Queen Mary, University of London
School of Electronic Engineering and Computer Science

A.I. Companions in Bioshock Infinite and The Last of Us

ECS7016P - Interactive Agents and Procedural Generation

Mughees Asif | 180288337

Due: 24 March 2022

Contents

1 Introduction 1

2 AI System 1

2.1 Bioshock Infinite 1

2.2 The Last of Us 4

3 Constraints 7

4 Conclusion 7

References 8

1 Introduction

Bioshock Infinite was released in 2013 as a first-person shooter role-playing video game set in 1912 in the fictional city of Columbia. The two main protagonists, Booker DeWitt (player-controlled) and Elizabeth (autonomous), make their way through the city using different weapons and tools to complete various objectives. Two weapons and ammunition per protagonist are allowed and can be collected from either the environment or fallen enemies. Health is dependent on medical kits and Booker is imbued with powers and abilities dependent on collecting different artifacts from the game environment. Elizabeth is a non-player character (NPC) built using artificial intelligence (AI) who acts as an accomplice for Booker providing assistance (non-combat and combat help) and sustenance (medic kits for health), when needed. The main objective of the AI companion was to not only be an active helper but to also develop an emotional bond via customised interactions with each player [1].

The Last of Us was released in 2013 as an action-adventure third-person perspective video game set in 2013 in a dystopian environment. The main protagonists, Joel, and Ellie, traverse several post-apocalyptic cities and environments to progress the story, whilst the player gets to control each character periodically throughout the course of the game. The AI element of the game revolved around companionship where if a player is assigned Joel, AI would control Ellie and vice versa, and hostile enemies reacting to the discovery of the players' characters with call to actions such as taking cover or requesting assistance from fellow enemies. The main objective of the AI system was to undermine "... the risk of [agents] turning into a tedious "escort" quest, generally getting underfoot, or turning into mindless drones with no agency in the world" [2].

2 AI System

2.1 Bioshock Infinite

Figure 1: The goal-side blocking system [3]

The AI system behind *Bioshock Infinite* was designed to enable Elizabeth (NPC) to engage with the environment using a "player-facing" approach that allows the NPC to perceive the work from the point-of-view of the player [3]. Player-facing in gaming can be defined as the NPC understanding the personality of the player instead of how to operate around them.

For the non-combat movement, as an alternative to using generic player facing techniques such as following the player with a scripted system or to follow the sole lead of the player, blocking was used to ensure Elizabeth followed the player accurately. Blocking is borrowed from the theatre sphere where it is a performance term for determining positioning and movement on-stage [3]. The developers at Irrational Games (the makers of *Bioshock Infinite*) implemented a reactive "goal-side blocking" system that required Elizabeth to stay in-between the player and the goal of any given scene (Fig. 1). The preference was always for Elizabeth to stay closer to the goal instead of the player, as staying too close or far to the player is unnatural and impersonal.

Figure 2: Hand placed markers for Elizabeth to concentrate on [3]

The NPC's interaction with the environment was also an important development point where attentive selection and relevancy were the salient features. The developers instilled Elizabeth with Systematic Design Principles (SDPs) including the need to avoid staring, ensure to look directly at the target(s), and not to look in one direction for an extended amount of time. Meshes were created of the game space environment and marker templates were hand placed/scripted around to enable certain behaviours or generate the correct 'look' [3]. Fig. 2 displays the positioning of hand placed markers for Elizabeth to concentrate on or pay attention to during the game.

For combat action, Elizabeth was programmed to always be visible but remain out of the line of friendly fire. This system was instituted by creating a path to the player via conic sections that added a cost (numerical scalar value) if the sections contained an enemy [3]. If the cost is high within a specified proximity of the player, she moved to a cover position using background servicing. Elizabeth was equipped with the ability to throw objects including bricks and block incoming attacks. Leveraging the congested environments of fighting sequences and the player's concentration directly on the enemy,

