תכנות מונחה עצמים – מטלה 4 (ואחרונה) – גרסה 0.2 26.12.2018

במטלה זו נרחיב את המטלות הקודמות (מטלות 2,3) לכדי משחק בסגנון פקמן. מטרת המטלה היא לסכם את כל הנושאים שנלמדו בקורס. בגלל אילוצי זמן בהחלט ייתכן שלא תספיקו את כל השלבים במטלה, חשוב להבדיל בין עיקר וטפל, המטלה היא לפתח קוד איכותי שמתועד היטב – גם אם המימוש אינו מושלם. מטלה זו תיבדק באופן פרונטלי כדי לאפשר לכם להגן על העבודות שלכם.

<u>מטרת המשחק:</u> לאכול את כל הפירות שבלוח, תוך כדי צבירת ניקוד מקסימאלי.

<u>שינויים בייחס למטלה 3:</u>

1. במטלה 4, עליכם להזיז שחקן אחד בלבד (שנקרא player, ומסומן באות M). השליטה בשחקן שלכם היא ע"י מתן כיוון במעלות לתנועה שלו. כאשר כל התנועה היא אך ורק בתוך האזור שהוגדר במטלה 3, וכל התנועה הוא בדו מימד בלבד (מימד הגובה הוא תמיד 0).
2. במטלה 4 יש גם מכשולים (מלבנים שחורים) שהשחקן שלכם לא יכול לעבור דרכם, למעשה על כל כניסה למלבן שחור או יציאה מחוץ לגבולות המגרש יורדת נקודה. המלבנים השחורים מסומנים בקובץ באות B.
3. במטלה 4, מעבר לשחקן שלכם (מסומן בוורוד בתמונה מטה) יש פקמנים נוספים (מסומנים בקבצים באות P, ובתמונה בצהוב) שרוצים לאכול לכם את הפירות. אתם יכולים לאכול את הפקמנים וכן את הפירות. על כל אכילה של פקמן או פרי מקבלים נקודות לפי שהוגדר בקובץ (בד"כ נקודה אחת).
4. במטלה 4 יש "רוחות" (Ghosts) אשר רודפות אחרי השחקן שלכם, (מסומנות בקבצים ב , G, ובתמונה באדום) אם כי השחקן שלכם מהיר יותר. אם זאת הרחות יכולות לעבור דרך מכשולים בעוד שהשחקן שלכם לא. בכל פעם שרוח מגיעה לשחקן שלכם יורדות 20 נקודות (כדי למנוע הורדה של נקודות רבות, לאחר הורדה של נקודות לשחקן שלכם יש 3 שניות "חסד" בהן הוא לא יאכל שוב).

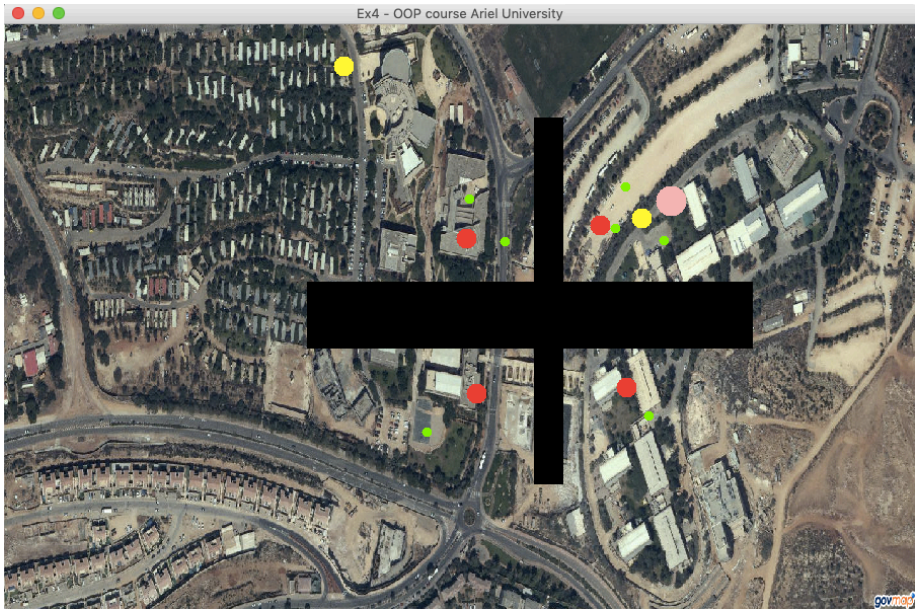
כדי לאפשר מידול של משחק אמיתי, בנינו לכם מערכת משחק שכוללת את המרכיבים הבאים:

- אוסף של 9 קבצים של תרחישים שונים – הקפידו לעשות בדיקות וניסויים בקבצים אילו – כך תוכלו למשל גם להשוות את התוצאות שלכם לאחרים.
- מערכת "שרת" שמאפשרת לכם לשחק מולה – בגדול המערכת תאפשר לכם לשלוט בשחקן שלכם, וכן תזיז את כל שאחר הפקמנים והרוחות. כדי לפשט את העניין על כל פעולה שאתם מבצעים השרת מבצע פעולה - ראו דוגמת הרצה בקובץ המצורף.
- מערכת איסוף מידע שתאפשר לכם לבצע השוואה אוטומטית של הביצועים של המערכת שלכם למול שאר הכיתה – פרטים יסופקו בהמשך.

