**תכנות מונחה עצמים מתקדם**

**עבודת הגשה מס 3'**

**נא לקרוא את כל המסמך עד סופו לפני תחילת העבודה!**

**דגשים להגשה**

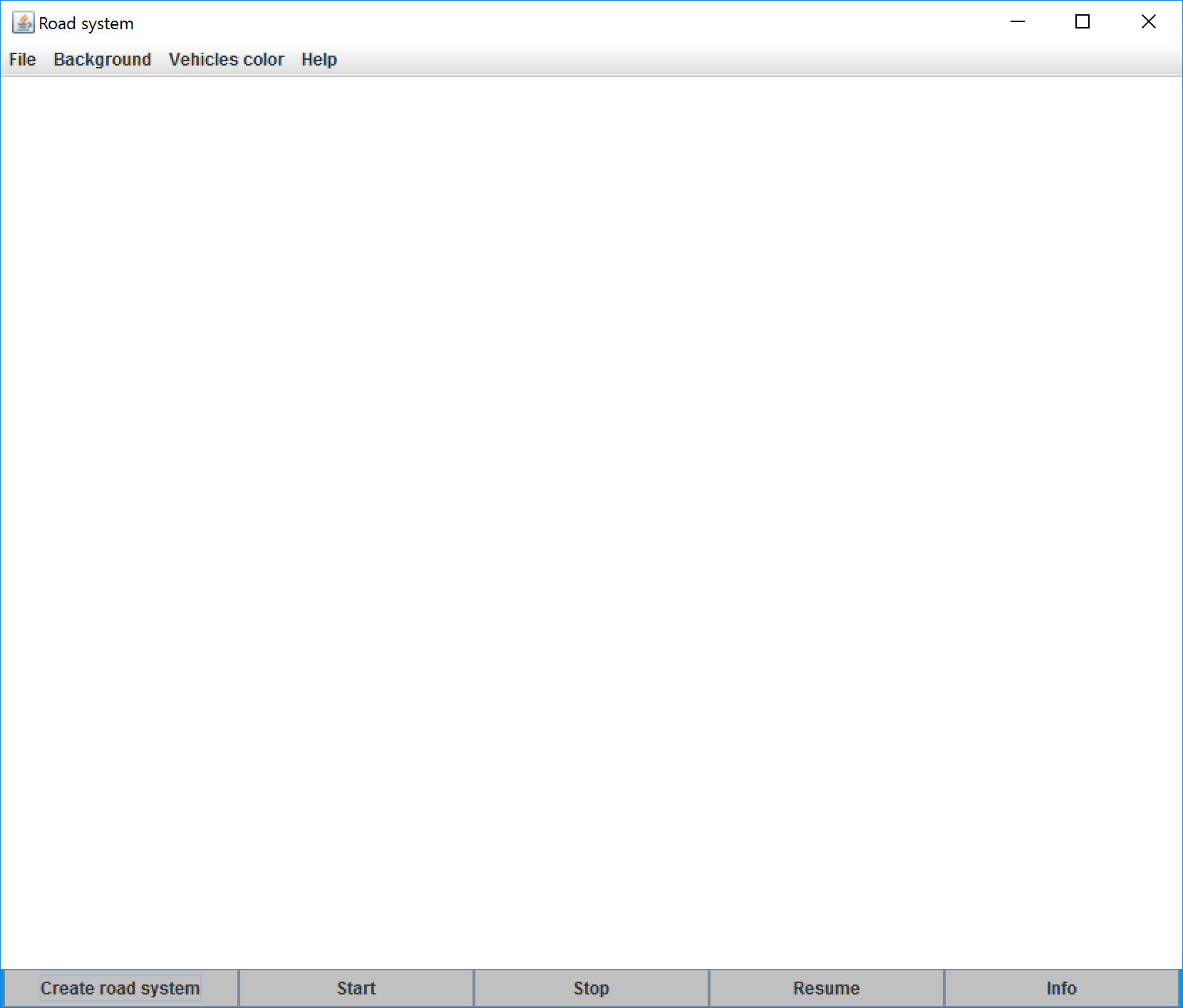
* ניתן להגיש עבודה זו בזוגות – רק אחד מהסטודנטים יגיש את העבודה במודל. בתיעוד של הקובץ יש לציין שם ות.ז. של מגיש\ים, בתוך תיעוד הjavadoc -
* לכל שאלה אנא עברו על מסמך ה[FAQ](https://docs.google.com/document/d/1Vw4qrZusj_L4Vm9z7VGecg6orfnonKR-eVoHZzj39a8/edit?usp=sharing) ובדקו האם היא נענתה כבר. במידה ולא, ניתן לפנות למתרגל האחראי – ילנה במייל elena.chk@gmail.com  
  **על כל פניה** להכיל את פרטי הסטודנט המלאים כולל ת.ז.
* חובה לתעד כל קובץ, מחלקה ופונקציה ע"י javaDoc  
  ניתן להיעזר [בתיעוד באתר oracle](http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html) או בקבצים הרלוונטיים במודל.

