Faculty of Computers, Informatics and Microelectronics

Technical University of Moldova

WINDOWS PROGRAMMING

Laboratory work #5

Complex application: Traffic light

Authors:
Irina Ungureanu
Daniel Surdu

Supervisor:
Irina Cojanu

Laboratory work #5

Purpose of the laboratory

Collaboration and review of all previously studied things

Laboratory Work Requirements

- **Basic Level (grade 5 - 6) you should be able to:**

  a) Simulate the work of a traffic light for an intersection with 4 traffic lights.

  b) The user should be able to increase the speed of traffic light (interval when colors are changed)

  c) The user may be able to increase the number of generated random cars

- **Normal Level (grade 7 - 8) you should be able to:**

  a) Realize the tasks from *Basic Level.*

  b) In the simulation should be present 2 crosswalks with people passing.

  c) Random special cars crossing (at least2; ex. ambulance, president care, police car, etc)

- **Advanced Level (grade 9 - 10) you should be able to:**

  a) Realize the tasks from *Normal Level.*

  b) Introducing of the car accident generated by the wrong crossing of the car/pedestrian or presence of drunk driver

Laboratory work implementation

Tasks and Points

- Simulate the work of a traffic light for an intersection with 4 traffic lights.

Done by Daniel.

To do this I just created in Photoshop three traffic lights. One with the working green, yellow and red color. Then in android studio I created a timer and set to change Textures of the sprite after a period of time.

- The user should be able to increase the speed of traffic light (interval when colors are changed)

Done by Daniel.

To do this part of laboratory work I find how to intercept the tuches on the screen and after that I devided the screen in 4 parts. When I tuch the ritght up corner of the screen I encreese by some seconds the time of the traffic light interval and when I tuch the right down corner of the scree I decrease it.

- The user may be able to increase the number of generated random cars

Done by Daniel.

By increasing the number of generated cars I mean showing them on the screen. They was already created and when I encrease the number of cars I just show them up.

- In the simulation should be present 2 crosswalks with people passing.

In order to create the animation for the people, I imported 4 png images that upon rendering will simulate a person's walking. Then, I created a class called Pedestrian and its constructor takes the parameters: name of the texture, the directoin in which the person is going and the speed. Then, depending on the direction, I set the staring x and y positions. Also, here I choose randomly one from the two crosswalks the go in one direction. Moreover, I specify the orientation and rotate the sprite accordingly.

Secondly, I created the move function. This is also done according to the direction in which the person is going. I just set a new position by incrementing or decrementing it with the speed. And here I specify that a pedestrian can move only if he is not near the crosswalk with a red traffic light. Otherwise, he stops at the crosswalk. In this case, I also set a flag to false in order to stop the walk animation. Also, when the pedestrian leaves the screen it is replaced with another randomly generated pedestrian. Furthermore, in order to animate the walk, I change the textures of the pedestrian according to the current texture with a few if statements.

Thirdly, in the class that renders the animation, I declared an array of 4 pedestrians and made a function that generates them at random positions adn with different speeds. In order for the pedestrians not to run into each other I created a collision detection function. It creates a rectangle slightly longer than the pedestrian and checks if the two rectangles overlap using a built-in function. If the collision happens, I stop the pedestrian.

- Random special cars crossing (at least2; ex. ambulance, president care, police car, etc)

We decided to include in our animation an ambulance and a police car. For this, I created a new class called SpecialCar. In the constructor I just assign it a texture, set the position and the orientation. Then, in the move function there are no restrictions such as pedestrains or traffic lights, so it just performs the movement by incrementing the position with the speed.

Then, in order to make the pedestrians and cars stop when such a car passes I declared in both the Pedestrain and Automobile classes a flag. This flag, if set to true will not allow them to move. Then, when the car is out of the screen I set the flag to false and everything comes back to normal.

- Introducing of the car accident generated by the wrong crossing of the car/pedestrian or presence of drunk driver

