Proceedings of CONF-MLA 2026 Symposium: Learning and Decision Making in Multi Agent Software Systems
DOI: 10.54254/2755-2721/2026.BA38730

AI-Assisted Adaptive Interface Design for Lowering Entry Barriers in Real-Time Strategy Games

Qianxi Men

Graduate School of Applied Information Technology, The Kyoto College of Graduate Studies for Informatics, Kyoto, Japan
st012640@m2.kcg.edu

Abstract. Real-time strategy (RTS) games require players to make strategic decisions while executing numerous commands. This paper presents a qualitative synthesis of recent academic research on game accessibility, human–Artificial Intelligence (AI) assistance, interface design, and community content. Six design directions are examined against accessibility, strategic depth, long-term engagement, and practical value. A layered control model is proposed with beginner, traditional, and competitive profiles sharing the same strategic rules. Routine automation requires player approval and remains visible and reversible, while tutoring uses information available to the player. Cooperative practice, hybrid interactions, and community content are considered as supporting design contexts. A conceptual mission walkthrough examines how the model could separate repeated operations from consequential decisions. The analysis yields design requirements and testable hypotheses; no prototype or new participant experiment was conducted. Controlled player studies, including unassisted transfer tasks and return-session measures, are needed to determine whether the model lowers entry barriers while preserving strategic depth.

Keywords: real-time strategy, adaptive interface, accessibility, artificial intelligence, cooperative gameplay

1. Introduction

RTS games are strategy games in which the game does not stop while the player is thinking. Players usually collect resources, construct buildings, research technology, produce units, explore the map, and fight opponents at the same time. A decision about the economy can immediately change the situation on the battlefield, and a small mistake can affect the rest of the match.

For interface design, two entry barriers deserve particular attention: learning what decisions matter and performing the operations needed to implement them. A formative study of experienced RTS players examined the content and structure of their explanations, providing a basis for connecting strategy-oriented annotations to visible actions [1]. An RTS AI survey distinguishes academic efforts to produce winning agents from the industry objective of player enjoyment [2]. These sources motivate separating strategic understanding from mechanical execution.

Recent work extends this distinction to assistance design. Research on inclusive gaming discusses AI co-pilots that could support control while preserving the player's decisions, but also identifies

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.BA38730

unresolved questions about agency and enjoyment [3]. Such findings do not establish that one assistance setting suits every player; they support examining configurable interfaces and control over delegation.

This study identifies learning cost and mechanical workload as the main entry barriers considered in its design analysis and finds that accessible interfaces, cooperative play, and optional artificial intelligence assistance address these barriers in complementary ways. A layered control model is proposed, in which optional automation, cooperative modes, and AI-based tutoring reduce repetitive operations while strategic decisions remain with the player. The model is assessed qualitatively against accessibility, strategic depth, long-term engagement, and practical value, and measurable indicators for future player studies are outlined.

2. Related work

Recent accessibility research provides a basis for considering alternative controls. Guzsvinecz analyzed accessibility features across commercial game genres and included input-support options such as key remapping, mouse settings, and controller options [4]. This study describes the availability of accessibility features; it does not demonstrate their effects on beginner RTS learning or compare RTS usability across devices. In the present design, configurable input is therefore treated as a feature that requires player evaluation.

Gallotta et al. organize the roles of large language models in games, including assistance and content creation, while highlighting hallucination and consistency problems [5]. Their synthesis supports considering AI as a tutor or design tool, but not assuming that generated advice is reliable. RTS assistance therefore needs access to valid game rules and current visible state, with advice separated from automatic execution.

Recent RTS systems illustrate possible architectures. Ahn et al. proposed Hierarchical Imitation Multi-Agent (HIMA), which coordinates specialized imitation-learning agents through a strategic planner in StarCraft II [6]. Its evaluation concerns autonomous agent performance, not beginner learning. Adaptive Command combines natural language interaction with action execution through behavior trees in StarCraft II and was evaluated in a user study involving 12 participants: four novice, four intermediate, and four expert players [7]. Its small sample and computer-opponent setting limit generalization, particularly to ranked human competition.

