Cautious Curiosity: A Novel Approach to a Human-Like Gameplay Agent

Chujin Zhou[1], Tiago Machado[2], Casper Harteveld[2]

[1] Macau University of Science and Technology
[2] Northeastern University
zcj2290151272@gmail.com, tiago.la.machado@gmail.com, c.harteveld@northeastern.edu

Abstract

We introduce a new reward function direction for intrinsically motivated reinforcement learning to mimic human behavior in the context of computer games. Similar to previous research, we focus on so-called "curiosity agents", which are agents whose intrinsic reward is based on the concept of curiosity. We designed our novel intrinsic reward, which we call "Cautious Curiosity" (CC) based on (1) a theory that proposes curiosity as a psychological definition called information gap, and (2) a recent study showing that the relationship between curiosity and information gap is an inverted U-curve. In this work, we compared our agent using the classic game Super Mario Bros. with (1) a random agent, (2) an agent based on the Asynchronous Advantage Actor Critic algorithm (A3C), (3) an agent based on the Intrinsic Curiosity Module (ICM), and (4) an average human player. We also asked participants (n = 100) to watch videos of these agents and rate how human-like they are. The main contribution of this work is that we present a reward function that, as perceived by humans, induces an agent to play a computer game similarly to a human, while maintaining its competitiveness and being more believable compared to other agents.

Introduction

Artificial intelligence (AI) has opened up a wide range of possibilities for game developers, from content creation (Togelius et al. 2011) to game debugging (Machado et al. 2018a). One of the most common applications of AI in games is the development of Game Playing Agents (GPAs), usually with two goals: as a test environment for AI methods and for automated game testing. From Min-Max chess players (McAllester 1988) to AlphaGo (Silver et al. 2017), games have created virtual environments challenging enough for advancing AI methods. AI has also helped developers automatically test their games, reducing development costs.

However, most of the GPAs available today report flaws, such as levels that are easy to defeat or platforms that cannot be reached in one jump (Smith, Whitehead, and Mateas 2010; Shaker, Shaker, and Togelius 2013). While these results reduce the workload of a human tester, it says little about the subjective aspects of the game and what areas remain unexplored. This happens because these agents are designed with the goal of winning at all costs. The agent will try to reach a winning status as quickly as possible, leaving behind much of the game content. Certain questions may never be answered, since the agent avoids circumstances such as confronting certain enemies or accessing difficult level areas that do not lead to a win status, but can bring the player rewards (coins, power-up items, etc.).

These limitations of GPAs have been studied by AI researchers. One possible solution is the use of human-like agents. Such agents are implementations to mimic human players and explore the game as a human would. However, this solution comes with some challenges, including the level of human likeness (sooner or later someone will figure out that they are not human at all) and the amount of training data generated by humans in order for the agents to play the game at an acceptable level.

More recently, agents using the Intrinsic Curiosity Model (ICM) have reduced the need to obtain data from humans and are able to learn how to play games on their own, with promising results, even in contexts previously considered difficult for agents to access (e.g., Montezuma's Return) (Aytar et al. 2018; Burda et al. 2018)). While ICM agents have shown good results playing games without relying on human data, their human-likeness is still in need of being better explored.

Motivated by ICM developments, we have developed a human-like agent to investigate "how human an ICM agent can be without losing its capabilities." To achieve this goal, we modeled the reward function of an agent based on the psychological theory of information gap curiosity (Dubey and Griffiths 2017; Loewenstein 1994), which states that the motivation of an agent increases proportionally to the degree of information learned. We hypothesized that this theory, in conjunction with recent ICM advances, would balance human-likeness and competitiveness.

To test our hypothesis, we implemented our "Cautious Curiosity" (CC) agent in the context of Super Mario Bros., compared its performance against baseline Game Playing Agents [1] and conducted a study with 100 participants who rated our agent on "how human it looks?"

Our results show that ICM combined with informa-

[1] https://github.com/AndyZCJ/Cautious-Curiosity-Agent

tion gap theory can potentially lead to the development of human-like agents that help developers test their games as a human would, without losing competitive features. In addition, we gained valuable insights into how game content can influence the further development of GPAs.

Background & Related Work

Game Play Agents

When designing games, designers must explore all possible actions and outcomes in the game. As their systems become more complex, it becomes difficult to control the range of all possible scenarios that may result from different players interacting with the system (de Mesentier Silva et al. 2017a). Automated playtesting helps designers develop more reliable game systems. In this scenario, the use of gameplay agents helps developers automate gameplay tasks, increasing the chances of crash-free games. It also reduces the hours of repetitive human tasks, which can lead to boredom and stress (Briziarelli 2016). However, automatic gameplay is not a mature method when subjective metrics such as challenge and game enjoyment need to be evaluated (Yannakakis and Togelius 2018; Pedersen, Togelius, and Yannakakis 2009, Chapter 5). Agents whose main goal is to win the game as fast as possible are not suitable for this task, as they do not guarantee to explore the game enough to consider states that do not lead to victory (Zhao et al. 2020; Shao et al. 2019; Aytemiz et al. 2020). Therefore, agents that can exhibit "as human as possible" behavior are necessary. In this way, designers can evaluate non-functional questions (Callele, Neufeld, and Schneider 2006) better than functional ones. However, designing an agent is a complex process. It involves fully understanding the game strategy, knowing the algorithm that fits the best strategy, and figuring out how to reward the agent when it performs the right action and how to punish it when it performs the wrong action (Yannakakis and Togelius 2018, Chapter 3). A typical gameplay agent can be built on different strategies, such as graph-based heuristic search (de Mesentier Silva et al. 2017b), evolution and genetic algorithms (Togelius, De Nardi, and Lucas 2006; Arulkumaran, Cully, and Togelius 2019), supervised unsupervised reinforcement learning (Chen 2016; Blog 2021), and imitation learning (Sestini et al. 2022; Hasegawa et al. 2013; Hosu and Rebedea 2016). For our gameplay agents, we focus on reinforcement learning with intrinsic motivation. This choice is consistent with our motivation to develop an agent that mimics humans while maintaining the competitiveness of the average player, i.e., it explores the game like a human without forgetting about winning. Finally, it is free from the huge amounts of data required to train other solutions presented in this section.

