Author: Arshiya Shaik & Priyanshu Guha

Product Backlog – Keys of Survival

Advanced Programming Topics

1. **Randomized Map Generation:** Algorithms to generate fair, unpredictable sequences of doors, keys, and zombies.
2. **UX (User Experience):** Intuitive controls, HUD clarity, visual feedback, and smooth game flow.

1. Randomized Map Generation

Topic Description:
This feature focuses on making algorithms that can generate **fair, unforeseeable sequences** of doors, keys, and zombies. The goal is to prevent repetition while keeping balance, speed and avoiding unplayable sequences of spawns in each run of the game. This involves exploring **procedural generation**, **difficulty curves**, and **seed-based randomization**.

Research Sources:

- Orcun Nisli. "*Endless Runners: Procedural Map Generation and Difficulty Curves*." *Orcunnisli*, 19 Oct. 2015, www.orcunnisli.com/post/2015/11/22/endless-runners-procedural-map-generation-and-difficulty-curves. Accessed 23 Oct. 2025.

- Unity Technologies. "*The Generation of Coins and Obstacles in Subway Surf*." *Unity Discussions*, 15 Mar. 2023, discussions.unity.com/t/the-generation-of-coins-and-obstacles-in-subway-surf/258316. Accessed 23 Oct. 2025.

- Gamesbeat. "Unity Endless Tutorial 0 Introduction [Tutorial][C#]." *Youtube.com*, 16 Mar. 2016, www.youtube.com/watch?v=7jdL5538bEo. Accessed 23 Oct. 2025.

Research Process & Insights:
These sources were found through keyword searches such as *"random generation endless runner algorithms"* and *"random map generation game balance"* on Google. We also took some inspiration from existing games like "Subway Surfers" and "Google Dinosaur Game" and researched Github repositories of their remakes.
Through these materials, we learned about:

- **Chunk-based generation**, where the game world is built using repeatable but randomized "segments"

- **Weighted randomness,** it was ensured that despite the random spawns of sprite, the game was still fair and playable by making sure no unfair sequences show, for e.g a door followed by a zombie immediately or impossible door-key combinations.
- **Difficulty scaling curves**, gradually increasing challenge as the player plays and the speed and reaction demands increase.

Influence on Design of Final Game:
The research directly influenced our implementation strategy:

- We designed a **randomised algorithm** that dynamically adjusts based on player progress i.e via speed and difficulty curves.
- We ensured **key-door pairing logic** is balanced (no unwinnable runs) and other such unfair sequences that could appear in a completely randomized algorithm.
- A **random seed** system was added for testing reproducibility.
- We also used a randomised algorithm for the mini games appearing at each obstacle. As there were 4 mini games (excluding the one at the zombie collision), they were randomly chosen for each door collision, but with an algorithm to ensure the player got to play all 4 games and not just one indefinitely.

Depth of Understanding:
We now understand both:

- The **theoretical principles** (randomization algorithms, procedural noise)
- The **practical challenges** of generating endless maps that feel designed yet unpredictable without any **unfairness** for the player of the game.

2. UX (User Experience)

Topic Description:
This topic focuses on creating a **visually clear, interactive and responsive** gameplay experience. This includes refining controls, HUD layout (hearts, score, keys), sound effects and visual feedback (e.g., "SPEED UP!" messages, gradients, mute/unmute options and animation smoothness).

Research Sources:

- Chepchieng, Amos. "*UI/UX :Java & Swing Explained*." *Medium*, 23 May 2017, medium.com/@keeptoo/ui-ux-on-java-and-swing-da7dbacb8e14. Accessed 23 Oct. 2025.
- "*Implementing Java Swing User Interfaces*." *Oracle.com*, 2025, docs.oracle.com/middleware/12213/jdev/user-guide/implementing-java-swing-user-interfaces.htm#OJDUG2085. Accessed 23 Oct. 2025.
- GeeksforGeeks. "*Introduction to Java Swing*." *GeeksforGeeks*, 15 Feb. 2022, www.geeksforgeeks.org/java/introduction-to-java-swing/. Accessed 23 Oct. 2025.