**דגשים לעבודה זו**

* העבודה מתבססת על עבודת הגשה 2 – עליכם לעדכן את הקבצים הקיימים ולהשתמש במחלקות שכבר בניתם. אפשר להוסיף שדות ומטודות לממשקים ומחלקות הקיימות.
* עליכם להוסיף GUI ותמיכה בהרצה מקבילית בעזרת ריבוי תהליכונים (multi-threading). בפרט:
  + לספק ממשק גרפי למשתמש ע"מ שיוכל לבנות את מערכת הכבישים ולהריץ סימולציה (ראה תיאור ותמונות לדוגמה בהמשך).
  + לאחר בניית מערכת כבישים (לפי הכללים של עבודה 2), משתמש יראה את כל הצמתים, כבישים, רמזורים ורכבים במצב ההתחלתי שלהם.
  + להפוך את הרכבים ורמזורים לתהליכונים ולהריץ אותם במקביל בסימולציה. כל רכב ורמזור צריך להיות תהליכון נפרד. גם המופע של Driving צריך להיות תהליכון שמפעיל בהתחלה את כל שאר התהליכונים ולאחר מכן מעדכן את הציור של מערכת הכבישים כל 100 מילישניות.
  + לאחר התחלת הסימולציה משתמש יראה את מערכת הכבישים עם כל הרכבים במיקום שלהם ואת כל הרמזורים במצב שלהם, שיתעדכן כל איטרציה (יחידת זמן של הסימולציה בהתאם לעבודה 2). יש להוסיף השהיה של 100 מילישניות לכל איטרציה ולצייר מחדש את המערכת הכבישים עם הרכבים כל 100 מילישניות.
  + יש להקטין את המהירות של כל הרכבים פי 10 על מנת שלא יעשו קפיצות בכביש בכל איטרציה ותנועת רכבים תהיה חלקה יותר.
  + הזמן ההשהיה של הרמזור בסימולציה יחושב לפי זמן השהיה של הרמזור מעבודה 2 כפול 100 מילישניות. כלומר, אם בעבודה 2 זמן השהיה היה 6 אז יש להשהות את התהליכון ל-600 מילישניות.
  + את ההמתנה של הרכבים בצמתים יש לממש בצורה יעילה (ללא busy waiting).
* על כל העבודה להיות פרויקט יחיד המחולק לpackages לפי המטלות.

