TED ÜNİVERSİTESİ

CMPE360

Project 1

Compositing Images

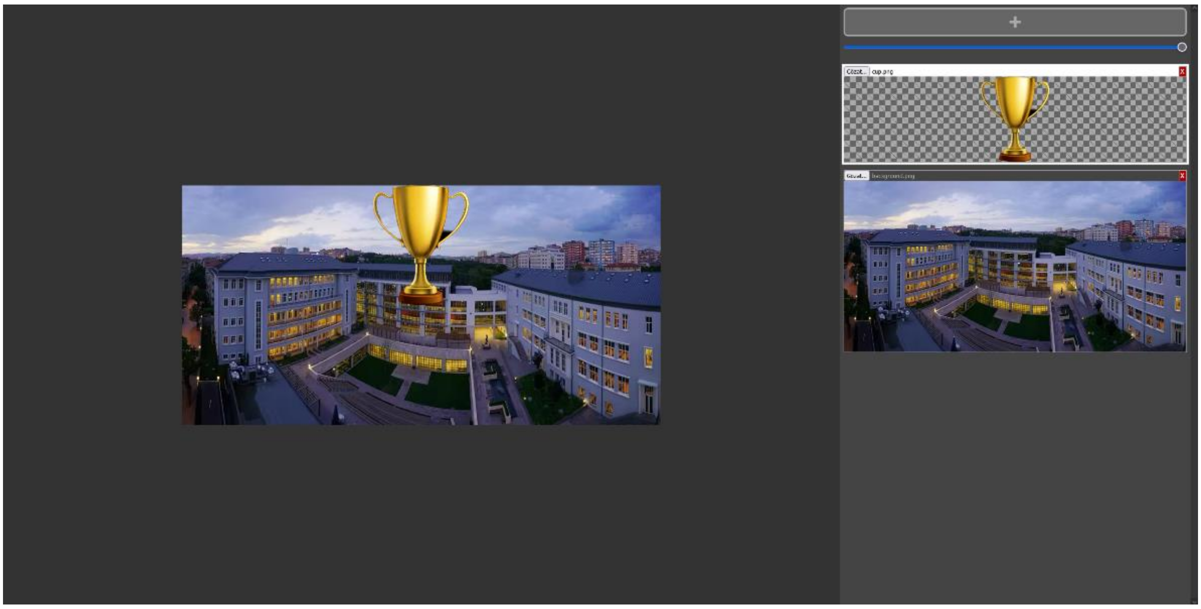
Section 02

Erkan Sancak

44293566706

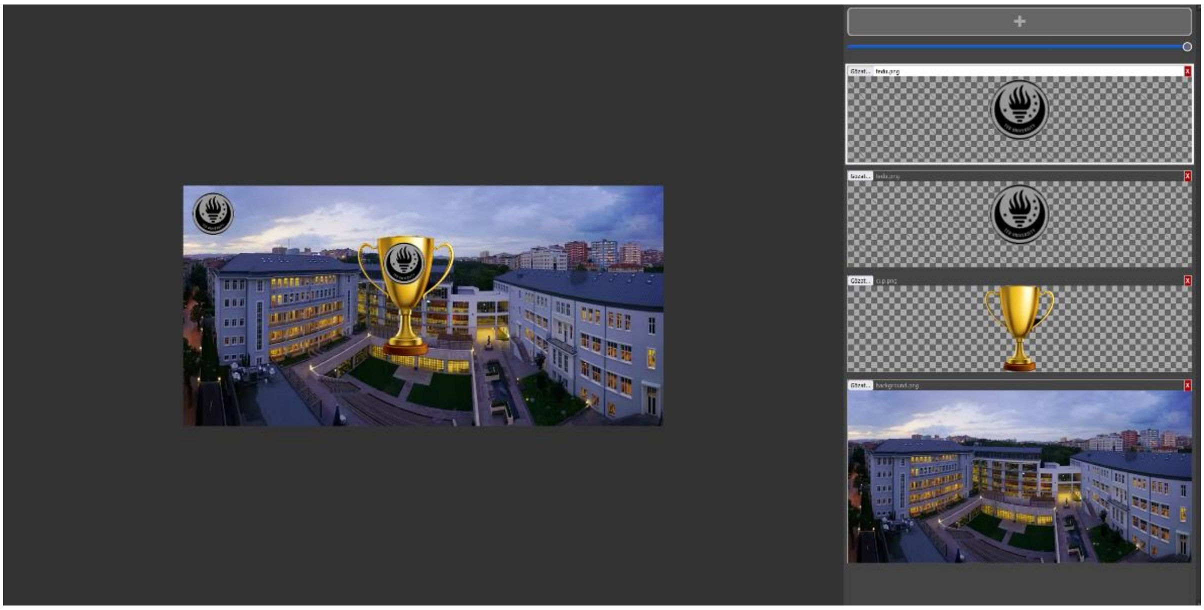
PART 1

- *image with cup.png and background.png:*



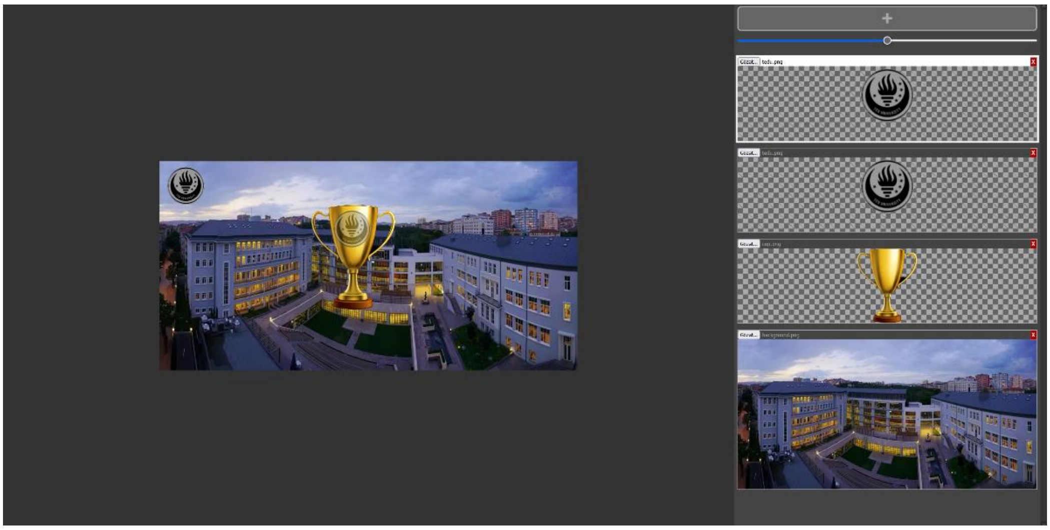
This process was done by uploading the background and cup png images and then placing the background png image at the bottom position on the site.

