Evolving Models of Player Decision Making: Personas versus Clones

Christoffer Holmgård[a,*], Antonios Liapis[b], Julian Togelius[c], Georgios N. Yannakakis[b]

[a]*Center for Computer Game Research, IT University of Copenhagen, Denmark*
[b]*Institute of Digital Games, University of Malta, Malta*
[c]*Game Innovation Lab, New York University, New York, United States of America*

Abstract

The current paper investigates multiple approaches to modeling human decision making styles for procedural play-testing. Building on decision and persona theory we evolve game playing agents representing human decision making styles. Three kinds of agents are evolved from the same representation: procedural personas, evolved from game designer expert knowledge, clones, evolved from observations of human play and aimed at general behavioral replication, and specialized agents, also evolved from observation, but aimed at determining the maximal behavioral replication ability of the representation. These three methods are then compared on their ability to represent individual human decision makers. Comparisons are conducted using three different proposed metrics that address the problem of matching decisions at the action, tactical, and strategic levels. Results indicate that a small gallery of personas evolved from designer intuitions can capture human decision making styles equally well as clones evolved from human play-traces for the testbed game *MiniDungeons*.

Keywords: decision making, procedural content generation, evolutionary computation, player modeling

*Corresponding author
Email address: `holmgard@itu.dk` (Christoffer Holmgård)
URL: `christoffer.holmgard.org` (Christoffer Holmgård)

1. Introduction

This paper investigates how to create models of human decision making styles in games using generative, game-playing agents for procedural play-testing. It proposes an evolution based framework for representing player decision making in games and a simulation based method for evaluating human likeness of game playing agents at three different levels. The framework is applied in two different ways: evolving in a top-down manner from designer-driven intuitions and evolving in a bottom-up, data-driven manner from play-traces. The evaluation method is then used on both applications of the framework to evaluate their performances. Finally, a possibility for combining the two applications of the framework, allowing for hybrid top-down/bottom-up decision modeling through generative agents is suggested.

Generative, game-playing agents that represent and replicate human decision making may be useful in games for many purposes e.g. as believable stand-ins for human players or as benchmark rivals for players to surpass.

This paper focuses on using game playing agents representing human decision making styles as stand-in players, supporting the traditional process of human play-testing.

Play-testing is typically an integral part of game development [1]. The complexity and cost of the play-testing depends on the kind of game under development, the stage in the development process, and the objectives of the play-testing. At one extreme play-testing may be conducted by the game designer herself by simply imagining how players might interact with the game, a feature or a piece content. At the other extreme play-testing may be conducted under highly instrumented laboratory conditions or at a massive scale in the wild by telemetrically collecting data from players after the launch of the game [2].

In this paper, we suggest there may be an opportunity for methods using generative agents to support designers in situations where new content is being developed, but access to human play-testers is limited or impossible. For example, when a level designer is implementing a new level for a game or making changes to an existing one, these changes might not be large enough to mandate a full play-test with human players. Still, it might be useful for the level designer to observe how different kinds of players would interact with the level. In situations like these, generative game playing agents based on models of human decision making might provide designers with surrogate play-traces to inform their design process and explore what parts of the game space players are likely to interact with and how, effectively delivering *procedural play-testing*.

When agents sufficiently simulate a particular archetypal human decision making style we call agents *procedural personas*. Integrated with content creation tools, we envision that procedural personas will allow for mixed-initiative game design tools [3] that yield immediate feedback during the design process, even if this feedback is not a completely accurate representation of how human players might play the game. Additionally, play-traces from procedural personas can be used as input for procedural content generation systems shaping the output in response to the generative player models [4]. In other words, agents that play like humans can help understand content by playing it as it is being created.

1.1. Research Questions

An important question then arises regarding which sources of information about player decision making styles are useful for constructing believable, accurate procedural personas. Do we need some amount of low level behavioral data from actual players or can we derive the same information from the expert knowledge of a game designer?

A second question is how general we can make the resulting models. Can they perform consistently on unseen content from which no human play-traces were sampled or which the game designer was not explicitly considering?

The work presented here addresses these questions by comparing two methods for realizing procedural personas in order to evaluate which method produces the best models for generating synthetic play-test data. One draws on designer expert knowledge and the other uses empirically gathered play test data.

1.2. Prior Work

In previous work we have designed a simple turn-based, tile-based dungeon crawling game, *MiniDungeons*, which features monsters, treasures and potions in mazes [5]. 38 players played 10 levels of this game and we recorded their every action. Next, we analyzed the design of the game to extract a number of possible affordances which we translated into partially conflicting objectives that a player might seek to fulfill (e.g. kill all monsters, avoid danger or get to the exit quickly). Using these *affordances* we trained agents to play the game rationally for each objective. Both Q-learning [5] and evolutionary algorithms [6] were used to train high-performing agents; the evolved agents have the benefit that they generalize to levels they were not trained on, in contrast to Q-learning agents which where unable to perform on levels they previously had not seen.

1.3. Metrics and Methods for Comparing Agents to Human Players

In previous work on *MiniDungeons* [5, 6], the agents' behaviors were compared to play-traces of the human players through a metric we call the *action agreement ratio* (AAR) which compares agents and humans at the action level — asking at each step of the human playthrough whether the agent would pursue the same *next action* as the player. But is this really the right level of analysis for comparing players to agents? It could be argued that the microscopic level of comparing actions gives a biased view of how well an agent's behavior reproduces player behavior; it may be more interesting to compare behaviors not on the level of atomic decisions, but rather at the level of tactical or strategic decisions. Further, are we right to assume that players exhibit boundedly rational behavior given some set of objectives? It might be that with the same agent representation, we could train agents that reproduce player behavior better by using the actual play-traces as training data instead of focusing on player objectives. The current paper tries to answer these two questions.

Expanding on previous work [7], we propose two new play-trace comparison methods, *tactical agreement ratio* (TAR) and *strategic agreement ratio* (SAR). While AAR considers whether an agent would perform the same singular action as the player in a given state, TAR and SAR consider whether it would choose to pursue the same *next affordance* or the same *overall outcome*, respectively.