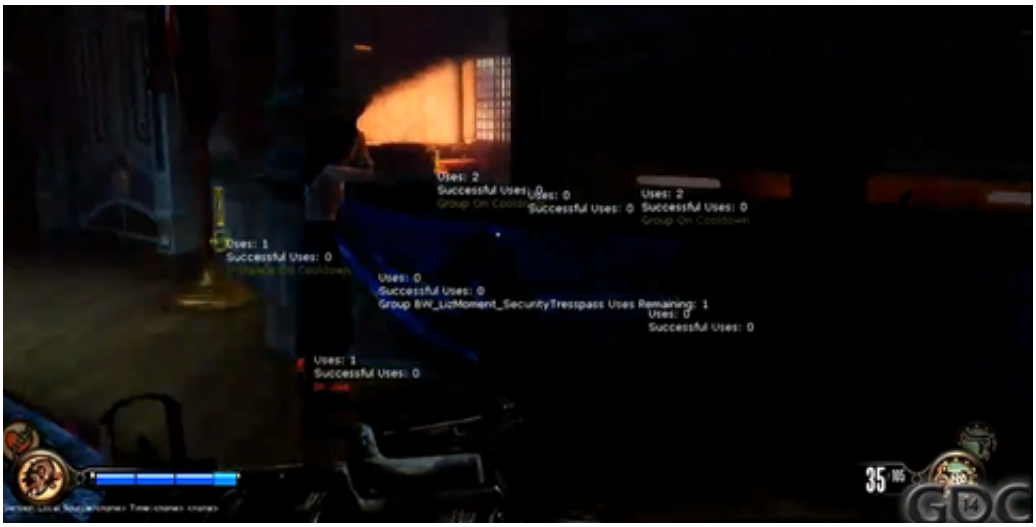


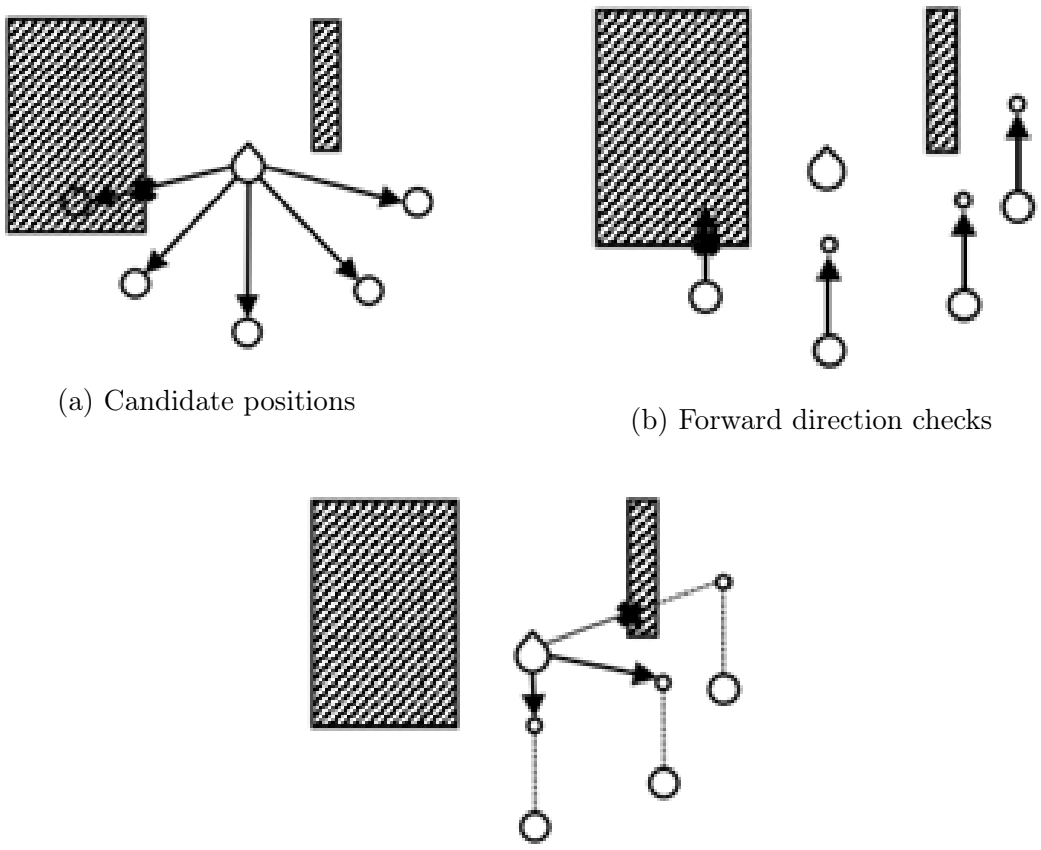
Figure 3: 'Smart' terrain [3]

