

FREE EXPLORATORY QA REPORT

PANIC TOWER 48

Browser playtest build - usability, gameplay clarity and visual quality

Prepared by: Libert Family Studio

Test format: Solo exploratory QA test (free sample report)

Platform: Windows PC - browser build

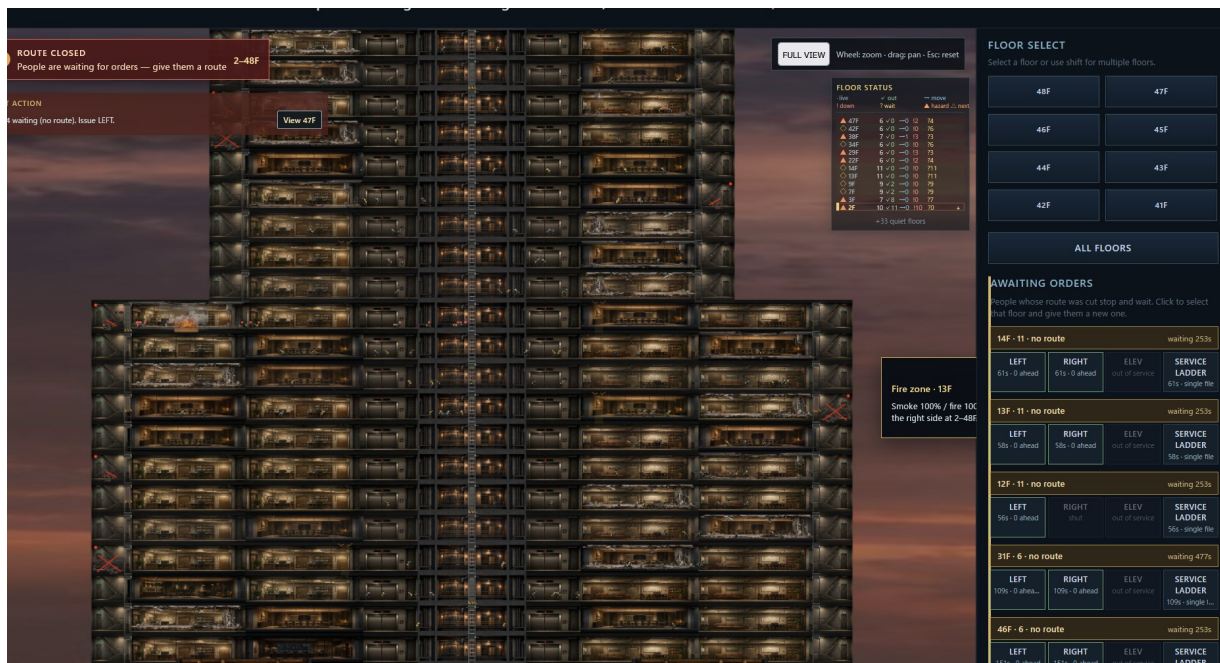
Scenario: 48-floor scenario

Test date: 29 August 2026

Scope: Single-session functional, visual and UX review

Overall assessment

The simulation has an interesting concept, but the tested build does not consistently explain hazards, route outcomes or rescue states. Pathing and positioning defects, weak visual integration and repetitive individual commands significantly reduce clarity and player confidence.



Evidence overview: 48-floor scenario with crowded interface and partially obscured contextual information.

Executive summary

Test objective: Assess whether a first-time player can understand the evacuation tools, issue reliable movement orders, identify hazards and interpret rescue outcomes.

Result: The session revealed multiple major clarity and pathfinding problems. The player often could not tell why characters collapsed, whether a command was safe, or why emergency services remained inactive.

Severity	Count	Meaning	Main impact
Major	6	High impact	Commands, pathing, hazard visibility and evacuation clarity
Moderate	4	Noticeable	Visual polish, navigation and actionable feedback

Highest-priority actions

- Make smoke and exposure risk visible before characters become incapacitated.
- Correct left/right routing, sprite orientation and stair/escalator path alignment.
- Explain rescue requirements for helicopters, incapacitated survivors and broken elevators.
- Reduce repetitive person-by-person management and improve navigation to waiting characters.
- Replace placeholder-looking fire visuals with effects integrated into the environment.

