמצאת ב – base, האחריות על קידום – של האלגוריתם)

# דוקומנטציה

## נבדקו ונערכו מחדש הדפים:

1. The process of implementing algorithm
   1. Create initial code
   2. NetworkElements
2. למחוק Inserting Data Needed for the algorithms
3. יש לערוך את Programming the algorithm ע"מ לסיים את הנושא

# תוספת מילון ל – NetworkElement

1. להוסיף בכל המחלקות את ה – Init של המילון
2. להוסיף בכל ה – enums את ה – Keys של המילון
3. לבדוק את החלונות הבאים:
   1. ElementInputWindow
   2. AddRemoveAlgorithmWindow
   3. Config window

# MessageQ

1. לטפל בנושא של שינוי סדר ההודעות ע"י התהליך
2. להעביר את Breakpoint, MessageQ, המחלקה החדשה ל – ElementAttributes

## אופן הפעולה:

הפעולה של הגרירה היא בחירה של אלמנט אחד , השמתו באינדקס אחר והזזת כל השאר

לכן לפעולה של הגרירה יש 2 פרמטרים:

אינדקס ראשון, אינדקס אחרון

1. בתחילת הגרירה צריך לבצע:
   1. שמירת אינדקס ראשון
2. בזמן ה – drop צריך לבצע 2 פעולות
   1. שינוי סדר ההודעות ב – MessageQ
   2. (כפוף לאישור) הקלטה של המצב

### אובייקט ChangeMessageOrderEvent

1. יורש מ – AttributeDictionary

#### תכונות:

* 1. רשימת פרמוטציות
  2. רשימת הודעות

#### פעולות

1. הקלטה
   1. שלב ראשון בזמן הלחיצה על העכבר – אם אין רשימה של סטטוס ראשוני להקליט סטטוס ראשני לרשימה חדשה
   2. שלב שני בזמן ה – drop
      1. יש שתי אופציות
         1. להמשיך בהקלטה
            1. לא לבצע כלום
         2. לסגור event
            1. ליצור אובייקט של event
            2. למחוק את המצב הראשוני
         3. Quit
            1. למחוק את המצב הראשוני
2. בדיקה
   1. הפעלה – כל פעם שהתור משתנה
   2. השווה בין התור המוקלט לתורים שקיימים (פונקציה חדשה ל – message שמחליטה מה להשוות)
   3. אם יש זהות – שאלה האם לשנות את התור
3. בזמן ה – drop שואלים האם לשמור
4. אם כן נשמרות כל ההודעות

# פיתוחים עתידיים

1. **ליצור מנגנון שכל ההשמות מהחלון לא ייצרו אלמנטים חדשים**
2. **תיקונים קטנים**
   1. **בחירת מספר רנדומלי כאשר מכניסים מעבד חדש**
3. **לפשט את האלגוריתם של LayYoung (בהתאם ליצירת משתני חבר אוטומטים)**
4. **עדכון FloatingSummary כך שיתמוך ברשימות ומילונים**
5. **רויזיה ביצירת אלגוריתם**
   1. **יצירת סוג חלון חדש שיקבל את הפרמטרים**
6. **רויזיה בהודעות:**
   1. **הוספת הודעות מהממשק**
   2. **שינוי סדר ההודעות בתור – באמצעות תוכנה ובאמצעות הממשק**
7. **לשים בהסבר שכל פעולה שנחוץ לבצע לפני השליחה של BaseAlgorithmMessage שמים ב – BuildBaseAlgorithmMessage כמו:**
   1. **עדכון פרמטרים**
   2. **עדכון ההודעה בשדות שידועים רק בזמן הריצה**
8. **להוסיף ב – help את אמצעי הדווח הבאים:**
   1. **MessageRouter.ReportMessage**
   2. **MessageRouter.CustomizedMessageBox**
   3. **שינוי הפרמטרים של התצוגה**
9. **לסיים את האלגוריתם של LayYoung**
10. **לתקן את ה – comments של base, template, layYoung כך שיכללו גם א סוג הפרמטר (למחוק את הפרמטרים מה – comment ולבצע עדכון ל – comment)**
11. **רבולוציה בפונקציות התצוגה של הפרמטרים**
12. **כאשר מתחילים debug היום ההודעות נמסרות לתהליך ששלח את ההודעות לפני העצירה. ואז התהליכים שולחים את ההודעה ע"מ לשחזר את התור של ההודעות. הבעיה היא שהשיטה הזאת לא שומרת על סדר ההודעות שהיו בתור לפני העצירה**
13. **פרמטרים ל – ElementWindow:**
    1. **presentExistingValue**
    2. **presentNewValue**
    3. **performValueChangeCheck**
14. **הסברים :**
    1. **תהליך בדיקת שינוי בערך ב – element window**
    2. **תהליך בדיקת ערכים בשלב check**
    3. **תהליך השמה והוספה של ערכים ו – Attributes**
    4. **Enums**
    5. **DeepCopy**
    6. **תהליך קביעת controls ותכונות עבור Attribute ו – ElementWindow (בדגש שלא כל הפרמטרים צריכים להיות ממולאים)**
15. **קביעת , שמירת, והצגת המיקום בתור של ההודעה:**
    1. **יתוסף פרמטר ב – header של ההודעה שיכיל את המיקום של ההודעה בתור**
    2. **הפרמטר יוצג בפונקציה Description של ההודעה**
    3. **הפרמטר יתעדכן כאשר :**
       1. **מוסיפים הודעה לתור (עדכון ההודעה בלבד)**
       2. **מוציאים הודעה מהתור (עדכון כל ההודעות)**
       3. **משנים את סדר ההודעות בתור**
       4. **השחזור של סדר ההודעות במצב debug יעשה לפי הפרמטר הזה**
16. **שינוי ערכים של Attributes מהחלון ע"י עקיפת ה - Permitionss**