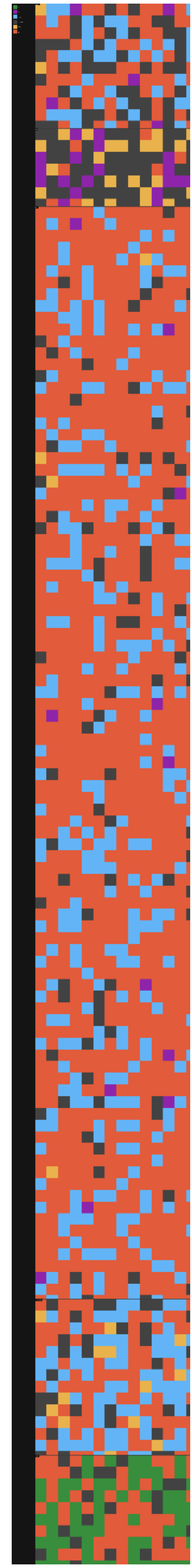
IAT 339 DIRECTED STUDIES REPORT

Phase 2: Diffusion Style Generation

Author: ZhongJun Qian
Student ID: 301541827
Instructor: Andrew Hawryshkewich

Abstract

Table of Contents



Phase 2 builds on data from Phase 1 by transforming spatial and semantic band data with RG encoding and semantic band labeling into fully procedural layouts. Using adjacency probabilities and element-geometry statistics derived from the wrangled data of Phase 1, I was able to create a modified Wave Function Collapse engine that generated noise, which I then used deterministic denoising passes to produce semantically meaningful design wireframes from which the website layout can be derived.

Introduction

Background and Aim

Phase 1 established a deterministic RGB capture system capable of translating webpage structure into analyzable spatial data. That phase demonstrated what real layouts look like statistically.

The goal of Phase 2 is not simply to randomize existing components but to test adjacency and proportion rules alone can reconstruct design through the denoising of probabilistically populated noise.

WFC as Generative Logic

Wave Function Collapse (WFC) is a probabilistic constraint-solving algorithm first developed for image texture synthesis. Instead of drawing pixels, it fills a discrete grid one cell at a time while ensuring that every tile’s neighbours follow statistical adjacency patterns learned from real data. In this project, each tile represents an abstract interface element such as text, media, or a button.

The method is intended to maintain layout coherence, meaning that adjacent tiles form visually and semantically compatible blocks. During early tests this expectation held only at a local level. Each cell obeyed its adjacency rules, but the overall page structure broke down. Headers mixed with main content, and footers appeared near the centre of the canvas. The result looked like coloured noise rather than a layout. This proved that adjacency data alone cannot express hierarchy or rhythm. Figure 1 illustrates this failure and explains why later stages of the pipeline include deterministic correction.

From Noise to Meaning through Hybridization

The collapse of hierarchy in the first trials led to a hybrid solution that combines probabilistic generation with deterministic correction. WFC still provides the initial noise lattice, which captures the statistical diversity of real layouts. A sequence of denoising passes then interprets the lattice in context, separating bands, removing improbable adjacencies, and expanding consistent regions. These operations convert statistical texture into a recognizable wireframe. WFC supplies variation and authenticity, while the deterministic passes restore readability and hierarchy. This combination mirrors the way human designers explore options and then refine them into purposeful structure.

Hierarchical Pipeline Architecture

The final generator is divided into three connected stages:

Stage 1: Probabilistic Generation.

Each semantic band: Header, Navigation, Main, Supplement, and Footer are solved separately with its own adjacency matrix and probability distribution. This maintains diversity within each band while preserving vertical order.

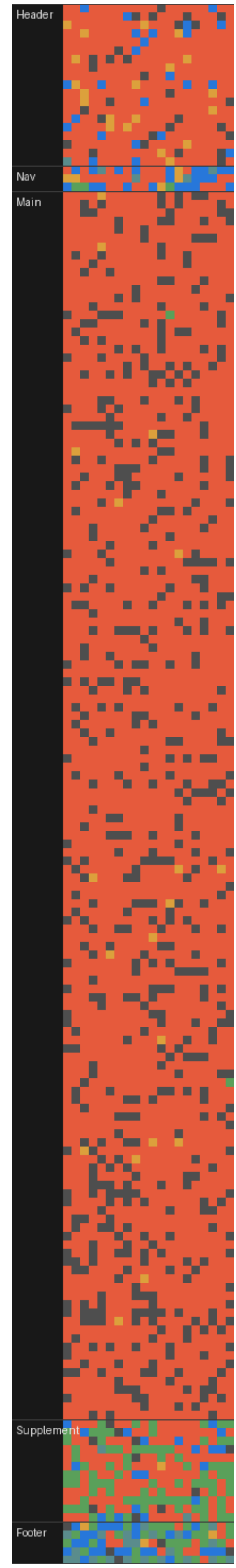
Stage 2: Zone Orchestration.