<u>שלבי המטלה:</u>

1. הורידו את המטלה + דוגמאות הקוד + הקבצים, הריצו והבינו היטב את אופן השימוש בשרת, לאחר מכן כתבו הסבר קצר במילים שלכם על המשחק.
2. כתבו ממשק גרפי (שמתבסס על מטלה 3) שמאפשר משחק של משתמש אנושי – אפליקציה זו תאפשר טעינה של "משחק" והפעלה שלו (בתעודות הזהות שלכם). הריצו את האפליקציה ושחקו איתה, נסו להגיע לתוצאה טובה עבור דוגמא 8, לאחר מכן כתבו בקצרה בתיעוד את התוצאה המיטבית שהצלחתם להגיע אליה (עבוד דוגמא 8).
3. תכננו מערכות אוטומטית שמשחקת מול השרת – שימו לב שלב זה דורש לתכנן מערכת ולבנות לה דיאגרמת מחלקות וכן הסבר על אופן העבודה של המערכת – ואינה דורשת לתכנת, את כל תכנון יש לכתוב בדיפי הwiki של הgithub שלכם.
4. ממשו מערכת שמאפשרת לשחקן שלכם לשחק מול השרת באופן אוטומטי (ללא התערבות).
5. כתבו מסמך שמסכם את איכות התוצאות ביחס לכל אחת מ 9 התרחישים. השווה בין התוצאות של שחקן "אנושי" לתוכנה שלכם, וכן כתבו מערכת תוכנה שניגשת לבסיס הנתונים ומשווה את ביצועים שלכם לתוצאות של העבודות האחרות בכיתה (נושא זה יוסבר בתרגולים).

הסבר והדגמות למטלה:



איור 1: דוגמא מספר 5: שכוללת פירות ירוקים, פקמנים צהובים, ורוחות אדומות. השחקן מסומן בוורוד. מטרת השחקן היא לאכול את כל הפירות ולקבל ניקוד מקסימאלי: השחקן יכול לאכול גם פקמנים (יעזור לו למנוע מהפקמנים לאכול את הפירות). מעבר לכך השחק צריך להתחמק מהרוחות ולהימנע מלגעת במלבנים.

`Play(java.lang.String g)`	Init the play from a file, please use one of the 9 given files.
`Play(Robot.Game g)`	

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method	Description
`java.util.ArrayList<java.lang.String>`	`getBoard()`	The main data information: returns an ArrayList of Strings, of all the elements:
`java.lang.String`	`getBoundingBox()`	returns a String representing the bounding box of the region of interest [min,max], each is a point of (lat,lon,alt).
`java.lang.String`	`getStatistics()`	Returns the current state of the play: time, score, time left, #of kills, #of out box-bump.
`boolean`	`isRuning()`	Returns the current state of the server, return true iff the server is running.
`boolean`	`rotate(double ang)`	The main player functionality, rotates the player to a global orientation between [0,360) degrees.
`void`	`setIDs(long id1)`	set the first ID.
`void`	`setIDs(long id1, long id2)`	sets the first and second IDs.
`void`	`setIDs(long id1, long id2, long id3)`	sets the maximal group of three members: ID1, ID2, ID3.
`boolean`	`setInitLocation(double lat, double lon)`	Sets the player init location to a given lat,lon location.
`void`	`start()`	Start the server for a given default time (i.e., 100 seconds).
`void`	`stop()`	Stops the server - not needed for production only for testing, simply force - remain time to be 0.

איור 2: רשימת הפונקציות הציבוריות של הממשק של play (גרסה 0.2 – נוספו מספר שינויים קלים מהגרסה המקרית 0.1). שימו לב שישנה כעת שיטה לקבלת הסטטוס – האם השרת עדיין רץ – ראו isRunning().

הערות: הסטודנט ליעד כהן מהקורס בנה אתר נחמד שמציג את כל המידע בבסיס נתונים כל רבע שעה ראה: https://liad.cloud/

כדי לדעת להשוות את התוצאות שלכם עם אחרים עליכם להבין מהו המספר המייצג כל קובץ,

לדוגמא את הקובץ מספר 2 מייצג המספר: 1149748017, כך תוכלו להשוות את התוצאות שלכם למול תוצאות של סטודנטים אחרים על אותו קובץ קלט. שימו לב שהמספר מתאים לתוכן הקובץ לפיכך אם תשנו את תוכן הקובץ תקבלו מספר שונו – לפיכך הימנעו מלעשות זאת.

בהצלחה!