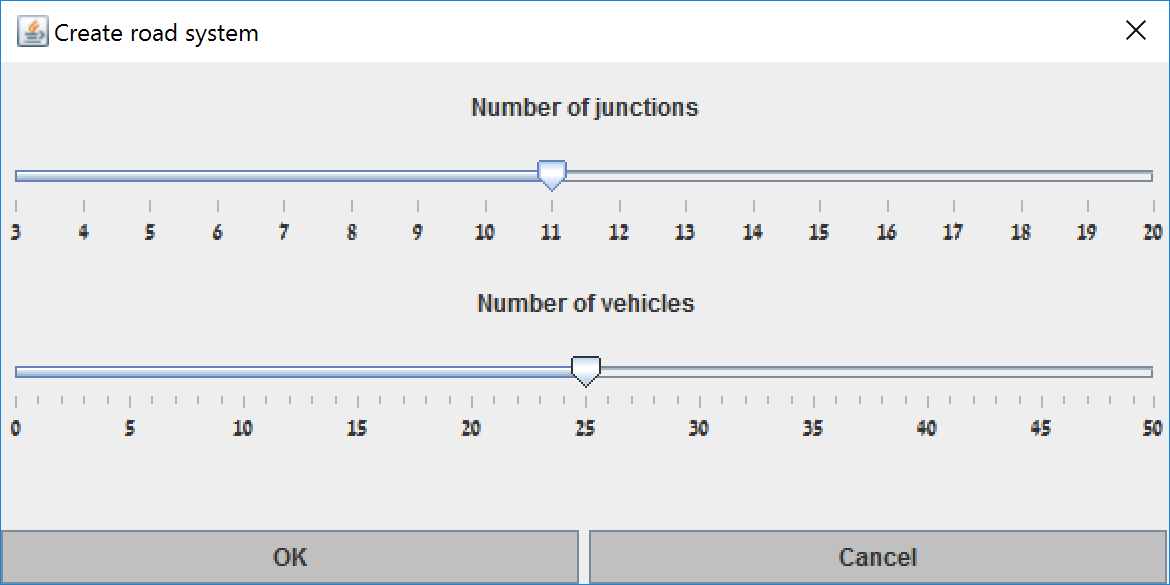
**תיאור של ה-GUI**

* מסך התחלתי – המידות של המסך ה-GUI צריך להתאים לגודל של מערכת הכבישים מעבודה 2.