Human-Like Agents

Many researchers have proposed algorithms that allow game agents to behave like a real human player. Attempts to develop human-like agents using Monte Carlo Tree Search (MCTS) have yielded promising results in recent years. Khalifa et al. (Khalifa et al. 2016) proposed a modified MCTS that can make the agent behavior more similar to a human. They tested their algorithm in games such as Zelda, Pacman, and Boulderdash and set up a Turing test-like user study. Their results show that their algorithm behaved more like a human from the user study participants' perspective than other MCTS-based methods. Devlin et al. (Devlin et al. 2016) biased MCTS with gameplay trace data collected from human players to emulate human gameplay. By analyzing the data from real humans, they define a general human gameplay style and indirectly imitate it. Their results show that their agent can behave like a human in the traditional card game of spades. In addition to MCTS, methods based on Deep Learning are receiving attention. Gudmundsson et al. (Gudmundsson et al. 2018) trained a convoluted neural network that takes player data from Candy Crush Saga (CCS) and Candy Crush Soda Saga (CCSS) as input to predict human actions in a given position. Their experiments show that this algorithm works well in CCS and CCSS, and it is also suitable for many games, especially where content creation is sequential. All in all, the field of human-like agents has shown great potential in recent years and generally attracted more attention. However, there are some issues surrounding the development of human-like agents that motivated us to conduct this research, which we will address in the following subsection.

Issues of Human-Like Agents

Human-like agent is a vague definition that has different meanings in different fields. We follow the definition of human-like agents in the game research literature, which describes human-like agents as agents that are indistinguishable from human players when compared by human judges. However, game players can be broadly classified into three types, master players, average players, and new players (Poels et al. 2012). One problem with developing human-like agents is that many agents behave like master players and tend to pass the game as quickly as possible, so they usually ignore game content (Lillicrap et al. 2015; Mnih et al. 2015). However, game developers want agents to help them find potential bugs and even analyze the difficulty level of the game (Ariyurek, Betin-Can, and Surer 2019). In this case, agents should try to interact with as much game content as possible, but agents with master skills cannot fulfill this requirement. Moreover, it will be easy for an agent to behave like a master player, since master players can complete the game quickly and with a low error rate. If we train our agent millions of times, it will behave even better than a master player, and sometimes it will even show some non-human behaviors. For example, in Starcraft, a well-trained agent can control more than 100 units at the same time and make each unit perform different activities (Vinyals et al. 2019). Such a level of capability cannot be achieved by humans. Unlike master players, average players are more like normal people. They can definitely pass the game, but they will not make mistakes like a new player or show skills like a master player. We try to avoid the two extremes: the master (because its superhuman abilities can be easily noticed by players), and the novice (because it is too inexperienced to provide useful information about the game). Therefore, our goal is to make our agent indistinguishable from the average player. This "competitive trait" is one of the differences of our work related to recent studies on human-like agents (Milani et al. 2023).

Reinforcement Learning

Reinforcement learning (RL) is based on the framework of Markov decision processes (MDPs). The goal of reinforcement learning is to maximize long-term cumulative reward through interaction with an observable or partially observable world (Mohan and Laird 2009). In traditional reinforcement learning, an agent observes the environment and learns a policy from the environment. The reward is determined by a task-specific reward function. Game agents learn from the reward the environment gives them and eventually form a "best" policy. In intrinsically motivated reinforcement learning, the agent observes the environment and then translates the observation and reinforcement signal, such as the game score, into intrinsic reinforcement, which is then optimized by traditional RL (Dossa et al. 2019). Instead of being motivated only by the external environment, the behavior of intrinsically motivated RL agents is motivated by the internal desire to do something for its own sake. In this case, the goal of agents is to maximize the combination of extrinsic and intrinsic rewards. Curiosity belongs to the intrinsic rewards group and plays an important role in tasks such as exploration. We will talk more about curiosity in the following subsection.

Curiosity in Reinforcement Learning

In the literature on intrinsic motivation as discussed in the context of games, learning, and artificial intelligence, a frequently discussed form of intrinsic motivation is *curiosity*. Its importance was highlighted as early as the early 1980s in a psychological study of intrinsic motivation and games by Malone (Malone 1981). According to Malone, curiosity, fantasy, and challenge are the three most important factors in games. However, both challenge and fantasy have strong connections with curiosity. For example, game developers incorporate interesting elements into games to attract players, and they also set goals or difficulties to encourage players. In this case, both fantasy and challenge can be explained by curiosity, which means that curiosity is the dominant motivation when people play games. Therefore, we believe that it is necessary to study curiosity as the first step towards intrinsically motivated human-like agents. Recent studies (Marvin and Shohamy 2016; Burda et al. 2018; Pathak et al. 2017; de Abril and Kanai 2018) about curiosity suggest that the gap between the expected reward and the received reward, i.e., the "information prediction error" or "information gap" determines curiosity. The definition of the "information gap" comes from Loewenstein (Loewenstein 1994). He stated that the information gap refers to the gap between "information I want to know" and "information I already know." People are more likely to be curious about topics where there is a large information gap.

The combination of reinforcement learning and curiosity has made good progress in recent years. Some of these have enabled agents to pass games without relying on extrinsic rewards (Pathak et al. 2017).

Curiosity with U-Shaped Curve

Loewenstein also proposed an idea in his theory that differs from the above definition. According to Loewenstein, people who acquire knowledge in a particular area are likely to become increasingly curious about the subjects which they know the most. In this case, people will not be interested in topics where there is a large information gap.

