CS 153 Assignment 2

Due: 10 February, 2025

Introduction to Image Composition

This is the second assignment for CS153 in Spring 2025. In this assignment you will be asked some conceptual questions focusing on image formation, as well as gain further practice with image handling and image representation, particularly through the use of alpha channels.

The conceptual questions will be submitted directly on Gradescope, and are worth 30 points total. Two of the written questions are directly related to coding component of this assignment, including submitting your own original composition. The code submission detailed in this document is worth 30 points.

Question 1: Composite Images

[30 points]

In class we saw a green screen demo where we extracted a target object from an image that was taken in front of a green screen, using colour logic to create an alpha channel. For this question we will explore a slightly simplified version of this task where the alpha channel is already provided to you, but your job will be to implement further downstream image handling to allow more control over the appearance of the composite image created.

You will be provided with a set of background images in the `bkg` folder, a set of character images in the `chars` folder, and a set of objects in the `objs` folder. Note that all images in the `chars` and `objs` folders are in `PNG` format, and include a transparency channel.

Write a function which places an element (character or object) into a specified location of a background image at a specified scale. Location will be specified as the center of the minimal bounding box of the element in *normalized image scale* coordinates. Normalized image scale coordinates are drawn from the range $[0, 1]$, and are then scaled to a pixel location based on the width and height of the background image. Scale will similarly be specified by the height of the element in normalized image scale (the width should be calculated such that the original aspect ratio of the element is maintained). Elements may have positions specified such that they extend beyond the bounds of the background; your function should not crash in this case, but rather should truncate the element to the bounds of the background image.

Your function should take the form `img_out = place_object(bkg_img, element, elmask, location, height)`, where

- `img_out` is an RGB image containing the composite output
- `bkg_img` is an RGB image into which the element should be inserted
- `element` is an RGB image containing the content channels of the element to be inserted
- `elmask` is the single channel alpha map corresponding to the `element` image
- `location` is a tuple in (x, y) format which specifies the horizontal and vertical position, respectively, in normalized image scale coordinates to which the centroid of the minimal bounding box of the element should be inserted
- `height` is a parameter which specifies the height of the minimal bounding box for the element in normalized image scale. The element's width should be scaled accordingly to maintain its aspect ratio

You are provided with a helper function `load_alpha` that takes an image path and returns the separate content and alpha channel images. Please note that if you call this on an image file that does *not* contain an alpha channel, it will throw an error. You can assume that any object or character image provided to your program will contain an alpha channel.

Note: all of the provided images have a common theme drawn from fantasy themed tabletop roleplaying games, but that is by no means the only theme you could use these tools on! Feel free to experiment and explore your own creations!