A Player-like Agent Reinforcement Learning Method For Automatic Evaluation of Game Map

Authors: Tan Jian

Date: 2021-12-21T20:00:22+00:00

Abstract

Game maps constitute a crucial human-computer interactive content-bearing platform in major games. With the application of cellular automata (CA) and Procedural Content Generation (PCG) in map generation, the spatial scale and data volume of contemporary game maps have increased substantially. However, in game map testing procedures, automated methods such as interactive test scripts are inadequate in both depth and application breadth, particularly lacking evaluation of game maps from a player experience perspective. This research proposes an automated game map testing methodology based on reinforcement learning agents. By establishing interactive action models for agents that represent different types of player behaviors within the map, agent actions enhance comprehensive evaluation of the map environment, which can optimize game map design from a player experience perspective with quantitative evaluation metrics. Finally, campus scenes constructed in Minecraft were utilized as experimental environments to verify the effectiveness of the proposed method.

Full Text

Preamble

Game maps serve as crucial human-computer interactive content platforms in major game titles. With the application of cellular automata (CA) and Procedural Content Generation (PCG) in map generation, the spatial scale and data volume of contemporary game maps have increased dramatically. However, current automatic testing methods, such as interactive test scripts, prove inadequate in both depth and breadth of application, particularly lacking evaluation of game maps from the player experience perspective. This research proposes an automatic game map testing method based on agent reinforcement learning. By establishing interactive action models that represent different player behavior types within the map, universal evaluation of the map environment is enhanced through agent actions, enabling optimization of game map design from a player

experience perspective with quantitative assessment of deficiencies. Finally, our campus scenes in Minecraft were employed as experimental environments to verify the effectiveness of the method.

CCS CONCEPTS • Games and Play • Computational Interaction
Additional Keywords and Phrases: Games/Play; Machine Learning; Programming/Development Support; Artifact or System; Method; Theory; Application Instrumentation/Usage Logs; Quantitative Methods; Usability Study

1. INTRODUCTION

This paper introduces a modified reinforcement learning model to solve the problem of evaluating game maps from the perspective of player experience using player-like agents. As the demand for increasingly larger, high-quality game maps in massive games rises, research into game map testing, particularly automatic testing from the player experience perspective, now deserves additional attention.

In recent years, the magnitude and complexity of modern game maps have exploded with the assistance of PCG methods. For example, Assassin's Creed, a AAA game franchise, has undergone constant updates over the past decade, with its game maps growing exponentially—1700 times from 0.13 square kilometers in Damascus to 230 square kilometers in the North Sea area of Europe. Automatic procedural modeling of game maps (primarily PCG) has remained an academic research frontier for over thirty years, resulting in high-quality procedures for specific game map features of any type [?], such as landscapes [?, ?], rivers [?], plant models [?] and vegetation distribution [?], road networks [?], urban environments [?], and building facades [?]. [?] introduced declarative modeling of virtual worlds that combines integrated use of various procedural modeling techniques with a semantics-driven model to capture designer intent. [?] enables the construction of 3D buildings using grown building footprints by L-system for amateur players to create custom game content. These PCG achievements have formed engineering application tools such as Houdini, which can produce massive, large-scale game maps quickly.

Many studies have emerged on ensuring the consistency and compatibility of various PCG map elements objectively. [?] proposed a shader-based system for real-time integration of Geographic Information Systems (GIS) vector features, such as roads and rivers, into a DEM. [?] presented an interactive simulation system for cities growing over time by expanding streets in the city' s road network, also proposing a dynamic system that connects geometrical with behavioral modeling. [?] applied evolutionary and other metaheuristic search algorithms to automatically generate content for games, both digital and non-digital (such as board games). [?] discussed that the consistency of all generated content across various procedural models goes far beyond the internals of individual procedural methods. These objective verification methods (also called static testing), often implemented through automated test scripts, evaluate generated game maps according to computable criteria (e.g., "Is there a path between the entrance and exit of the dungeon?" [?], or "Does the tree have proportions within a certain range?" or a fully automated process using image processing techniques to compare and judge examples [?]). If the test fails, all or part of the candidate game map is discarded and regenerated, and this process continues until the content is satisfactory [?].

However, subjective game map testing has failed to match the level of automatic generation achieved by PCG maps. Subjective evaluation (also called dynamic testing) for game maps may involve a human observer who specifies which individual players survive in each map generation [?, ?]. Two traditional types of manual subjective testing methods for game maps—public testing and internal testing—have hindered PCG applications in the game industry. Public testing has high efficiency in testing the scope of game content (not limited to maps) but requires substantial manpower and advertising costs to promote public participation. Internal testing by game company personnel cannot cover large map ranges adequately. Moreover, internal testers cannot evaluate map design schemes on behalf of public players, which explains why MMO games like World of Warcraft and JX Online 3 have public betas in recent years.

Two possible solutions have not yet solved the problem of subjective testing of game maps. One involves incorporating human players into the PCG process rather than in subsequent testing. In PCG Design Metaphors, the PLAYER EXPERT [?] is supposed to encompass any analysis, interpretation, and adaptation suggestions specifically related to player experience in any PCG use that employs player behavior and experience as input. Kazmi and Palmer [?] describe a system embodying both a PLAYER EXPERT and a DESIGNER, premised on analyzing and interpreting player actions in terms of player skill and style. [?] proposed an interactive process between the player and computer that allows the player to guide evolving equations by observing results and providing aesthetic information at each step of the procedural models, achieving flexible complexity. Such solutions slow down the entire map generation process and require PLAYER EXPERTs with considerable PCG knowledge. Moreover, a few PLAYER EXPERTs involved in PCG cannot represent all public players and cannot validate game maps by themselves; expensive public testing remains the most reliable game map testing method.

