

קורס מונחה עצמים – מטלה 4 ואחרונה : The Maze of Waze

כללית: במטלה זו נשתמש בתשתית של מטלות 2,3 (מטלה 2: מבני נתונים ואלגוריתמים עבור גרף ממושקל, מכון, מטלה 3: מערכת שמאפשרת ניהוג ידני ואוטומטי של קבוצת רובוטים במשימות "איסוף פירות). היעד העיקרי במטלה הוא לטייב את מטלה 3 מבחינת הלוגיקה של המשחק (קבוצת רובוטים שמבצעת משימה של איסוף "מטבעות" ע"ג גרף ממושקל), וכן לבצע השוואה למול תוצאות של סטודנטים אחרים בקורס.

במטלה 4 הוספנו פונקציונליות נוספת לשרת כדי לאפשר שיתוף תוצאות בשרת DB. למעשה בכל הקשור למשחק אין שינוי למעט היכולת שלכם כעת לעשות login עם מספר תעודת הזהות שלכם וכך לאפשר לכם לפרסם את תוצאות הריצה שלכם בשרת (נעשה באופן אוטומטי):
להלן שתי התוספות (מסומנות בקו תחתון) שנוספו לממשק game_service כפי שניתנה לכם בדוגמא המצורפת :

```
int scenario_num = 9; // current "stage: is 9, can play[0,9], can
NOT 10 or above
int id = 999;
Game_Server.Login(id);
game_service game = Game_Server.getServer(scenario_num);
// ...
String remark = "This string should be a KML file!!";
game.sendKML(remark); // Should be your KML
```

מעבר לכך בניגוד למטלה 3, במטלה 4 תצטרכו לעבור את השלבים לפי סדר (להתחיל בשלב 0 ורק אם תצליחו לעבור את התוצאה הנדרשת ובפחות קריאות לשרת תוכלו להריץ גם את השלב הבא). בטבלה מטה ניתן לראות את כל 11 השלבים שעליכם לעבור. להלן רשימת המשימות שעליכם להוסיף למטלה 3:

שלבי עבודה:

1. החליפו את ה jar בגרסה החדשה: התקינו אותה וודאו שהקוד של מטלה 3 רץ היטב על ה jar החדש, לאחר מכן הריצו את שתי הדוגמאות שמצורפות לגרסה של מטלה 4: אחת הדוגמאות מסבירה כיצד להשתמש בפונקציונליות החדשה (שתי פקודות חדשות), המחלקה השנייה מדגימה כיצד לקרוא מבסיס הנתונים את נתוני המשחקים של כלל הסטודנטים (שימו לב שאם עשיתם login(id) לפני יצירת ה game_service התוצאה של ההרצה של המשחק שלכם תעלה לשרת ותוכלו לראות את הנתונים ע"י הרצה של הדוגמא של קריאה מה DB).
2. בדקו את עצמכם: האם הפתרון שלכם מצליח לפתור את התרחישים (ראו טבלה מטה) בציון הנדרש ובלא יותר פעולות. עליכם לבצע login(id) לפני תחילת המשחק (ראו דוגמא) ולהתחיל משלב 0 וכל פעם להתקדם – הקפידו לכתוב ל DB עבור כל אחת מ-24 רמות המשחק קובץ KML מיטבי (ראו דוגמאות כיצד לעשות זאת).
להלן טבלת "השלבים" של מטלה 4: הטבלה כוללת 11 שלבים, מתחילים בשלב 0 וכל פעם שמצלחים להריץ משחק בניקוד גבוה מהנדרש ובלא יותר פעולות תעברו שלב אחד קדימה. להלן הטבלה המעודכנת של השלבים.

| Stage | Case | Grade | Moves |
|-------|------|-------|-------|
| 0 | 0 | 125 | 290 |
| 1 | 1 | 436 | 580 |

| | | | |
|----|----|------|------|
| 3 | 2 | 713 | 580 |
| 5 | 3 | 570 | 500 |
| 9 | 4 | 480 | 580 |
| 11 | 5 | 1050 | 580 |
| 13 | 6 | 310 | 580 |
| 16 | 7 | 235 | 290 |
| 19 | 8 | 250 | 580 |
| 20 | 9 | 200 | 290 |
| 23 | 11 | 1000 | 1140 |

3. הוסיפו לממשק הגרפי (ה GUI שפיתחתם עבור מטלה 3) אפשרות להציג (מבסיס הנתונים) את התוצאות שלכם ביחס למשתתפים האחרים לרבות:

a. מידע על המשחקים שלכם: כמה משחקים שיחקתם בשרת, את השלב הנוכחי שלכם. ועבור כל שלב את התוצאה הכי טובה שלכם (שעומדת בלא יותר פעולות מהמתואר בטבלה).

b. עבור כל אחד מה 11 שלבים את המיקום שלכם בכיתה.

4. שפרו את התיעוד של מטלה 3 כולל התייחסות מפורטת לאלג', ניתוח התוצאות, עשו מאמץ להשלים את הפרויקט כך שיהיו כתוב ומוסבר היטב (הן מבחינת הנדסת תוכנה והן מבחינת איכות התיעוד).

5. יצאו את מטלה 4 שלכם ל executable jar בשם Ex4.jar, שימוקם בתיקיית השורש של הפרויקט.

הנחייה כללית:

- מטלה זו פתוחה למימוש שלכם, התוצרים המרכזיים במטלה הם מערכת משחק שתפתחו, קובצי KML של תוצאות, ותיעוד הפרויקט ב github.
- עליכם לעשות שימוש בשרת (כפי שמודגם במחלקה המצורפת).
- כרגיל הקפידו להוסיף מחלקת בדיקה (JUnit) לכל מחלקה לוגית שאתם כותבים (לא צריך לכתוב בדיקה למחלקות GUI, אבל לממיר קבצים כמו KML – כמובן שצריך).
- המטלה תיבדק בבדיקה פרונטלית – הקפידו שכל המגשים מכירים היטב את הקוד, את אופן השימוש בו ואת אופן ההרצה.
- את המטלה יש להגיש כפרויקט github (אפשר ורצוי לעשות זאת באותו מקום של מטלה 3 רק בגרסת commit אחרת).

מטלה אחרונה ודי!!
בהצלחה.