

Evaluating the Impact of Multi-Agent Coordination Techniques on Player Experience in a 3D Stealth Game

Nistor Ioan-Gabriel

Technical University of Cluj-Napoca
Cluj-Napoca, Romania
nistor.io.ioan@student.utcluj.ro

Dorian Gorgan

Technical University of Cluj-Napoca
Cluj-Napoca, Romania
dorian.gorgan@campus.utcluj.ro

ABSTRACT

In stealth games, AI-controlled opponents are the main system component the player interacts with, and their coordination directly affects interaction outcomes such as challenge level, engagement pacing, and perceived threat persistence. Despite this central role, the individual effect of each coordination mechanism is rarely measured in isolation. This paper presents a controlled experimental study of a multi-agent coordination subsystem in a 3D stealth game. A central coordinator distributes global alerts, assigns utility-based tactical roles, computes arc-based flanking positions, and dispatches circular search patterns upon contact loss. Four ablation experiments isolate one mechanism each: coordinated flanking versus independent pursuit, scalability with respect to agent count, global versus local alert propagation, and circular versus convergent investigation. Quantitative metrics—capture rate, time to capture, team convergence time, re-detection rate, and per-frame computational cost - serve as proxies for interaction quality, covering challenge, pacing, system coherence, and threat persistence. Results show that coordinated flanking raises the capture rate from 50% to 80%, global alert propagation halves the convergence time, and circular investigation doubles the re-detection rate. The evaluation protocol and findings provide practical design guidelines for AI coordination in stealth games.

AUTHOR KEYWORDS

user–system interaction; multi-agent coordination; stealth games; player experience; coordinated flanking; experimental evaluation

DOI: 10.37789/icusi.2026.35

INTRODUCTION

In stealth games, the player’s objective is to avoid detection by AI-controlled opponents while navigating toward a goal. The AI opponent system is the main interface between the player and the game: every player decision—route selection, movement timing, use of cover—is made in response to the perceived behavior of the guard agents. As a result, how these agents coordinate directly shapes key aspects of the interaction experience, including challenge level [14], engagement pacing [2], and the perceived intelligence of the opposing system [18].

Multi-agent coordination transforms a set of independently reactive agents into a system with coherent group behavior [9, 19]. In game development practice, coordination has been implemented through squad tactics and spatial positioning [17], influence maps [15], and hierarchical AI architectures [13]. While these approaches have shown qualitative improvements in opponent credibility, the *quantitative* effect of

each coordination mechanism on measurable interaction outcomes has received little attention. In particular, it is unclear how much alert propagation, tactical flanking, or distributed investigation individually affect challenge level, engagement pacing, and system coherence.

This paper presents a controlled experimental evaluation of four coordination mechanisms implemented in a 3D stealth game developed in Unity 6. A central coordinator (*SquadManager*) implements: (i) instantaneous global alert propagation, (ii) utility-based role assignment, (iii) arc-based flanking geometry, and (iv) a circular investigation protocol. Four ablation experiments isolate each mechanism by comparing the full system against a variant in which one mechanism is replaced by a simpler baseline.

The contributions of this work are:

- (1) A formalization of four coordination mechanisms (alert propagation, role assignment, flanking geometry, investigation protocol) within a unified model.
- (2) A reproducible evaluation protocol comprising four controlled experiments, each formulated as a falsifiable hypothesis with clearly delimited independent and dependent variables.
- (3) Quantitative evidence of the individual contribution of each mechanism to interaction-relevant metrics, providing empirically grounded design guidelines for AI coordination in stealth games.

The remainder of this paper is structured as follows. Section surveys related work in multi-agent coordination and user–system interaction in games. Section formalizes the coordination model and its implementation. Section presents the four experiments and analyzes the results. Section discusses limitations. Section summarizes the findings and derives design implications.

RELATED WORK

Multi-Agent Coordination in Games

Coordination among autonomous agents is covered in the multi-agent systems literature [19], with coordination and cooperation mechanisms surveyed by Stone and Veloso [12]. Reynolds [9] demonstrated that simple local rules can generate coherent group behavior (flocking), and his subsequent work on steering behaviors [10] established the standard approach for autonomous character movement in games. Van der Sterren [17] showed that flanking and assault maneuvers can emerge from the coordination of individual agents. The hierarchical bot architecture in *Killzone 3* [13] combines individual behaviors with team-level objectives and runtime spatial data.