- *image with cup.png , tedu.png and background image:*



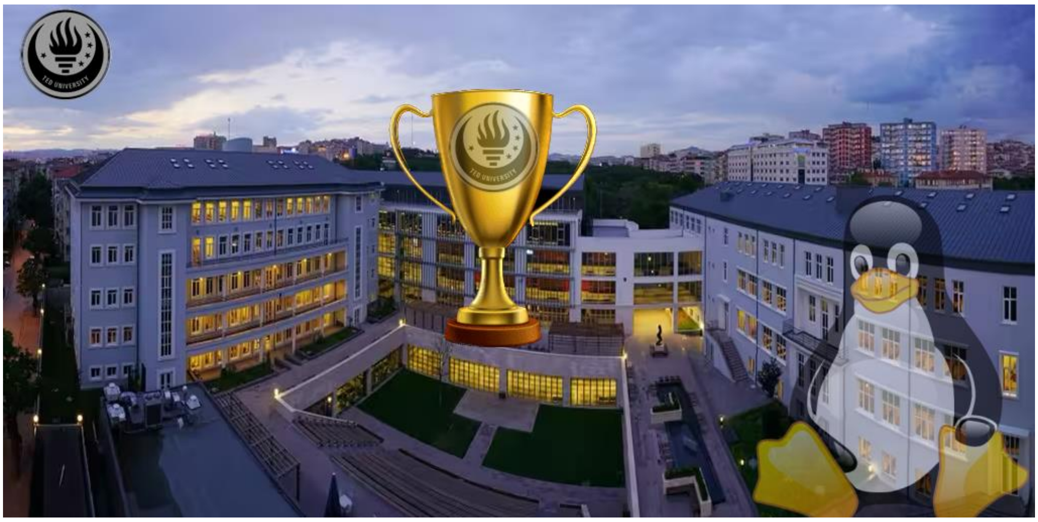
This process was done by uploading the background, cup and tedu png images, then placing the background png image at the bottom position of the site, then placing the cup png image above the background and placing the tedu png images at the top position.