Research Process & Insights:
We discovered these resources through **academic search engines (Google Scholar)** and **developer forums (StackOverflow)** when researching *Java Swing UI optimization* and *game UX feedback*.
The research highlighted:

- The importance of **visual hierarchy** (contrast, gradients, padding) in improving readability and hence improving user experience.

- The role of **responsive feedback**, e.g., animations or text prompts that react to player performance, for interactivity for the user.

- Techniques for **layering UI elements** in **Swing** (using gradients, alpha blending, and layout managers), for responsive gameplay from the user.

Influence on Design:

- *Core Gameplay Experience:*
    - The **runner movement system** was refined to provide **smooth lane transitions** with minimal input delay, ensuring that arrow key responses feel immediate and natural.
    - Visual feedback accompanies these actions through **sprite animations**, emphasizing direction and motion.
    - The **Health and Revival system** connects directly to the UX ideology of giving players agency: using hearts to continue after failure reinforces fairness and emotional engagement.
- *Interface and Feedback:*
    - A **semi-transparent gradient HUD** ensures the visibility of health hearts, keys, and scores even against detailed backgrounds.
    - The **taskbar layout** was optimized for even spacing, contrast, and readability, preventing clutter while keeping vital information visible.
    - **"SPEED UP!"** notifications appear in the center of the screen with glowing, fading text: providing strong **positive reinforcement** and pacing awareness.
    - **Dynamic score and progress tracking** ensures players always see their growth, encouraging replayability.
- *Control and Usability:*
    - The **pause screen (P or ESC)** uses a clean, centered overlay that clearly indicates the game's state without breaking immersion.
    - A **mute button** allows instant sound control while maintaining consistent iconography and accessibility for different player environments.
    - **Sound design**: including door creaks, zombie growls, and key collection chimes: reinforces player actions and builds immersion through consistent auditory feedback.

Depth of Understanding:
We developed a strong understanding of how UX and user psychology intersect: the importance of instant feedback, clarity, and creation of a visual atmosphere. We also gained hands-on familiarity with **Java Swing**'s rendering pipeline, enabling us to produce more visually balanced interfaces.

Sources of Images:

MLA 9 Citations of Sources for Images (the sprites not mentioned here were made by the programmers using Piskel.com):

Menu Page: "Generate an image of the menu page with this main player (image attached) looking upset at the audience and holding a red board with a zombie apocalypse atmosphere" prompt. ChatGPT, 4o, OpenAI, 17 Oct. 2025, chat.openai.com/chat

Source People Image (later pixelated): miah, Rahim. “Twee Abstract Figuren Omarmen Symboliseert Vriendschap Eenheid En Ondersteuning in Zwart Silhouet.” Vecteezy, 2025, nl.vecteezy.com/vector-kunst/60468389-twee-abstract-figuren-omarmen-symboliseert-vriendschap-eenheid-en-ondersteuning-in-zwart-silhouet. Accessed 23 Oct. 2025.

Prioritised Backlog Items

Priority 1 – Core Gameplay

1. Runner Movement in 3 Lanes – UX
    - Demo: Player presses left/right arrow keys to switch between three lanes. Runner sprite updates position smoothly with visible lane transitions.
    - Notes:
        - Must feel responsive and intuitive to ensure enjoyable control.
        - Serves as the foundation for all other interactions such as avoiding zombies or collecting keys.
2. Door and Key System – Random Generation
    - Demo: Randomly spawned locked door blocks the lane. If the player has the matching key, the door opens; if not, the player triggers a mini-game or game over.
    - Notes:
        - Demonstrates logic handling (key-door color matching, state checking).
        - Central challenge that combines player timing, decision-making, and resource management.
3. Key Collection – Random Generation
    - Demo: Keys of various colors appear randomly in lanes; when the runner touches a key, it disappears, and the HUD inventory updates immediately.

- Notes:
  - Visually distinct keys ensure quick player recognition and correct color association.
  - Encourages exploration of lanes and risk-reward decision-making.

4. Zombies as Obstacles – Random Generation

   - Demo: Zombies spawn unpredictably across lanes. Colliding with one ends the run or triggers a revival if health is available.

   - Notes:
     - Adds unpredictability and tension, forcing players to react quickly.
     - Placement is randomized but ensures fair gameplay by always providing an escape lane.

