ENHANCING ARCHITECTURAL SAFETY DESIGN THROUGH A 3D NPC-DRIVEN EMERGENCY EVACUATION SIMULATION

Malcolm Qiu [1], Moddwyn Andaya [2]

[1] Northwood High School, 4515 Portola Pkwy, Irvine, CA 92620
[2] California State Polytechnic University, Pomona, CA 91768

ABSTRACT

This simulation's purpose overall helps complete the requirements of a proper design. With its features of imposing problems, the user must adjust to help improve the safety and usability of their ideas, for example making sure that the NPC will always reach the exit no matter what and the features and objects in a building do not prevent evacuation in case of emergencies. One disaster imposed so far is in the case of a fire,theNPC must be able to leave safely and without casualties.The user will test their ideas by copying their layout onto the 3d simulation using the build blocks/furniture's provided by the simulation and then placing NPCs inside. Eexperiments' involving NPCsshow that they recognize and act on special limitations most of the time and will always head towards an available exit if they can reach it and stand still if there is none.However,cannot scale heights both up and down without stairs.When that is patched the user will be able to work with similar limitations of humans thus making sure that safety is ensured with realistic logic. By testing their designs through the simulation, it helps produce and revise a more ideal final draft of a concept by figuring how to fix/expand on the starting layout.

KEYWORDS

Emergency evacuation, Simulation design, NPC behavior modeling, Architectural safety

1. INTRODUCTION

Design failure is a problem in a lot of buildings that has resulted in a lot of incidents which can cause people to be unable to leave the building in time thus dying in fires to earthquakes or many other disasters many design flaws that have occurred include [11]:

Missing or inadequate fire/carbon monoxide detection or suppression, systems lack of sufficient egress options in the event of a fire, Improper Grounding of electrical systems, improper electrical junction systems, or improper labeling of circuits and insufficient GFCI receptacles, Inadequate building ventilation, Improper building framing, Missing handrails and railings, or open-ended banisters, Improperly secured decking or balconies, Insufficient waterproofing of windows, doors, and other entries.

And all safety issues may not be able to be resolved to tested properly without using significant time therefore this simulation aims to preemptively find problems in designs and determine if they still want to go through with their idea or not. The program will help the designers understand the dangers of their designs and certain limitations of the structure; this will allow

David C. Wyld et al. (Eds): CSML, AISCA, DNLP, SOEA, NET, BDHI, SIPO – 2026
pp. 52-59, 2026. CS & IT - CSCP 2026 DOI: 10.5121/csit.2026.160205

users to how to best design something that is emergency proof so they can optimize the layout suit the needs of the people using it better.

The designer copies over one's ideas into this 3D simulation, assisted with having the basic layout digitized and converted into 3D onto the software, producing a 3D mesh and represents the designers ideas, with the general structure created the user can make corrections and add on features to make it match their ideas, after the users deem that they've made a close enough replicate they are then able to fun simulations with NPCs in order to test that the design is practical and does not compromise safety, or point out the in considerations of their design and give them an idea of how they should better adjust in order to make a better design, by looking at the failures with safety requirements for the NPC unable to be met because of the layout, by being able to replicate ones ideas into a 3D version they will be able to better understand the structure when visualized, therefore letting them understand the effectiveness of their designs better so they would be able to receive feedback on to accuracy and realism of their processes [12].

This simulation allows the user to replicate the structural layout of their blueprint to test space management and if the design has interfered in regulating the building's safety. This is also useful for creating the most flexible structure that can be used for its purpose without having to hinder evacuation in cases of emergencies in order to make sure that the design does not reduce safety. This program helps the user revise their design to have multiple ways to travel in and out,up or down the building to ensure that people are not trapped in danger because of space restraints of design flaws. My project works towards letting people figure how to make the most visualizing appeal and containing the most of user desired features without making it impractical to live or work in. I will include features that will point out the limitations of the users ideas with letting them know if there are things they left unaccounted for when they designed their ideas or if certain structures get in the way of things that would make it undesirable to stay in. by adding a sense of realism with safety hazards and condition requirements through good lighting convenient to get around structure that is pleasing to the eye at every angle and consistent proportions.