The other possible solution is to make objective automatic tests more subjective—that is, to empower automated test scripts with prior knowledge so that these artificial players [?] have evaluation abilities closer to human player experience of game maps without losing efficiency or increasing costs. Game-playing agents are beneficial in play testing by reducing costs and the need for human play testers [?]. Methods such as MCTS and reinforcement learning (RL) can provide automated play testing without human player intervention. AI agents have proven useful in finding bugs [?] and game parameter tuning [?]. RL agents exhibit behaviors more closely resembling those of human players than traditional objective verification methods, thus increasing the probability of finding bugs and exploits. Recent techniques have tackled these scenarios using either a single model learning the dynamics of the whole game [?], or two models focusing on specific domains respectively (navigation and combat) [?]. Devlin et al. showed how observations of human play data can be used to bias MCTS to play the card game Spades [?]. They used a relative entropy measure to assess the similarity of playing styles to traces of human players. Zook et al. limited the computational resources of MCTS to simulate player skill for various games [?], with similar findings reported by Nelson [?]. Another approach to biasing the MCTS search process to be more similar to human players is described by Khalifa et al. [?]. Christoffer Holmgard et al. [?] bias MCTS using evolution applying designer-defined utility to produce a set of personas that show what different playstyles might look like in MiniDungeons 2. [?] introduced a self-learning mechanism to FPS-type game testing, where the required sum of game frames to reach a certain percentage of maximum reward (when the agent is well-trained) is regarded as a quantitative indicator of game environment difficulty.

The shortcoming of previous studies is that the behaviors of these agents do not directly adhere to real player behavior but are reinforced by reward guidance under different navigation targets. These navigation targets are not the same as players' interaction goals in game maps, and the training environments or methods are simplified to varying degrees compared to real games.

Therefore, our work advances the state of automatic game map testing through model-based reinforcement learning with player-like agents. The workflow of this research is shown as follows: 1) Game Player Behavior analysis and Clustering for game map testing; 2) Constructing Player-like Experience Evaluation Model of Game Map; 3) Modifying a Model-based reinforcement learning for game map evaluation; 4) Experiments in Minecraft map testing.

2. GAME PLAYER BEHAVIOR CLUSTERING FOR GAME MAP TESTING

In our research, the player behavior model focuses exclusively on automated testing of game maps, establishing a map-related behavior MCT (Monte Carlo Tree) model that can drive an artificial agent as a policy function.

Human player behaviors in any game can be regarded as sequential decision-making. A Markov Decision Process (MDP) represents a formal framework to describe such processes, modeling the possible interaction between an arbitrary agent and its environment over time. The MDP method requires the human player behavior model to accurately define various states and direct actions. Generally, Monte Carlo Tree Search (MCTS) is an alternative method to solve MDPs. It estimates the optimal action by building a tree of possible future (game) states and rewards, with each tree node corresponding to the state resulting from an explored action [?]. Obviously, for different human players and different game types, the structure of the Monte Carlo tree could be very different.

Map-related player behaviors have two characteristics [?]: 1) widely existent; 2) elementary interactions. First, according to Bakkes, player behavior modeling can be divided into four levels: player analysis, strategic level, tactical level, and action level [?]. This behavior system, migrated from military command, can be found in various games. Among these four behavioral levels across different games, map interaction is indispensable. Secondly, the basic element characteristics of map interaction behavior are also obvious. As a typical discrete spatial area, game maps support a limited number of player behaviors, including spatial dimension switching, speed switching, and switching frequency changes [?]. Pure map interaction elements are simple and identical in the analysis of game interaction across various studies [?]. Discrete definitions of degree of freedom and moving distance in map space include upper, lower, left, and right movement and moving step size [?, ?], further defining spatial intersection, spatial aggregation, etc., to generate interactive behavior with other game elements (shooting targets, treasure boxes). The spatial position, movement speed, and current direction of the avatar constitute game map-related player behaviors without any other social properties such as level, health, strength or attractions in game scenes such as reward items, flags, or monsters, which are diversified across different game types. Among these, movement through virtual worlds is one of the primary mechanics in open-world (sandbox) games [?]; in other words, MoveDistance is a player behavior highly related to the desire for game exploration [?]. [?] proposed that landmarks are usually used by players for pathfinding, with each player type having specific moving patterns (spatial decision trees) about transition probabilities between landmarks.

Thus, we propose Pure Spatial Monte Carlo Tree (PSMCT) as the basic framework of the map test behavior model. The basic map interaction elements are closer to the game character' s own space roaming capabilities and more aligned with the player interaction behavior model purely for game map automatic testing purposes. PSMCT contains only the definition of basic interactive elements and basic states of the game map.