We also train a second class of agents to behave as similarly as possible to human players on *unseen* levels using play-traces as objectives, again evaluated on the three levels of comparison: the action level, the tactical level, and the strategic level. We call such agents *clones*.

Finally, we train a third class of agents to behave as similarly as possible to human players on *previously seen* levels in order to explore the maximal performance of our chosen representation. We call such agents *specialized agents* as they are likely to be the closest fit of the representation to an individual play-trace, but are trained for just one particular level.

1.4. Modeling Bounded Rationality

Grounded in contemporary decision science, this paper has two central assumptions about human players' decision making: The first is that players' decisions are guided by their expected *utility* for a given decision; i.e. the amount of experienced value they expect to derive from the consequences of a decision.

The second is that human players exhibit *bounded rationality* i.e. players allocate limited amounts of cognitive resources to decisions in games either due to innate limitations or because they only apply part of their cognitive capacity to the decision due to conscious or subliminal reasons. Decision making style in games thus depends not only on preferences in outcomes, but also the resources the player is willing or able to allocate to the decision making task. Our approach to simulating a decision maker in the form of a generative agent is to represent these two characteristics, the player's rational utility function and the player's cognitive bounds, in the implementation of the agent.

In the following sections we outline the relations between persona theory, decision theory, player modeling, and the resulting concept of procedural personas. We briefly describe our testbed game, *MiniDungeons*, and the methods we used to create game playing personas and clones, before we present the results from comparing the resulting agents to the human players.

2. Related Work

In this section, we review decision theory, the concept of personas as applied to (digital) games, player modeling, and the relations between the three areas in this study.

2.1. Decision Theory and Utility

The personas used for expressing designer notions of archetypal player behavior in *MiniDungeons* are structured around the central concepts of decision theory. Decision theory states that whenever a human makes a *rational decision* in a given situation, the decision is a result of an attempt to optimize the expected *utility* [8]. Utility describes any positive outcome for the decision maker and is fundamentally assumed to be idiosyncratic. This means that in principle no definite assumptions can be made about what can provide utility to the decision maker. The problem is further complicated by the fact that the effort a decision maker directs toward attaining maximum utility from a decision can be contingent on the expected utility itself. For problems that are expected to provide low utility even in the best case, humans are prone to rely more heavily on heuristics and biases for the decision making process, further bounding the rational analysis applied to the problem [9, 10, 11, 12].

In practice, however, for structured, well-defined problems, such as many games, insights from e.g. psychology or contextual information about the decision maker or the decision problem may provide us with opportunities for assuming which decisions are important and which outcomes may be of utility to the decision maker. As decision spaces, most games are special cases since the available decisions and their consequences are highly structured by the game's mechanics and evaluation mechanisms. Games, through their design, often provide specific affordances [13, 14] to the player, and suggest utility for various outcomes. This perspective forms the basis for our understanding of player behavior in our testbed game, as we assume that players are interacting with the game in accordance with the rules, understanding and responding to the affordances of our game. That, in turn, motivates our use of utility for attaining game rule based affordances as the defining characteristics of the personas we develop. Similar theoretical perspectives have been described by other authors, notably Dave Mark in [15]. When attempting to characterize player decision making styles in games using utilities, it is important to consider the level of decision making relevant for the game, as described in [16]. Here, we model players at the individual *action level*, at the more *tactical level* of game affordances, and at the *strategic level* of aggregate outcomes.

In the following section, we suggest how the concept of play-personas can be used to arrive at a selection of utility configurations for a particular game.

2.2. Personas

The concept of personas was first adapted to the domain of (digital) games under the headline of *play-personas* by Canossa and Drachen who define play-personas as "clusters of preferential interaction *(what)* and navigation *(where)* attitudes, temporally expressed *(when)*, that coalesce around different kinds of inscribed affordances in the artefacts provided by game designers" [17]. Their work focuses on how assumptions about such player preferences can be used as metaphors for imagined player behavior during the design process or patterns in observed player behavior can be used to form lenses on the game's design during play-testing.

Applying the perspective of decision theory further narrows the play-persona concept. Rather than consider any arbitrary reason for player preferences, decision theory operationalizes the backgrounds for preferences into combinations of affordances and utilities. For any spatio-temporal configuration of a given game, a limited number of plausible affordances can be determined using information about the game mechanics and reward structures. Based on these affordances, different hypothetical combinations of utilities can be used to create metaphors for typical player behavior. To the extent that these metaphors match what actual human players decided, they can be considered lenses on the players' decision making styles with utilities explaining how player preferences are distributed between the available affordances. Our long term research agenda is to operationalize the play-persona concept into actual game playing procedural personas, by building generative models of player behavior from designer metaphors, actual play data, or combinations of the two.

In the following section, we argue for the use of game playing agents to provide such representations of possible configurations of utilities, drawn from play-personas.

2.3. Player Modeling

Generative models of player behavior can be learned using a number of different methods. A key dichotomy in any player modeling approach lies in the influence of theory (vs. data) for the construction of the player model [18]. On one end, *model-based* approaches rely on a theoretical framework (in our case persona theory or expert domain knowledge) and on the other hand, computational models are built in a *model-free*, data-driven fashion. In this paper, *personas* represent the model-based approach while what we term *clones* represent the data-driven approach. Within the model-free approach, a fundamental distinction is between *direct* and *indirect* player imitation, where the former uses supervised learning methods to train agents directly on play-traces, and the latter uses some form of reinforcement learning to train agents to behave in a way that agrees with high-level features extracted from the play-traces [19]. In several investigations, direct and indirect comparisons have been compared for imitating player behavior in racing games [19, 20] and platform games [21]. Model-free player modeling can be done by imitating the player directly, using supervised learning methods on the play-traces, or indirectly using some form of reinforcement learning to train agents to behave in a way that agrees with high-level features extracted from the play-traces [19]. Evolutionary computation can be used to optimize an agent to behave similarly to a play-trace or optimize it to exhibit the same macro-properties as said play-trace [19, 20, 21]. Direct imitation is prone to a form of over-fitting where the agent only learns to cope with situations which exist in the play-traces, and might behave erratically when faced with new situations. Indirect imitation to a large extent solves this problem by learning a more robust, general strategy, which could be termed a decision making style. Here, we investigate this problem by comparing clones that are directly trained on play-traces, but tested on unseen maps, to the personas. Finally, we use specialized agents directly trained and tested on seen maps as a best case, but context dependent, performance of the chosen agent representation.

