

קורס מונחה עצמים – מטלה 2: גרפים מכוונים ואתגר "פוקימונים"

במטלה זו נשפר את מטלה 1, ע"י הכללה של מבנה הנתונים שפיתחתם כך שיכול לתמוך בגרפים מכוונים. לאחר התאמת מבנה הנתונים נממש מספר אלגוריתמים על הגרפים מכוונים (וממשוקלים) לרבות יכולת שמירה ושיחזור של הגרף מקובץ בפורמט json, חישוב מסלול מכוון קצר ביותר, בדיקת קישוריות מכוונות ועוד. בשלב השני של המטלה נשתמש במבנה הנתונים והאלגוריתמים שפיתחתם כדי לממש משחק "אתגר הפוקימונים" – שכולל תהליכים, ממשק גרפי, ומערכת אוטומטית למשחק נגד "שרת". מטלה זו מחולקת לשני חלקים:

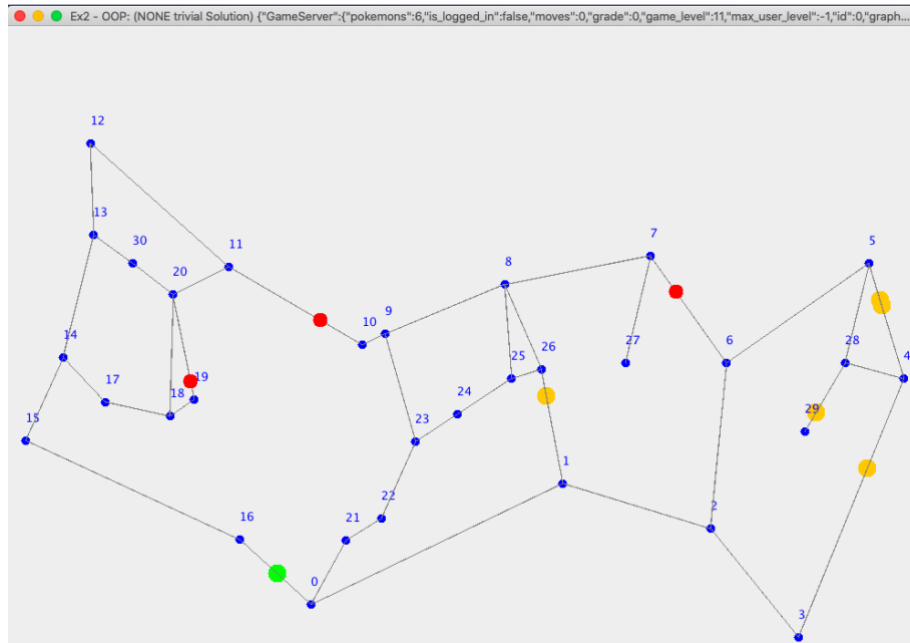
חלק ראשון:

1. בשלב הראשון עליכם להכיר את העקרונות של מבנה הנתונים של גרף מכוון (ממושקל חיובי), ולאחר מכן לתכנן כיצד אתם רוצים לממש אותו – דוגמאות והסברים נתנו בהרצאות והתרגולים. כלל המידע לגבי המטלה נמצא (ומתעדכן) בקישור הבא:
https://github.com/simon-pikalov/Ariel_OOP_2020/tree/master/Assignments/Ex2
2. פתחו פרויקט חדש ב-github
3. ממשו את מחלקה DWGraph_DS שמממשת את הממשק [directed weighted graph](#), עשו זאת בעזרת מימוש [הממשקים](#) הנוספים של צלע וקודקוד. (שימו לב שהמימוש הפנימי של רשימת הצלעות \ קודקודים שכנים של כל קודקוד – נתון לבחירתכם, וכך גם אופן המימוש של המחלקות צלע וקודקוד – שאינם צריכים להיות מחלקות פנימיות).
4. כתבו את המחלקה DWGraph_Algo שמממשת את הממשק [dw_graph algorithms](#) שמייצג את האלגוריתמים הבאים על גרפים ממושקלים חיוביים: מרחק מכוון בין זוג סדור של קודקודים, העתקה, שמירה ושחזור (בפורמט json) מקובץ, ובדיקת קישורות חזקה.
5. כתבו מחלקת בדיקה לכל מחלקה – ודאו שאתם מצליחים לעבור את כל הבדיקות, אתם נדרשים להוסיף בדיקות לפי שיקולכם כדי להגדיל את הביטחון שלכם בקוד.
6. כתבו תיעוד + הסברים מפורטים לגבי מבנה הנתונים, האלגוריתמים, מערכת התצוגה, וכמובן אופן השימוש בפרויקט מבחינת הורדה, והרצה. מומלץ לכתוב את ההסברים הכלליים בקובץ README.txt את התיעוד ספציפי יש לעשות על הקוד עצמו – יודגש יש לתעד כל מחלקה וכל שיטה לא טריוויאלית באופן שיאפשר למתכנתים אחרים להבין את המימוש שלכם ולהשתמש בקוד באופן קל ומדויק.