The raw tile grid is grouped into functional zones such as logo areas, hero banners, navigation rows, and card grids. Rule files derived from common web patterns from the data gathered in Phase 1 define how many zones appear and how they divide horizontal space. Figure 2 to the right shows the identification of major semantic bands

Stage 3: Element Population and Geometry Guidance.

Each zone is populated with child elements whose sizes and counts follow median width, height, and density values measured in Phase 1. This stage ensures proportion, spacing, and rhythm consistent with real interfaces.

These three layers translate statistical adjacency into an interpretable hierarchical composition while keeping the process procedural and repeatable.



Methodology

Rules and Dataset

The data collected in Phase 1 contained labelled screenshots that showed where text, images, and buttons usually appear. From these images, I calculated how often different element types touched each other and how much vertical space each band used.
These counts were turned into probability tables that describe common spatial relationships, such as text appearing under a header or images beside text. This gave the generator a simple rulebook based on real layouts rather than manual design choices.

Creating the Design

The Wave Function Collapse (WFC) algorithm used these probabilities to fill a blank grid with element labels. Each grid cell chose its label by checking what surrounded it so that local patterns matched the data. At first, this created coloured noise instead of layouts. As seen in Figure 1. Bands mixed together and nothing looked organised, showing that adjacency rules alone cannot form a proper page. To fix this, each band was solved separately, and small tolerance settings were added so the algorithm could finish without getting stuck. This made the results more stable and produced clear bands that roughly followed real website structure.

Solving the grid at full size caused messy detail and slow results. Running it at a smaller scale first, then enlarging it, kept the overall shape correct while removing fine-grain noise. This step improved speed and gave cleaner base layouts for the later stages since upscaling with a deterministic function is faster compared to running WFC at higher resolution.

The cleaned grid was then grouped into larger “zones” that represent real interface parts such as logo areas, navigation bars, or hero sections. Rules stored in a small configuration file defined how these zones should be arranged within each band. This step added the missing hierarchy that the first WFC passes lacked, turning random clusters into recognizable website sections.

Finally, each zone was filled with smaller elements using average sizes measured from the Phase 1 dataset. Text blocks, images, and buttons were placed in reasonable numbers and proportions. This stage produced layouts that looked balanced and readable instead of crowded or empty. Quality checks measured how much of each band was filled and whether spacing matched typical sites. The final outputs showed consistent hierarchy and proportion across multiple random seeds. Figure 3 shows the final wireframe results.