3. PLAYER-LIKE EXPERIENCE EVALUATION MODEL OF GAME MAP

From the perspective of human-computer interaction, the game experience is a highly personalized and comprehensive concept containing rich elements from arousal of endogenous emotion [?] to engagement expressed through external player game duration and frequency [?]. Most previous research relies upon the assumption that player emotions can be inferred via the association of player self-reports (Subjective Player Experience Modeling, SPED) and game context variables (Objective Player Experience Modeling, OPED) [?, ?]. However, significant experimental noise usually exists in SPED, potentially caused by player learning and self-deception effects. Additionally, self-reports in SPED can be intrusive if questionnaire items are injected during gameplay sessions, while post-experience questionnaire items suffer from minimal post-experience effects

[?]. The objective PEM approach can be model-based or model-free. Model-based refers to emotional models derived from emotion theories (e.g., cognitive appraisal theory [?], usability theory [?], belief-desire-intention model, the cognitive theory by Ortony, Clore, & Collins [?], Skinner' s model [?], Scherer' s theory [?]), but there are also theories about player affect specific to games, such as Malone' s design components for fun games [?], Koster' s theory of fun [?], and game-specific interpretations of Csikszentmihalyi's concept of Flow [?], such as the popular emotional dimensions of arousal and valence [?, ?]. Model-free PEM refers to constructing an unknown mapping (model) between modalities of player input and an emotional state representation via player-annotated data [?]. Key limitations of the OPEM approach include high intrusiveness, low practicality (specific to games combined with high complexity), and questionable feasibility.

Our study selects only game map exploration within a limited time span as the objective evaluation indicator of game map experience. The reason is that game map exploration constitutes the main basis of high-level game experience. Although game experience metrics of OPEM and SPEM differ significantly, the level of experience is recognized. Spatio-temporal features of game interaction (in our study, PCMCT) are usually mapped to levels of cognitive states such as attention, challenge, and engagement [?], and the player' s cognitive processing patterns and cognitive focus may influence emotions (affective states: fun, challenge, frustration, predictability, anxiety, and boredom [?]). Ferro et al. [?] proposed the Game Experience and Elements (GEM) framework. Through exploratory factor analysis (EFA), they determined that game map exploration is the basis of game experience and the most important cognitive element.

To facilitate calculation and cooperation with test agents traversing PCMCT, this study proposes an Exploration-Based Game Map Experience Function (EBGMEF). The calculation formula for game map exploration degree is based on three assumptions: **A.** The game map is spatially uniformly discretized, such as a uniform hexagonal grid (as in Civilization 6, Total War, etc.), a uniform quadrilateral or cube (Flame Heraldic Series, Minecraft). This assumption decomposes the overall experience value of the map into the sum of experience values of each uniform discrete unit. **B.** Game exploration is time-relative, influenced by the player' s total game time. In previous research on player involvement or fatigue, important indicators like operation frequency are counted within a specified time [?, ?]. In game map testing, utilizing limited play time (perhaps defined by how long human players play on average per session) to count the size of the explored map scope is clear and coherent with human player feelings. **C.** Game exploration relates to players' memory. The experience of game maps varies with players' ability of spatial memory [?, ?], and this spatial memory is the remaining value of players' map-seeking, especially the instant impressions of fog-of-war games (such as Star Wars, Age of Empires) after map exploration. Obviously, considering the spatial memory ability of different players can better illustrate the experience value of game maps than without it.

In the EBGMEF formula, a game map consists of n discrete units, each with an initial experience value of 1. When the player' s agent roams to a map unit for the first time, the quality value of that unit is calculated only once (meaning exploration). k is the maximum number of memorized map units for a certain player type, j indicates the j-th map unit on this memorized path. For example, if a player-like agent can remember 10 map units, k is 10 and j is in $\{0 \sim 9\}$. k is determined by map memory rate γ and a memory threshold which does not appear in the equation. A memory threshold eliminates map units with little impression. For instance, if the memory threshold is set to 0.01 and γ is 0.8, then after 20 map units, the remaining memory of the 21st unit is less than 0.01, making the maximum remembered map units $k = 20$. Obviously, the farther the map unit is from the current unit ($i = 0$), the less spatial memory value remains (short-term memory characteristic of human beings [?]).

Two points about the formula are noteworthy. First, the experience value of a discrete map unit is non-renewable; calculation does not occur when the player-like agent passes again, conforming to the common sense of exploration as one-time discovery. The more frequently the player-like agent passes, the more boring the game map design becomes [?, ?]. Second, the total experience value of the game map is positively correlated with total agent exploration time. Various spatial traversal algorithms based on greedy algorithms [?] can explore a map completely given enough time. Obviously, time-related exploration efficiency better reflects the player-like experience of a game map.

4. MODEL-BASED REINFORCEMENT LEARNING FOR GAME MAP EVALUATION

Model-free reinforcement learning (RL) can learn effective policies for complex tasks with basic interactions between agents and environments with reward rules, such as AI playing Atari games [?] from image observations. However, this typically requires very large amounts of interaction data and lengthy computing processes for agents to learn, such as OpenAI 5 using about 10,000 years of equivalent human game time to outperform human world champions at the esports game Dota 2 [?].

Model-based reinforcement learning can use known behavior or environment models to set agent action policies, conduct automatic learning in specific types of data enhancement, or shape the hidden space in the time domain with substantially improved efficiency by applying predefined models [?]. Using models of environments, or informally giving the agent the ability to predict its future, has fundamental appeal for reinforcement learning [?, ?]. The spectrum of possible applications is vast, including learning policies from the model [?], capturing important details of the scene [?], encouraging exploration [?], creating intrinsic motivation [?], and counterfactual reasoning [?].