The experiments overall reflect that the NPC has the awareness of a human and recognizes when it can or cannot reach the exit and does not move when recognizing there is no exit however their abilities showed to be limited as the NPC's capabilities do not go beyond the ability to walk, climbing and jumping from short distances seemed to be out of their current reach which does not make them identical to the abilities and inabilities of a normal human. To be able to provide the most accurate simulation possible it is needed to use the most realistic expectations of human ability as a metric to properly safety proof and expand on the layout. Therefore, with the verification of the best replication of real world limitations and factors designer makers can acquire the best insight about their concepts and how or if they should expand on that or whether or not their idea choices are feasible when confronted with human needs.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Free Placement for Realistic Outcomes

Skeptical Question: Why would you let users place props anywhere, even if it blocks exits or makes escape impossible?

Response: That's actually the point of the app — to simulate real-world design consequences. The placement system gives users full freedom so they can test and learn from poor design choices. If a prop blocks an exit and NPCs can't escape, the simulation will reflect that failure with injury or death results. This helps users understand how small layout choices can impact safety during disasters.

2.2. Reliable NPC Pathfinding Safeguards

Skeptical Question: What if the NPCs can't find a valid path to the exit or glitch into walls?

Response: The NPCs use Unity's NavMesh pathfinding system which ensures they only follow navigable terrain. Before committing to a destination, the system checks if a full path exists. If not, it logs a warning and triggers a fallback response. This avoids NPCs getting stuck or behaving unrealistically. The system also uses animations to show the escape behavior visually, making the simulation more accurate.

2.3. Trustworthy Insights Despite Simplification

Skeptical Question: How can you trust the results of the simulation if it's just a simplified version of real life?

Response: While it's true the simulation can't capture every real-life detail, it focuses on key factors like layout, exit placement, and environmental hazards to give a realistic overview of how a space performs under stress. It uses Unity's built-in physics and AI navigation, which are reliable for general movement and collision. The goal isn't perfect realism, but useful insight — enough to flag major safety flaws before a design is built in the real world.

3. SOLUTION

On the main menu, you can access the level selector where you can choose different floor plans you have created. When opening or creating a new floor plan, you are able to edit it as you wish with extensive decoration, furniture, essentials, or structural related models. This is made to allow you to closely match your own desired floor plan. You can also place people (AI NPCs) for the simulation results to be relevant [13]. Once finished, you can select different disasters to simulate on your floor plan and watch, using physics and real life conditions, what will happen for some time during this disaster. This is the simulation itself, and right after some duration or conditions, the simulation ends given a result of "how safe" the floor plan is with different result data such as how many people were injured or deceased during the run.

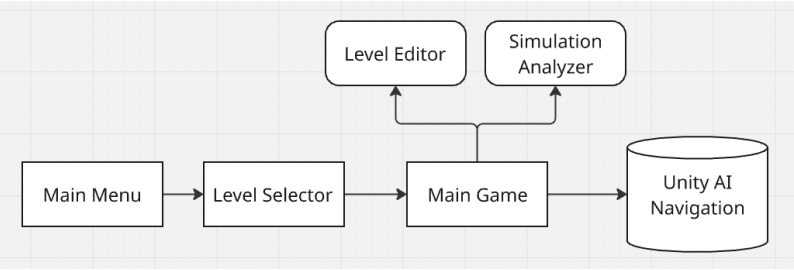

Figure 1. Overview of the solution

The first most important component of the app is the level editor. The level editor is composed of two parts, the structure editor and the prop/decoration editor. You can create structures like walls to layout your floor plan and then place down many different types of props and furniture organized by categories of location to design.