In the following section we describe the *MiniDungeons* testbed game in further detail.

Figure 1: Heat-maps of six selected human play-traces in Level 2 of *MiniDungeons*, showing a diversity of player decision making styles. Note that in two heat-maps, top center and bottom right, the player died before completing the level.

3. MiniDungeons

The testbed game, *MiniDungeons*, implements the fundamental mechanics of a dungeon exploration game: the player navigates an avatar through a dungeon containing enemies, power-ups, and rewards. The turn-based game puts the player in a top-down viewed dungeon containing monsters, potions, and treasures on a grid of 12 by 12 tiles. Wall tiles block the avatar's movement, while passable tiles may contain enemies or items for the player. The avatar begins the level at the entrance tile, and finishes the level (loading the next level) when it reaches the exit tile. All of the level is visible to the player who can move freely between passable tiles. When the player moves to a tile occupied by a monster or item, immediately the monster is fought or the item is collected and applied. The player has a health counter of 40 *hit points* (HP) and dies if this drops to zero. Monsters randomly deal between 5 and 14 HP of damage while potions heal 10 HP up to a maximum of 40 HP. Treasures have no game mechanical effect other than adding to a counter of collected treasure. The game contains one tutorial level and 10 "real" levels. For further details on the test-bed game and discussion of its properties, we refer to [5]. The necessary data for developing and evaluating the agents was collected from 38 anonymous users who played *MiniDungeons* on-line; this resulted in 380 individual play-traces on the 10 *MiniDungeons* levels provided. The data was subsequently used to evolve clones and specialized agents as described below. Figure 1 shows Level 2 from the game, along with human play-traces from the level, exemplifying the diversity of human decision making styles expressed in even a simple game like this. In Section 5.1 we give a brief introduction to our method of representation, but first we introduce three metrics that we propose for evaluating to which degree personas, clones, and specialized agents successfully enact human decision making styles in *MiniDungeons*.

4. Agreement Ratios for Evaluating Player Models

In this section, we present the three different metrics used to evaluate the performance of the agents. Each metric was constructed to capture a different level of game play, ranging from the specific and atomic to the general and aggregated.

4.1. Action Agreement Ratio

The first metric used to evaluate agent to human likeness is the *action agreement ratio* (AAR). AAR considers each step of a human play-trace a distinct decision. To produce the AAR between an agent and a human player, all distinct game states of the human play-traces are reconstructed. For each game state, the agent being tested is inserted into the game state and queried for the next preferred action, essentially asking: "What would you do?". If the action is the same as the actual next human action, the agent is awarded one point. Finally, the AAR is computed by dividing the points with the number of decisions in the human play-trace. As such, a perfect AAR score of 1.0 represents an agent that for every situation in the player's play trace decided to take exactly the same action as the player did.

4.2. Tactical Agreement Ratio

The second metric used for evaluating the likeness between agents and humans is the *tactical agreement ratio* (TAR). TAR only considers reaching each distinct affordance in the level a significant decision, ignoring the individual actions in between. For *MiniDungeons* the affordances considered relevant are: fighting a monster, drinking a potion, collecting a treasure, or exiting a level. For each affordance reached in the human play-trace, the resulting game state is reconstructed and the agent being tested is inserted into the game state. The agent is then allowed as many actions as necessary to reach the next affordance, asking the question "What affordance would you go for next?" at the tactical level. If the next encountered affordance, in terms of both type and location, matches the actual next human one exactly, the agent is awarded a point. Finally, the TAR is computed by dividing the points with the number of affordances reached in the human play-trace. As such, a perfect TAR score of 1.0 represents an agent that visits every affordance in the level in the same order as the player originally did.

4.3. Strategic Agreement Ratio

The third metric used for evaluating the likeness between agents and humans is the *strategic agreement ratio* (SAR). Operating at the general and aggregate level, SAR considers the total amount of affordances engaged with for each level. The affordances considered by SAR are: the number of monsters fought, the number of treasures collected, the number of actions taken, and whether the agent reached the exit and was alive at the end of the level. For each affordance the absolute difference between the agent's measure and the player's measure (e.g. monsters killed) is calculated and normalized by the maximal possible number for the level or, in the case of the number of moves, in relation to the number of moves in the player's play trace. These statistics are then summed, divided by the number of statistics (in this case five). The score, which is an expression of how different the agent's statistics are from the player's, is subtracted from 1.0 to produce the SAR. As such, a perfect SAR score of 1.0 would indicate an agent that fought exactly the same number of monsters, collected exactly the same number of treasures, died in combat or exited the level just like the player, and did so in exactly the same number of actions. In other words, the SAR asks the question "How often would you go for each affordance in this level?"

In the following section we describe the controllers that were evolved and evaluated using the metrics as fitness functions.

