

Escape From Ganivi

Saba Kandashvili, Saba Shengelia, Saba Tsikarishvili, Valiko Bugadze

e-mail: saba.kandashvili981@ens.tsu.edu.ge, saba.shengelia647@ens.tsu.edu.ge,
saba.tsikarishvili771@ens.tsu.edu.ge, valiko.bugadze597@ens.tsu.edu.ge

^a Computer Science, Faculty of Exact and Natural Sciences, Tbilisi State University, 1 Ilia Tchavtchavadze Avenue, Academic Building

^b Pinky Punishers, Address 2 University St, Tbilisi

This paper discusses the software architecture of "Escape from Ganivi," a Survival Horror video game inspired by the chaotic and labyrinthine architecture of TSU Building 11. The primary challenge of the project is to simulate a disorienting, infinite environment within a limited digital space, a task that standard Euclidean geometry and traditional game engine tools (Unity) struggle to address efficiently. To overcome these limitations, the team developed a hybrid generation system utilizing three distinct custom algorithms: CGSME (Coherent Guided Stochastic Maze Engine), WWFC (Weighted Wave Function Collapse), and SHMG (Stochastic Hybrid Maze Generator).

These algorithms are implemented in C for high performance and rely on specific stochastic distribution models to ensure structural variety and logic. The mathematical foundations for these generation models are described below:

CGSME:

$$\mathcal{G}_{final} = K \left(\Psi \left(\bigcup_{z=0}^H \left(D_p^+ \circ C_{max} \left(\left\{ (x, y) \in \mathbb{Z}^2 \mid (1 - |2 \cdot P(\vec{v} + a \nabla P(\vec{v})) - 1|)^2 \geq \Theta(p)\right\} \right) \times \{z\} \right) \right) \right)$$

WWFC:

$$d(x, y)_i = D_{max} - \sqrt{(x - x_i)^2 + (y - y_i)^2}, \quad P(x, y)_i = \frac{d(x, y)_i}{\sum_j^n d(x, y)_j}$$

SHMG:

$$L_{t+1} = \{ y \notin \bigcup_{k=0}^t L_k \mid \exists x \in L_t, V(x, y) = 1 \}$$

Spatial limits are resolved via a Non-Euclidean Portal system using relative transformation matrices to create a "Tardis Effect." Visually, custom shaders employ dynamic projection algorithms to render complex 3D depth onto 2D surfaces, governed by the following projection logic:

$$UV_{final} = \frac{(M_{proj} \cdot M_{view} \cdot P_{world})_{xy}}{2w} + 0.5, \quad P_{n+1} = P_n + f(P_n) \cdot D$$

Additionally, a "Native Hot-Reload" pipeline was developed to bypass Unity's DLL locking, streamlining the C workflow. The result is a high-performance engine capable of generating unique, real-time surreal horror experiences.

References

- [1] M. Gumin, "WaveFunctionCollapse," GitHub repository, 2016
- [2] L. Alkhazishvili and A. Gamkrelidze, "Design of Algorithms," Lecture Notes, Faculty of Exact and Natural Sciences, Ivane Javakishvili Tbilisi State University, Tbilisi, Georgia.
- [3] L. Nyhoff, ADTs, Data Structures, and Problem Solving with C++, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2005.