- *image with cup.png, tedu.png and background image with changing transparency:*



This process was done as a step after the previous process by making a Tedu logo transparent with the transparency slider on the HTML site.

- *combined image by adding cup.png, tedu.png and background.png and any additional .png image by choice*



This process was done on a png image found on the internet using the same method as in the previous process.

PART 2

```
// BackGround is the background image to be changed.
// ForeGround is the foreground image.
// ForeGroundOpacity is the opacity of the foreground image.
// ForeGroundPosition is the The foreground image's location, measured in pixels. It can
be negative, and the alignment of the foreground and background's top-left pixels is
indicated by (0,0).

function composite(BackGround, ForeGround, ForeGroundOpacity, ForeGroundPosition) {
  var bgData = BackGround.data;
  var fgData = ForeGround.data;
  var width  = BackGround.width;
  var height = BackGround.height;

  for (var y = 0; y < ForeGround.height; y++) {
    for (var x = 0; x < ForeGround.width; x++) {

      var bgX = ForeGroundPosition.x + x;
      var bgY = ForeGroundPosition.y + y;
      var bgIndex = (bgY * width + bgX) * 4;
      var fgIndex = (y * ForeGround.width + x) * 4;

      if (bgX >= 0 && bgX < BackGround.width && bgY >= 0 && bgY < BackGround.height) {
        var alpha = fgData[fgIndex + 3] * (ForeGroundOpacity / 255);
        var beta = 1 - alpha;

        bgData[bgIndex] = Math.round(alpha * fgData[fgIndex] + beta * bgData[bgIndex]);
        bgData[bgIndex + 1] = Math.round(alpha * fgData[fgIndex + 1] + beta *
bgData[bgIndex + 1]);
        bgData[bgIndex + 2] = Math.round(alpha * fgData[fgIndex + 2] + beta *
bgData[bgIndex + 2]);
        bgData[bgIndex + 3] = Math.round(fgData[fgIndex + 3] + beta * bgData[bgIndex +
3]);
      }
    }
  }
}
```

- The function ***composite*** takes four arguments: ***BackGround***, ***ForeGround***, ***ForeGroundOpacity***, and ***ForeGroundPosition***.

- ***bgData*** and ***fgData*** stores the data of the background and foreground images. In addittion to that height and width stores dimensons of the images.

- The purpose of the ***for loops*** is to process each pixel in the foreground image and blend it with the corresponding pixel in the background image, taking into account the opacity and position of the foreground image.

- ***bgX*** represents the X-coordinate of the pixel in the background image. ***bgY*** represents the Y-coordinate of the pixel in the background image. ***bgIndex*** is used to access the correct position in the background image ***(bgData). (bgY * width + bgX) * 4*** calculates index in background image. **4** is for R,G,B,A. ***fgIndex*** is used to access the correct position in foreground image ***(fgData).***
- The condition ***if(bgX >= 0 && bgX < BackGround.width && bgY >= 0 && bgY < BackGround.height)*** is used to check whether the current pixel in the foreground image is within the bounds of the background image

- The last part of the code calculates the alpha and beta values to perform alpha blending, ensuring that the foreground pixel is combined with the background pixel based on their alpha values and the user-defined opacity.