Test limitations

- Exploratory single-session test; this is not an exhaustive compatibility or regression pass.
- Findings reflect the browser build and the 48-floor scenario observed during the session.
- Language availability was not evaluated because the tester used the English interface.

Detailed findings

1. Contextual action window is hidden behind the side menu

MAJOR | User interface

Observed: When certain people near the right side of the building are selected, the contextual action/information window opens beneath the main side menu. Part of the window becomes unreadable or inaccessible.

Expected: The contextual window should remain fully visible and usable regardless of the selected character's location.

Recommendation: Automatically reposition the window to the left of the character or clamp it inside the usable viewport.

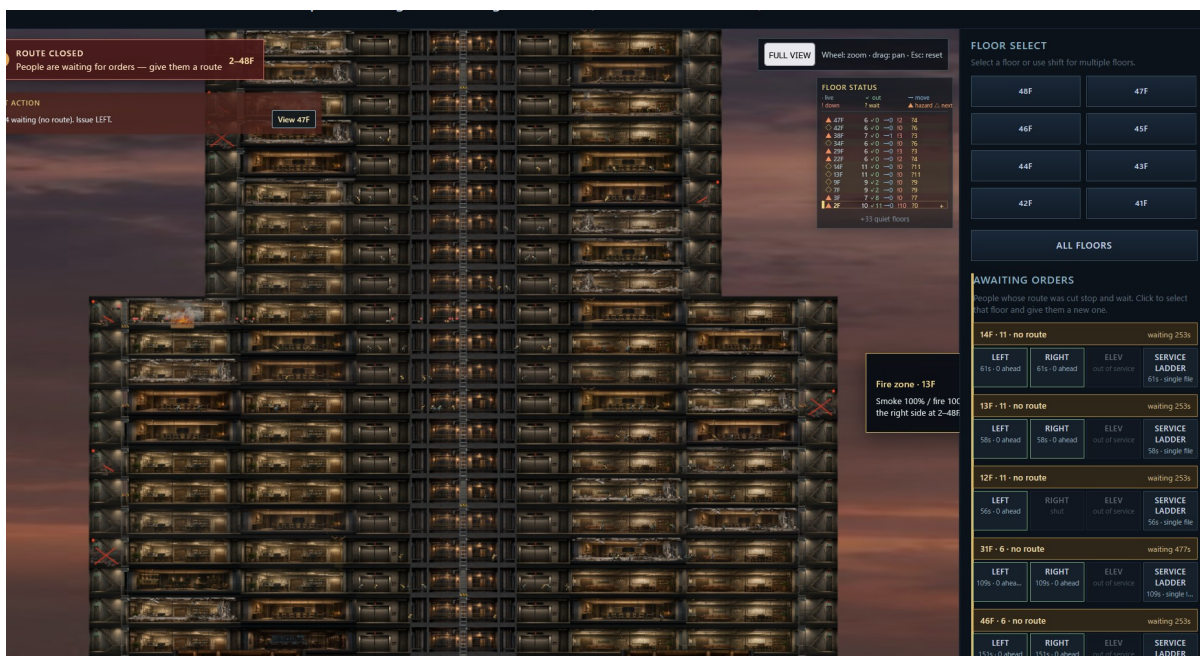


Figure 1 - Contextual information is partially obscured by the right-side interface.

2. RIGHT route command can send a character left

MAJOR | Controls / pathfinding

Observed: After a character is selected and the RIGHT route is chosen, the character can begin moving left. The result makes the command appear reversed or disconnected from the route label.

Expected: RIGHT should send the character toward the right-side route, staircase or exit.

Recommendation: Verify route-node mapping on multiple floors and confirm whether the labels indicate screen direction or a named exit. Make the distinction explicit.

3. Firefighters do not follow the escalator or floor geometry

MAJOR | Pathfinding / positioning

Observed: Firefighter sprites move horizontally through open space while the escalator descends diagonally. They appear suspended between floors rather than walking on the escalator or a valid path.

Expected: Emergency responders should follow the visible slope and remain attached to valid floor or stair geometry.

Recommendation: Align movement nodes with the escalator path and rotate or animate sprites to match the direction of travel.



