

internal object design

מטרתנו הייתה ליצור צומת בה יש מספר רב של כבישים.
יש לנו consumer אחד, והוא "הרמזור". קצב הוצאת המכוניות הוא הזמן שהוא בירוק, והוא נקבע על ידי המשתמש ב-consumerRate.
מצד שני, יש לנו producer אחד והוא הרכבים שנכנסים. קצב הכנסתם לכל כביש נקבע על ידי המשתמש.
לכל כביש יש לו תור, וכל פעם נוצר אובייקט מכונית חדש.
הקוד שלנו עובד על מספר רב של מכוניות, אך להזיז את המכונית לכיוונה המתאים והזזת כל התור הייתה משימה קשה.
התמודדנו על deadlocks על ידי יצירת mutex לכלל נתיב. כל פעם ששיחררנו מכוניות נעלנו את המנעול, וכשנכנסו מכוניות בדקנו שהוא פתוח, נעלנו, הכנסנו עוד ואז שיחררנו את המנעול.

```
private void Producer()
{
    while (true)
    {
        foreach (var lineBuffer in lineBuffers)
        {
            string direction = lineBuffer.Key;
            Queue<Car> buffer = lineBuffer.Value;
            Mutex mutexLine = mutexDict[direction];
            mutexLine.WaitOne();
            for (int i = 0; i < producerRates[direction]; i++)
            {
                Car car = new Car(); // Create a new car object
                _uc.AddCar(car, direction);
                buffer.Enqueue(car); // Add the car to the line buffer
            }
            mutexLine.ReleaseMutex();
            Thread.Sleep(10);
        }
    }
}
```

הכנסת מכוניות לנתיב. לפני הקטע קוד הקריטי נעלנו אותו.
_uc.AddCar היא הפונקציה אשר יוצרת את הגרפיקה.

```
private void Consumer()
{
    while (true)
    {
        foreach (var cycleDir in cycle) //south to north, south to west
        {
            string dir1 = cycleDir.Value[0];
            string dir2 = cycleDir.Value[1];
            Queue<Car> buffer1 = lineBuffers[dir1];
            Queue<Car> buffer2 = lineBuffers[dir2];
            Mutex mutexDir1 = mutexDict[dir1];
            Mutex mutexDir2 = mutexDict[dir2];
            mutexDir1.WaitOne();
            mutexDir2.WaitOne();
            if (buffer1.Count >= consumerRate || buffer2.Count >= consumerRate)
            {
                for (int i = 0; i < consumerRate; i++)
                {
                    if (buffer1.Count > 0)
                    {
                        Car car = buffer1.Dequeue(); // Remove car from the line buffer
                        _uc.RemoveCar(car, dir1);
                    }
                    if (buffer2.Count > 0)
                    {
                        Car car = buffer2.Dequeue(); // Remove car from the line buffer
                        _uc.RemoveCar(car, dir2);
                    }
                }
            }
            mutexDir1.ReleaseMutex();
            mutexDir2.ReleaseMutex();
            Thread.Sleep(30);
        }
    }
}
```

כאן הוצאת המכוניות מהנתיב, ומחיקתן.
על כל אור ירוק רמזור יש שני נתיבים שיוצאים לדרך, כיוון לכיוון הנגדי וכיוון לכיוונו השמאלי.

באתחול הצומת ניתן להגדיר את הכיוונים שנשלחים כל פעם.

```
// Set theSizeMode property to adjust the image display
pictureBox.SizeMode = PictureBoxSizeMode.StretchImage; // Adjust the size mode as required

pictureBox.Size = pictureBox.Image.Size;

switch (direction)
{
    case "NorthToSouth":
        pictureBox.Location = new Point(300, 200);
        pictureBox.Image.RotateFlip(RotateFlipType.Rotate270FlipNone);
        pictureBox.Image = new Bitmap(pictureBox.Image, desiredHeight, desiredWidth);
        pictureBox.SizeMode = PictureBoxSizeMode.StretchImage;
        pictureBox.Size = pictureBox.Image.Size;
        break;
    case "NorthToEast":
        pictureBox.Location = new Point(330, 200);
        pictureBox.Image.RotateFlip(RotateFlipType.Rotate270FlipNone);
        pictureBox.SizeMode = PictureBoxSizeMode.StretchImage;
        pictureBox.Size = pictureBox.Image.Size;
        break;
    case "SouthToNorth":
        pictureBox.Location = new Point(465, 360);
        pictureBox.Image.RotateFlip(RotateFlipType.Rotate90FlipNone);
        pictureBox.Image = new Bitmap(pictureBox.Image, desiredHeight, desiredWidth);
        pictureBox.SizeMode = PictureBoxSizeMode.StretchImage;
        pictureBox.Size = pictureBox.Image.Size;
        break;
    case "SouthToWest":
        pictureBox.Location = new Point(400, 360);
        pictureBox.Image.RotateFlip(RotateFlipType.Rotate90FlipNone);
        pictureBox.SizeMode = PictureBoxSizeMode.StretchImage;
        pictureBox.Size = pictureBox.Image.Size;
        break;
    case "EastToWest":
        pictureBox.Location = new Point(500, 244);
        break;
    case "WestToEast":
        pictureBox.Location = new Point(200, 320);
        pictureBox.Image.RotateFlip(RotateFlipType.Rotate180FlipNone);
        pictureBox.SizeMode = PictureBoxSizeMode.StretchImage;
```

קטע קוד ליצירת המכונית על פי כיוונה. בגלל שהכיוון משתנה יש צורך לסובב את המכונית בהתאם לכיוון הנסיעה. ולאתחלה לתחילת הנתיב הנכון.