Relatedly, Dubey et al. (Dubey and Griffiths 2017) investigated the relationship between curiosity and information gap (i.e., confidence, see next section for details) and suggested that it can be represented as an inverted U-shaped curve, meaning that curiosity increases with the information gap at the beginning and then decreases when the gap is larger than a "threshold." We decided to combine the information gap theory and the theory of Dubey et al. to design our reward function because it details a recent theory of curiosity based on cognitive studies that has not yet been implemented as a game agent.

Methods

Reward Function

Intrinsic Curiosity Module Our "Cautious Curiosity" (CC) agent is composed of two parts: an intrinsic reward generator that outputs the curiosity-driven intrinsic reward and an extrinsic reward receiver that receives an extrinsic reward from the environment. The policy of our agent is to perform a series of actions that can maximize the sum of intrinsic and extrinsic rewards. First of all, let the intrinsic curiosity reward generated by the agent at time t be r_t^i and the extrinsic reward be r_t^e. The sum of these two rewards can be represented as $r_t = r_t^i + r_t^e$.

We use the Intrinsic Curiosity Module (ICM) (Pathak et al. 2017) to build the agent and use the Asynchronous Advantage Actor-Critic policy gradient (Mnih et al. 2016) for policy training. The policy is represented as $\pi(S_t;\theta_p)$ with a parameter θ_p. At time t, we have state S_t and an action a_t generated by policy $a_t \sim \pi(S_t;\theta_p)$. θ_p is optimized to get the max expected sum rewards.

$$\max_{\theta_p} E_{\pi(S_t;\theta_p)}[\textstyle\sum_t r_t] \tag{1}$$

To get a good intrinsic reward from the prediction error in the learned feature space, our deep neural network can be separated into two modules. The first module will encode state S_t into a feature vector $\phi(S_t)$ and the second module takes two consequent state $\phi(S_t)$ and $\phi(S_{t+1})$ then predict the action a_t. The function g of the neural network can be defined as

$$\widehat{a} = g(s_t, s_{t+1}; \theta_I) \tag{2}$$

where $\widehat{a}$ is the predicted estimate of the action a_t. θ_I is a parameter of our deep neural network that is trained to optimize. So we have:

$$\min_{\theta_I} L_I(\widehat{a}_t, a_t) \tag{3}$$

where L_I is the lost function. It shows the difference between the actual action and the predicted action. Following the action prediction, we have another neural network to predict the feature encoding of the state at time $t + 1$

$$\widehat{\phi}(S_{t+1}) = f\left(\widehat{\phi}(S_t), a_t; \theta_F\right) \quad (4)$$

where $\widehat{\phi}(S_{t+1})$ is the estimate state of $\phi(S_{t+1})$. Parameters θ_F are optimized by minimizing the lost function L:

$$L_F(\phi(S_t), \widehat{\phi}(S_{t+1})) = \frac{1}{2}||\widehat{\phi}(S_{t+1}) - \phi(S_{t+1})||_2^2 \quad (5)$$

Finally, the prediction error P can be represented as:

$$P = ||\widehat{\phi}(S_{t+1}) - \phi(S_{t+1})||_2^2 \quad (6)$$

From 1 to 6, we are using exactly the same equations presented in Pathak et al. (Pathak et al. 2017). Compared to the intrinsic reward function of Pathak et al., which directly links the size of the intrinsic reward to the size of the prediction error, we used a different way to measure the intrinsic reward.

Information Gap and Curiosity Reward From the psychological literature, especially Loewenstein's paper (Loewenstein 1994), we know that people have a strong interest in subjects with large information gaps. The bigger the gap, the bigger the curiosity. However, at the same time, many experiments have shown that people are more curious about subjects with smaller information gaps. Therefore, based on the above viewpoints, we believe that people will lose interest in subjects that they are not familiar with (i.e., subjects with too large information gaps). Curiosity increases with the information gap when things are within a "range." When subjects go beyond this "range," curiosity will decrease due to the increase of the information gap. A recent study about curiosity supports this. Dubey et al. (Dubey and Griffiths 2017) showed that the relationship between curiosity and confidence can be shown as an inverted U-shaped curve. The graph is shown in Figure 1.

As we can see from Figure 1, curiosity increases with confidence at the very beginning and decreases after it reaches a point. That is, when people are very confident or not confident in a subject, they are usually not interested in it. Confidence or not represents the degree of understanding of a subject. This is consistent with the definition of the information gap. If the information gap is large, it means that people do not understand a certain subject.

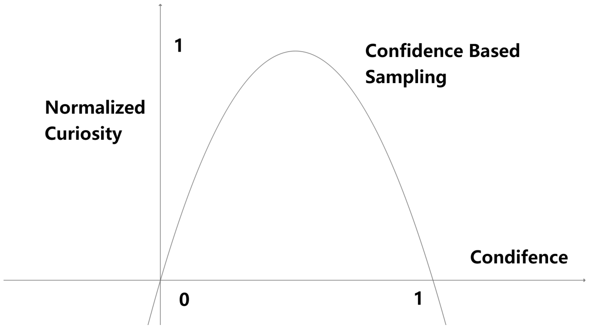

Figure 1: Relationship between confidence (i.e., information gap) and curiosity based on Dubey et al. (Dubey and Griffiths 2017).