Figure 2 - Firefighters travel horizontally through the escalator area instead of following its slope.

4. Helicopter does not evacuate incapacitated survivors

MAJOR | Rescue system / gameplay clarity

Observed: The helicopter is positioned on the rooftop helipad with incapacitated survivors directly beneath it. Survivors remain on the ground, the helicopter performs no visible rescue action and no message explains the blocking condition.

Expected: The helicopter or an emergency crew should collect eligible survivors. If another action is required, the game should state the requirement and show the current evacuation status.

Recommendation: Add a visible boarding/rescue state, eligibility explanation and clear feedback when evacuation is blocked.



Figure 3 - Incapacitated survivors remain beneath an inactive helicopter on the helipad.

5. Characters become incapacitated without visible smoke

MAJOR | Hazard visibility / feedback

Observed: Characters collapse or become incapacitated from smoke exposure even when no smoke or fire is visibly present around them. The player cannot identify the danger or react before the collapse.

Expected: Hazardous smoke should be visible and characters should receive a clear exposure warning before becoming incapacitated.

Recommendation: Add room-level smoke overlays, exposure bars or warning icons, plus a brief cause label when a character collapses.

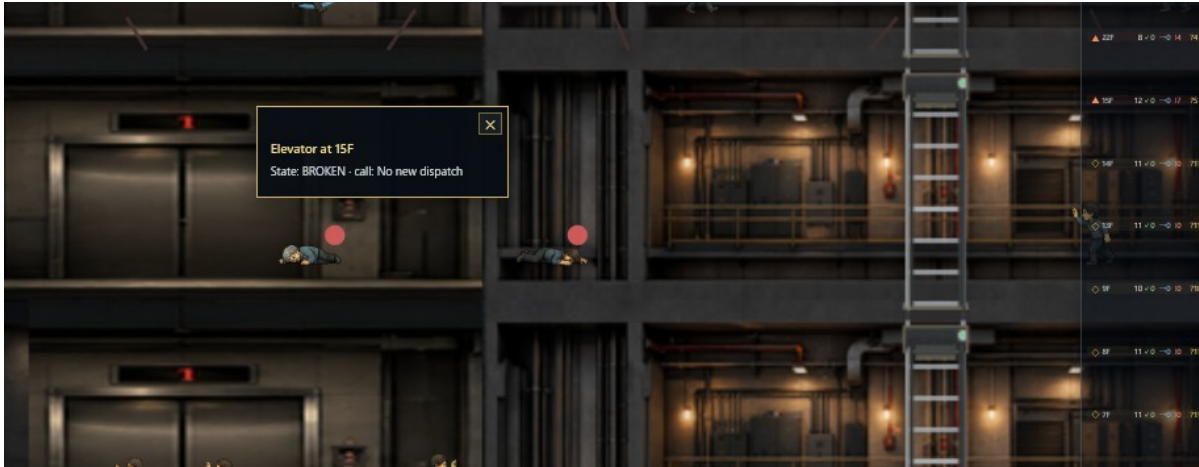


Figure 4 - Two incapacitated characters with no visible smoke or fire in the immediate area.

6. Core loop becomes repetitive and consequences are difficult to understand

MAJOR | Gameplay / user experience

Observed: The player repeatedly selects individuals and chooses left or right based on apparent stair access. It remains difficult to know whether a choice is correct, whether it improved the outcome, or why people later collapse.

Expected: Orders should produce immediate, understandable feedback and the game should communicate objectives, progress, risks and rescue outcomes.

Recommendation: Support group orders, clearer safe-route previews, objective milestones and a concise event log that connects decisions to consequences.

7. Character sprites float, clip and face the wrong direction

MODERATE | Visual positioning / animation

Observed: Characters can appear several pixels above the floor, partially clipped by floor boundaries, or facing away from their movement direction. Running characters may move without visible leg animation, which makes them appear to glide.

Expected: Feet should remain aligned with the floor, complete sprites should remain visible, and orientation/animation should match movement.

Recommendation: Standardize sprite anchor points and floor baselines, then validate facing and animation state changes on every route type.



Figure 5 - Character feet are visibly suspended above the floor; another sprite is clipped at the lower edge.

