Laboratory work 7

1 Objectives

The objective of this laboratory is to use the vectors and matrixes operations defined in previous laboratories, together with the transformation matrixes, in order to perform different transformations over a 2D figure.

2 Using SDL application

In order to exemplify graphically the 2D transformations, we will use an application very similar to the one from Laboratory 1, based on the SDL (Simple DirectMedia Layer) library.

3 SDL Renderer

SDL_Renderer is a struct that handles all rendering. It is tied to a SDL_Window so it can only render within that SDL_Window. It also keeps track of the settings related to the rendering.

In order to create a renderer related to the application window, we will use the following code:

```
SDL_Renderer *windowRenderer;

windowRenderer = SDL_CreateRenderer(window, -1, SDL_RENDERER_ACCELERATED);
```

There are several important functions tied to the SDL_Renderer:

1. **SDL_SetRenderDrawColor** sets the color that will be used in all drawing operations, until another call to the function is performed.

```
SDL_SetRenderDrawColor(renderer, r, g, b, a);
```

2. **SDL_RenderClear** clears the entire window display area using the current active color, previously set with SDL_SetRenderDrawColor function.

```
SDL_RenderClear(renderer);
```

3. **SDL_RenderPresent** will display everything drawn in the renderer to the screen. Until this function is called, all the drawing takes place in a hidden buffer that is not visible to the user. The call to SDL_RenderPresent should be made only once, after all the drawing functions have been called.

```
SDL_RenderPresent(renderer);
```

4 Drawing a line

Drawing a line using SDL Renderer requires the following steps:

1. Define the color that will be used (example: blue)

```
SDL_SetRenderDrawColor(renderer, 0, 0, 255, 255);
```

2. Define the start and end points of the line. For this we will use vec3 variables:

```
vec3 P1(100, 100, 1), P2(400, 100, 1);
```

3. Draw the line in the renderer

```
SDL_RenderDrawLine(windowRenderer, P1.x, P1.y, P2.x, P2.y);
```

4. Display the content of the renderer on the screen

```
SDL_RenderPresent(renderer);
```

5 Assignment

Download and run the application from the laboratory website. Try to understand the basic example and then extend the application with the following functionality:

- Include into the application your implementation files (.cpp).
- Define an initial rectangle with the top-left corner in P_1(100, 100) and bottom-right corner in P_2(400, 200).
- Rotate the rectangle around its center (diagonals intersection) by 10 degrees clockwise when RIGHT_ARROW is pressed and 10 degrees counterclockwise when LEFT_ARROW is pressed.
- Scale the rectangle having the top-left corner as a reference, using UP_ARROW and DOWN_ARROW keys.