Important to note is that different people can have different understandings of the same subject, as a result of different life experiences. Therefore, the "range" mentioned above is different from person to person. In the game context, we call this "range" "threshold," denoted by ψ. In the beginning, the agent's understanding of the world is only the initial observation. So we have:

$$\psi_0 = \phi_0 \quad (7)$$

As the agent continues to move forward, there are constantly new observations. The new "knowledge" broadens its perception. Here, "knowledge," denoted by K, is expressed as the difference between the new observation and the old observation. We have

$$K = ||\phi(S_t) - \phi(S_{t-1})||_2^2 \quad (8)$$

The "threshold" is constantly updated as the agent progresses, and as the knowledge gained continues to grow, the thresholds will keep getting bigger.

$$\psi_t = \psi_{t-1} + K \quad (9)$$

When the prediction error is smaller than the threshold ψ_t, the reward will be:

$$r_t^i = \eta * ||\widehat{\phi}(S_{t+1}) - \phi(S_{t+1})||_2^2 \quad (10)$$

where η is a scaling factor larger than 0. When the prediction is larger than the threshold, we have:

$$r_t^i = -\eta * ||\widehat{\phi}(S_{t+1}) - \phi(S_{t+1})||_2^2 + 2 * \psi_t \quad (11)$$

The image of the reward function is shown in Figure 2. It can be seen that when the prediction error is less than the threshold, the reward increases with the increase of the error. When the prediction error is greater than the threshold, the reward decreases as the error increases. Finally, we need to jointly optimize (1), (3), and (5). Together, we have:

$$\min_{\theta_I, \theta_P, \theta_F} [-\lambda E_{\pi(S_t;\theta_p)}[\textstyle\sum_t r_t] + (1-\beta)L_I + \beta L_F] \quad (12)$$

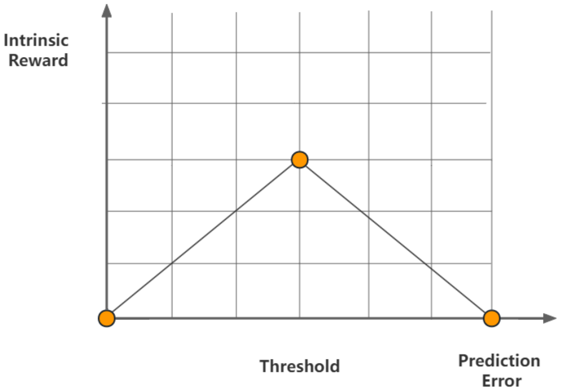

Figure 2: Our inverse V-shaped intrinsic reward function.

where $0 \leq \beta \leq 1$ is a scalar that weighs the inverse model loss against the forward model loss and $\lambda > 0$ is a scalar that weighs the importance of the policy gradient loss against the importance of learning the intrinsic reward signal.

It is important to note that Dubey et al.'s work does not explain the mathematical relationship between curiosity and the information gap because their conclusion was derived through a questionnaire survey. In this case, we cannot directly translate the conclusions of Dubey et al.'s work into a reward function. Therefore, the resulting graph of the proposed reward function exhibits an inverted V shape that can be used to approximate the inverted U-shaped curve.

User Study

To measure the degree of human-likeness, we compared our method to other methods and real humans in a Turing-Test-like user study. We evaluated our method using Super Mario Bros. We selected relevant state-of-the-art methods as our algorithmic benchmarking, in addition to game videos of real humans as human benchmarking.

Algorithmic Benchmark We chose Intrinsic Curiosity Module (ICM) to be the (algorithmic) benchmark. This work not only successfully combines curiosity with reinforcement learning, it also demonstrates that curiosity-driven agents can explore environments efficiently without extrinsic rewards or with sparse extrinsic rewards.

Baselines We first chose to use a random agent which can only choose an action randomly as our baseline. In addition, the ICM mentioned above is based on A3C (Mnih et al. 2016), which is a deep reinforcement learning algorithm. As such, we considered that including A3C would also be meaningful as a baseline.

Human Benchmark We opted to include three average players who are familiar with the game, but who are not master players, as the human benchmark.

Game Video Our team implemented the "Cautious Curiosity" (CC) agent based on the newly designed reward function and let it run in Super Mario Bros. At the same time, we also implemented a random agent, an A3C agent, and an ICM agent, which we used to compare the human-likeness level with the CC agent. We also recruited three average human players to compare with. We recorded three one-minute game videos of each agent (CC, random, A3C, and human player), giving us a total of 15 one-minute videos.

Participants We published our survey on Amazon Mturk. We gave each participant $3 for finishing the survey. In total, 100 people participated in the experiment.

Study Procedure The study is a web-based survey where participants had to watch 15 videos. For each video, the following questions were asked:

1. Please watch this video then rank the human-likeness of the player. (From 0 to 10, 0 means the player is not a human at all and 10 means the player is a real human.)

2. State briefly (in one or two sentences) why you think the player above is a human or an AI player.

Experiment Setup

Training Details All agents in this work are trained with visual inputs that are preprocessed similarly to ICM's work (Pathak et al. 2017). The RGB input images for the agents, including A3C, ICM, and CC, are converted to grayscale and reformatted to 84×84. All agents were trained for 50 episodes in Super Mario Bros.'s level 1-1, with each episode representing a complete game trajectory. We considered the models from the last training episode as the experimental result shown in this paper.

A3C Agent The input state is passed through a sequence of four convolution layers with 32 filters each, kernel size of 3x3, stride of 2, and padding of 1. An exponential linear unit (ELU) (Clevert, Unterthiner, and Hochreiter 2015) is used after each convolution layer. The output of the last convolution layer is fed into an LSTM with 512 units. Two separated fully connected layers are used to predict the value function and the action from the LSTM feature representation.

A3C Architecture for ICM and CC The input state is passed through a sequence of four convolution layers with 32 filters each, kernel size of 3x3, stride of 2, and padding of 1. An exponential linear unit (Clevert, Unterthiner, and Hochreiter 2015) is used after each convolution layer. The output of the last convolution layer is fed into a GRU with 512 units. Two separated fully connected layers are used to predict the value function and the action from the GRU feature representation.