5. Health and Revival System – UX

   - Demo: Every five doors opened grants +1 heart. Upon death, a pop-up asks to use three hearts to continue or end the game.

   - Notes:
     - Adds long-term progression and strategic decision-making.
     - Connects survival mechanics to the theme of saving others for self-preservation.

<u>Priority 2 – Progression and User Experience</u>

- Randomized Map Generation – Random Generation (Advanced Topic)

  - Demo: Each session generates a unique pattern of doors, keys, and zombies. Restarting the game shows a new arrangement.

  - Notes:
    - Ensures replayability by preventing repetition and predictable patterns.
    - Implements algorithmic randomization to create fair yet varied gameplay experiences.

- Score and Progress Tracking – UX

  - Demo: The HUD displays the number of doors opened, survivors saved, and total score, updating dynamically as the game progresses.

  - Notes:
    - Provides clear feedback and measurable progress to the player.

        - Encourages replayability by making achievements visible and rewarding.

- HUD (Health, Keys, Score) – UX

    - Demo: Player sees health hearts, collected keys, and score displayed clearly during gameplay. All values update in real time.

    - Notes:
        - Designed for readability against the dark background using contrast and layout.
        - Keeps players aware of essential information without cluttering the screen.

- Speed Progression – UX / Random Generation

    - Demo: Game speed increases gradually over time: ×2 at 2 mins, ×2.5 at 6 mins, ×3 at 10 mins, ×3.5 at 14 mins, ×4 at 18 mins, ×4.5 at 22 mins, ×5 at 26 mins.

    - Notes:
        - Provides dynamic difficulty scaling that keeps gameplay challenging and engaging.
        - Demonstrates variable manipulation and timer-based progression mechanics.

- Pause Screen – UX

    - Demo: Pressing P or ESC pauses the game and shows a centered “PAUSED” message; pressing again resumes gameplay.

    - Notes:
        - Enhances player control and usability during play sessions.
        - Visually clean overlay ensures the player knows the game is paused without breaking immersion.

- Mute Button – UX

    - Demo: Clickable icon at top-right toggles all game sounds. The icon visually switches between muted and unmuted states.

    - Notes:
        - Improves accessibility and lets players control their sound environment.
        - Maintains immersion with simple, intuitive toggle behavior.

Priority 3 – Immersion and Aesthetic

- Zombie Apocalypse Background – UX
    - Demo: Game starts with scrolling urban ruins, cracked streets, fog, and debris in the backdrop, moving continuously to simulate motion.
    - Notes:
        - Establishes atmosphere and setting immediately.
        - Uses muted tones to maintain visual clarity and highlight gameplay elements.
- Color-Coded Doors and Keys – UX
    - Demo: Different colored doors (Red, Green, Blue, Yellow, etc.) appear with matching key colors. The correct key opens the corresponding door.
    - Notes:
        - Reinforces visual learning and quick recognition.
        - Maintains thematic consistency with apocalyptic visuals.
- Sprite Animations (Runner and Doors) – UX
    - Demo: Runner's walking animation cycles through frames; doors show opening animation upon unlocking.
    - Notes:
        - Adds polish and visual depth to gameplay.
        - Makes interactions feel dynamic and alive, enhancing player engagement.
- Sound Effects and Music – UX
    - Demo: Key pickup plays a chime, door opening creaks, zombie collisions grunt, and ambient music loops softly in the background.
    - Notes:
        - Reinforces player feedback for actions and events.
        - Builds immersion through atmospheric audio design.

- Mini-Games Integration – Random Generation / UX
    - Demo: Hitting a locked door or obstacle without a key triggers a random mini-game (Tic Tac Toe, Tap Match Fruits, Connect Wires, Button Sequence).
    - Notes:
        - Diversifies gameplay and adds narrative depth to survival theme.
        - Keeps tension high while rewarding quick reflexes and logic.
- Door Colors Expansion – Random Generation / UX
    - Demo: New door colors are introduced progressively: brown (3 min), red (4 min), white (5 min), icy blue (6 min).
    - Notes:
        - Gradual color expansion scales difficulty and variety.
        - Demonstrates use of time-based state changes and progression logic.