Human reliance on advice is a separate problem. In a Connect Four experiment, Dunning et al. found that strong AI advice could improve immediate performance while players followed it uncritically; the tested display formats did not improve learning [8]. This is evidence from a turn-based game, so transfer to RTS remains a hypothesis. It nevertheless motivates testing whether players can explain and apply a strategy after assistance is removed.

The literature therefore offers technical and interaction foundations, but limited direct evidence that an integrated RTS interface improves lasting learning. The present model combines these foundations into explicit assistance boundaries and a scenario-based design assessment.

3. Methodology

3.1. Search and case selection

This study uses a targeted narrative review of English-language journal and conference papers, prioritizing material published from 2023 onward while retaining a small number of earlier foundational studies on explanation and input design. Sources were identified through targeted web

DOI: 10.54254/2755-2721/2026.BA38730

searches and checked against publisher or official conference records. Search terms included real-time strategy, StarCraft II, game accessibility, player assistance, large language models and games, and user-generated content. Additional checks addressed human reliance on AI advice and community modding. The references were verified and updated on 5 October 2026.

Inclusion required relevance to RTS interaction, player assistance, game accessibility, transferable human–AI advice, or game creation and community participation. Developer webpages, unpublished preprints, unsupported commentary, and redundant records were excluded. Selection considered the study setting, method, reported findings, and limitations. The final purposive corpus comprises 13 formally published journal or conference papers, ten of which were published from 2023 onward. This is a qualitative narrative synthesis rather than an exhaustive systematic review or meta-analysis.

Research cases were selected for their relevance to the four analytical criteria. Ray and Sturtevant's navigation study provides a technical example of delegating unit movement to autonomous agents [9]. The Virtual Reality (VR) game Steady the Drums illustrates the combination of RTS and rhythm mechanics [10]. Duan et al.'s survey provides a framework for analyzing user-generated content and game editors, while Reisinho et al.'s review identifies barriers to community modding [11, 12]. ExInCOACH provides an example of AI tutoring evaluated in team-based StarCraft II matches [13]. These sources differ in evidence type and do not collectively validate the proposed layered interface.

3.2. Evidence synthesis and analytical criteria

The synthesis distinguishes documented features, reported study findings, and proposed benefits. Table 1 records the recent academic evidence most relevant to assistance. Four criteria guide the comparison: accessibility means reducing operational or learning obstacles; strategic depth means retaining consequential choices; engagement means providing reasons to return; practical value concerns implementation, maintenance, and verification costs. Table 2 applies these criteria to six directions. Its entries are qualitative design judgments, not measured effect sizes or a ranking.

Table 1. Evidence supporting assistance design

Source	Evidence type	Relevant finding	Limit for interface design
Dwyer et al. [3]	Inclusive gaming perspective	Co-piloting and co-design may support access and player agency.	Proposed opportunities; no validated RTS learning effect.
Gallotta et al. [5]	Game AI synthesis	Assistance and creation are distinct roles; hallucination remains a concern.	A research roadmap does not demonstrate tutoring efficacy.
Ahn et al. [6]	RTS agent benchmark	A strategic planner coordinates specialized imitation-learning agents.	Computer-opponent performance does not measure human learning.
Ma et al. [7]	RTS assistance user study	Language interface and behavior trees; 12 participants at three experience levels.	Small study against built-in AI; limited competitive transfer.
Dunning et al. [8]	Connect Four experiment	Strong advice improved immediate performance but encouraged uncritical reliance.	Turn-based setting; relevance to RTS needs testing.

Evidence types remain separate: a perspective, survey, agent benchmark, and user study cannot be treated as equivalent proof of learning. The comparison summarizes proposed mechanisms and

DOI: 10.54254/2755-2721/2026.BA38730

limits. Engagement and learning benefits require player evaluation.