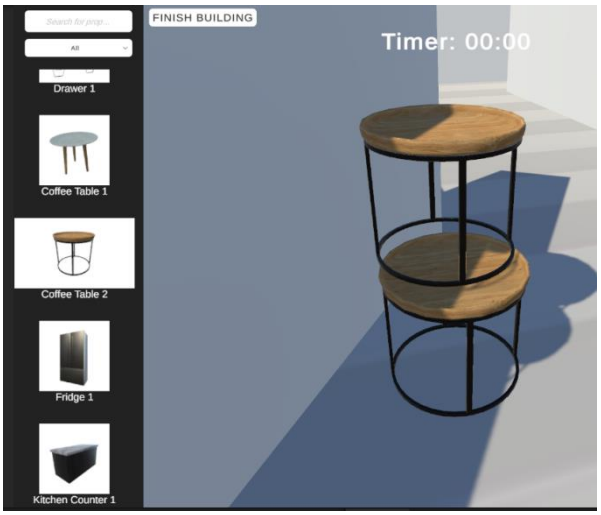

Figure 2. Screenshot of the game 1

```
void PlacementUpdate()
{
    ray = cam.ScreenPointToRay(Input.mousePosition);
    if (Physics.Raycast(ray, out hit, 1000, layerMask))
    {
        Vector3 pos = hit.point + Vector3.up * heightOffset;

        Prop currentPropSelected = propsGhost[propIndex];

        if (lastGhostPropShown != null && lastGhostPropShown != currentPropSelected)
        {
            lastGhostPropShown.gameObject.SetActive(false);
        }

        lastGhostPropShown = currentPropSelected;
        currentPropSelected.transform.position = pos;
        currentPropSelected.gameObject.SetActive(true);

        if (Input.GetMouseButtonDown(0) && propLibrary.hovering == false)
        {
            Prop newProp = Instantiate(currentSelectedProp, pos, Quaternion.Euler(currentPropSelected.transform.rotation.eulerAngles));
            newProp.ActivateGhost(false);
            // If there is no obstacle component on the prop, lets activate its physics (its not an obstacle, so it can be moved by physics)
            if(newProp.GetComponent<NavMeshObstacle>() == false)
                newProp.ActivatePhysics(true);
            else
                newProp.ActivatePhysics(false);
        }
    }
}
```

Figure 3. Screenshot of code 1

The function PlacementUpdate() handles the placement of props (game objects) in the world using mouse input and raycasting. The ray is cast from the camera through the mouse position onto a defined layer. If it hits something, the placement position is calculated with an upward height offset. A ghost version of the selected prop (used for previewing placement) is fetched from the propsGhost array based on the current prop index. If a different ghost was shown previously, it is deactivated before activating the new one and updating its position. When the user left-clicks (mouse button down) and no UI element is being hovered over (propLibrary .hovering == false), a new instance of the selected prop is created at the specified position with the same rotation as the ghost. The prop is initialized in a non-ghost state. The code then checks if the object has a NavMeshObstacle component. If not, it enables physics, allowing the prop to be affected by the physics system; otherwise, it disables physics to keep it static. This script allows dynamic, visual placement of props in the scene during level design, making it easier for designers to test and build levels interactively [14].

Another part of the app before the simulation, is playing down the people or NPCs. These people will simulate the actual escape or actions that will happen during disasters. Their actions and escape depend on the floor plan like where the exit is and what's the fastest way out.