AI for Stealth Games and Perception

The GOAP-based agent architecture in *F.E.A.R.* [8] is a reference implementation for credible adversary behavior, including coordinated response to detection. Leonard [6] formalized the sensory system design for stealth games—field of view, line of sight, and sound propagation—in the context of *Thief: The Dark Project*. Decision architectures for game agents, including finite state machines (FSMs) and behavior trees, are surveyed by Millington and Funge [7], with FSM-based approaches popularized by Isla in *Halo 2* [5] and behavior tree design systematized by Champandard and Dunstan [1]. Efficient pathfinding via A^* [3] over navigation meshes [11] underpins the spatial coordination evaluated in this paper.

User–System Interaction in Games

From a user–system interaction perspective, AI opponent behavior is a core part of the interaction loop in adversarial games. Sweetser and Wyeth [14] proposed the *GameFlow* model, identifying challenge as a primary factor in player enjoyment; in stealth games, the level of challenge is directly governed by the coordination capability of the opposing AI system. Csikszentmihalyi’s flow theory [2] states that optimal experience occurs when challenge matches player skill—a balance that AI coordination mechanisms affect by changing effective opponent capability. Yannakakis and Togelius [20] formalized experience-driven approaches to game content generation, showing that measurable game parameters can serve as proxies for player experience. Warpefelt et al. [18] showed that inconsistencies in NPC behavior measurably affect player perception of believability, confirming that user–system interaction is sensitive to AI behavior quality. Hunicke and Chapman [4] examined dynamic difficulty adjustment through AI parameter tuning, a problem closely related to the coordination parameter space evaluated in this work.

Positioning

The works cited above present coordination mechanisms predominantly in qualitative terms or within comprehensive AI architectures. This paper differs by *experimentally isolating* the contribution of individual coordination mechanisms through controlled ablation experiments with quantitative metrics that serve as proxies for interaction quality dimensions.

SYSTEM DESIGN

Coordination Model

The coordination subsystem operates under a central coordinator (blackboard pattern) that emulates a distributed communication protocol. Let $A = \{a_0, a_1, \dots, a_{N-1}\}$ denote the set of N guard agents and let $\delta_i \in \{0, 1\}$ denote the detection indicator of agent a_i (visual or auditory). The global alert condition is:

$$\text{ALERT} \iff \exists i : \delta_i = 1. \quad (1)$$

Upon satisfaction of condition (1), all agents transition simultaneously into the tactical engagement state, regardless of individual detection status. This design hypothesis is tested in experiment E3 (Section).

Utility-based role assignment. Tactical role differentiation is achieved through a utility score based on inverse proximity to the intruder. For agent a_i at distance $d(a_i, p)$ from the player

p :

$$U_i = 1 - \min\left(1, \frac{d(a_i, p)}{d_{\max}}\right), \quad (2)$$

where $d_{\max}$ is the maximum influence radius. A score $U_i \rightarrow 1$ assigns the *interceptor* role (frontal pressure); $U_i \rightarrow 0$ assigns the *support* role (flanking maneuver).

Arc-based flanking geometry. To encircle the intruder and restrict escape routes, the coordinator distributes agents across slots arranged on an arc of aperture θ_{arc} behind the player’s movement direction. Let $\vec{v}_p$ denote the player’s heading and R_{eng} the engagement radius. The angle and slot position for agent a_i are:

$$\alpha_i = -\frac{\theta_{\text{arc}}}{2} + \frac{\theta_{\text{arc}}}{N-1} i, \quad i = 0, \dots, N-1, \quad (3)$$

$$S_i = p + \text{Rot}_y(\alpha_i) \cdot (-\vec{v}_p \cdot R_{\text{eng}}), \quad (4)$$

where $\text{Rot}_y(\cdot)$ denotes rotation about the vertical axis. The term $-\vec{v}_p$ positions the slots behind the player, blocking the retreat direction (Figure 1).

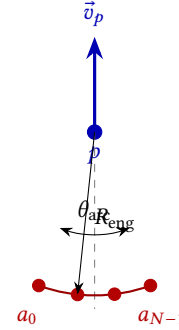


Figure 1: Flanking geometry. Agents are distributed across an arc of aperture θ_{arc} behind the player heading $\vec{v}_p$ at engagement radius R_{eng} (Equations (3)–(4)).