the developers made Elizabeth available anywhere in the environment space by using background services to determine her exact location, if she is needed [3]. Elizabeth was also provided with the ability to initiate aforementioned tactile behaviours during combat scenes such as spotlighting the enemy by pointing out large threats with exclamations. To pinpoint the exact location of the hostility, Elizabeth was instilled with "theatre-like" spotlighting (Fig. 4) to ensure validity of the methodology when the player is unfamiliar with the geometry of the environment i.e., they do not know where the enemy is in a specific environment space, where exclamations including "look right" or "near the door" become redundant.

Figure 4: Theatre-like spotlighting with captions and enemy identifying icons [3]

2.2 The Last of Us

The Last of Us AI system was developed to enable autonomous ambient following of the player's character (PC) and combat utilities for NPCs such as taking cover and armed combat. The core concept behind the system was to ensure spatial proximity with the PC and the NPC (mostly played by the character, Ellie), thereby offsetting the need to repetitively recalibrate the trajectories of all relevant characters during combat sequences or whilst travelling together through the game environment. The auxiliary objectives were to instil the NPC with behaviours pertinent to the character exposition, enable utility including combat and non-combat actions, and provide an autonomous, incremental progression in combat vocalizations and conversation tracks [2].



(a) Candidate positions

(b) Forward direction checks

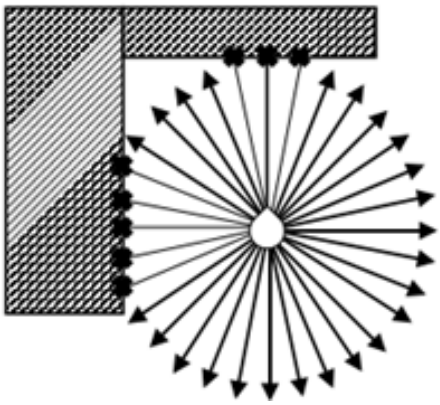
(c) Future positions

Figure 5: Navigation Mesh (NavMesh) vectors for path-planning [2]

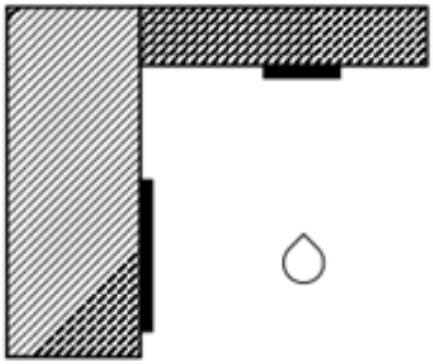
The ambient following was enabled by creating a follow system that developed and tracked all possible trajectories towards the PC and then followed the most efficient route. The trajectories were established in a torus region surrounding the PC and certain candidate positions were then evaluated for quality by casting three sets of navigation mesh (NavMesh) vectors as shown in Fig. 1 [2]. Fig. 5a represents the initial development of the candidate positions where there is no obstacle between the PC and NPC. Fig. 5b represents the second set of vectors that ensure the NPC does not face a wall/obstacle as this was found to increase behavioural abnormality. Lastly, Fig. 5c represents the vectors

to enable forward motion of the PC and NPC together for example standing on the same side of the fence as the PC. These vectors are quantified for each frame and an optimum route is chosen dependent on several metrics including proximity to the PC, visibility of environmental variables that enable achievement of the current objectives, and distance from other NPCs [2].