Therefore, we propose a model-based reinforcement learning approach based on PCMCT and EBGMEF. This reinforcement learning differs from previous game

RL models in several ways: 1) **Action strategy player-like**: The agent i' s action strategy function comes from the fixed action strategy model (PCMCT) of a specific player type i. The agent does not need training to improve its action strategy, ensuring the agent' s behavior remains close to human players. 2) **Experience reward player-like**: The reward R_i obtained by agent i' s roaming action through EBGMEF reflects human player experience of the game map as exploration memory, rather than serving as a stimulant for training agent behavior. 3) **Spatial memory player-like**: The value of map unit Q_i comes from the direct action reward R_i of agent i with the spatial memory rate. If Player i is more proficient at playing games, the spatial memory rate is higher [**?**, **?**], ensuring the experience assessment remains close to human players. 4) **Map total evaluation player-like**: According to our RL evaluation model, the total value of an identical game map varies with agent types, and total values of different game maps differentiate for the same agent type i, cohering with human player testing.

5. EXPERIMENTS

This study uses Minecraft as the test environment. First, Minecraft is popular in the game research community with great potential for automated world map testing [**?**]. Numerous Minecraft maps are shared online, including Hogwarts School of Magic, King' s Landing from Game of Thrones, UC Berkeley, and Beijing University of Posts and Telecommunications, providing almost unlimited game map test resources. Second, Minecraft map automatic test computation is simple. The Minecraft map is a standard octree discrete space with uniform unit size [**?**]. Player roaming actions in Minecraft are clear, and state-action calculation is straightforward. Third, the total development workload for Minecraft map automatic iterative testing is low. Microsoft has published the Malmo reinforcement learning environment and open-sourced its code, which we modified to implement our player-like reinforcement learning model.

The Malmo version used is 0.37.0, with JAVA as the programming language for rewriting. The modification includes three steps:

A. Extend the map base class of Malmo. First, the map unit has an initial exploration value (represented by a red rose on the map block) as a default attribute. Then, each map unit saves its own experience value only once during an agent test when the agent first passes the map unit.

B. Build the map test agent with an internal PCMCT. First, the PCMCT of the agent is consistent with the clustering results of the player survey in our experiment, representing the roaming behavior of a certain player type. Second, the agent maintains the map units memory queue. The queue length depends on the memory rate and forgetting threshold. For example, if the memory rate is 0.8 and the forgetting threshold is 0.01, map units beyond 20 cells do not meet the forgetting threshold ($0.0821001 < 0.01$), and the agent' s memory queue is set to 20.

C. Add global test configuration, including current test map files, number of test agents, and test time.

Beyond Malmo modifications, the experiment' s emphasis is establishing the player-like PCMCT. In this study, the transfer probability of map state and agent action in player PSMCT modeling is obtained through questionnaire, as player survey is more operable and universal in map testing tasks than other methods so far. Sharma et al. [**?**] proposed a higher-order classification of player modeling, distinguishing between (1) direct-measurement approaches (e.g., utilizing biometric data) and (2) indirect-measurement approaches (e.g., inferring player skill level from in-game observations). [**?**] analyzing game log data shows that experienced players often try more spatial choices in games. [**?**] established the potential field of game scenarios through multiple statistics of player behavior in specific game scenarios, then drove AI agents by potential field gravity in different regions. While multidimensional clustering methods [**?**] can effectively handle game behavior log data, log data contents differ significantly across game types. For example, the location of treasure boxes or monsters, which may not exist in a PCG game map, affects player behavior. In summary, for game map testing, guaranteeing representativeness and universality of state-action learning through behavior log data acquisition or in-game observation of any specific game is difficult.

Therefore, we invited human players to answer a questionnaire using a Delphi method. The questionnaire includes two question types: classification questions on player experience [**?**] and map state-action questions. In 2015, Rafet Sifa et al. [**?**] found that players' game time determines player behaviors as the dominant feature through statistics of large-scale player data on the Steam platform. However, due to differences caused by game types, their research does not involve roaming behavior clustering in game maps. In the StABLE player behavior model proposed by Fragoso et al. [**?**], advanced and non-advanced players divided by game experience show differentiation in playing behaviors (interaction frequency, moving distance, etc.), with high stability across all scenarios. Referring to these studies, our player experience classification uses total game duration, number of games played, and game playing frequency as criteria for player classification.

Based on the PSMCT, our Minecraft game map state-action possibility assessment proceeds in three steps: first, define various map states represented by representative landmarks in Minecraft; second, investigate players' state-action selection and action range (agent speed controls); finally, cluster Minecraft map behavior data by player type according to answers and establish state-action functions of PSMCT through sampling probability.

Participants were recruited at Beijing University of Posts and Telecommunications in November 2021, with varied player experience and only one participant below age 20. Approximately 3/4 were male, 1/4 female. Thirty-four players participated, with 29 validated answers returned after anomaly checking. Processed by SPSS, the answer distribution shows significant 3-clustering. All testers' behaviors in game maps are classified by hierarchical clustering according to their game experience. Even within one map state, behavior choice has a significant relationship with player type, verifying differentiation theory of player behavior in references and directly aiding establishment of three different PSMCTs. The state-action probability of each tree node derives from the sampling probability of a certain player type. The map memory rate comes from the arithmetic average of such players (Problem 7).

Test maps were selected from Beijing University of Posts and Telecommunications in Minecraft, comprising three game scenes with obvious appearance differences but identical total map unit counts. To avoid spawn point effects, all player-like agents appear in scene centers, and RL traversal time is set to an identical 10 minutes.

In cross agent-map tests, the final player-like experience of game maps differs generally for each area and agent type. From the player-like value table obtained in experiments, Agent 1 (representing experienced players) shows relatively high values for all map areas, with the main teaching building area (having highest spatial complexity) scoring highest comparatively. For Agent 2 (representing players with general experience), the highest-valued spatial area is the Second Canteen area, featuring both small buildings and flats. For Agent 3 (representing novice gamers), the most valuable map area is West School Gate area, which is completely flat.