Table 2. Qualitative comparison against the four criteria

Direction	Accessibility	Strategic depth	Long-term engagement	Practical value
Interface and automation	May delegate routine movement; entry benefits untested	Retained if priorities remain player-selected	Unmeasured; easier entry may help	Navigation prototype; player control and logging need validation
Cooperative play	Shared objectives; entry benefits unmeasured	Coordinated tactical choices	Team play opportunities; retention unmeasured	Needs coordination and tutoring validation
AI tutoring	May explain decisions and routines	Risk of uncritical delegation	Learning support; retention unmeasured	Needs validation and timely advice
Hybrid genres	Alternative input; entry benefits unmeasured	Additional systems may obscure strategy	Alternative play contexts; unmeasured	Higher integration and balance costs
Community content	Scenario creation needs usable tools	Custom rules can expand choices	More content; lifespan effect unmeasured	Requires developer support and community management
Cross-platform input	More input options; device-specific tests	Equivalent choices require clear controls	Broader access; retention unmeasured	Separate layouts and fairness rules

4. Recent development directions

4.1. Accessibility and interface modernization

Traditional RTS controls can require hotkeys, repeated economic actions, camera movement, and simultaneous control of several armies. Beginners may devote attention to operating the interface before understanding the strategy. Clearer information and optional routine support may free attention for decision-making, provided players can inspect and override the assistance.

Ray and Sturtevant proposed an autonomous navigation architecture combining hierarchical pathfinding with a local reinforcement-learning planner for RTS environments [9]. Their agent improved navigation performance relative to the evaluated baselines. This provides a technical basis for delegating routine movement, but the study does not establish benefits for beginner learning or player retention. In the proposed model, strategic goals remain player-selected, while approved routine execution may be delegated.

4.2. Cooperative gameplay

Cooperative play provides a possible setting for shared practice. Hua et al. evaluated ExInCOACH in two-versus-two StarCraft II matches [13]. Their results concern AI-assisted tutoring within cooperative play; they do not isolate whether cooperation itself reduces anxiety or improves RTS skills. The proposed design therefore treats those benefits as hypotheses requiring further evaluation.

A learning-oriented RTS mission could allow one player to manage production while another controls the frontline. Responsibilities should be visible and transferable so that beginners can gradually practice unfamiliar tasks. Shared play can also hide skill gaps when a stronger partner makes most decisions, so later individual performance should be measured.

DOI: 10.54254/2755-2721/2026.BA38730

4.3. Artificial intelligence and player assistance

AI could provide practice opponents, situational hints, or feedback after a match. The recent literature shows that language-based strategy and action execution can be combined, but agent performance alone is insufficient evidence of tutoring quality [6, 7]. An assistant should explain an option and its trade-off before offering execution.

For example, it could explain that additional food supports worker production while extra wood supports buildings. Advice should be checked against current rules, resource availability, and information the player can observe. Uncertain or stale advice should be withheld. Player acceptance, rejection, and later unassisted decisions should be logged to evaluate useful reliance rather than simple obedience [8].

If the AI selects expansions, technologies, army composition, and attack timing without approval, it changes the strategic role of the player. Those choices therefore remain outside routine automation. Delegation is optional, reversible, and displayed in the interface.

4.4. Hybridization with other genres

RTS mechanics can be combined with rhythm-based interaction. Pesek et al. presented the VR game Steady the Drums, which combines RTS and rhythm elements in a tower-defense setting [10]. In the full game, rhythmic patterns on virtual drums trigger actions such as attacking, defending, and summoning units. However, the educational version used for rhythm training removed strategic choices and selected army actions automatically. The experiment evaluated rhythmic perception and performance, not beginner RTS learning, strategic depth, or long-term engagement.

Additional progression systems may also create extra learning requirements. A useful hybrid feature should support the central strategic decisions rather than obscure them. The comparison in Table 2 consequently treats both its accessibility benefit and its effect on depth as conditional.