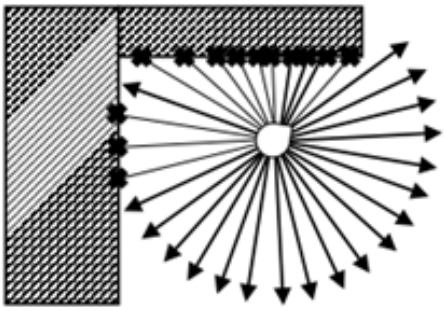
Once the optimal route was selected, realistic movement towards the destination was enabled by adding filters such as preventing small-distance movements, developing adaptive movements speeds i.e., walk when in non-combat situations and vice versa, and applying a 25% scalable speed to positively modulate the NPCs walking/running, in accordance with the movement of the PC [2]. To establish personal space boundaries, the NPC was instructed to move out of the way, if the PC gets too close and imbued with vocalization to admonish the PC. This increased the interactivity and authenticity of the NPC and subconsciously influence the PC to not encroach on the NPCs personal space.



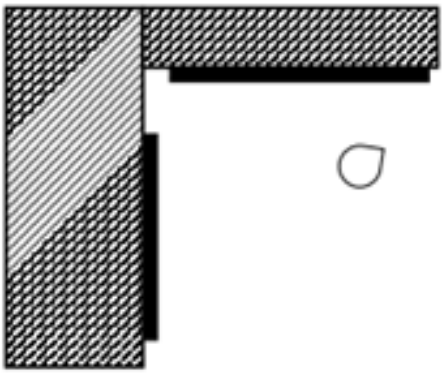
(a) Raycasts generated to determine the nearest collisions



(b) Combination of similar raycasts to form cover edge features



(c) Regenerate raycasts upon movement



(d) Modulation the *size* of the cover edge features

Figure 6: Procedural generation of cover-taking vectors [2]

For the combat sequences, due to the cover-based nature of the game, novel cover taking techniques, such as using the hybrid approach of Runtime Cover Generation (Fig. 6), were developed. Fig. 6a displays the generation of raycasts that return a Boolean value associated with either hitting or missing a collision. Using this information, Fig. 6b represents the truncation of all nearby collision points into a cache for future that uses two sets of cover edge features: static (compare the normal of the collision plane and the

corresponding location) and procedural (combine all nearby points with a similar normal). Fig. 6c and 6d represent the movement of the PC with a repetition of steps 6a and 6b. These cover edge features are then rated to choose an ideal edge on which the NPC can take cover, where the relevant criteria involved a dot-product check of the visibility to enemies and the NPCs proximity to the PC [2].

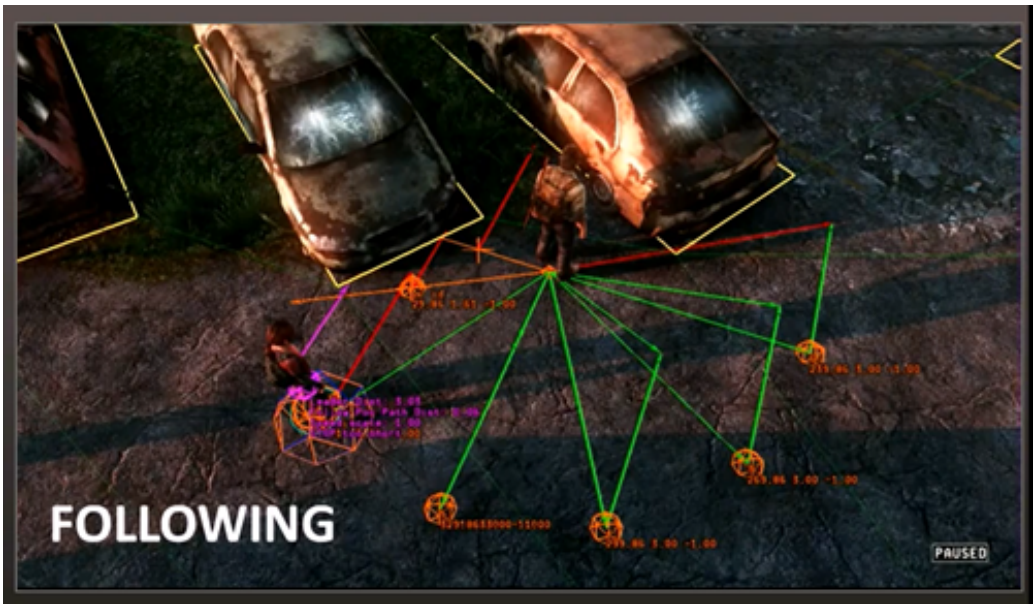


Figure 7: In-game generation of raycasts [2]

