**Drawing Houses with Turtle Design Document**

**Overview:** Our team desires to find sleek, clean solutions to the current lack of drawn houses. We want to develop reusable functions that can even be used to make other structures, such as office buildings, or even forests. This solution employs the turtle library in python to draw a house with doors, windows, garage doors, and trees. A variety of functions were defined providing primitive shapes that are used to draw the desired entities.

**Background:** Turtle is a simple drawing tool built-into Python. We will use Turtle to create a house-image generation tool. The program offers a reliable, reproducible alternative to drawing a house by hand.

**Goals:** Create a simple house using the Turtle library in Python. Do so by defining a set of functions that draw primitive shapes with Turtle. Call these primitive functions in functions that define higher-level objects in the house scene (e.g., windows, trees, doors, etc.). Finally, define a main module that leverages all the higher-level modules to draw a scene.

**Non-goals:** It can be tempting to add additional objects to the house scene (e.g., a sun, a car, a person). However, to limit the scope of this project, we will not draw any other structures besides those requested in the assignment.

**Future Goals:** There are a lot of features of the house that will not be editable during the first iteration of the software. For example, the color of the house, roof, trees, and cloud are fixed. During future development of this house-drawing software, we can adjust our code to accommodate more customization of the house scene, allowing the user to specify colors, types of trees, types of clouds, etc.

**Detailed Design:** The house scene should include the following components:

* One house
* One front door
* Two garage doors
* Four windows
* Two trees
* One cloud

We will draw these components in a scene with green grass and a blue sky. The scene will be drawn using the Turtle package in Python. To begin, we will define functions with two primitive shapes: a square and a circle. We will use these primitive shapes to draw higher-level entities in the scene. Higher-level entities will be defined in their own functions, including ones for a door, a window, a garage door, a house frame, and a tree. Each higher-order entity will be defined with an anchoring position, specified in its function’s docstring. Finally, we will stitch together the entities of the house using their functions to form a house scene in a main file called `draw\_scene.py`. We will decompose functions into basic, self-contained steps and will document the functions to make them easier to understand. The application needs to be highly secure and private, so we will create all functions using built-in Python libraries and without requiring use of the internet.

**Python environment:** We will include a `make.sh` file that will conveniently install the Python 3.9.12 environment required for drawing the house.

**Third Party Dependencies:** This function does not have third-party dependencies outside of Python 3.9.12 and its built-in libraries.

**Work estimate:** 6 hours for each of four teammates, for a total of **20 labor hours**. This time includes response to code review, but does not include the code review itself.

**Alternative approaches:** We could have used the R Turtle library instead of the Python library. Alternatively, we could have used GUI-based drawing tools like AutoCAD and Adobe Illustrator to draw the house. We could have alternatively drawn the house by hand with pencil and paper; however, this option lacks the reproducibility of our Turtle code.

**Related work:** DALL-E is an AI-based image generator by OpenAI that draws scenes from human-readable text. DALL-E could take in a description of the house scene and can make the scene from scratch.

**Designation of work:**

Design Document – Joe and Thomas

Make file and dependency management – Joe

Primitive Shapes (circle and square) – Thomas

Door and garage door functions – Thomas

Window functions – Joe

Tree and cloud function – Ariana

Roof and base – Joe, Angelika

Make scene – Joe, Thomas, Ariana, Angelika

Video – Ariana and Angelika