6. DISCUSSION

Aiming at endless PCG game map testing for infinite playable value evaluation, this study presents a modified reinforcement learning model utilizing player-like agents to replace human players in map testing, greatly reducing testing workload, time, and financial costs. The contributions include:

1. This study proposes a feasible definition of agent behavior for map testing from the perspective of player behavior modeling. Player behavior modeling based on game data and questionnaire surveys has been studied in different specific games, but different game types make models highly complicated. In fact, agent behavior purely for map testing does not require much complexity. Based on game space interaction design principles, this study proposes a special pure behavior tree structure for game map testing, providing a unified player behavior model for testing various game maps.

2. From the player experience perspective, this study proposes a map value definition model. Whether agents can obtain experience value in the game, and the convenience and magnitude of obtaining it, indicate game map design quality. Previous studies coupled specific game types, making direct evaluation of map design quality difficult without experience values from other interaction elements. Starting from the commonality of interactive experience in game maps, this study decomposes overall spatial map design

quality into cumulative quality of each grid, and decomposes each grid' s quality into direct exploration and spatial memory according to game psychology theory, providing a unified player experience evaluation model for testing various game maps.

3. Based on model-based reinforcement learning, this study establishes a reinforcement learning model dedicated to map testing. The RL model in this paper differs from previous studies in that: 1) The agent' s behavior itself is not variable during the learning process, while the game map' s experience value is variable during iterative learning. The RL purpose is to automatically enhance accuracy and comprehensiveness of map experience value evaluation results. 2) The player experience value of the map is obviously player-oriented. If the agent' s behavior model represents different player types, the experience value of an identical map differs. 3) Effective map evaluation requires limitation on agent action counts. Maximizing experiential value is not the goal of training the RL model in this study. For agents with fixed player-like behavior patterns, unlimited action counts will definitely improve map experiential value, but this biases the purpose of map testing. Effective map RL testing must occur within limited time or limited agent action counts.

Through the reinforcement learning model proposed, we can select different player types to test maps automatically (three types as shown above). In our experiments, evaluations of identical game maps differ according to player types, and total experience values of different maps differ for the same player-like agent, effectively evaluating and comparing interactive values of map designs from target player type perspectives. Moreover, differences between player types help PCG designers improve existing maps or generate new maps according to target players.

Two deficiencies remain: First, the proposed RL model does not couple with PCG in iteration for automatic game map design. In this study, map evaluation is independent of player-like agent testing behavior without any spatial structure alteration of the map itself. Future work could automatically and intelligently iteratively update PCG game map design according to RL evaluation to maximize player-like experience value, advancing this study further toward artificial intelligence game design.

Second, experiments are only conducted in Minecraft, where PSMCT and EBG-MEF calculations are simpler than in other AAA games. Minecraft maps are three-dimensional volumes based on classical octrees, and agent behavior modeling is simple, with overall computational workload much smaller than complex 3D maps. However, current mainstream games, especially AAA titles, employ high-precision 3D maps where player state-action modes are more complex than Minecraft. Migrating and promoting this study into other game types requires further research in state-action strategy definition, map experience calculation, and test computing optimization. Particularly, fast extraction methods of player state-action models through game log data [**?**, **?**] are needed to replace the current independent and inefficient player questionnaire.

7. CONCLUSIONS

Reviewing current literature, game map testing struggles to match PCG development with an automatic pattern. Objective, rapid automatic testing can only reflect superficial map indicators and cannot evaluate map advantages and disadvantages from player perspectives. Game map testing still requires substantial manual participation, raising costs for the game industry. While previous literature has focused on reinforcement learning applications in games, particularly solutions for AI agents playing various games, few studies have addressed game testing assistance.

In general, this study provides new ideas and computational frameworks for automated game map testing. The contribution is presenting a modified reinforcement learning model combining objective and subjective testing, ensuring effective game map test results, including the proposed agent behavior tree model (PSMCT) and player experience evaluation function for map testing (EBGMEF). In Minecraft experiments, through player surveys, three agent types acting in three test scenes automatically evaluated and scored game maps with distinct player-like perspectives. Experimental results are more subjective than former automatic script map test methods and offer more extensive map testing capabilities than some game-specific AI models. Moreover, scope exists for further research mixing player-like AI testing with PCG methods to realize iterative automatic game design, enabling co-evolution.

References:

[1] R., M.S., et al., A declarative approach to procedural modeling of virtual worlds. *Computers & Graphics*, 2011. 35(2).

[2] GSP, M., The definition and rendering of terrain maps, in *SIGGRAPH ' 86: proceedings of the 13th annual conference on computer graphics and interactive techniques*. 1986, ACM: New York, NY, USA.

[3] Musgrave, F.K., Methods for realistic landscape imaging. 1993, Yale University.

[4] Kelley, A.D., M.C. Malin and G.M. Nielson, Terrain simulation using a model of stream erosion, in *SIGGRAPH ' 88: proceedings of the 15th annual conference on computer graphics and interactive techniques*. 1988, ACM: New York, NY, USA.

[5] Prusinkiewicz, P. and M. Hammel, A fractal model of mountains with rivers. *Proceeding of graphics interface ' 93*, 1993.