8. Fire effect appears as an opaque rectangular image

MODERATE | Visual presentation

Observed: The fire is displayed inside a clearly visible rectangle with an opaque background. Its lighting, perspective and visual style do not match the surrounding room, making it look pasted over the environment.

Expected: Fire and smoke should blend naturally with the room, use transparency and visually affect nearby surfaces.

Recommendation: Replace the placeholder-style image with layered transparent effects and local light/smoke treatment consistent with the building art.



Figure 6 - Fire artwork has a visible square boundary and does not blend with the room.

9. Waiting-person indicator is not an effective navigation control

MODERATE | Navigation / usability

Observed: A yellow number indicates that a person is waiting after a WAIT order, but it does not identify the person or take the player to their location. In a 48-floor scenario, the player may no longer remember who was told to wait.

Expected: The indicator should help the player locate and manage every waiting person.

Recommendation: Make the indicator clickable. Open the correct floor, center and highlight the character, and cycle through multiple waiting people when necessary.

10. Broken elevator status provides no actionable guidance

MODERATE | Interface / gameplay clarity

Observed: The elevator status reports 'BROKEN - Call: No new dispatch', but does not explain whether the elevator can be repaired, whether help is coming, or which alternative action is expected.

Expected: A failure state should explain its consequence and the player's available response.

Recommendation: Show repair eligibility, estimated response, affected routes and a direct alternative-route action.

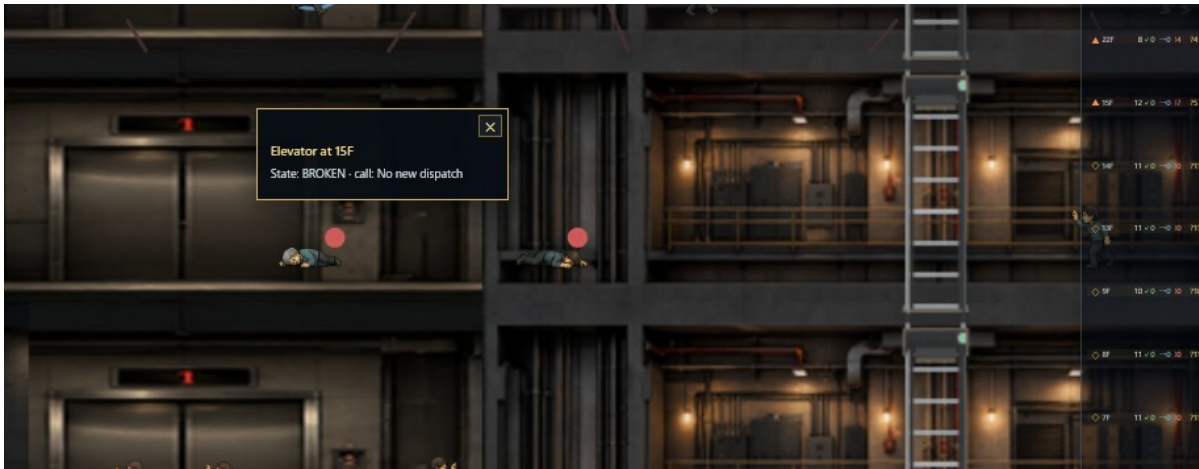


Figure 7 - Broken elevator status is visible, but the next action is not explained.

Player experience conclusion

PANIC TOWER 48 creates an initially interesting large-scale evacuation scenario, and the detailed building presentation gives the simulation a strong first impression. During extended play, however, the experience becomes tiring because the player cannot reliably connect orders with outcomes. Invisible smoke exposure, inconsistent route behaviour and inactive rescue systems make failures feel arbitrary rather than instructive.

The most valuable next step is not adding more scenario complexity. It is improving the clarity of the existing loop: show hazards, explain blocked actions, make routes predictable and help the player navigate directly to people who require attention.

Recommended retest focus

- Confirm that LEFT and RIGHT commands always select the intended route.
- Verify that smoke is visible before exposure becomes dangerous.
- Test firefighter movement on stairs, escalators and service ladders.
- Confirm the complete rescue flow for incapacitated rooftop survivors.
- Verify that every contextual panel remains visible at all camera positions.

Prepared by Libert Family Studio

Independent family QA and playtesting