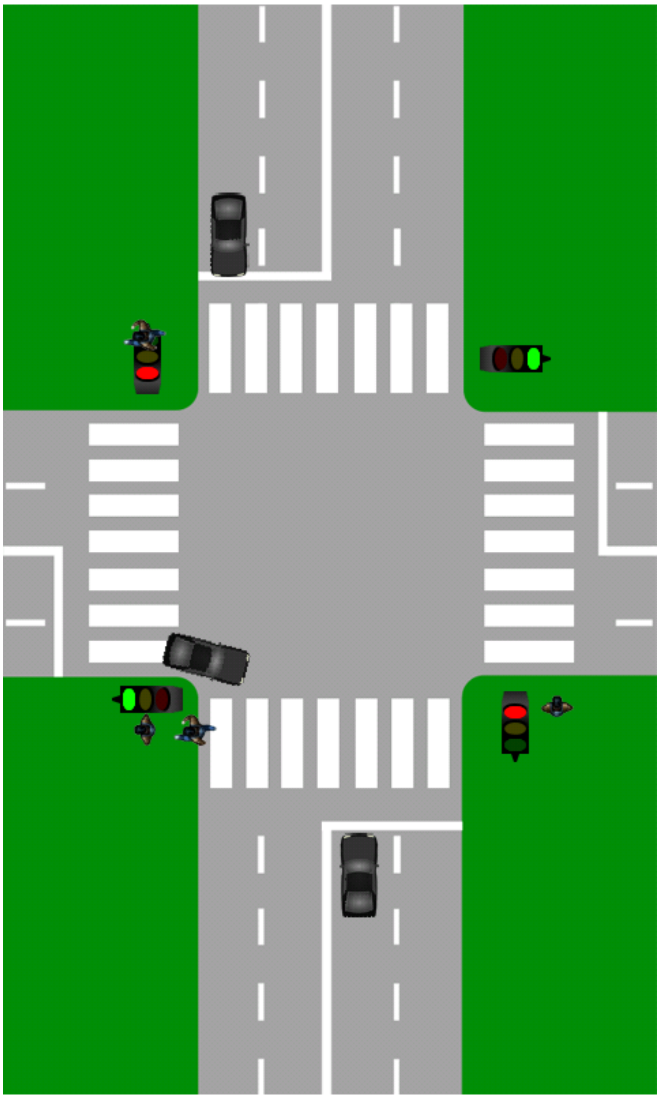
Done by Daniel.

This I realize by introducing an boolean variable "accident" It changes the status from false to true after about 40 seconds, and when it happensit generate a random number from 0 to nr_cars and this car take the parameter accident. After that this car start to move a lot faster tha others and it doesn't care about traffic lights. Whet it hits something it stops and stops the object that was hit. And because of this can be created traffic jams.

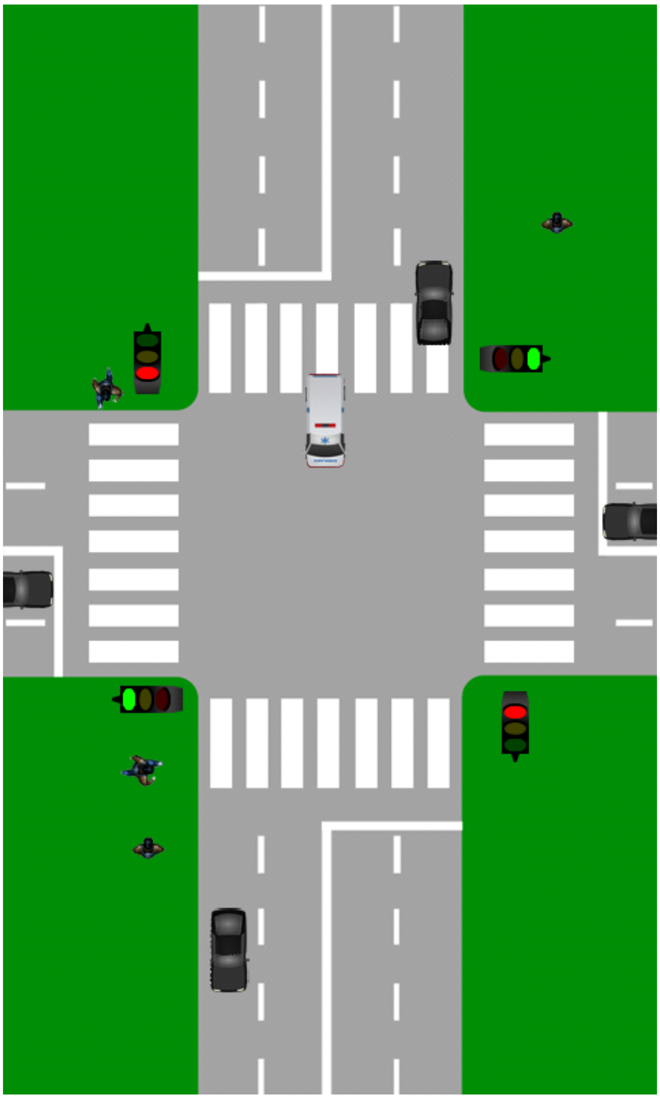
Laboratory work analysis

https://github.com/taurrielle/WPlab5

Prove your work with screens



The basic animation



Special car passing

Conclusions

During this laboratory work, I learned how to do a complex animation and how to implement a real-life model in a program. We used a gaming library so we also studied the basics of game developement. Moreover, we worked toghether, which gave us an idea of how we will do it in the future, being a part of a developement team.

References

1 libGDX, https://libgdx.badlogicgames.com/