[6] Teoh, S.T., River and coastal action in automatic terrain generation, in *CGVR 2008: proceedings of the 2008 international conference on CG & VR*. 2008, CSREA Press: Las Vegas, Nevada, USA.

[7] Prusinkiewicz, P. and A. Lindenmayer, *The algorithmic beauty of plants*. 1990, New York, NY, USA: Springer-Verlag.

[8] Deussen, O., et al., Realistic modeling and rendering of plant ecosystems, in *SIGGRAPH ' 98: proceedings of the 25th annual conference on computer*

graphics and interactive techniques. 1998, ACM: New York, NY, USA.

[9] Sun, J., et al., Template-based generation of road networks for virtual city modeling, in *VRST' 02: proceedings of the ACM symposium on virtual reality software and technology*. 2002, ACM: New York, NY, USA.

[10] Parish, Y. and P. Müller, Procedural modeling of cities, in *SIGGRAPH ' 01: proceedings of the 28th annual conference on computer graphics and interactive techniques*. 2001, ACM: New York, NY, USA.

[11] Wonka, P., et al., Instant architecture, in *SIGGRAPH ' 03: proceedings of the 30th annual conference on computer graphics and interactive techniques*. 2003, ACM: New York, NY, USA.

[12] Müller, P., et al., Procedural modeling of buildings, in *SIGGRAPH ' 06: proceedings of the 33rd annual conference on computer graphics and interactive techniques*. 2006, ACM: New York, NY, USA.

[13] Finkenzeller, D., Detailed building facades. *IEEE Computer Graphics and Applications*, 2008.

[14] Dumim, Y. and K. Kyung-Joong, 3D Game Model and Texture Generation Using Interactive Genetic Algorithm. *Computers in Entertainment (CIE)*, 2016. 14(1).

[15] Smelik, R.M., et al., A Survey on Procedural Modelling for Virtual Worlds. *Computer Graphics Forum*, 2014. 33(6): p. 31-50.

[16] Tessler, C., et al. A Deep Hierarchical Approach to Lifelong Learning in Minecraft. in *AAAI Publications, Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*. 2016.

[17] Cook, M., S. Colton and J. Gow, Automating Game Design In Three Dimensions. 2014.

[18] Bruneton, E. and F. Neyret, Real-time rendering and editing of vector-based terrains, in *Computer graphics forum: eurographics 2008 proceedings*. 2008: Crete, Greece.

[19] Vanegas, C.A., et al., Interactive design of urban spaces using geometrical and behavioral modeling, in *ACM TOG: proceedings of ACM SIGGRAPH Asia*. 2009, ACM.

[20] J., T., et al., Search-Based Procedural Content Generation: A Taxonomy and Survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 2011. 3(3): p. 172-186.

[21] Huang, H., Intelligent Pathfinding Algorithm in Web Games. 2020: Cyber Security Intelligence and Analytics.

[22] Machado, P. and A. Cardoso. Computing aesthetics. in *Lecture Notes in Artificial Intelligence*.

[23] Sims and Karl, Artificial evolution for computer graphics. ACM, 1991: p. 319-328.

[24] Secretan, J., et al. Picbreeder: evolving pictures collaboratively online. in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2008.

[25] Khaled, R., M.J. Nelson and P. Barr. Design metaphors for procedural content generation in games. in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2013.

[26] Kazmi, S. and I.J. Palmer, Action Recognition for Support of Adaptive Gameplay: A Case Study of a First Person Shooter. *International Journal of Computer Games Technology*, 2010.

[27] Sims, K., Interactive evolution of equations for procedural models. *The Visual Computer*, 2005. 9: p. 466-476.

[28] Vítek, A. Cross-Game Modeling of Player' s Behaviour in Free-To-Play Games. in *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*. 2020. Genoa, Italy: Association for Computing Machinery.

[29] Roohi, S., et al., Predicting Game Difficulty and Engagement Using AI Players. *Proc. ACM Hum.-Comput. Interact.*, 2021. 5(CHI PLAY): p. Article 231.

[30] Zhu, J. and S. Ontañón. Player-Centered AI for Automatic Game Personalization: Open Problems. in *International Conference on the Foundations of Digital Games*. 2020. Bugibba, Malta: Association for Computing Machinery.

[31] Ariyurek, S., A. Betin-Can and E. Surer, Automated Video Game Testing Using Synthetic and Human-Like Agents. 2019.

[32] Fernando, D., et al. AI-based playtesting of contemporary board games. in *International Conference*. 2017.

[33] Isaksen, A., G. Dan and A. Nealen, Exploring Game Space Using Survival Analysis. 2015.

[34] Harmer, J., et al. Imitation Learning with Concurrent Actions in 3D Games. in *2018 IEEE Conference on Computational Intelligence and Games (CIG)*. 2018.

[35] Lample, G. and D.S. Chaplot. Playing FPS Games with Deep Reinforcement Learning. in *31st AAAI Conference on Artificial Intelligence (AAAI-17)*, San Francisco, USA. 2016.

[36] Cowling, P.I., Combining Gameplay Data with Monte Carlo Tree Search to Emulate Human Play.

[37] Zook, A., B. Harrison and M.O. Riedl, Monte-Carlo Tree Search for Simulation-based Strategy Analysis. 2019.

[38] Nelson, M.J. Investigating Vanilla MCTS Scaling on the GVG-AI Game Corpus. in *2016 IEEE Conference on Computational Intelligence and Games (CIG)*. 2017.

[39] Khalifa, A., et al., Modifying MCTS for Human-Like General Video Game Playing. AAAI Press.