4.5. User-generated content and community development

Community-created content can provide additional scenarios and rules. Duan et al. reviewed user-generated content and game editors, including tools that allow players to modify gameplay and game resources [11]. Their survey provides a framework for analyzing creation tools and community participation. It does not establish that introducing these tools into the proposed RTS would extend the game's lifespan.

AI-assisted authoring could generate a draft scenario for manual correction, as suggested by research on language models in game creation [5]. Validation remains necessary: a scenario can be syntactically valid yet strategically unbalanced, inaccessible, or inconsistent with its stated objective.

Developer support also matters. Reisinho et al.'s review identified limited support from game companies and negative community responses as factors that can discourage mod creation [12]. For the proposed RTS design, this motivates planning continued support for creators alongside the development of community tools.

4.6. Cross-platform development

Cross-platform development should preserve the same strategic options while adapting how commands are accessed. Guzsvinecz's analysis includes motor accessibility features such as key

DOI: 10.54254/2755-2721/2026.BA38730

remapping, mouse settings, and controller options [4]. These features inform the proposed input design, but the study does not establish RTS usability across devices.

Each supported device should be evaluated for command accuracy, workload, screen readability, and access to consequential choices. Different screen sizes and input layouts may require separate interface designs. Competitive rules should make permitted automation and information access explicit, and each input configuration should allow players to inspect and override delegated actions.

5. Discussion

5.1. Learning cost and mechanical workload

Learning cost and mechanical workload are related but distinct. A player may understand why an expansion is useful yet fail to execute it while handling an attack. Conversely, automation may improve immediate survival without teaching resource tradeoffs. The interface should reveal why assistance was offered and how the player can reproduce the action manually.

Skill differences and limited session time are additional concerns that the model may help address. The small assistance study suggests that different experience groups may value support differently [7]. This motivates configurable levels, but does not establish the size of a population-wide skill gap.

5.2. A coherent combination of accessibility, cooperation, and AI assistance

Accessible interfaces, cooperative play, and optional AI assistance may provide a coherent combination because they address different obstacles: command execution, opportunities for shared practice, and interpretation of decisions. Hybrid systems and community content can supply additional contexts for using these capabilities.

Together, these systems may make the first hours of an RTS less intimidating while preserving consequential decisions. That expectation remains conditional on advice quality, player control, and opportunities to practice independently. More successful assisted play should not be equated with improved learning.

5.3. The layered control model

Figure 1 defines three selectable control profiles over the same economy, technology, and combat systems. The beginner profile offers guided information, approved worker routines, co-op tasks, and optional hints. The traditional profile defaults to manual execution with advice on request. The competitive profile uses transparent ranked rules, permits input accessibility settings, and confines AI tutoring to practice or replay review.

Adaptation changes the visibility and frequency of support, not the strategic rules. Observable events, such as idle workers or a missed production queue, can trigger a short suggestion. The player chooses whether to accept it and can change profiles between sessions. Automation records its actions and stops when overridden. Fog-of-war information is never expanded, and the system cannot initiate an attack or spend resources on a new strategic objective without the player's decision.