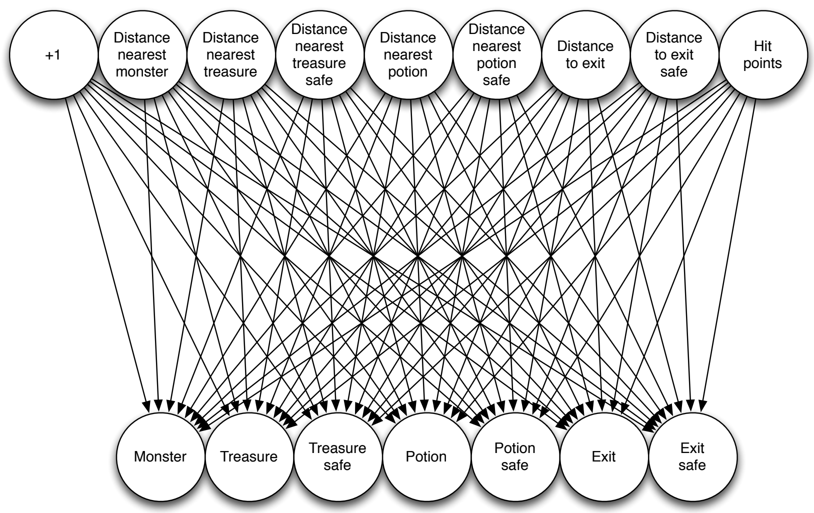
5. Generative Agents

This section describes the general controller framework used to evolve personas, clones, and specialized agents, and the evolutionary algorithm used to learn behaviors from utilities in the case of personas and from the AAR, TAR, and SAR metrics in the cases of clones and specialized agents. All agents were evolved using the same algorithm; what differentiates personas, clones, and specialized agents is the fitness function and whether they are evolved across multiple levels (personas and clones) or on just one level (specialized agents).

5.1. Evolving Agent Controllers

Each agent controller is represented as seven linear perceptrons. Each perceptron takes 8 inputs describing safe and risky path distances to the nearest affordances in the map as illustrated in Figure 2. By only considering the nearest affordances, the agent controller simulates bounded rationality in the sense that it only looks one affordance ahead. For the sake of simplicity this bound is not varied in these experiments, but potentially this horizon could be changed to represent different degrees of bounded rationality. Further details of the controller representation is given in [6]. Controllers are evolved using a $(\mu + \alpha)$ evolutionary strategy without self-adaptation. For each generation the best 2% individuals remain unchanged, the lowest performing half of the remaining population is removed, and single-parent offspring from the remaining individuals are produced to maintain the population size. Finally all individuals not in the elite are mutated. Mutation is accomplished by changing each connection weight in the network with a random number drawn from a Gaussian distribution centered around zero with a standard variation of 0.3, a value confirmed as useful for this game by informal experimentation. All experiments are done using a population size of 100 individuals, evolved for 100 generations. Controllers

Figure 2: The controller network used for all agents: personas, clones, and specialized agents. The weights of the connections are determined through evolution as described in Section 5.1. For every action it takes in the game, a controller uses the current state of the game, represented by the 8 input nodes (disregarding the bias node), and the weights to select which of the 7 strategies represented by the output nodes to pursue.



are initialized with random connection weights for all connections in the linear perceptrons. The topology shared by all controllers is illustrated in Figure 2.

5.2. Personas

For the purpose of the experiments, five individual personas with different utility configurations were defined, based on designer interpretations of likely game-play in *MiniDungeons*. The personas were intended to represent five hypothetical extreme decision making styles in interacting with the game: an *Exit* (E) persona who simply tries to escape the level, a *Runner* (R) persona who tries to escape the level in as few steps as possible, a *Survivalist* (S) persona who tries to avoid risk, a *Monster Killer* (MK) persona who tries to kill all monsters and escape the level, and a *Treasure Collector* (TC) persona who attempts to collect all treasures and escape the level. The decision making styles are defined by the utility weights presented in Table 1, and serve as a metaphor for the relative importance of the affordances to the archetypal player represented by the persona. The values were assigned by the authors, as the designers of the game, by imagining how different archetypal player types would weigh the various affordances. When assigned to personas as a fitness score during the evolutionary process, utility points attained from a level are normalized by the maximally attainable utility for the same level. E.g. killing three monsters in a level with eight monsters will yield a Monster Killer persona 0.375 utility points, while yielding a Treasure Collector persona no points. During evolution, personas are exposed to and evaluated on 9 of the 10 *MiniDungeons* levels. The 10th level is

Table 1: Utility weights for the five designed personas.

Affordances	E	R	S	MK	TC
Move	-0.01	-0.02	-0.01	-0.01	-0.01
Monster				1	
Treasure					1
Death			-1		
Exit	0.5	0.5	0.5	0.5	0.5

subsequently reserved for comparison with human players. In total, 50 personas were evolved for this study. More details can be found in [6].

5.3. Clones

Clones, like personas, are evolved by exposing them to 9 of the 10 levels of *MiniDungeons*. Their fitness value is computed as the average normalized AAR, TAR or SAR across all 9 seen levels. One clone per player per map is evolved, yielding 380 agents per evaluation metric, in total 1140 clones. All subsequent tests are done comparing the clones to the players they were cloned from on their individually unseen levels.

5.4. Specialized Agents

In order to find the likely closest possible fit of the perceptron-based representation, a set of specialized agents is evolved. Again, one agent for each human play-trace for each evaluation metric is evolved, resulting in 1140 total. These are evolved on a single level of *MiniDungeons* each. Their fitness scores are computed directly from AAR, TAR or SAR on that same level in an attempt to establish the closest fit to each human player the representation can achieve.

6. Results

This section compares the three presented evaluation metrics, and compares the ability of personas, clones, and specialized agents to represent human decision making styles in *MiniDungeons*. It presents a breakdown of how the application of different evaluation metrics change which personas are mapped to individual players. Finally, it provides an example of a single player's play-trace and the personas, clones, and specialized agents derived from that player on a particular level.

Table 2 shows the mean of the agreement ratios for each kind of agent evolved, compared to each other, using the AAR, TAR, and SAR metrics. In the case of personas, the best matching persona from the 5 persona gallery, meaning the one with the highest agreement ratio (for AAR, TAR, or SAR respectively) is identified for each player on each level and used in the analysis. The distributions of best matches are presented below in Table 3.