[40] Holmgård, C., et al., Automated Playtesting With Procedural Personas Through MCTS With Evolved Heuristics. *IEEE Transactions on Games*, 2019. 11(4): p. 352-362.

[41] Bergdahl, J., et al., Augmenting Automated Game Testing with Deep Reinforcement Learning.

[42] Bakkes, S., P. Spronck and G.V. Lankveld, Player behavioural modelling for video games. *Entertainment Computing*, 2012. 3(3): p. 71-79.

[43] Drachen, A. and M. Schubert, Spatial game analytics and visualization. 2013. p. 1-8.

[44] Fragoso, L. and K.G. Stanley, StABLE: Analyzing Player Movement

Similarity Using Text Mining.
[45] Laviola, J.J. and R.L. Marks, An introduction to 3D spatial interaction with video game motion controllers. ACM, 2010: p. 1-78.
[46] Jr, L.V. and D.F. Keefe. Course: 3D Spatial Interaction: Applications for Art, Design, and Science. in *ACM SIGGRAPH 2011*. 2011.
[47] Mueller, S., et al. HeapCraft: interactive data exploration and visualization tools for understanding and influencing player behavior in Minecraft. in *the 8th ACM SIGGRAPH Conference*. 2015.
[48] Thawonmas, R., et al. Clustering of Online Game Users Based on Their Trails Using Self-organizing Map. in *Entertainment Computing - ICEC 2006, 5th International Conference, Cambridge, UK, September 20-22, 2006, Proceedings*. 2006.
[49] Melhart, D., A. Liapis and A.G.N. Yannakakis., Towards General Models of Player Experience: A Study Within Genres., *IEEE Conference on Games*. Retrieved https://again.institutedigitalgames.com/. 2021.
[50] Gratch, J.M. and S.C. Marsella, Evaluating a Computational Model of Emotion. *Autonomous Agents and Multi-Agent Systems*, 2005.
[51] Conati, C., Probabilistic assessment of user' s emotions in educational games. *Applied Artificial Intelligence*, 2002. 16(7): p. 555-575.
[52] Drachen, A., et al., Correlation between heart rate, electrodermal activity and player experience in first-person shooter games. Dragon Consulting; Department of Computer Science University of Saskatchewan; Center for Computer Games Research IT University Copenhagen; Center for Computer Games Research IT University of Copenhagen, 2011.
[53] Pagulayan, R.J., et al., User-centered design in games. L. Erlbaum Associates Inc., 2002.
[54] Yannakakis, G.N. Preference learning for affective modeling. in *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*. 2009.
[55] Frijda, N.H., *The Emotions*. Studies in Emotion & Social Interaction, 1986. 1(5): p. 583-584.
[56] Isbister, K., *Game Usability*. 2008.
[57] Ortony, A., G.L. Clore and A. Collins, *The Cognitive Structure of Emotions. Contemporary Sociology*, 1988. 18(6): p. 2147-2153.
[58] Skinner, B.F., *The behavior of organisms: An experimental analysis*. appleton century new york smith a, 1938.
[59] Scherer and R. Klaus, Studying the emotion-antecedent appraisal process: An expert system approach. *Cognition & Emotion*, 1993. 7(3-4): p. 325-355.
[60] Malone, T.W. What makes things fun to learn? heuristics for designing instructional computer games. in *Acm Sigsmall Symposium & the First Sigpc Symposium on Small Systems*. 1980.
[61] Koster, R., *A Theory of Fun for Game Design*. Paraglyph Press, 2004.
[62] Csikszentmihalyi, M., *Flow: The Psychology of Optimal Experience. Design Issues*, 1991. 8(1).
[63] Feldman, L.A., Valence Focus and Arousal Focus: Individual Differences in the Structure of Affective Experience. *Journal of Personality and Social*

Psychology, 1995. 69(1): p. 153-166.
[64] Russell, J.A., Core Affect and the Psychological Construction of Emotion. *Journal of Behavioral Finance*, 2003.
[65] Asteriadis, S., et al., A natural head pose and eye gaze dataset. ACM, 2009: p. 1-4.
[66] Shaker, N., G.N. Yannakakis and J. Togelius. Towards Automatic Personalized Content Generation for Platform Games. in *Proceedings of the Sixth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE 2010, October 11-13, 2010, Stanford, California, USA*. 2010.
[67] Ferro, L.S., The Game Element and Mechanic (GEM) Framework: a structural approach for implementing game elements and mechanics into game experiences. *Entertainment Computing*, 2020.
[68] Fragoso, S. Interface design strategies and disruptions of gameplay: notes from a qualitative study with first-person gamers. in *International Conference on Human-Computer Interaction*. 2014.
[69] Toups, Z., et al., Making Maps Available for Play: Analyzing the Design of Game Cartography Interfaces. *ACM transactions on computer-human interaction*, 2019. 26(5): p. 1-43.
[70] Matsumoto, Y. and R. Thawonmas. MMOG Player Classification Using Hidden Markov Models. 2004. Berlin, Heidelberg: Springer Berlin Heidelberg.
[71] Stahlke, S.N. and P. Mirza-Babaei, Usertesting Without the User: Opportunities and Challenges of an AI-Driven Approach in Games User Research. *Comput. Entertain.*, 2018. 16(2).
[72] Ashlock, D. and C. Salge, Automatic Generation of Level Maps with the Do What' s Possible Representation. 2019.
[73] Mnih, V., et al., Playing Atari with Deep Reinforcement Learning. *Computer Science*, 2013.
[74] Berner, C., et al., Dota 2 with Large Scale Deep Reinforcement Learning. 2019.
[75] Doll, B.B., D.A. Simon and N.D. Daw, The ubiquity of model-based reinforcement learning. *Current opinion in neurobiology*, 2012. 22(6): p. 1075-1081.
[76] Kaiser, L., et al., Model-Based Reinforcement Learning for Atari. *ArXiv*, 2020. abs/1903.00374.
[77] Hafner, D., et al., Learning Latent Dynamics for Planning from Pixels. 2018.
[78] Piergiovanni, A.J., A. Wu and M.S. Ryoo, Learning Real-World Robot Policies by Dreaming.
[79] Rybkin, O., et al., Unsupervised Learning of Sensorimotor Affordances by Stochastic Future Prediction. 2018.
[80] Ebert, F., et al., Self-Supervised Visual Planning with Temporal Skip Connections. 2017.
[81] Finn, C., et al. Deep spatial autoencoders for visuomotor learning. in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016.
[82] Watter, M., et al., Embed to Control: A Locally Linear Latent Dynamics