ICM Agent and CC Agent Due to the fact that the CC agent is based on the ICM agent and the only difference between them is the reward function, they have the same architecture. Both of them contain an intrinsic curiosity module that consists of two parts. The first part is an inverse model that maps an input state s_t into a feature vector $\phi(s_t)$. The inverse model is made of four convolution layers, each with 32 filters, kernel size 3x3, stride of 2, and padding of 1. An ELU is used after each layer. After that, the inverse model will concatenate $\phi(s_t)$ and $\phi(s_{t+1})$ into a single feature vector. Then, this vector is passed as input into a fully connected layer of 256 units followed by an output fully connected layer with 4 units to predict a possible action from the action space. The second part is the forward model, which is used to concatenate $\phi(s_t)$ with a_t and then passed to a sequence of two fully connected layers with 256 and 288 units, respectively. The output of the forward model is the estimation of the next state $\widehat{\phi}(S_{t+1})$. Finally, output from the inverse model and forward model were used to compute the intrinsic reward.

Results

Benchmarking

We evaluated the performance of the learned policy with the proposed intrinsic curiosity reward function in Super Mario Bros. Our experiments with Mario were trained with the curiosity reward and dense extrinsic reward. We also trained the ICM agent with the same extrinsic reward setting. We repeatedly ran the CC agent and the ICM agent in the level 1-1 environment in Super Mario Bros. and compared the performance of the two agents. First, the CC agent prefers

Name	Pass the game	Time used, in M(SD)	Mistakes, in M (SD)
CC	Yes	65 (2.82)	4 (0)
ICM	Yes	58.3 (0.94)	1 (0.81)
A3C	Yes	60 (0)	1.33 (0.47)
Human	Yes	96 (0.94)	3.67 (0.47)
Random	No	400 (= time limit)	

Table 1: Average time used and average mistakes made by each player.

to stay on the ground compared to the ICM agent. The ICM agent often jumps on the rocks and moves on them, while the CC agent prefers to move on the ground, even if the probability of hitting a Goomba is higher. We believe that the reason for this phenomenon is that the reward function of the CC agent makes it "cautious," hence we named the agent "Cautious Curiosity." Because of the reward function, the agent is more inclined to explore states that are familiar to it. The area above the bricks is relatively unfamiliar to it, so it prefers to stay on the ground. Second, compared to the ICM agent, the CC agent makes more "mistakes." The CC agent often jumps and bumps into obstacles such as water pipes or bricks, while the ICM agent can easily avoid most of them. This also results in the ICM agent reaching the target faster. After observing that the average pass time of the ICM agent is shorter than that of the CC agent, we decided to compare the average pass time of all agents (average human, CC agent, A3C agent, ICM agent, random agent) participating in this experiment. The results are shown in Table 1.

After the human player, the CC agent finished the game in the longest time. ICM is the best among all the players, as it hardly makes any mistakes and can play through the game in the shortest time. It is worth noting that the A3C agent completed the game at a similar speed to the ICM agent and their average mistakes are close. The CC agent, although based on ICM, made many more mistakes than the A3C agent and the ICM agent. Based on this data, we can conclude that our reward function makes the behavior of the CC agent different from that of the ICM agent.

Video Ratings

To verify that our algorithms perform in a more human-like manner than the ICM algorithm, we performed a user study. With 100 participants we received 1,500 data points in total (i.e., 15 videos rated by every participant). Table 2 shows the result of our study.

	Video 1	Video 2	Video 3
CC	7 (5)	7 (6)	7 (5)
ICM	5 (6)	6 (6)	4.5 (6)
A3C	5 (6.25)	6 (6)	7 (5)
Human	7 (5.5)	7 (5)	7 (5)
Random	3 (8)	4 (8)	3 (8)

Table 2: Median of ratings (0 to 10), in *Mdn* (*IQR*)

	A3C	Human	ICM	Random	CC
A3C	1	0.025	0.18	1.9e-5	0.16
Human	0.025	1	0.002	2.9e-10	0.16
ICM	0.18	0.002	1	7.1e-4	0.03
Random	1.9e-5	2.9e-10	7.1e-4	1	1.05e-7
CC	0.16	0.16	0.03	1.05e-7	1

Table 3: Details of the post hoc dunn test

Each cell shows the median score of the video indicated by the participants. From this table, it can be seen that the human has the highest median score and the random agent has the lowest median score. Except for the human and the random agent, the CC agent received the same median score as the human agent. However, the A3C agent scored higher than the ICM agent. This is contrary to our expectations. Although the difference between the two scores is not large, it shows that the vast majority of participants think that the A3C agent has more human characteristics than the ICM agent.

As for the significance of the data, we first tested the normality assumption. The result showed that the data were not normally distributed. Therefore, we decided to use the Kruskal-Wallis test instead of ANOVA. The Kruskal-Wallis test showed that there was a difference between the score of each agent, $H(4) = 50.29$, $p < .001$. Post-hoc Dunn tests with a two-stage FDR adjustment were used to compare all groups. The difference between the CC agent and Human was not significant, $p = .15$. However, other comparisons showed significant results after two-stage FDR adjustment ($p < .01$). The details of the post-hoc Dunn test and boxplots can be found in Table 3. To test the effect size, we used the epsilon squared test. The result gave us the effect size = .0335, which corresponds to a "weak" effect.

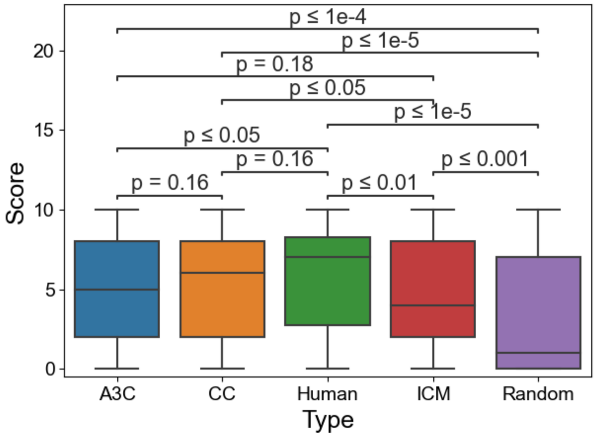

Figure 3: Boxplots of the user study data.