The achieved ratios are generally highest for AAR, followed by SAR, with TAR producing the lowest ratios. However, while the ratios share some semantic

Table 2: Means and standard deviations (SD) of agreement ratios attained for best matching personas, clones, and agents evolved using the described fitness functions. It is worth noting that across all three metrics, personas generally exhibit performance close to that of the clones. Additionally, it is worth noting that while specialized agents evolved using TAR and SAR as fitness functions perform best in the metric they were evolved from, this is not the case for agents evolved using the AAR metric as a fitness function. Using the AAR metric, the TAR-evolved specialized agents exhibit the best performance.

		Evaluation metric					
		AAR		**TAR**		**SAR**	
Agent type	**Fitness**	Mean	SD	Mean	SD	Mean	SD
Personas	Utilities	0.75	0.08	0.62	0.13	0.77	0.16
Clones	AAR	0.77	0.08	0.66	0.13	0.68	0.23
Clones	TAR	0.76	0.09	0.65	0.13	0.67	0.23
Clones	SAR	0.69	0.10	0.53	0.17	0.72	0.22
Specialized	AAR	0.81	0.09	0.73	0.17	0.68	0.24
Specialized	TAR	0.84	0.07	0.86	0.09	0.71	0.24
Specialized	SAR	0.73	0.10	0.57	0.18	0.85	0.21

properties such as an upper perfect match represented by 1.0, it should be noted that the ratios are not directly comparable.

No agents, not even the specialized ones, attain a perfect agreement ratio. This indicates that the chosen agent control architecture seems incapable of matching players perfectly. This may be due to an inability of the seven-perceptron network to represent and hence learn the decision making preferences of actual players or it may be due to the fact that controllers are only provided with information about the distance to the nearest kind of each affordance. While the linear perceptron network has the advantage of having easily inspectable and interpretable weights, other networks with greater representational power (such as non-linear multilayer perceptrons) may provide a better matching of individual player from the same information.

An interesting perspective is to compare across agent types within the different evaluation metrics. If we look at how fitness functions based on AAR, TAR, and SAR respectively perform when cross-evaluated against each other within clones and specialized agents, we see that AAR and TAR produce comparable results. In other words, a clone or a specialized agent evolved using AAR as a fitness function also performs well when evaluated by TAR and vice versa. Agents evolved using SAR on the other hand, only perform relatively well when evaluated by SAR. For the other metrics, clones and specialized agents evolved from SAR perform worse than the other members of their groups. This may indicate that evolving agents using the high-level SAR as a metric, results in some loss of information about the decision making styles that players are enacting. When agents are only evolved from information about the aggregate outcome for each level, they might not learn about the order in which a player prefers to pursue affordances. SAR results also show higher standard deviations, which may be attributable to the loss of information from human play-traces making it harder to produce reliable matches. This suggests that AAR and TAR may be the most relevant metrics for modeling player decision making styles in *MiniDungeons*.

Next, we investigate the differences between how well personas and clones agree with player decision making styles. In the following analyses, specialized agents are omitted as they are assumed to represent the likely closest fit of the representation and indeed produce the highest agreement ratios when evaluated on the same metric from which they were developed.

Using the AAR evaluation metric across personas and clones, a one-way ANOVA shows significant differences ($F(3, 1516) = 62.24, p < 0.001$) between the means of the four groups: the best matching personas and the three kinds of clones. Similar differences were established when using the TAR ($F(3, 1516) = 68.16, p < 0.001$) and the SAR evaluation metrics ($F(3, 1516) = 16.29, p < 0.001$), in spite of large standard deviations in the latter case.

For all evaluation metrics, Tukey HSD tests for post-hoc analysis reveal that the personas are significantly different ($p < 0.05$) from all clones, regardless of the fitness function used to evolve the clones. The only exception is when personas are compared to clones evolved from TAR, evaluated by AAR. With AARs of 0.75 and 0.76, respectively, there is no significant difference between these two groups.

When examining the specific mean agreement ratios of the personas in contrast to the clones it is clear that the actual differences in the case of AAR, even though significant, are minor and that the personas for all practical purposes achieve the same performance as the clones, perhaps with the exception of clones evolved through SAR which perform the worst out of the group.

A similar pattern is repeated when examining the mean agreement ratios calculated using TAR, though the personas in this case rank lower than clones evolved from both AAR and TAR, while clones evolved using SAR again produce the lowest agreement ratio.

Finally, when applying the SAR metric, the personas outperform all clone types with a sizable performance difference between the personas and the worst performing clones, the ones evolved using SAR.

Taken together the above results indicate that a small gallery of five personas, defined using utility theory by game designers, are capable of representing a corpus of 38 players across 10 levels with roughly the same performance as individually evolved clones. The personas do perform worse than specialized agents evolved to specifically copy player behavior on one particular level, but this is to be expected. Measured by the action and strategic metrics, the personas come relatively close to the specialized agents, while they lag further behind the specialized agents when applying the tactical metric.

Table 3 shows which personas best captured human-play-traces for each *MiniDungeons* level and in total. For each human play-trace, the personas with the highest AAR, TAR, and SAR respectively, are identified. All three metrics generally favor the Treasure Collector persona as the best match for most play-traces, although there is some discrepancy among the three measures. Notably, the SAR metric yields best matches that are quite different from the matches

Table 3: Best persona matches based on Action Agreement Ratio (AAR), Tactical Agreement Ratio (TAR), and Strategic Agreement Ratio (SAR), respectively

AAR

Persona/Level	1	2	3	4	5	6	7	8	9	10	Total
Exit	0	2	5	1	0	1	5	1	2	3	20
Runner	0	0	0	0	0	0	0	0	0	0	0
Survivalist	0	1	0	0	0	0	0	0	0	0	1
Monster Killer	8	8	0	2	3	1	7	2	2	0	33
Treasure Collector	30	27	33	35	35	36	26	35	34	35	326
Total	38	38	38	38	38	38	38	38	38	38	380

TAR

Persona/Level	1	2	3	4	5	6	7	8	9	10	Total
Exit	0	0	5	1	0	1	4	0	2	0	13
Runner	0	0	0	0	0	0	0	0	0	0	0
Survivalist	0	0	0	0	0	0	0	0	0	0	0
Monster Killer	5	15	3	3	4	3	4	5	0	0	42
Treasure Collector	33	23	30	34	34	34	30	33	36	38	325
Total	38	38	38	38	38	38	38	38	38	38	380