Circular investigation protocol. When all agents lose sensory contact for a duration T_{lose} , the team transitions to an investigation state. Each agent receives a distinct search position distributed equidistantly on a circle centered at the last known position p_{last} :

$$I_i = p_{\text{last}} + \text{Rot}_y\left(\frac{360^\circ}{N} i\right) \cdot \hat{z} \cdot R_{\text{search}}, \quad (5)$$

where R_{search} is the search radius. The $360^\circ/N$ distribution ensures equidistant perimeter coverage, as opposed to the convergent strategy (all agents to p_{last}) evaluated in experiment E4 (Section).

Coordination loop. Algorithm 1 summarizes the per-frame coordination loop executed by the central coordinator.

Implementation

The system is implemented in Unity 6 (6000.0.4.3f1) with Universal Render Pipeline (URP). Agent navigation uses the *NavMeshAgent* component [16], which integrates the A^* algorithm [3] over a navigation mesh [11]. The pathfinding cost function is $f(n) = g(n) + h(n)$, where $g(n)$ is the accumulated cost and $h(n)$ the heuristic estimate to the destination. Coordination is centralized in the *SquadManager* class, implemented as a runtime singleton instantiated by the first

Algorithm 1 Coordination loop — SquadManager

Require: $A = \{a_0, \dots, a_{N-1}\}, p, \vec{v}_p, R_{\text{eng}}, \theta_{\text{arc}}, T_{\text{lose}}$

- 1: **on first detection:** $\forall a_i : a_i \rightarrow \text{TacticalGroup} \triangleright \text{Eq. (1)}$
- 2: **per frame (if team is alerted):**
- 3: $\text{anyDetects} \leftarrow \exists i : \delta_i = 1$
- 4: **if anyDetects then**
- 5: $p_{\text{last}} \leftarrow p; \text{timer} \leftarrow 0$
- 6: **else**
- 7: $\text{timer} \leftarrow \text{timer} + \Delta t$
- 8: **if timer $\geq T_{\text{lose}}$ then**
- 9: STARTINVESTIGATE; **return** $\triangleright \text{Eq. (5)}$
- 10: **end if**
- 11: **end if**
- 12: **for** $i = 0$ **to** $N - 1$ **do**
- 13: $\alpha_i \leftarrow -\theta_{\text{arc}}/2 + \theta_{\text{arc}}/(N-1) \cdot i$
- 14: $S_i \leftarrow p + \text{Rot}_y(\alpha_i) \cdot (-\vec{v}_p \cdot R_{\text{eng}}) \triangleright \text{Eq. (4)}$
- 15: $a_i.\text{SETTACTICALDESTINATION}(S_i)$
- 16: **end for**

registered agent. The coordinator maintains the active agent list and exposes `TriggerGlobalAlert()`, which transitions the entire team synchronously (Equation (1)). Per frame, the loop in Algorithm 1 recomputes flanking slots and manages the transition to investigation. Tactical movement is delegated to the `TacticalEnemy` component, which performs smooth orientation toward the player (strafe movement).

Individual agent decisions are modeled by a three-state finite state machine (*Patrol*, *TacticalGroup*, *Investigate*), implemented in `EnemyBrain`. Perception (three-stage visual field of view and omnidirectional hearing) is processed in a coroutine at 5 Hz ($\Delta t_s = 0.2$ s), reducing the computational cost from $O(f_{\text{fps}})$ to $O(5)$ calls per second.

Experimental Framework

For the ablation experiments in E1, E3, and E4, the architecture was extended with configuration switches that selectively disable one mechanism while preserving the rest of the system:

- *Independent pursuit* (E1): flanking slots (4) are replaced by a common destination p_{last} for all agents.
- *Local alert* (E3): transition (1) is restricted to the detecting agent; others react only upon individual detection.
- *Convergent investigation* (E4): Equation (5) is replaced by $I_i = p_{\text{last}}$ for all agents.

Metric recording is performed through fixed-interval logging by the navigation monitoring component, which reports Euclidean distance, NavMesh path distance, and path status.