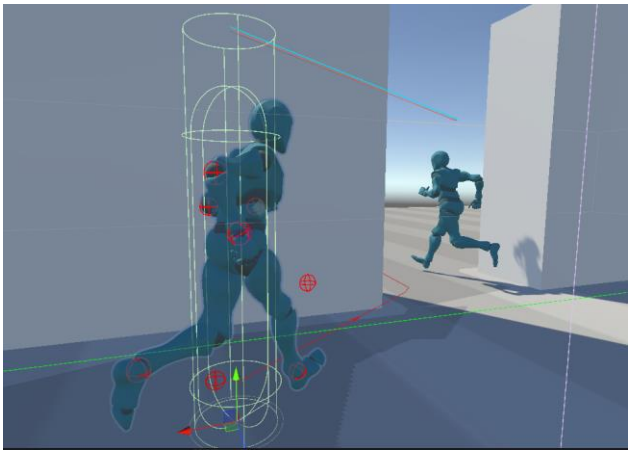

Figure 4. Screenshot of game 2

```
25      [Button]
        1 reference
26      public void Exit()
27      {
28          animator = GetComponent<Animator>();
29          Transform nearestExit = ExitPoints.Instance.GetNearestPoint(transform);
30          if (nearestExit != null)
31          {
32              NavMeshPath path = new NavMeshPath();
33              bool pathFound = agent.CalculatePath(nearestExit.position, path);
34
35              if (pathFound && path.status == NavMeshPathStatus.PathComplete)
36              {
37                  agent.SetDestination(nearestExit.position);
38                  animator.SetBool("Exit", true);
39              }
40              else
41              {
42                  Debug.LogWarning("Can't reach the destination!");
43                  animator.SetBool("Exit", false);
44                  OnPathFail?.Invoke();
45              }
46          }
47      }
48
```

Figure 5. Screenshot of code 2

This code is for controlling an NPC (non-player character) to escape a building by finding and moving toward the nearest exit using Unity's NavMesh system. When triggered, it first retrieves the NPC's Animator component and finds the closest exit point using ExitPoints .Instance .GetNearestPoint(). If a valid exit is found, the NPC attempts to calculate a navigable path to it. A new NavMeshPath object is created, and CalculatePath() checks if the destination is reachable. If a valid path is found and its status is PathComplete, the NPC sets the destination using SetDestination() and sets the "Exit" animation flag to true, to trigger a running animation. If the path cannot be completed, it sets the "Exit" flag to false, and invokes a failure callback OnPathFail. This setup ensures the NPC only attempts to escape if a valid path exists, handling failure cases gracefully.

The last most important is the disaster itself. Disasters can be many things. One of them being some kind of fire where NPCs will be damaged in a certain intensity of the fire. The way things like exits are placed and where fires are, will impact the decision making of an NPC.

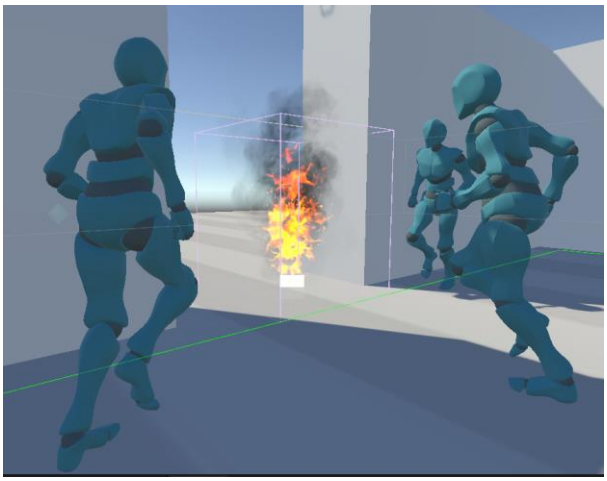

Figure 6. Screenshot of game 3

```
0 references
void OnTriggerEnter(Collider other)
{
    Health health = other.GetComponent<Health>();
    if (health && !targets.Contains(health))
    {
        targets.Add(health);
        StartCoroutine(DoDamage(health));
    }
}

0 references
void OnTriggerExit(Collider other)
{
    Health health = other.GetComponent<Health>();
    if (health && targets.Contains(health))
    {
        targets.Remove(health);
    }
}

1 reference
IEnumerator DoDamage(Health target)
{
    while (targets.Contains(target))
    {
        target.ChangeHealth(-damageAmount);
        yield return new WaitForSeconds(damageInterval);
    }
}
```

Figure 7. Screenshot of code 3

This script handles damaging NPCs when they enter a hazardous area—like a fire—using Unity's trigger system. When an NPC with a Health component enters the fire's trigger collider, OnTriggerEnteris called. If the NPCisn't already tracked in the targets list, it's added, and a coroutine (DoDamage) starts applying damage to it over time. When the NPC leaves the fire area, OnTriggerExit removes it from the targets list, which stops further damage by breaking the coroutine's while-loop condition. The DoDamagecoroutine keeps running as long as the target is still inside the trigger. It repeatedly calls ChangeHealth() with a negative value (representing damage), waits for a set time (damageInterval), and repeats. This setup simulates continuous environmental damage, like standing in fire, in a performance-friendly way. It avoids starting duplicate damage routines for the same target and cleanly stops damage when they leave.