SAR

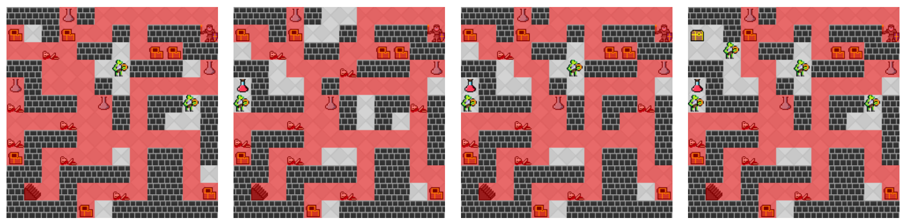
Persona/Level	1	2	3	4	5	6	7	8	9	10	Total
Exit	7	7	10	11	2	15	11	1	7	0	71
Runner	0	0	0	1	0	2	2	0	0	0	5
Survivalist	0	0	0	0	0	1	0	3	0	1	5
Monster Killer	5	10	5	7	3	7	18	6	14	3	78
Treasure Collector	26	21	23	19	33	13	7	28	17	34	221
Total	38	38	38	38	38	38	38	38	38	38	380

yielded by the AAR and TAR metrics. This underlines the fact that using an aggregate strategic level metric allows for larger degrees of variability in the decision order, since different orderings may lead to the same aggregate results.

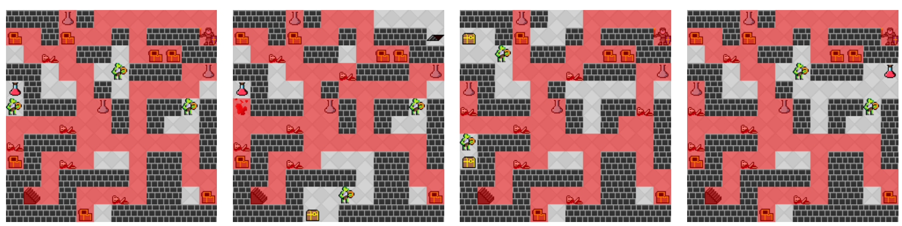
Finally, Figure 3 shows a particular player's original play trace along with the best matched personas, using each metric, and the derived clones and specialized agents. The player was matched by the Treasure Collector by all three metrics, as was indeed the typical case in the data set. The heat-maps illustrate how most agents provide a relatively close match to the visited areas of the level. The Treasure Collector comes close to matching the visitation pattern of the player, but fights several monsters the player did not fight, as monsters have no bearing on the utility attained by the Treasure Collector as long as they do not endanger the persona's chances of completing the level. The AAR metric does not consider the order of the affordances in the level directly, but only the immediate next atomic action. As a consequence, the clone and the specialized agent derived from this metric miss visiting tiles occupied by monsters and potions visited by the player, but in general exhibit a visitation pattern close to that of the player. The TAR metric does consider the kinds of affordances it visits and the order of them and hence visits more of the same specific monsters, treasures and potions of the level as the player. As a consequence the visitation

Figure 3: Heat-maps from one player and all best matching personas, all clones, and all specialized agents on Level 2. For each agent the maximally attained agreement on each metric is indicated in the order AAR; TAR; SAR. The metric used to drive the evolution of each agent, excepting the persona, is indicated in bold. The best matching persona, by all metrics, was the Treasure Collector.

(a) Player | (b) Personas (TC) 0.78; 0.61; 0.94 | (c) AAR Clone **0.83**; 0.78; 0.57 | (d) AAR Specialized **0.86**; 0.77; 0.53



(e) TAR Clone 0.85; **0.78**; 0.56 | (f) TAR Specialized 0.85; **0.89**; 0.84 | (g) SAR Clone 0.60; 0.39; **0.47** | (h) SAR Specialized 0.82; 0.78; **1.00**



pattern also looks the same. The SAR metric only considers outputting the same number of affordance interactions as the player and does not concern itself with neither the order, nor the location, of these affordances. The SAR based clone ends up with a relatively (and uncharacteristically) poor performance, possibly by choosing an early order of actions that makes it difficult for it to achieve the same statistics as the player. Meanwhile, the SAR specialized agent attains perfect performance [1], but does so through a quite different visitation pattern, demonstrating how the SAR metric allows for greater variation in the underlying implementation of actions leading to the same SAR value.

7. Discussion

In the following discussion we revisit and evaluate the results from the experiments. Two important observations must be made in relation to the experiment used to collect the play-traces. Firstly, all 38 players in the study were playing *MiniDungeons* for the first time. Though they were given unlimited plays on the tutorial level, they were not familiar with the rules of the games or the

[1] A potion is missed, but these are not included in the SAR metric.

monsters' damage range. The play-traces collected are most likely subject to learning effects, as the players moved from being novices to developing some expertise, which in turn may have impacted their decision making styles. It might also have made players change decision making styles along the way, as their expertise increased. However, *MiniDungeons* is a relatively easy game to learn, making it likely that any learning effects beyond the first few levels would be limited. Future work could attempt to measure such learning effects or try to counter-act them by randomizing level order or operating with a block experimental design.

Secondly, each level in the game was played independently of the preceding levels. Each level let the player start with full hit-points. This may have lead players to favor the Treasure Collector decision making style, simply because risk was perceived as low. If a player died before reaching the exit, the player would start the next level with full health.

Below, we discuss the observed properties and performances of the three types of metrics used. We then briefly discuss the used controller architecture and whether its performance, as indicated by the metrics, suggests that it is sufficient for playing *MiniDungeons*. Afterwards, we discuss the implications of personas and clones attaining relatively similar performances and whether either method hypothetically would be transferable to other games. Finally, we suggest future work, including adapting both personas and clones further to observed play-traces by fitting utility values through multi-objective evolution.