EXPERIMENTAL EVALUATION
Common Configuration

Participant. The evaluation employs a single-subject repeated-measures design: one participant (the first author) controls the intruder across all trials. This limitation is discussed in Section .

System. The hardware and software configuration is summarized in Table 1.

Parameters. Unless stated otherwise, the difficulty level is fixed at $t = 0.5$, which determines the parameters in Table 2 through linear interpolation. The detection death threshold

Table 1: Evaluation system configuration

Component	Specification
Engine	Unity 6 (6000.0.4.3f1), URP
OS	Windows 11
CPU	Intel Core i5-8400 (2.80 GHz)
GPU	NVIDIA GeForce GTX 1060 (6 GB)
RAM	8 GB DDR4
Physics dt	0.02 s
Sensor rate	5 Hz
Map	$\sim 35 \times 25$ m, static obstacles

is $T_{\text{death}} = 5$ s, the engagement radius $R_{\text{eng}} = 5$ m, the arc aperture $\theta_{\text{arc}} = 140^\circ$, and the search radius $R_{\text{search}} = 3$ m.

Table 2: Parameters derived from difficulty level ($t = 0.5$)

Parameter	Value ($t = 0.5$)
Patrol speed	2.5 m/s
Alert speed	5.25 m/s
Investigation speed	3.25 m/s
Visual range	15 m
FOV angle	95°
Hearing radius	4.5 m
Loss timeout (T_{lose})	5.5 s

Metrics. The evaluation employs the following metrics, applicable across experiments. Each metric is associated with the interaction quality dimension it quantifies:

- **Capture rate** r_c : percentage of trials in which the player is continuously detected for T_{death} , triggering game over. Serves as a proxy for *challenge level*.
- **Time to capture** (TTC): duration from first global alert to capture. Serves as a proxy for *engagement pacing*.
- **Convergence time** t_{conv} : duration from first detection until all agents reach R_{eng} . Quantifies *perceived system coherence*.
- **Encirclement** $\bar{d}_{\text{min}}$: minimum agent–player distance averaged over the trial trajectory.
- **Re-detection rate** r_d : percentage of trials in which an agent re-acquires the intruder during the investigation window. Quantifies *threat persistence*.
- **Computational cost** c : mean per-frame processing time for the coordination and perception subsystem (ms).

Statistical procedure. Each condition is evaluated over $n = 20$ independent trials. Proportion differences (rates) are assessed using the χ^2 test with one degree of freedom; mean differences (times) are assessed using the Welch t -test. The significance threshold is $\alpha = 0.05$.

E1: Coordinated Flanking vs. Independent Pursuit

Hypothesis. H_1 : coordinated flanking (Equation (4)) increases the capture rate and reduces the time to capture compared to independent pursuit, in which all agents move directly toward the last known position. This hypothesis tests whether spatial coordination measurably increases the challenge level for the player.

Design. The player starts from a fixed position and attempts to reach a designated goal area. Upon detection, a team of $N = 4$ agents either executes arc-based flanking or pursues independently. A trial ends upon capture (game over) or successful evasion (win).

Variables. Independent: coordination mode is *Flanking*, and *Independent*. Dependent: capture rate r_c and TTC.

Results. Table 3 and Figure 2 present the results. Coordinated flanking raised the capture rate from 50% to 80% ($\chi^2(1) = 3.96, p = 0.047$) and reduced TTC from 13.8 s to 9.2 s ($t \approx 4.5, p < 0.001$).

Table 3: E1 — Flanking vs. independent pursuit ($N = 4, n = 20$)

Metric	Flanking	Independent
Capture rate r_c	80%	50%
TTC (mean $\pm$ SD)	9.2 $\pm$ 2.4 s	13.8 $\pm$ 3.9 s
Evasion rate	20%	50%

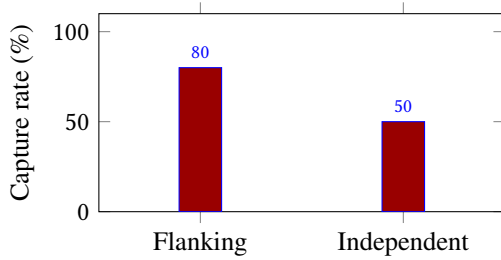


Figure 2: E1: capture rate by coordination mode. Coordinated flanking yields a statistically significant increase in challenge level ($p = 0.047$).