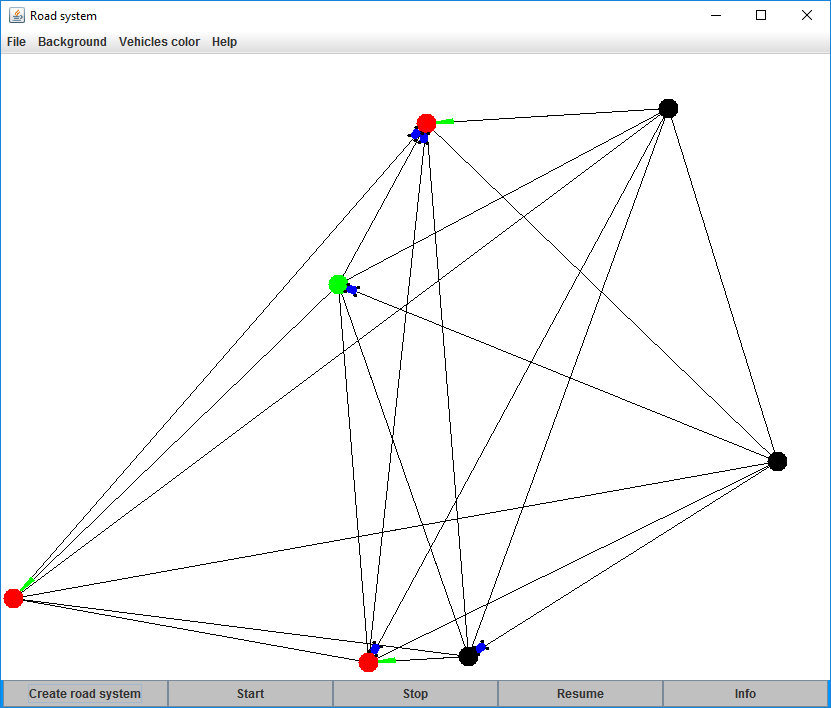


* על מנת להתחיל את הסימולציה יש לבנות תחילה מערכת כבישים ע"י לחיצת על הכפתור

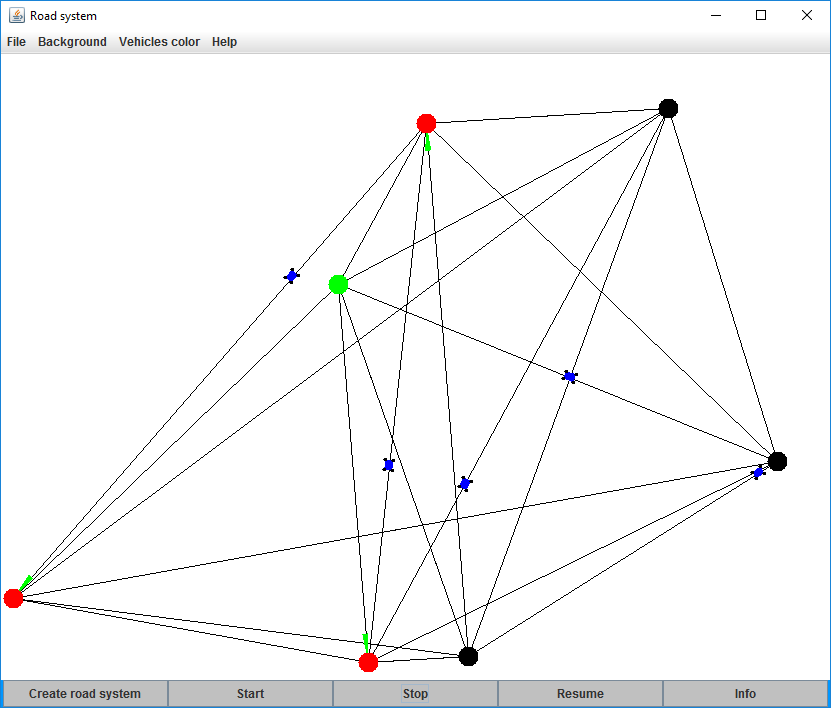
**"Create road system"**. לאחר לחיצת על הכפתור ייפתח חלון (דיאלוג) הבא עם שני slider-ים:



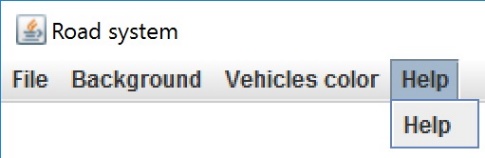
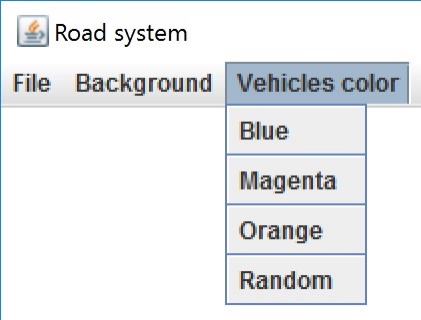
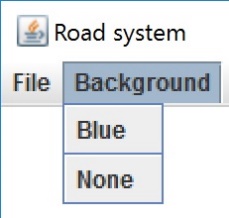
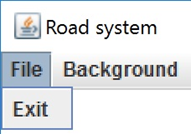
* הגדלים של הדיאלוג הם 600x300.
* הדיאלוג מאפשר לבחור כמות הצמתים בין 3 ל-20 וכמות הרכבים בין 0 ל-50.
* לאחר בחירת כמות צמתים ורכבים יש ללחוץ על OK.
* למשל, אם נבחר כמות הצמתים 7 וכמות הרכבים 5 אנחנו יכולים לקבל תמונה הבאה (מכיוון שכל המערכת נוצרת בצורה רנדומלית נקבל תמונה שונה כל פעם שנבנה את המערכת הכבישים):



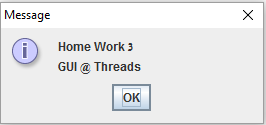
* את הצמתים יש לייצג ע"י עיגולים עם רדיוס 10.
* את הצמתים הרגילים יש לצבוע בצבע שחור, את הצמתים LightedJunction עם הרמזור כבוי יש לצבוע בירוק ואת LightedJunction עם הרמזור פעיל יש לצבוע באדום ואת הקצה של הכביש (שיש בו צבע ירוק) הנוגע בצומת יש לסמן במשולש ירוק.
* את הכבישים יש לייצג ע"י הקווים המחברים בין הצמתים.
* לפני תחילת ריצה של הסימולציה יש למקם את הרכבים בתחילת הכבישים ההתחלתיים שלהם, ליד הצמתים (כפי שנראה בדוגמא למעלה).
* את הרכבים אשר נמצאים על הכבישים יש לייצג ע"י מלבנים בגודל 10x8עם גלגלים שחורים בעלי רדיוס 2. שימו לב שיש לסובב את הרכב בהתאם לכיוון של הכביש. הקוד של הרכב מצורף בסוף הקובץ.
* את הרכבים אשר נמצאים בצמתים יש לייצג ע"י ריבוים רגילים (ללא כיוון) בגודל 8x8 עם גלגלים ולמקם אותם במרכז הרמזור. במידה ויש מספר רכבים באותו צומת אין צורך להדגיש זאת.
* צבע ברירת מחדל של הרכבים יהיה כחול (אפשר יהיה לשנות את הצבע בהמשך).
* על מנת להתחיל את הסימולציה צריך ללחוץ על הכפתור **“Start”**:



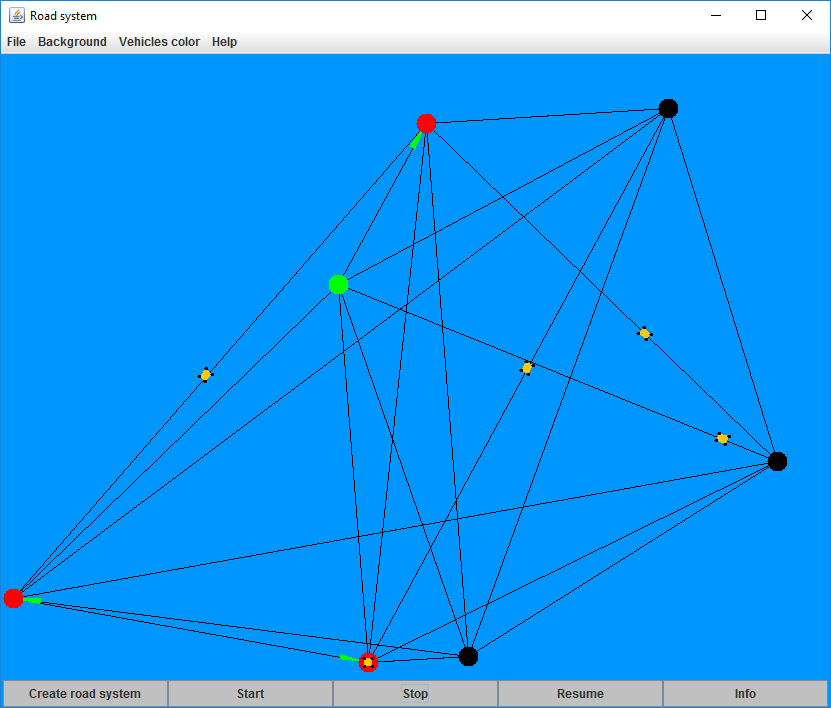
* לאחר תחילת הסימולציה, התהליכונים של הרכבים והרמזורים מתחילים לרוץ. הרכבים יישנו את המיקום שלהם והרמזורים יישנו את הסימון של הכביש עם הצבע הירוק (יש להקטין את המהירות ולעשות השהיה בהתאם לדרישות שפורטו בתחילת התרגיל). חוקי התנועה של הרכבים בכבישים וצמתים הם בהתאם לחוקים שהוגדרו בעבודה 2.
* כפתור **“Stop” –** עוצר זמנית את כל התהליכונים של המערכת כבישים. יש לממש בעזרת שימוש ב-wait().
* כפתור **“Resume”** – מחזיר את כל התהליכונים לרוץ. יש לממש ע"י שימוש ב-notify().
* אם לוחצים על **"Create road system"** במהלך הסימולציה יש לעצור את כל התהליכונים (בצורה תקינה ללא הריגת תהליכון) ורק אז יש ליצור מערכת רכבים חדשה.
* ב- **menu** יש להגדיר אפשרויות הבאות:



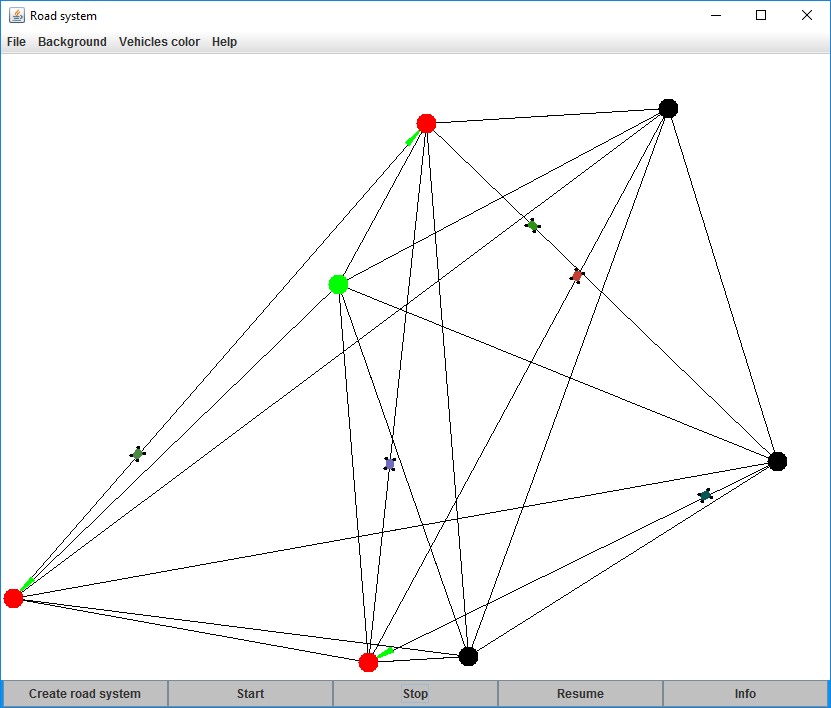
* **File -> Exit** – סגירה של התכנית
* **Background –** מאפשר לשנות רקע לכחול ולהחזיר ללא רקע.
* **Help** – לפתיחת דיאלוג בצורה הבאה:

****

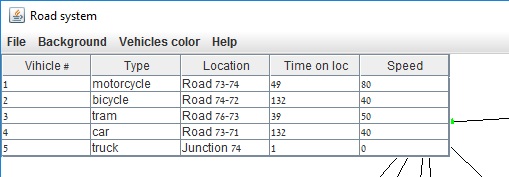
* **Vehicle color –** מאפשר לשנות צבע של הרכב לכחול, כתום, סגול לצבוע רכבים בצבעים רנדומליים. למשל אם נשנה רקע לכחול וצבע רכבים לכתום נקבל:



אם נחזיר רקע ללא צבע וצבעים רנדומאליים נקבל:



* כפתור **“Info”** – פותח טבלה עם מידע של כל רכב:
* מספר של הרכב.
* סוג של הרכב.
* מיקום של הרכב. אם מקום של הרכב הוא צומת אז צריך לציין את מספר הצומת, ואם מיקום של הרכב הוא כביש, צריך לציין את מספר הצומת ההתחלה והסוף של הכביש.
* זמן שהרכב נמצא ברכיב הנוכחי של המסלול שלו.
* מהירות של הרכב ברכיב הנוכחי.
* על מנת לסגור את הטבלה יש ללחוץ פעם נוספת על **“Info”.**



**נספח**

הפונקציה שיוצרת את הרכב במקום מסוים בכביש:

**private** **void** drawRotetedVehicle(Graphics g, **int** x1, **int** y1, **int** x2, **int** y2, **int** d, **int** h)

**int** dx = x2 - x1, dy = y2 - y1, delta = 10;

**double** D = Math.*sqrt*(dx\*dx + dy\*dy);

**double** xm = delta, xn = xm, ym = h, yn = -h, x;

**double** xm1 = delta + d, xn1 = xm1, ym1 = h, yn1 = -h, xx;

**double** sin = dy / D, cos = dx / D;

x = xm\*cos - ym\*sin + x1;

xx = xm1\*cos - ym1\*sin + x1;

ym = xm\*sin + ym\*cos + y1;

ym1 = xm1\*sin + ym1\*cos + y1;

xm = x;

xm1 = xx;

x = xn\*cos - yn\*sin + x1;

xx = xn1\*cos - yn1\*sin + x1;

yn = xn\*sin + yn\*cos + y1;

yn1 = xn1\*sin + yn1\*cos + y1;

xn = x;

xn1 = xx;

**int**[] xpoints = {(**int**) xm1, (**int**) xn1, (**int**) xn, (**int**) xm};

**int**[] ypoints = {(**int**) ym1, (**int**) yn1, (**int**) yn, (**int**) ym};

g.fillPolygon(xpoints, ypoints, 4);

g.setColor(Color.***BLACK***);

g.fillOval((**int**) xm1-2,(**int**) ym1-2,4,4);

g.fillOval((**int**) xn1-2,(**int**) yn1-2,4,4);

g.fillOval((**int**) xm-2,(**int**) ym-2,4,4);

g.fillOval((**int**) xn-2,(**int**) yn-2,4,4);

}

x1, y1 – מיקום של הרכב בכביש.

x2, y2 – מיקום של הצומת הסופי של הכביש.

d – אורך של הרכב (10)

h- חצי מהרוחב של הרכב (4(

**עבודה נעימה!!!**