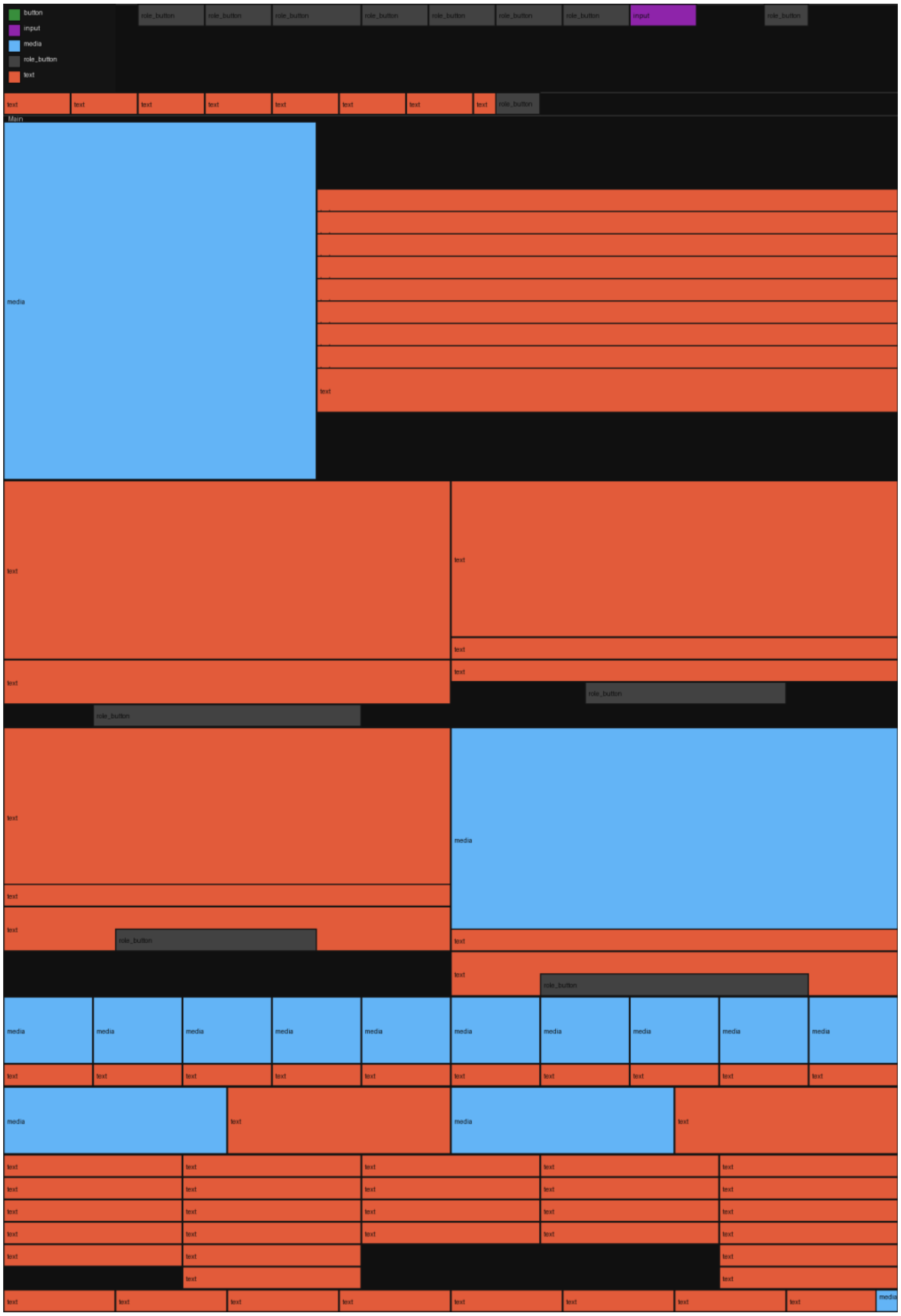


Figure 3

Results

Outputs

The generator produced complete multi-band layouts from the Phase 1 dataset.
Each output contained a header, navigation row, main content, supplement, and footer.
The early WFC pass looked like random coloured noise, confirming that adjacency rules alone cannot form structure.

Design Coherence

The final layouts displayed patterns similar to those found in the reference dataset.
Headers stayed near the top, main sections occupied most of the vertical space, and footers remained consistent in width and alignment. Text elements filled central areas, while media blocks appeared near upper or middle regions, echoing common documentation-style pages.
Coverage checks confirmed that between 70 and 85 percent of each band contained expected element types, as confirmed in Figure 3.