Keyword	Number of occurrence (133 in total)	Distribution $(CC \mid ICM \mid Human \mid A3C \mid Random)$
Mistakes	59 (44%)	$19 \mid 11 \mid 16 \mid 13 \mid 0$
Normal speed and reaction	41 (31%)	$9 \mid 11 \mid 15 \mid 5 \mid 1$
Poor skill	14 (11%)	$1 \mid 5 \mid 2 \mid 1 \mid 5$
Hesitation	8 (6%)	$4 \mid 3 \mid 0 \mid 1 \mid 0$
Others	11 (8%)	$4 \mid 4 \mid 3 \mid 0 \mid 0$

*Others involve overaggressive, unpredictable, and different game patterns.

Table 4: The frequency of top qualitative keywords that identify that the player is a Human.

Keyword	Number of occurrence (341 in total)	Distribution $(CC \mid ICM \mid Human \mid A3C \mid Random)$
Fast speed and reaction	127 (37%)	$26 \mid 40 \mid 18 \mid 42 \mid 1$
Useless Movement	105 (31%)	$10 \mid 9 \mid 1 \mid 2 \mid 83$
Perfect skill	50 (15%)	$5 \mid 14 \mid 13 \mid 18 \mid 0$
Ignore game content	39 (11%)	$9 \mid 8 \mid 9 \mid 13 \mid 0$
Jitterness	12 (4%)	$0 \mid 0 \mid 0 \mid 0 \mid 12$
Others	8 (2%)	$1 \mid 2 \mid 3 \mid 2 \mid 0$

*Others involve constant speed, purpose and inconsistent skill.

Table 5: The frequency of top qualitative keywords that identify that the player is an AI.

Open Responses

We also analyzed the participants' open responses. We first discarded answers that were not useful such as "this is human" or "I don't know why but I think it is a robot," and then considered the 474 useful responses to find an answer as to why participants believe the player is a human or not. Through qualitative analysis, keywords were identified and coded in the responses. Table 4 and Table 5 show the frequency of the top qualitative keywords for identifying a player as human or AI, respectively.

From the tables, we can see that participants considered "mistakes" as an important criterion for judging whether a player is human. In Super Mario Bros., if a player makes frequent mistakes, such as falling into a pit or hitting a brick, participants perceive the player as a human being. On the other hand, participants believed that human players would not have extremely fast reactions and speed. Thus, if the player played the game at a slower speed and had a slow reaction, participants thought the player was a human. In the previous sections, we found that the CC agent made more mistakes than the ICM agent. In addition, the average completion time of the CC agent is longer. Both aspects (i.e., more mistakes and longer completion time) may explain the higher ratings of the CC agent. It also explains why the rating of the ICM agent is lower than the A3C agent.

Aside from fast reactions and speed, we can see from Table 5 that too many useless moves make participants think the player is an AI. However, most of these responses come from the responses to random agents. Since the actions of random agents are randomly selected from all valid actions, useless movement is expected. Nevertheless, some participants judged that the player is an AI based on the jump action of the ICM agent and the CC agent at the beginning of the game. The jump action is useless because Mario stands on the ground at the beginning of the game.

Interestingly, many participants considered the player's interaction with the game environment as a basis for judgment. About 40 responses pointed out that collecting gold coins, collecting mushrooms, and killing enemies are actions that human players take, but since the player in the video only focuses on completing the game, the player is likely to be an AI. Therefore, future research should focus on agents that actively interact with the game environment.

Discussion

Speed, Reaction, and Skill

To our surprise, apart from the random agent, the ICM agent has the lowest score in the survey. From the open response analysis, we know that "speed and reaction" is an indicator for participants to judge whether the agent is human. If an agent completes the game too fast, the participants are likely to think it is AI. Compared to CC and A3C, the ICM agent can complete the game in the shortest time, which means the ICM agent is more likely to be considered an AI. Also, many participants will think that the agent is AI because of its "perfect skill" characteristics. There were 14 participants who thought the ICM agent is a perfect player while only 5 participants thought the CC agent played the game perfectly. We believe that the curiosity reward greatly improves the performance of the ICM agent. The ICM agent moves seemingly pre-calculated and completes the game with an inhuman speed. In contrast, our reward function lowers the performance, as demonstrated by the results of the CC agent.

Mistakes and Useless Movement

It seems strange to discuss "mistakes" and "useless movement" together. However, if we look closely at the random agent, it makes mistakes too often. Participants perceive these actions as "useless movements." On the other hand, participants will not consider human mistakes as useless movements. The reason why this happens is: "mistakes" means the agent is trying to do something useful like jumping through a pit or kill an enemy, but does not succeed. "Useless movements" then means the agent is doing something that will not bring any gains, involve any risks, or result in consequences of any nature to the agent or the environment. If we look at the CC agent and the ICM agent, both agents make some useless movements. For example, both the CC agent and the ICM agent like jumping at the beginning without any purpose. Therefore, a good way to improve human-likeness is to decrease the frequency of making useless movements. Humans making mistakes when playing is a natural behavior. Thus, another way to improve human-likeness is to have agents mimic such behavior, i.e., have them make mistakes.

Game Content

We got 39 responses about "ignore game content" and 8 responses about "hesitation." The reason we discuss the two together is that "hesitation" means the agent is confused by the game content around it and does not know what to do next, while "ignore game content" means the agent has a purpose and skips game content. Due to the nature of reinforcement learning, agents have formed policies that maximize rewards, and as such, they will only focus on completing the game rather than interacting with the environment. Human-like hesitations, such as people hesitating whether to kill the enemy or eat the mushrooms first, were not seen in this study. Although we could make the agent actively interact with the environment by modulating the extrinsic reward (for example, increasing the reward for interacting with mushrooms), this is a departure from our original intention. Our research focused on making agents more human-like by designing a new intrinsic reward function. However, future work should consider how to make the agent interact with the environment to improve human-likeness.