DOI: 10.54254/2755-2721/2026.BA38730

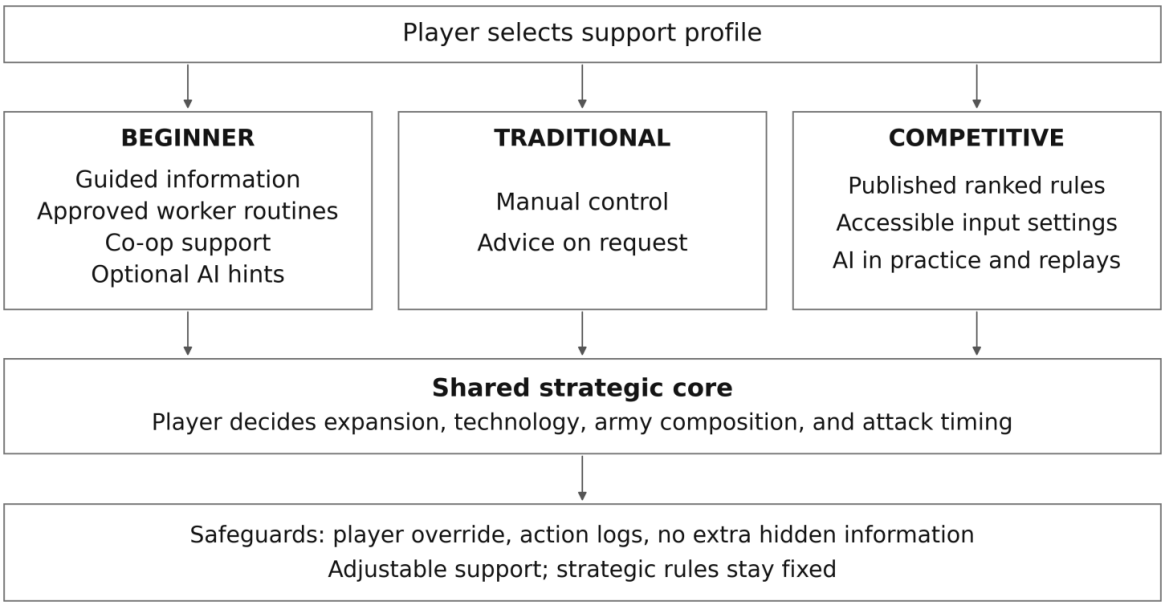


Figure 1. Layered control model and function modules (photo credit: original)

5.4. Conceptual scenario walkthrough

The model was examined through an analytical walkthrough of a hypothetical opening mission: establish an economy, scout an uncertain route, and respond to a raid while deciding whether to expand. The same resources, opponent events, strategic options, and victory condition are assumed across profiles. Table 3 compares required manual attention with the beginner profile's permitted support. It records design reasoning, not observed behavior or measured performance; no prototype or new participant experiment was conducted.

Table 3. Conceptual walkthrough of a shared RTS mission

Stage	Manual baseline	Beginner profile response	Conceptual finding
Opening economy	Player notices idle workers and assigns each task.	Offer an approved resource-ratio routine; show its assignments.	Routine execution can be delegated while the resource priority remains a choice.
Scouting uncertainty	Player decides whether to scout or defend.	Explain the unscouted area using visible information; request a decision.	Tutoring can clarify uncertainty without revealing hidden enemies.
Raid and expansion	Player switches between defense and production.	Maintain approved worker routines; present defense and expansion tradeoffs.	Assistance can separate repeated work from a strategic priority conflict.
Recovery and replay	Player interprets losses and adjusts the plan.	Explain recorded decisions, then repeat the task with assistance off.	Transfer performance is needed to distinguish learning from dependence.

The walkthrough identifies three requirements: approval before delegation, explanations grounded in observable state, and practice without assistance. It also exposes failure modes. Unclear logs can conceal automated actions; repeated hints can interrupt attention; automatic resource spending can remove tradeoffs. Traditional play retains manual execution, while competitive

DOI: 10.54254/2755-2721/2026.BA38730

practice uses replay tutoring instead of live ranked advice. These are model properties and testable hypotheses, not evidence that players will learn faster.

5.5. Future player evaluation

A future study should compare a manual baseline with optional routine automation and with automation plus tutoring, using the same map, opponent, and strategic choices. Tutorial completion time and operational errors would assess entry difficulty; an unassisted transfer task would test learning. Perceived control, workload ratings, strategic choices, and return sessions would assess agency, depth, and engagement. Counterbalanced conditions and a delayed transfer test would help separate immediate assistance from practice effects. Novices and experienced players should be analyzed separately, with sample size determined before recruitment.