The fighting actions were designed to augment the ambient movement patterns with precise and realistic help by introducing a combat utility. The first item in the unarmed combat utility was the throwing action (NPC throws a brick) which was developed to be minimally used by "... hooking into the enemy perception and movement systems and predicting if they will be able to see the player in the near future" [2]. Secondly, melee grapples were also implemented through data by design where the NPC character was instilled with custom animations to escape grapples. Lastly, the non-combat action of delivering gifts/artifacts to the PC by the NPC was developed using a complex drop system (supplies procured from fallen enemies).

For the combat utility suite, the NPC was created with the ability to use a gun, where the positioning was executed as discussed previously and shown in-game in Fig. 7. For most practical combat situations, the NPC was defaulted to mirror the PC's actions or to take cover. Shooting was inhibited for the NPC as the character is portrayed a small girl but as the story progressed, logical statements were used to make the NPC more receptive to using a gun [2]. Instances where the PC is in danger or if the PC is actively engaging enemies, the logical statement outlined the NPCs reaction. To prevent the NPC from giving up the position of the PC in a stealth setting, the comparison between the PC's actual location and the NPC's perception of the PC's location, served as a useful heuristic to determine when to enter/exit stealth mode. Additionally, vocalizations were also added to the NPC to alert the PC about seen/unseen threats, alike.

3 Constraints

Bioshock Infinite was a complex game with multiple combat scenes. One major challenge for the developers was to ensure realistic anthropomorphic navigation of Elizabeth and to enable her to react accordingly during combat scenes. Developing robust intelligence that is able to cope with a myriad of combat levels whilst engaging in optimal companionship was accomplished with the amalgamation of several methodologies from different disciplines such as theatre studies and multi-platform game development.

The development of the NPC in *The Last of Us* faced issues related to the concept of realistic "cheating". Examples of cheating in game development include addressing the NPC's location during a combat scene and when to enable teleportation. The first issue was tackled by implementing Runtime Cover Generation, and the latter was addressed by limiting the instances when the teleportation of the NPC next to the player were warranted such as during melee attacks, and to bring the NPC closer to the player, if the distance between the two exceeds the predefined limit. Another salient point is the discovery of the NPC by enemy agents which was resolved by concealing the position of the NPC if the player was in stealth mode themselves.

4 Conclusion

Irrational Games, the developers behind *Bioshock Infinite*, wanted to create a player-facing NPC that was sufficiently intuitive to understand the personality of the real player and then modulate its actions accordingly. The developers wanted to deliver an experience not bounded by algorithmic resources but rather provided the player with a true companion based on real-life activity through the combination of theatre, language, and movement. The author of this research believes that Irrational Games achieved a high standard of NPC interactivity and reliability whilst substantially constraining behaviours deemed robotic.

Naughty Dog, the developers for *The Last of Us*, focused on enabling instinctive and natural abilities for Ellie to enable fluid movement in combat and non-combat scenes, alike. The premise of the game was to enable *true* companionship, where the player developed a realistic and genuine emotional bond with the NPC. The developers concentrated on positioning of the NPC within the game environment and utility roles such as functional movement patterns and vocalization adjustments. The author of this research believes that the developers accomplished a perfect balance between anthropomorphic movements and artificially intelligent characterisations.

References

[1] Stephany Nunneley. *BioShock Infinite - Elizabeth's role in the game deepened as development progressed*. Dec. 2012. URL: https://www.vg247.com/bioshock-infinite-elizabeths-role-in-the-game-deepened-as-development-progressed (page 1).

[2] Max Dyckhoff. "Ellie: Buddy AI in The Last of Us". In: *Game AI Pro 360: Guide to Character Behavior*. CRC Press, 2019, pp. 25–36 (pages 1, 4–6).

[3] John Abercombie. *Bringing bioshock infinite's elizabeth to life: An AI development postmortem*. URL: https://www.gdcvault.com/play/1020831/Bringing-BioShock-Infinite-s-Elizabeth (pages 1–3).