AI Ethics

Despite the automation we propose, the AI method does not aim to replace human testers. The main idea is to provide resources for teams who cannot afford human testers and/or to save human testers from the burden of repetitive tasks. Human testers act as consumers of the information the agent produces. Based on the produced information, human testers can decide to test parts by themselves that they feel the AI did not cover correctly, or request the AI to focus on particular areas of interest. As such, this method facilitates human-AI collaboration with the objective to gain better outcomes than a human or AI can achieve on their own (Politowski, Petrillo, and Guéhéneuc 2021).

Limitations

First, our research was limited due to the lack of support from the community. At the very beginning, we planned to use the winner of the Mario AI competition as the benchmark (Shaker et al. 2013). However, the competition was discontinued after 2012, and the winner did not keep the code of their agent. We tried to find other state-of-the-art papers for benchmarking, but these papers either used a different version of Super Mario Bros., or their authors only gave us limited instructions on their implementation. Therefore, we could not find a suitable algorithmic benchmark of a human-like agent from existing work with Super Mario Bros.

Second, we tested our agent in other levels such as 1-2, 1-3, and 1-4. The experiment showed that our agent does not avoid obstacles or enemies, and dies at the beginning of the game. Although it failed to generalize to other levels, our work demonstrates how people feel about a curiosity-based human-like agent and what criteria people use to determine whether an agent is human or not. Based on the outcomes of this work, we can further redesign our reward function to improve both its generalization and its human-likeness.

We did not test our agent as a debugging assistant because our primary goal was to focus on assessing the human-likeness. Future work may consider such use and be inspired by leveraging metrics (Machado et al. 2018b) for this kind of assistance.

Conclusion

Reinforcement learning is currently one of the most successful algorithms in games. It has achieved impressive results in the GVGAI competition in recent years (Perez-Liebana et al. 2016). Curiosity is considered an important emotion for people who are playing games like Super Mario Bros. As such, attempts to combine curiosity with reinforcement learning started decades ago for developing so-called "curiosity agents." We combined both approaches to design a novel intrinsic reward to create the "Cautious Curiosity" (CC) agent, an agent that attempts to behave more human-like. We based the reward function and the resulting CC agent on (1) a psychological theory that suggests curiosity is formed by the information gap between "what people know" and "what people want to know" and (2) a recent study that showed that the relationship between curiosity and the information gap is an inverted U-shape curve. Based on comparing the CC agent to other methods (random, A3C, ICM, and average human player) in a Super Mario Bros. environment with human judges (n = 100), we find that our agent is the only agent who received similar ratings compared to the average human player. Although much further scrutiny is required (including generalization to other Super Mario Bros. levels and other games), a key contribution is that we advanced the area of human-like agents based on the use of psychological theory. We demonstrate promising results that with the proposed reward function, an agent can play a computer game similarly to a human, while maintaining its competitiveness and being more believable compared to other agents.

References

Ariyurek, S.; Betin-Can, A.; and Surer, E. 2019. Automated video game testing using synthetic and humanlike agents. *IEEE Transactions on Games*, 13(1): 50–67.

Arulkumaran, K.; Cully, A.; and Togelius, J. 2019. Alphastar: An evolutionary computation perspective. In *Proceedings of the genetic and evolutionary computation conference companion*, 314–315.

Aytar, Y.; Pfaff, T.; Budden, D.; Paine, T.; Wang, Z.; and De Freitas, N. 2018. Playing hard exploration games by watching youtube. *Advances in neural information processing systems*, 31.

Aytemiz, B.; Shu, X.; Hu, E.; and Smith, A. 2020. Your buddy, the grandmaster: Repurposing the game-playing ai surplus for inclusivity. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 16, 17–23.

Blog, G. A. 2021. Quickly Training Game-Playing Agents with Data-Efficient Reinforcement Learning. https://ai.googleblog.com/2021/06/quickly-training-game-playing-agents.html. [Online; accessed 21-February-2023].

Briziarelli, M. 2016. Invisible play and invisible game: Video game testers or the unsung heroes of knowledge working. *tripleC: Communication, Capitalism & Critique. Open Access Journal for a Global Sustainable Information Society*, 14(1): 249–259.

Burda, Y.; Edwards, H.; Pathak, D.; Storkey, A.; Darrell, T.; and Efros, A. A. 2018. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*.

Callele, D.; Neufeld, E.; and Schneider, K. 2006. Emotional requirements in video games. In *14th IEEE International Requirements Engineering Conference (RE'06)*, 299–302. IEEE.

Chen, J. X. 2016. The evolution of computing: AlphaGo. *Computing in Science & Engineering*, 18(4): 4–7.

Clevert, D.-A.; Unterthiner, T.; and Hochreiter, S. 2015. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

de Abril, I. M.; and Kanai, R. 2018. A unified strategy for implementing curiosity and empowerment driven reinforcement learning. *arXiv preprint arXiv:1806.06505*.

de Mesentier Silva, F.; Lee, S.; Togelius, J.; and Nealen, A. 2017a. AI-based playtesting of contemporary board games. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, 1–10.

de Mesentier Silva, F.; Lee, S.; Togelius, J.; and Nealen, A. 2017b. AI-Based Playtesting of Contemporary Board Games. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, FDG '17. New York, NY, USA: Association for Computing Machinery. ISBN 9781450353199.

Devlin, S.; Anspoka, A.; Sephton, N.; Cowling, P.; and Rollason, J. 2016. Combining gameplay data with monte carlo tree search to emulate human play. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 12, 16–22.

Dossa, R. F. J.; Lian, X.; Nomoto, H.; Matsubara, T.; and Uehara, K. 2019. A human-like agent based on a hybrid of reinforcement and imitation learning. In *2019 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Dubey, R.; and Griffiths, T. L. 2017. A rational analysis of curiosity. *arXiv preprint arXiv:1705.04351*.