Model for Control from Raw Images. *Advances in neural information processing systems*, 2015.
[83] Ha, D. and J. Schmidhuber, Recurrent World Models Facilitate Policy Evolution. 2018.
[84] Jeong, K. and J. Choi, Deep Recurrent Neural Network. *Communications of the Korean Institute of Information Scientists and Engineers*, 2015. 33.
[85] Schmidhuber, J., Formal Theory of Fun and Creativity. DBLP, 2010.
[86] Buesing, L., et al., Woulda, Coulda, Shoulda: Counterfactually-Guided Policy Search. 2018.
[87] Nebel, S., S. Schneider and G.D. Rey, Mining Learning and Crafting Scientific Experiments: A Literature Review on the Use of Minecraft in Education and Research. *Journal of Educational Technology & Society*, 2016. 19(2): p. 355-366.
[88] Salge, C., et al., Generative Design in Minecraft: Chronicle Challenge. 2019.
[89] Sharma, M., et al., Player modeling evaluation for interactive fiction. 2009.
[90] Moll, P., et al., How Players Play Games: Observing the Influences of Game Mechanics. 2019.
[91] Stefan, S., et al., Learning human-like Movement Behavior for Computer Games. 2004, MIT Press. p. 315-323.
[92] Bauckhage, C., A. Drachen and R. Sifa, Clustering Game Behavior Data. *IEEE Transactions on Computational Intelligence and AI in Games*, 2015. 7(3): p. 266-278.
[93] Brühlmann, F. and E. Mekler, Surveys in Games User Research. 2018.
[94] Sifa, R., A. Drachen and C. Bauckhage. Large-Scale Cross-Game Player Behavior Analysis on Steam. in *AIIDE*. 2015.
[95] Lopes, R., T. Tutenel and R. Bidarra. Using gameplay semantics to procedurally generate player-matching game worlds. in *Proceedings of the The third workshop on Procedural Content Generation in Games*. 2012. Raleigh, NC, USA: Association for Computing Machinery.
[96] (Reference appears incomplete in original)

Figures

Source: ChinaXiv —Machine translation. Verify with original.

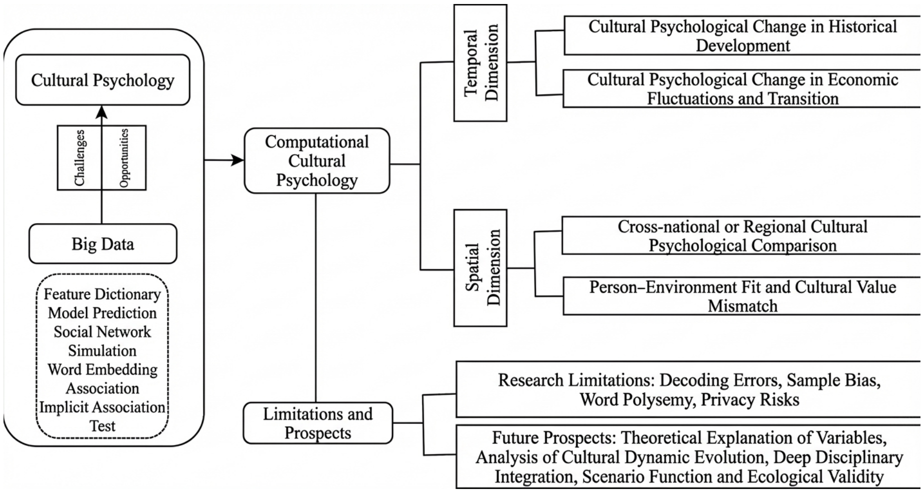
Cultural Psychology
Challenges
Opportunities
Big Data
Feature Dictionary
Model Prediction
Social Network Simulation
Word Embedding Association
Implicit Association Test
Computational Cultural Psychology
Temporal Dimension
Cultural Psychological Change in Historical Development
Cultural Psychological Change in Economic Fluctuations and Transition
Spatial Dimension
Cross-national or Regional Cultural Psychological Comparison
Person–Environment Fit and Cultural Value Mismatch
Limitations and Prospects
Research Limitations: Decoding Errors, Sample Bias, Word Polysemy, Privacy Risks
Future Prospects: Theoretical Explanation of Variables, Analysis of Cultural Dynamic Evolution, Deep Disciplinary Integration, Scenario Function and Ecological Validity

Figure 1: Figure 1