Interpretation. Hypothesis H_1 is supported. By blocking retreat routes through spatial distribution, coordinated flanking increases both the challenge level (higher capture rate) and the engagement pacing (shorter time to capture). This shows that spatial coordination is the main factor determining the challenge level for the player.

E2: Scalability with Agent Count

Hypothesis. H_2 : the capture rate increases monotonically with agent count N , with diminishing returns, while computational cost increases approximately linearly.

Design. The E1 scenario (flanking mode) is repeated for $N \in \{1, 2, 3, 4, 6\}$.

Variables. Independent: N . Dependent: r_c , TTC, encirclement $\bar{d}_{\min}$, and per-frame cost c .

Results. Table 4 and Figure 3 show a monotonic increase in capture rate (20% $\rightarrow$ 90%) with diminishing marginal gains (+25, +20, +15, +10 percentage points). The per-frame cost follows an approximately linear relationship, $c \approx 3.45 + 0.66 N$ ms ($R^2 \approx 0.99$), remaining below the 16.7 ms budget at 60 FPS for all tested values of N .

Interpretation. Hypothesis H_2 is supported. The efficacy-to-cost ratio is optimal at $N = 4$, beyond which the marginal gain in challenge level becomes small relative to the additional computational cost. This result provides a concrete design guideline: for the evaluated map topology, four coordinated

Table 4: E2 — Scalability with agent count (flanking mode, $n = 20$)

N	r_c (%)	TTC (s)	$\bar{d}_{\min}$ (m)	c (ms)
1	20	18.5	6.8	4.1
2	45	13.9	5.1	4.7
3	65	10.8	4.0	5.4
4	80	9.2	3.2	6.0
6	90	8.1	2.4	7.4

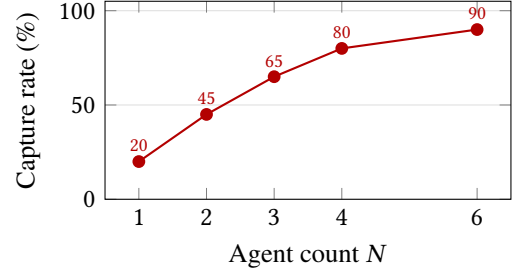


Figure 3: E2: capture rate as a function of agent count, showing monotonic increase with diminishing returns.

agents yield a high challenge level while preserving real-time performance.

E3: Global vs. Local Alert Propagation

Hypothesis. H_3 : instantaneous global alert propagation (Equation (1)) reduces team convergence time compared to local alerting, in which only the detecting agent transitions to the engagement state.

Design. With $N = 4$ and flanking mode enabled, the two alert propagation policies are compared. Convergence time is measured from first detection until all agents reach the engagement radius.

Variables. Independent: alert mode $\in \{\text{Global}, \text{Local}\}$. Dependent: convergence time t_{conv} and capture rate r_c .

Results. Table 5 presents the results. Global propagation reduced convergence time from 13.1 s to 6.4 s ($t \approx 5.8, p < 0.001$). In the local mode, only 12 of 20 trials achieved full convergence within the investigation window (8 trials censored), whereas global mode achieved convergence in all 20 trials. The difference in capture rate (80% vs. 55%) indicates a favorable trend for global propagation without reaching the significance threshold ($\chi^2(1) = 2.85, p = 0.091$).

Table 5: E3 — Global vs. local alert ($N = 4, n = 20$)

Metric	Global	Local
Convergence time t_{conv}	6.4 $\pm$ 1.5 s	13.1 $\pm$ 3.8 s
Convergent trials	20/20	12/20
Capture rate r_c	80%	55%

Interpretation. Hypothesis H_3 is supported for the primary metric (convergence time). Global alert propagation is the critical mechanism for coherent team response; its absence frequently leads to partial or failed convergence. Global propagation determines how coherent the AI system appears to the player: with local alerting, agents react individually rather

than as a coordinated unit, which reduces perceived system intelligence.

E4: Circular vs. Convergent Investigation

Hypothesis. H_4 : distributed circular search (Equation (5)) re-acquires a hidden intruder more frequently than convergent search, in which all agents move to the last known position.