Gudmundsson, S. F.; Eisen, P.; Poromaa, E.; Nodet, A.; Purmonen, S.; Kozakowski, B.; Meurling, R.; and Cao, L. 2018. Human-like playtesting with deep learning. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, 1–8. IEEE.

Hasegawa, K.; Tanaka, N.; Emoto, R.; Sugihara, Y.; Ngonphachanh, A.; Ichino, J.; and Hashiyama, T. 2013. Action selection for game play agents using genetic algorithms in platform game computational intelligence competitions. *Journal of Advanced Computational Intelligence Vol*, 17(2).

Hosu, I.-A.; and Rebedea, T. 2016. Playing atari games with deep reinforcement learning and human checkpoint replay. *arXiv preprint arXiv:1607.05077*.

Khalifa, A.; Isaksen, A.; Togelius, J.; and Nealen, A. 2016. Modifying MCTS for Human-like General Video Game Playing. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, 2514–2520. AAAI Press. ISBN 9781577357704.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Loewenstein, G. 1994. The psychology of curiosity: A review and reinterpretation. *Psychological bulletin*, 116(1): 75.

Machado, T.; Gopstein, D.; Nealen, A.; Nov, O.; and Togelius, J. 2018a. AI-Assisted Game Debugging with Cicero. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, 1–8.

Machado, T.; Gopstein, D.; Nealen, A.; Nov, O.; and Togelius, J. 2018b. AI-Assisted Game Debugging with Cicero. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, 1–8.

Malone, T. W. 1981. Toward a theory of intrinsically motivating instruction. *Cognitive science*, 5(4): 333–369.

Marvin, C. B.; and Shohamy, D. 2016. Curiosity and reward: Valence predicts choice and information prediction errors enhance learning. *Journal of Experimental Psychology: General*, 145(3): 266.

McAllester, D. A. 1988. Conspiracy numbers for min-max search. *Artificial Intelligence*, 35(3): 287–310.

Milani, S.; Juliani, A.; Momennejad, I.; Georgescu, R.; Rzpecki, J.; Shaw, A.; Costello, G.; Fang, F.; Devlin, S.; and Hofmann, K. 2023. Navigates Like Me: Understanding How People Evaluate Human-Like AI in Video Games. *arXiv preprint arXiv:2303.02160*.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, 1928–1937. PMLR.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

Mohan, S.; and Laird, J. E. 2009. Learning to play Mario. *Tech. Rep. CCA-TR-2009-03*.

Pathak, D.; Agrawal, P.; Efros, A. A.; and Darrell, T. 2017. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, 2778–2787. PMLR.

Pedersen, C.; Togelius, J.; and Yannakakis, G. N. 2009. Modeling player experience in super mario bros. In *2009 IEEE Symposium on Computational Intelligence and Games*, 132–139. IEEE.

Perez-Liebana, D.; Samothrakis, S.; Togelius, J.; Schaul, T.; and Lucas, S. M. 2016. General video game ai: Competition, challenges and opportunities. In *Thirtieth AAAI conference on artificial intelligence*.

Poels, Y.; Annema, J. H.; Verstraete, M.; Zaman, B.; and De Grooff, D. 2012. Are you a gamer? A qualititive study on the parameters for categorizing casual and hardcore gamers. *Iadis International Journal on www/internet*, (1): 1–16.

Politowski, C.; Petrillo, F.; and Guéhéneuc, Y.-G. 2021. A survey of video game testing. In *2021 IEEE/ACM International Conference on Automation of Software Test (AST)*, 90–99. IEEE.

Sestini, A.; Bergdahl, J.; Tollmar, K.; Bagdanov, A. D.; and Gisslén, L. 2022. Towards Informed Design and Validation Assistance in Computer Games Using Imitation Learning. *arXiv preprint arXiv:2208.07811*.

Shaker, N.; Shaker, M.; and Togelius, J. 2013. Ropossum: An authoring tool for designing, optimizing and solving cut the rope levels. In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Shaker, N.; Togelius, J.; Yannakakis, G. N.; Poovanna, L.; Ethiraj, V. S.; Johansson, S. J.; Reynolds, R. G.; Heether, L. K.; Schumann, T.; and Gallagher, M. 2013. The turing test track of the 2012 mario ai championship: entries and evaluation. In *2013 IEEE Conference on Computational Inteligence in Games (CIG)*, 1–8. IEEE.

Shao, K.; Tang, Z.; Zhu, Y.; Li, N.; and Zhao, D. 2019. A survey of deep reinforcement learning in video games. *arXiv preprint arXiv:1912.10944*.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. 2017. Mastering the game of go without human knowledge. *nature*, 550(7676): 354–359.

Smith, G.; Whitehead, J.; and Mateas, M. 2010. Tanagra: A mixed-initiative level design tool. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, 209–216.

Togelius, J.; De Nardi, R.; and Lucas, S. M. 2006. *Making racing fun through player modeling and track evolution*. SAB'06 Workshop on Adaptive Approaches for Optimizing Player Satisfaction in Computer and Physical Games.

Togelius, J.; Yannakakis, G. N.; Stanley, K. O.; and Browne, C. 2011. Search-Based Procedural Content Generation: A Taxonomy and Survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3): 172–186.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354.

Yannakakis, G. N.; and Togelius, J. 2018. *Artificial intelligence and games*, volume 2. Springer.

Zhao, Y.; Borovikov, I.; de Mesentier Silva, F.; Beirami, A.; Rupert, J.; Somers, C.; Harder, J.; Kolen, J.; Pinto, J.; Pourabolghasem, R.; et al. 2020. Winning is not everything: Enhancing game development with intelligent agents. *IEEE Transactions on Games*, 12(2): 199–212.