6. Conclusion

This paper presents a conceptual approach to RTS interface design that aims to make mechanical execution more flexible while preserving consequential strategic choices. It treats learning cost and execution workload as potential entry barriers and synthesizes recent research on game accessibility, AI assistance, interface features, and community content, drawing on evidence from navigation agents, large language model tutoring, rhythm-based hybrid genres, and user-generated content tools.

The proposed layered model connects beginner support, manual play, and competitive practice through explicit limits on delegation. Across the three profiles, the same economy, technology, and combat systems remain shared, while the visibility and frequency of assistance change. The conceptual walkthrough shows how routine assistance and explanations can be organized around the same strategic task while leaving expansion, technology, and attack decisions with the player; it also surfaces failure modes such as concealed automation, repeated hints, and automatic resource spending that a future prototype would need to guard against.

These results are design propositions, not evidence of improved retention or learning. Their value lies in a testable structure: configurable support, visible automation, information parity, and unassisted evaluation. A prototype and controlled player study, incorporating counterbalanced conditions, delayed transfer tasks, and separate analysis of novice and experienced players, are needed before claiming that the model lowers entry barriers in practice.

References

[1] Metoyer, R., Stumpf, S., Neumann, C., Dodge, J., Cao, J., & Schnabel, A. (2010). Explaining how to play real-time strategy games. *Knowledge-Based Systems, 23*(4), 295–301.

[2] Robertson, G., & Watson, I. (2014). A review of real-time strategy game AI. *AI Magazine, 35*(4), 75–104.

[3] Dwyer, N., Harrison, M., O'Mara, B., & Harley, K. (2024). Inclusive gaming through AI: A perspective for identifying opportunities and obstacles through co-design with people living with MND. *Frontiers in Computer Science, 6*, Article 1379559.

[4] Crenshaw, N., Orzech, S., Wong, W. S., & Holloway, A. (2011). On creating a native real-time-strategy game user interface for multi-touch devices. In *Proceedings of the 6th International Conference on Foundations of Digital Games* (pp. 283–285). Association for Computing Machinery.

[5] Gallotta, R., Todd, G., Zammit, M., Earle, S., Liapis, A., Togelius, J., & Yannakakis, G. N. (2024). Large language models and games: A survey and roadmap. *IEEE Transactions on Games*. Advance online publication.

[6] Ahn, D., Kim, S., & Choi, J. (2025). Society of mind meets real-time strategy: A hierarchical multi-agent framework for strategic reasoning. In *Proceedings of the Second Conference on Language Modeling (COLM 2025)*.

DOI: 10.54254/2755-2721/2026.BA38730

[7] Ma, W., Xu, D., Lin, S., Zhang, H., & Wang, J. (2024). Adaptive command: Real-time policy adjustment via language models in StarCraft II. In *Proceedings of the 2024 6th International Conference on Distributed Artificial Intelligences* (pp. 22–30). Association for Computing Machinery.

[8] Dunning, R. E., Fischhoff, B., & Davis, A. L. (2024). When do humans heed AI agents' advice? When should they? *Human Factors, 66*(7), 1914–1927.

[9] Ray, D., & Sturtevant, N. R. (2023). Navigation in adversarial environments guided by PRA* and a local RL planner. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 19*(1), 343–351.

[10] Pesek, M., Hirci, N., Žnideršič, K., & Marolt, M. (2024). Enhancing music rhythmic perception and performance with a VR game. *Virtual Reality, 28*, Article 118.

[11] Duan, H., Liu, Y., & Cai, W. (2026). User-generated content and editors in games: A comprehensive survey. *ACM Transactions on Multimedia Computing, Communications, and Applications, 22*(6), Article 180.

[12] Reisinho, P., Raposo, R., & Zagalo, N. (2024). The ethical dilemma of modding digital games: A literature review of the creation and distribution of mods. *Convergence, 30*(2), 860–881.

[13] Hua, R., Huang, Z., Lu, J., Li, Y., & Zhao, N. (2026). ExInCOACH: Strategic exploration meets interactive tutoring for context-aware game onboarding. *Information Fusion, 131*, Article 104151.