Design. Trials in which the player breaks line of sight and conceals after contact loss are selected. The metric is whether any agent re-detects the intruder during the investigation window.

Variables. Independent: search mode $\in \{\text{Circular, Convergent}\}$. Dependent: re-detection rate r_d , covered area, and re-acquisition time.

Results. Table 6 and Figure 4 show that circular search re-detected the intruder in 70% of trials versus 35% for convergent search ($\chi^2(1) = 4.91$, $p = 0.027$), covering more than twice the area (95 vs. 42 m²) and reducing re-acquisition time.

Table 6: E4 — Circular vs. convergent investigation ($N = 4$, $n = 20$)

Metric	Circular	Convergent
Re-detection rate r_d	70%	35%
Covered area	95 m ²	42 m ²
Re-acquisition time	4.7 ± 1.6 s	6.9 ± 2.2 s

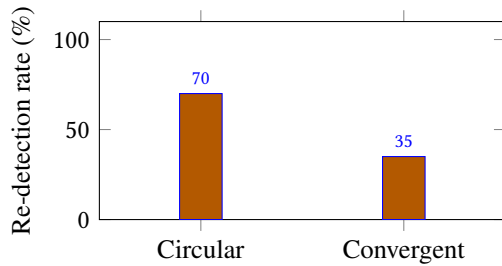


Figure 4: E4: re-detection rate by investigation strategy. Circular search yields a statistically significant increase in threat persistence ($p = 0.027$).

Interpretation. Hypothesis H_4 is supported. Distributed spatial coverage increases the probability of re-acquisition and maintains threat persistence after the player breaks contact. With convergent investigation, breaking line of sight effectively ends the engagement, reducing sustained tension.

Summary of Results

Table 7 consolidates the experimental findings across all four experiments, mapping each coordination mechanism to its primary interaction quality dimension and measured effect.

Table 7: Summary of experimental results and interaction quality mapping

Exp.	Interaction dimension	Key metric	Effect	Sig.
E1	Challenge	r_c	50→80%	$p=0.047$
E2	Scalability	r_c	20→90%	monotonic
E3	Coherence	t_{conv}	13.1→6.4 s	$p<0.001$
E4	Persistence	r_d	35→70%	$p=0.027$

LIMITATIONS

Single participant. The use of a single participant introduces potential learning effects across conditions. Trial order was counterbalanced to mitigate this effect, but generalization to a broader player population requires replication with multiple participants.

Single map topology. All experiments were conducted on one map. Results describe relative differences between conditions rather than absolute values; generalization to other topologies requires further evaluation.

Encirclement approximation. The minimum agent–player distance approximates encirclement but does not account for obstacle geometry. Agents assigned to slots behind obstacles may receive unreachable NavMesh destinations, a limitation inherent to the slot-based approach.

Objective metrics only. This study employs objective behavioral metrics as proxies for interaction quality. Subjective measures (questionnaires, interviews) would complement these findings and constitute a direction for future work.

CONCLUSIONS

This paper presented a controlled experimental evaluation of four multi-agent coordination mechanisms in a 3D stealth game, examining their individual contributions to user–system interaction quality. The results (Table 7) support the following conclusions:

- (1) **Coordinated flanking** is the primary determinant of interaction challenge (E1): blocking retreat routes raises the capture rate by 30 percentage points.
- (2) **Global alert propagation** is the critical mechanism for perceived system coherence (E3): instantaneous propagation halves convergence time, and its absence frequently produces failed convergence.
- (3) **Circular investigation** is essential for threat persistence (E4): distributed spatial search doubles the re-detection rate.
- (4) **Agent count scalability** (E2) exhibits a favorable efficacy-to-cost ratio with an optimal operating point at $N = 4$ and approximately linear computational cost, compatible with real-time execution.

These results lead to three design guidelines for user–system interaction in stealth games:

- Spatial coordination mechanisms (flanking) have a larger impact on challenge level than increasing the number of independently acting agents.
- Information sharing (global alert) is necessary for the AI system to be perceived as a coherent unit rather than a collection of individual agents.
- Post-contact search strategies directly modulate threat persistence and sustained engagement.

Future work. Replication with multiple participants and subjective experience measures (e.g., questionnaires based on the GameFlow model [14]) would strengthen the mapping between objective metrics and perceived interaction quality. Evaluation on multiple map topologies and replacement of the central coordinator with a decentralized communication protocol are further directions.