7.1. AAR, TAR, and SAR metrics

The three different metrics presented in this paper are not directly comparable as they operate on three different levels of analysis, relative to the mechanics and affordances of the *MiniDungeons* game. As the results above indicate, each metric yields somewhat different behavior when applied to the evolution of game playing agents representative of human players, each capturing different aspects of the original player's decision making style.

The metrics operate at three different levels of analysis, which is rooted in the theoretical notion that, even for simple games, decision making takes place at multiple levels simultaneously, focusing on the individual decision, the order of individual decisions, and the aggregate outcome of a set of decisions irrespective of the individual decisions or their order.

However, the specific implementation of the three metrics are tailored to the affordances and game design of *MiniDungeons*, and not rooted in a theoretical system of analysis. This could be potentially be addressed by applying existing formal systems for identifying affordances, decisions, and actions in games, such as e.g. through the notion of game design patterns [22]. The cost of taking this approach would be a longer, more involved analytical process before metrics could be defined for a particular game, but the gain would potentially be a greater validity of the chosen affordances. Depending on the purpose of the modeling, this may or may not be a desirable trade-off: a single developer looking to reduce time and resource consumption spent on play-testing may be perfectly content with using her own expertise as a valid foundation. In contrast, a team working in a larger organization or doing an academic study of one or multiple games might prefer grounding the selected affordances and decisions in a larger corpus of examples collected from other games.

Finally, while these metrics are specifically targeted at modeling decision making, other play-trace comparison metrics could be used to compare agent behavior to human player behavior: e.g. action/edit-distance based methods such as the *Gamalyzer* metric [23].

7.2. Controller Architecture Performance

The results indicate that a relatively small gallery of personas, crafted analytically by a game designer, may provide representational performance comparable to that achieved by the cloning approach. Still, the evolution of specialized agents that represent individual players on individual levels provides a reproductive fidelity that neither personas nor clones can match. On the other hand, specialized agents suffer from the problem that they are optimized for one particular level and are not well suited for generalizing to new levels, for which they have not been evolved.

The performances of the specialized agents reveal that the chosen representation using seven linear perceptrons is incapable of learning a human play-trace perfectly. This could motivate two changes to the controller to achieve better performance. Firstly, the use of a more complex controller, such as e.g. a controller based on multilayer perceptrons configured using Neuro-Evolution with Augmenting Topologies [24], would most likely exhibit a greater ability to learn individual play-traces. Secondly, controller performance might be improved by increasing the environment sensing capabilities of the controller. In the current implementation, the controller is only informed about risky and safe path distances to the nearest affordance of each kind. As such, it can be interpreted as representing the assumption that players only look to the nearest affordances that they can reach when making decisions. This is most likely an over-simplification which, while having the advantage of simplifying the implementation and intelligibility of the model, may be too aggressive. Future work should explore more complicated agent control architectures.

7.3. Personas and Clones for MiniDungeons and Beyond

The fact that the top-down approach of persona construction and the bottom-up approach of cloning yield similar performances provides some indication that using procedural personas for synthesizing player models and play-test data for games might be feasible at least for games of the same scope as *MiniDungeons*.

We suggest that this might be useful in multiple situations: one example could be when game designers need quick access to potential interactions with new content during an iterative design process. By considering only the basic affordances of their game, and producing different preference orderings of these by assigning utilities, they may quickly sketch out different play-styles and see them in action. Another example could be for games using procedural content generation to an extent where full human curation of the generated content is impossible. Here, procedural personas can act as proxy critics of a human designer's intentions [4].

The persona method is less play-trace-dependent and computationally expensive than the cloning method, but needs an expert game designer. Still, some players may exhibit decision making styles that cannot be captured by the designer's intuition, and would be captured better by the cloning approach.

MiniDungeons is arguably a game with a small scope, compared to most commercial games (even most made by single, independent developers). This begs the question of whether the method is extensible to larger games. While this is an open question at the moment, some strategies may be imagined. Any game which may be abstracted to a discrete graph of decisions may in principle be subject to the method. How this abstraction could be appropriately implemented would again be dependent on the design of the game in question and possibly subordinate modeling steps could be added to the method. If e.g. a first person shooter were being modeled, a first step could be to reduce the game's spatial areas into specific decision points, based on the affordances they contain. A collection of design patterns for the genre, such as those described in [25], might aid in this process. Once the design patterns were identified, personas would be implemented to choose between these areas based on their play style preferences, choosing e.g. between a sniper area or a close combat area. Decision making styles within each area could then be simulated using a separate application of the persona method, a probabilistic model, or simple scripted behavior. The major challenge in this case would be to arrive at a successful abstraction of the game space into appropriate affordances with an acceptable amount of effort. Future work will focus on formalizing this process.

7.4. Combining Personas and Clones

The fact that the persona and the cloning methods seem to perform equally well for *MiniDungeons* raises the question of whether the persona method has any value once a sufficient amount of human play-traces have been collected for modeling. We suggest that even in this situation, procedural personas may offer two advantages to game and level designers.

Firstly, each persona defines a decision making style grounded in the designer's expectations. Individual or groups of human players may be described by their distances to these a priori defined decision making styles. This means that the designer is not interpreting the play traces based on their observed decision making styles alone, but also based on her preconceptions of the affordances of the game's current design. This helps contrast what the designer expected to what the players actually did.

Secondly, once a relation between personas and clones has been established, it becomes possible to interpolate between personas and individual clones or points defined by clusters of clones. This allows a designer to define new personas that better match the observed play-traces, but still follow her own observations, expectations and wishes for decision making styles within the game space.

Distances between personas and clones can be defined according to any of the three proposed metrics, depending on the designer's agenda, or according to an aggregation of all three metrics. While the designer could define these new adapted personas manually, a more efficient approach would be to fit utility values automatically to maximize one, two or all three of the defined agreement ratios. If a designer wanted to maximize agreement for all three metrics, multi-objective evolutionary methods [20] could be used to arrive at suitable utility weight configurations. Once these had been determined computationally, the designer could then inspect these utility values to understand the differences between the original personas and the adapted ones and further adjust them manually to her preference.