Variation Across Seeds

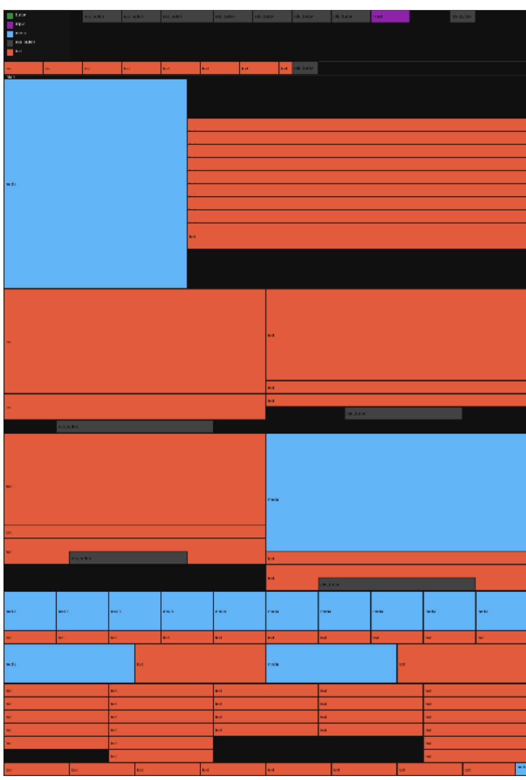


Figure 4

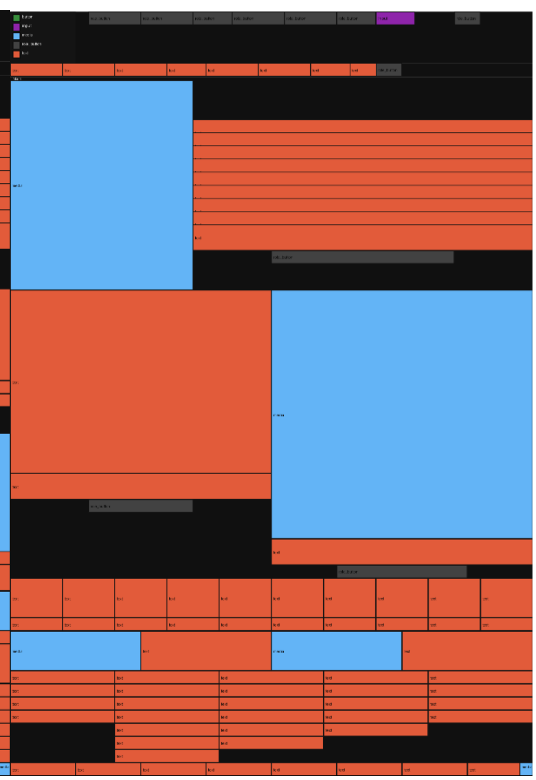


Figure 5

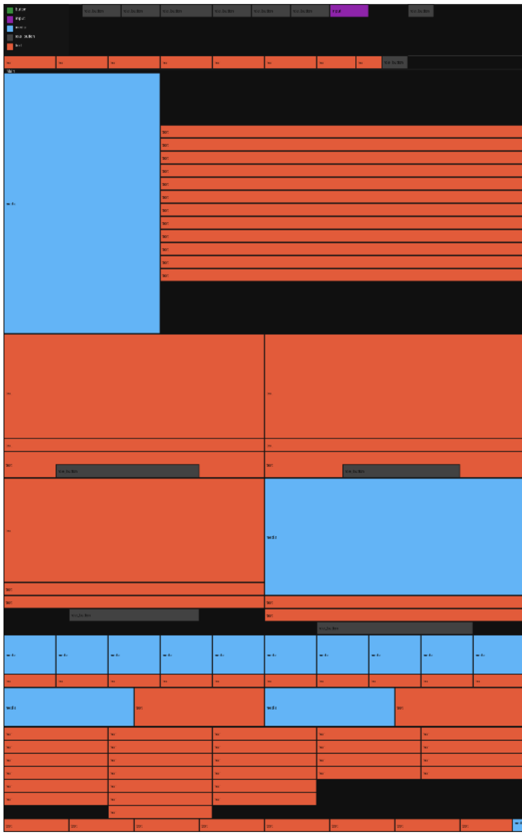


Figure 6

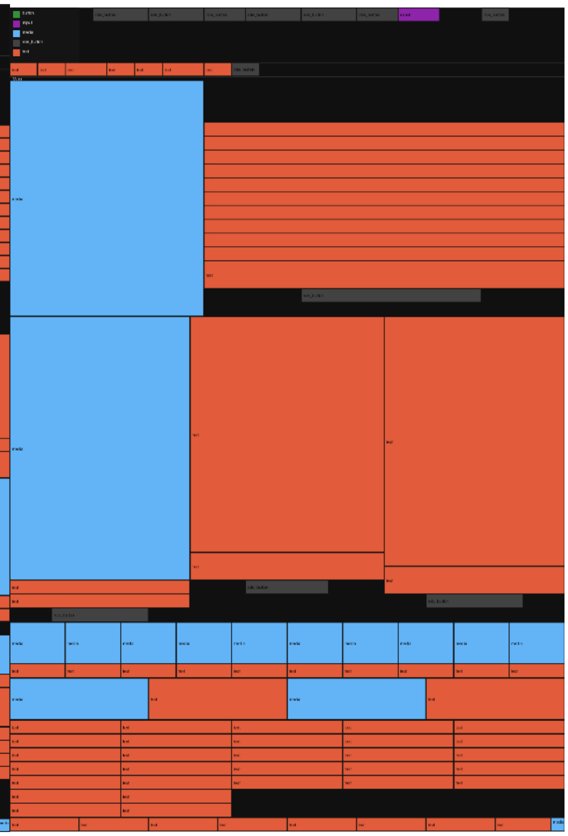


Figure 7

Figure 4-7 shows the variance of seed of WFC does generate variable layouts as is expected from the product of deterministic denoising and probabilistic WFC noise generation.

Limitations and Future Work

Current Limitations

Although the current pipeline generates layouts that look structured, it still relies on fixed statistical rules. The system understands where elements usually appear but not *why* they appear there.

Because of this, it cannot yet evaluate design quality beyond geometric balance.

Content within each class, such as the meaning of text or the tone of imagery, is not considered.

The model also assumes that all reference sites share similar visual grammar, which may reduce diversity when applied to different design styles.

Future Work

Future stages will address these limitations by introducing semantic reasoning and adaptive rule weighting. Text and image generation models could supply realistic content for each placeholder, allowing future tests to measure both visual and semantic plausibility. Additional metrics, such as alignment accuracy and white-space balance, can provide more objective evaluation of design quality. Improving the orchestration rules to handle responsive layouts would also bring the results closer to real web practice. These changes will make the system capable of generating complete, data-informed phishing pages that blend both structural and semantic realism.