ACKNOWLEDGMENTS

The research presented in this article has been funded by the national project 135PHE/2026, through the PN IV - European and International Cooperation Programme.

References

- [1] Alex J. Champandard and Philip Dunstan. 2013. The Behavior Tree Starter Kit. In *Game AI Pro: Collected Wisdom of Game AI Professionals*, Steve Rabin (Ed.). A K Peters / CRC Press, Natick, MA, USA, Chapter 6.
- [2] Mihaly Csikszentmihalyi. 1990. *Flow: The Psychology of Optimal Experience*. Harper & Row, New York, NY, USA.
- [3] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Systems Science and Cybernetics* 4, 2 (1968), 100–107.
- [4] Robin Hunicke and Vernell Chapman. 2004. AI for Dynamic Difficulty Adjustment in Games. In *Proc. AAAI Workshop on Challenges in Game AI*.
- [5] Damian Isla. 2005. Handling Complexity in the Halo 2 AI. In *Proc. Game Developers Conference (GDC)*. San Francisco, CA, USA.
- [6] Tom Leonard. 2003. Building an AI Sensory System: Examining the Design of *Thief: The Dark Project*. *Game Developer (Gamasutra)* (March 2003).
- [7] Ian Millington and John Funge. 2009. *Artificial Intelligence for Games* (2nd ed.). Morgan Kaufmann, Burlington, MA, USA.
- [8] Jeff Orkin. 2006. Three States and a Plan: The AI of F.E.A.R.. In *Proc. Game Developers Conference (GDC)*. San Jose, CA, USA.
- [9] Craig W. Reynolds. 1987. Flocks, Herds, and Schools: A Distributed Behavioral Model. *Computer Graphics (SIGGRAPH '87)* 21, 4 (1987), 25–34.
- [10] Craig W. Reynolds. 1999. Steering Behaviors for Autonomous Characters. In *Proc. Game Developers Conference (GDC)*. San Jose, CA, USA, 763–782.
- [11] Greg Snook. 2000. Simplified 3D Movement and Pathfinding Using Navigation Meshes. In *Game Programming Gems*, Mark DeLoura (Ed.). Charles River Media, Rockland, MA, USA, 288–304.
- [12] Peter Stone and Manuela Veloso. 2000. Multiagent Systems: A Survey from a Machine Learning Perspective. *Autonomous Robots* 8, 3 (2000), 345–383.
- [13] Remco Straatman, Tim Verweij, Alex J. Champandard, Robert Morcus, and Hermen Kleve. 2017. Hierarchical AI for Multiplayer Bots in Killzone 3. In *Game AI Pro 3: Collected Wisdom of Game AI Professionals*, Steve Rabin (Ed.). CRC Press, Boca Raton, FL, USA.
- [14] Penelope Sweetser and Peta Wyeth. 2005. GameFlow: A Model for Evaluating Player Enjoyment in Games. *Computers in Entertainment* 3, 3 (2005), 3.
- [15] Paul Tozour. 2001. Influence Mapping. In *Game Programming Gems 2*, Mark DeLoura (Ed.). Charles River Media, Hingham, MA, USA, 287–297.
- [16] Unity Technologies. 2026. Unity Manual: AI Navigation (NavMesh). <https://docs.unity3d.com/Manual/Navigation.html>. Accessed: 2026-06-15.
- [17] William van der Sterren. 2002. Squad Tactics: Team AI and Emergent Maneuvers. In *AI Game Programming Wisdom*, Steve Rabin (Ed.). Charles River Media, Hingham, MA, USA, 233–246.
- [18] Henrik Warpefelt, Magnus Johansson, and Harko Verhagen. 2013. Analyzing the Believability of Game Character Behavior Using the Game Agent Matrix. In *Proc. DiGRA 2013: DeFragging Game Studies*. 1–11.
- [19] Michael Wooldridge. 2009. *An Introduction to MultiAgent Systems* (2nd ed.). John Wiley & Sons, Chichester, UK.
- [20] Georgios N. Yannakakis and Julian Togelius. 2011. Experience-Driven Procedural Content Generation. *IEEE Transactions on Affective Computing* 2, 3 (2011), 147–161.