8. Conclusion

This paper presented a framework for modeling player decision making styles. This framework was implemented in three different manners: One was based on *personas*, evolved from designer expert knowledge, another was based on *clones*, based on human play-traces, while the third used *specialized agents* to replicate human play-traces while sacrificing generalizability. Three metrics were used to evaluate the agents' ability to represent human decision making styles, focused on the *action*, *tactical*, and *strategic* levels, respectively.

Personas and clones were shown to represent human decision making styles almost equally well when compared at the action and tactical levels. At the strategic level, personas were somewhat better at representing human decision making styles, compared to clones. The three different metrics showed how focusing on different analytical levels of a game can be used to characterize play traces differently and to generate variation in game playing agents.

Two advantages of using personas over clones is that we lose little accuracy from using personas instead of clones and we do not need to collect empirical data from players before we can start modeling. Clones, on the other hand, may learn examples of decision making styles that we as designers might not imagine on our own. The advantage of using either personas or clones over specialized agents is that — even though they are less accurate — they may be used on novel content, which is critical in enabling procedural play-testing.

Based on the results, we conclude that using the top-down approach procedural personas for representing player decision making styles is comparable to using the bottom-up approach of cloning. Either approach may be useful for synthesizing play-test data for new content as it is being generated which may be of use to game designers authoring content as well as game designers building procedural content generation systems.

Acknowledgments

We would thank the players of the game for providing data and our reviewers for providing a number of observations and suggestions that significantly improved this paper. The research was supported, in part, by the FP7 ICT project C2Learn (project no: 318480) and by the FP7 Marie Curie CIG project AutoGameDesign (project no: 630665).

References

[1] T. Fullerton, C. Swain, S. Hoffman, Game Design Workshop: Designing, Prototyping, and Playtesting Games, Focal Press, 2004.

[2] M. S. El-Nasr, A. Drachen, A. Canossa, Game Analytics: Maximizing the Value of Player Data, Springer Science & Business Media, 2013.

[3] G. N. Yannakakis, A. Liapis, C. Alexopoulos, Mixed-Initiative Co-Creativity, in: Foundations of Digital Games, 2014.

[4] A. Liapis, C. Holmgård, G. N. Yannakakis, J. Togelius, Procedural Personas as Critics for Dungeon Generation, in: Applications of Evolutionary Computation, Vol. 9028, LNCS, Springer, 2015.

[5] C. Holmgård, A. Liapis, J. Togelius, G. N. Yannakakis, Generative Agents for Player Decision Modeling in Games, in: Foundations of Digital Games, 2014.

[6] C. Holmgård, A. Liapis, J. Togelius, G. N. Yannakakis, Evolving Personas for Player Decision Modeling, in: IEEE Conference on Computational Intelligence and Games, 2014.

[7] C. Holmgård, A. Liapis, J. Togelius, G. N. Yannakakis, Personas versus Clones for Player Decision Modeling, in: Entertainment Computing–ICEC 2014, Springer, 2014, pp. 159–166.

[8] D. Kahneman, A. Tversky, Prospect Theory: An Analysis of Decision under Risk, Econometrica: Journal of the Econometric Society (1979) 263–291.

[9] H. A. Simon, A Behavioral Model of Rational Choice, The Quarterly Journal of Economics (1955) 99–118.

[10] A. Rubinstein, Modeling Bounded Rationality, Vol. 1, MIT Press, 1998.

[11] D. Kahneman, Maps of Bounded Rationality: Psychology for Behavioral Economics, American Economic Review (2003) 1449–1475.

[12] G. Gigerenzer, W. Gaissmaier, Heuristic Decision Making, Annual Review of Psychology 62 (2011) 451–482.

[13] J. Gibson, The Concept of Affordances, Perceiving, Acting, and Knowing (1977) 67–82.

[14] G. S. Elias, R. Garfield, K. R. Gutschera, Characteristics of Games, MIT Press, 2012.

[15] D. Mark, Behavioral Mathematics for Game AI, Course Technology Cengage Learning, 2009.

[16] A. Canossa, Y.-G. Cheong, Between Intention and Improvisation: Limits of Gameplay Metrics Analysis and Phenomenological Debugging, DiGRA Think, Design, Play.

[17] A. Canossa, A. Drachen, Patterns of Play: Play-Personas in User-Centred Game Development, in: Proceedings of the 2009 DiGRA International Conference, 2009.

[18] G. N. Yannakakis, P. Spronck, D. Loiacono, E. André, Player Modeling, in: Artificial and Computational Intelligence in Games, Dagstuhl Publishing, Saarbrücken/Wadern, 2013, pp. 45–55.

[19] J. Togelius, R. De Nardi, S. M. Lucas, Towards Automatic Personalised Content Creation for Racing Games, in: Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on, IEEE, 2007, pp. 252–259.

[20] N. Van Hoorn, J. Togelius, D. Wierstra, J. Schmidhuber, Robust Player Imitation using Multiobjective Evolution, in: Evolutionary Computation, 2009. CEC'09. IEEE Congress on, IEEE, 2009, pp. 652–659.

[21] J. Ortega, N. Shaker, J. Togelius, G. N. Yannakakis, Imitating Human Playing Styles in Super Mario Bros, Entertainment Computing 4 (2) (2013) 93–104.

[22] S. Bjork, J. Holopainen, Patterns in Game Design, Charles River Media, 2004.

[23] J. C. Osborn, M. Mateas, A Game-Independent Play Trace Dissimilarity Metric, in: Foundations of Digital Games, 2014.

[24] K. O. Stanley, R. Miikkulainen, Evolving Neural Networks through Augmenting Topologies, Evolutionary Computation 10 (2) (2002) 99–127.

[25] K. Hullett, J. Whitehead, Design Patterns in FPS Levels, in: Foundations of Digital Games, 2010.