4. EXPERIMENT

In order for a simulation to provide the most promising data I need to ensure the NPC will act accordingly to that of a real person in all situations that it is placed in,because realistic reactions are crucial for producing the most accurate results and to avoid false positives/negatives from misleading the user.

I will test them with experiments that humans will be able to get though/get stuck from and depending on if the NPC recognizes a path forward or not I will be able to determine if the ai is humanlike enough and what needs to be changed once they do something a human is physically unable to do(phase though small gap I will limit their hitboxes).

If they do not pursue the exit despite there being an open space their path find will need to be rearranged.

Scenerio	Description	Expectation	Result
1	A simple doorway between NPC spawn and exit	NPC go through the door to reach exit	Success
2	A simple doorway with pushable tables between NPC spawn and exit	NPC go through the door pushing tables to reach exit	Success
3	cramp	NPC pushes chair and move it out of the way to get to exit	fail
4	Put pillars in front of exit that block it	Npc recognize there is no space to move	success
5	Put npc in area surrounded by blocks left with a little squeezable gap	Npc recognize there is way out	success
6	Place exit on a pillar that is very much at a height npc can reach	Npc does not move	success
7	Place npc on a pillar that is very short and safe to get down from	Npc does not move	success
8	Place exit in air	Npc does not recognize it	Success
9	Place door and then place wall right next to it	Npc goes around	Success
10	Place npc far from exit and place big wall in front of them but not around it leaving them enough space	Npc starts moving towards it	Success

Figure 8. Table of experiment

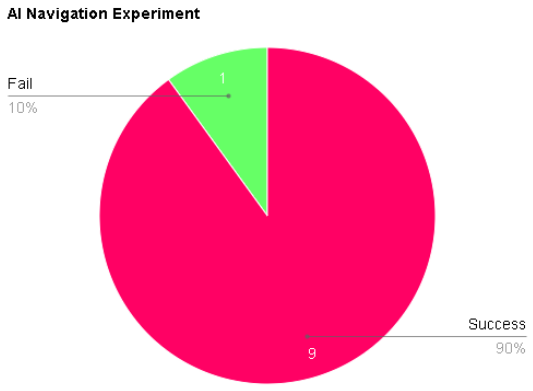

Figure 9. Figure of experiment

We realized that in some situations the NPCs does not recognize a legitimate pathway to the exit which means that it is necessary to implement new movement and adaptive response when faced with complications so that the NPC can give out the expected reaction of a normal human so that users will receive the most accurate evacuation process as the NPC’s response is near identical to human response by ensuring their capabilities are not above nor below what it should be.However, we did have success in proving NPCs capabilities do not exceed projected limitations as they often recognize that they do not have a path towards the exit and stops moving.Ultimately the results reflect that while in obvious situations where the path to the exit is direct, they will behave like a human but in situations where the path to the exit is not a straightforward path the NPC lack the ability to navigate their area.

5. RELATED WORK

This research paper looking into the game prison architect which is about a prison designing 2D simulation “You can build cell blocks, canteens, kitchens, yards, and yes -- even shower facilities. Prisoners will be hiding shivs and poison in their pockets, ready for a riot or break-out attempt, and your guards need to catch them [2].” The paper allowing the player to customize their own prison, my program like it also covers the design features but in 3D and brings about more concerns for building designs as it has more in depth safety demands and needs for NPCs which incentives the designer to have to adapt to make the most optimal layout, it has more in depth details instead of being condensed to fit into 2D and having to meet a large scale.

The second paper comparing it to the study of roman architecture via 3D modeling to find more specific mathematics that the structure was based upon “The project took on the ambitious task of scanning Roman structures, leveraging cutting-edge technology and academic insights to uncover the secrets of these millennia-old constructions [3].” However I believe that my simulation is able to provide more precise data as when historical information is being analyzed assumptions about the incomplete ruins needs to be made to properly 3D model it due to what has been lost in time therefore my project has more potential for precision due to the user being allowed to work with more given information regarding the structure and materials that will be used instead of working with something incomplete.