חלק שני:

7. הריצו את "משחק הפוקימונים". הקפידו להשתמש בגרסה המעודכנת של השרת והקוד המצורף.
8. נסו לתכנן פתרון מערכתי פשוט – כתבו את הפתרון כהסבר באנגלית וצרפו אותו לדפי ה-wiki של הפרויקט שלכם ב-github.
9. כתבו מערכת שמאפשרת להריץ את המשחק (הקפידו שהתוכנית הראשית שתהיה בשם Ex2.java ותמוקם באותה חבילה של gameClient).
10. כתבו מחלקה שמאפשרת הצגה גרפית של המשחק, להלן רשימת דרישות ממחלקה כזו:
 - a. הממשק הגרפי צריך לאפשר טעינה של תרחישים שונים (מספר שלם), את התרחישים אפשר יהיה להפעיל גם מפקודה ישירה של כפי נכתב בסעיף 9.
 - b. החלון הגרפי יאפשר הצגה של הגרף, מיקום הסוכנים, והפוקימונים, הציונים של כל סוכן, והזמן שנותר עד סוף המשחק, הממשק יופעל מהקובץ הראשי Ex2.java.
 - c. החלון הגרפי צריך להיות resizable – משמע שניתן יהיה להגדיל ולהקטין את החלון – והגרף והתנועה עליו צריכים להתעדכן בהתאם.
11. נסו להגיע לתוצאות גבוהות בכל אחד מהתרחישים [0-23] ודווחו על כך לשרת ע"י ביצוע login עם מספר ת"ז שלכם, שימו לב שהציון grade כולל את הסכימה של משקולות הפוקימונים "שנאספו". מעבר לכך נשמרים גם כמות הקריאות לשרת (moves) – תוצאה של יותר מ-10 קריאות לשנייה – נחשבת לא טובה, וככול שכמות הקריאות לשרת יותר נמוכה כך התוצאה יותר טובה.
12. סכמו את המטלה בדפי ה-wiki של ה-github שלה, כולל טבלה של התוצאות שלכם ביחס לכל תרחיש.

13. ייצרו קובץ הרצה (jar) בשם Ex2.jar שכולל ממשק גרפי שמאפשר להכניס את שם המשתמש ואת התרחיש, והרצה של התוכנית עם פרמטרים של <id, game_number> לפי הדוגמא הבאה: `java -jar Ex2.jar 12345678 12`. הקפידו למקם את הקובץ בתקייה הראשית של הפרויקט שלכם 12345678 בתרחיש מספר 12. הנחיות בנושא ניתן למצוא בקישור [הבא](#):



איור 1: צילום מסך של התוכנית Ex2_Client שמריצה את תרחיש 11, שכולל 3 "סוכנים" (Agents – מסומנים עגולים אדומים) ו 6 "פוקימונים" (מסומנים כעגולים צהובים וירוקים). הפוקימונים שנמצאים על צלע "עולה" (המקור קטן מהיעד) מסומנים בירוק (לפי השדה type), ופוקימונים שנמצאים על צלע יורדת (המקור גדול מהיעד) מסומנים בצהוב.

הנחייה כללית:

- מטלה זו מוגדרת בעיקר ע"י מספר ממשקים שמגדירים את ה api הנדרש ממחלקות, לנוחיותכם מימשנו כבר חלק המחלקות הנדרשות, בפרט כדי לאפשר לכם להריץ את הדוגמאות על החלק השני של המטלה. אתם רשאים לעשות שימוש בקוד המצורף אבל לא ניתן להניח שהוא מלא או נכון ואתם חייבים לבדוק אותו.
- חובה לעשות שימוש בJUnit, לכל מחלקה לוגית שאתם כותבים. עליכם לעשות שימוש בממשקים - אותם אינכם יכולים לשנות!
- המטלה מיועדת לזוגות – חשוב מאוד ללמוד לעבוד בזוגות ע"ג אותו repository.
- מותר לכם בהחלט להשתמש בקוד המצורף, אבל שימו לב שהקוד שצורף לכם אינו מלא ויש צורך להשלים \ לתקן אותו, כמובן אתן מוזמנים לכתוב את הפתרון שלכם בעצמכם מבלי להשתמש בדוגמאות הקוד – הכל לשיקולכם – אבל בכל מקרה של שימוש בקוד ממקורות חיצוניים חייבים לציין זאת בפרוש וחייבים להבין אותו.
- הימנעו מביצוע reverse engineering לקובצי ה jar שמצורפים לכם, יש שם מימוש חלקי של הפתרון שלכם אבל המטרה שתעשו את המטלה בעצמכם – למען הסר ספק: שימוש כזה יחשב כהעתקה!.
- הנחיות הגשה: את המטלה יש להגיש כפרויקט github, וכן לבדוק אותו במערכת בדיקת המטלות לפי ההנחיות מפורטות שהודגמו בתרגול – מטלות שלא תוגשנה לפי ההנחיות לא תזכנה בציון מלא.

בהצלחה!