The article looking at the game Architect Life: A House Design Simulator which like this simulation gives the player a set number of materials to build with and sets requirements for the user’s design “In Architect Life’s Career Mode, the gameplay loop is based on the construction stages of a real architectural project. For each mission they choose, players must follow precise phases while ensuring they respect the allocated budget [4].” However, the limitations placed in this simulation is better for overall data collection to make the best possible layout because a fixed budget is not something that will always be the case and safety necessities will always be something to look out for.

6. CONCLUSIONS

Things are this simulation may not be able to cover are would be having physics that are 100% compatible with real life situations, break in simulations may not be as predictable as it would be unable to properly simulate how a real person would act, the ai may underestimated the intelligence of a real person which would make certain shortcomings overlooked if the design allow for the NPCs to be saved while in the exact same irl situation they wouldn’t if the intruder was more competent than the ai, during the test of the 3D simulation we found out that flaws involving the NPCs include but not limited to, being unable to scale short walls or recognizing

short and safe falls when available causing the NPC to stand still when there is a way to reach the exit [15].

REFERENCES

[1] Abbott, Howard, and Mark Tyler. Safer by Design: A guide to the management and law of designing for product safety. Routledge, 2017.

[2] Jewkes, Yvonne. "The aesthetics and anaesthetics of prison architecture." Architecture and Justice. Routledge, 2016. 33-46.

[3] Williams, Kim, and Michael J. Ostwald, eds. Architecture and Mathematics from Antiquity to the Future. Vol. 2. Cham: Birkhäuser, 2015.

[4] Doma, OğuzOrkun. Architectural experience in video games: spatial, temporal and narrative. Diss. BahçeşehirÜniversitesi Fen BilimleriEnstitüsü, 2014.

[5] Østergård, Torben, Rasmus L. Jensen, and Steffen E. Maagaard. "Building simulations supporting decision making in early design – A review." Renewable and Sustainable Energy Reviews 61 (2016): 187-201.

[6] Clayton, Mark J., Robert B. Warden, and Thomas W. Parker. "Virtual construction of architecture using 3D CAD and simulation." Automation in Construction 11.2 (2002): 227-235.

[7] Joy, Emmanuel, Aadhithian R, and Christhu Raja. "Digital 3D modeling for preconstruction real-time visualization of home interior design through virtual reality." Construction Innovation 24.2 (2024): 643-653.

[8] Webster, Helena. "The architectural review: A study of ritual, acculturation and reproduction in architectural education." Arts and humanities in higher education 4.3 (2005): 265-282.

[9] Tellios, Anastasios, PanagiotaKoulali, and KalliopiValsamidou. "Designing tomorrow: AI and the future of architectural design process." Interdisciplinary Journal of Architecture And Built Environment 27 (2023): 22-25.

[10] Schlack, Marilyn. "The Arcadia Commons Partnership: The Community College and Economic Redevelopment." (1993).

[11] Lachheb, Ahmed, and Elizabeth Boling. "How do they define design failure? An investigation of design failure in instructional design practice from the practitioners' perspective." Educational technology research and development 72.2 (2024): 687-725.

[12] De Melo, Mirella Santos Pessoa, et al. "Analysis and comparison of robotics 3d simulators." 2019 21st Symposium on Virtual and Augmented Reality (SVR). IEEE, 2019.

[13] Hubble, Adam, Jack Moorin, and Arjab Singh Khuman. "Artificial intelligence in fps games: NPC difficulty effects on gameplay." Fuzzy Logic: Recent Applications and Developments. Cham: Springer International Publishing, 2021. 165-190.

[14] Karlsson, Tobias, Jenny Brusk, and Henrik Engström. "Level design processes and challenges: a cross section of game development." Games and Culture 18.6 (2023): 821-849.

[15] Bakic, Predrag R., et al. "Mammogram synthesis using a 3D simulation. I. Breast tissue model and image acquisition simulation." Medical physics 29.9 (2002): 2131-2139.