

Evolving Personas for Player Decision Modeling

Christoffer Holmgård, Antonios Liapis, Julian Togelius
Center for Computer Games Research
IT University of Copenhagen
Copenhagen, Denmark
Email: {holmgard,anli}@itu.dk; julian@togelius.com

Georgios N. Yannakakis
Institute of Digital Games
University of Malta
Msida, Malta
Email: georgios.yannakakis@um.edu.mt

Abstract—This paper explores how evolved game playing agents can be used to represent a priori defined archetypical ways of playing a test-bed game, as procedural personas. The end goal of such procedural personas is substituting players when authoring game content manually, procedurally, or both (in a mixed-initiative setting). Building on previous work, we compare the performance of newly evolved agents to agents trained via Q-learning as well as a number of baseline agents. Comparisons are performed on the grounds of game playing ability, generalizability, and conformity among agents. Finally, all agents’ decision making styles are matched to the decision making styles of human players in order to investigate whether the different methods can yield agents who mimic or differ from human decision making in similar ways. The experiments performed in this paper conclude that agents developed from a priori defined objectives can express human decision making styles and that they are more generalizable and versatile than Q-learning and hand-crafted agents.

I. INTRODUCTION

Decision making is a central aspect of almost any interesting agonistic [1] game, as noted by Sid Meier who famously stated that “a game is a series of interesting choices” [2]. Typically, play sessions in agonistic games can be described as chains of decisions by one or more players in a proactively and reactively changing environment. Players make decisions while the environment either observes, responds, or proceeds agnostically, or enacts some combination of the three. Capturing, describing, modeling, and reproducing chains of decisions is of interest to game researchers, developers, and players for several reasons.

One reason can be to characterize typical chains of decisions as being representative of certain *decision making styles* in playing particular games. They represent certain ways of navigating the *decision space* of the game at a chosen level of abstraction. The appropriate level of abstraction naturally differs among games and one game may have several levels of abstraction where decision making can be characterized. For instance, playing a game of Mario entails decisions at the aesthetic level (do I prefer fireball or raccoon tail?) and the tactical level (do I attempt to engage or evade enemies?) as well as the atomic level (do I press up or down?). Decision style characterization should hypothetically be possible at all three levels [3], and the relevant level of analysis must be determined by the purpose of the decision style characterization.

In this paper we address the problem of modeling human decision making in games in the following two ways: Firstly, we attempt to represent archetypical decision making styles in a test-bed game via game playing agents which we call

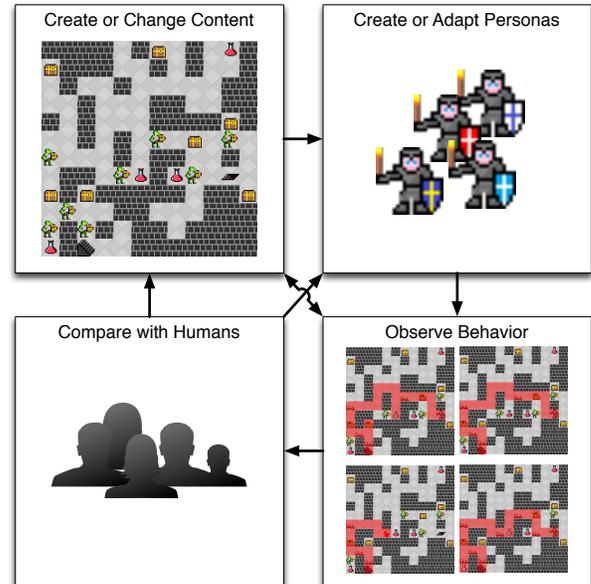


Fig. 1: The intended designer use of procedural personas.

procedural personas. Secondly, we map agent decision making styles to human ones in order to measure to which extent our personas are capable of expressing typical human ways of making decisions within our test-bed game. The chief motivation for the work presented is to use agents that express decision making styles to test manually or procedurally/co-creatively [4] generated game content such as levels [5]. This could support the game design process by providing low-cost, low-fidelity mock playtesting of content during development. Such a method could e.g. be of use and interest to level designers during the design of a new level, allowing for quick impressions of what decisions archetypical players might make in the level. By continuously comparing procedural persona behavior with human behavior, and adjusting persona decision making styles accordingly, the personas might be continuously refined during iterative development and playtesting cycles to better represent identified subgroups of human players, as outlined in our previous work [3]. The envisioned process is illustrated in Fig. 1.

In order to model decision making styles, this paper contributes to and expands on previous work [6] by developing personas based on linear perceptrons and comparing their performance against previously developed personas based on

Q-learning as well as several baseline agents. Our previous approach was limited in terms of performance, generalizability, and scalability; the approach presented here is an attempt to address these problems. To describe this process, we first present related work and outline how our approach is based on psychological decision theory. Secondly, we briefly present the reinforcement learning experiments which we are building on. Thirdly, we describe our method of and fitness functions for evolving linear perceptrons to represent archetypical decision making styles for procedural personas. Fourthly, we present our experiments and results in using these evolved personas to express decision making styles and capture typical player styles in decision making and comparing them to the reinforcement learning agents. We conclude by arguing that evolutionary methods are better suited to replicating observed player behavior than previously applied td-learning techniques, and we point out limitations and under-explored aspects of the method.

II. RELATED WORK

This paper builds upon a theoretical framework of human decision-making; additionally, it is related to player modeling as well as the simulation-based evaluation of procedurally generated content. A brief survey of these domains follows.

A. Decision Theory

As described in [6], decision theory [7] deals with human decision making under risk and uncertainty. One of its fundamental assumptions is that human decision making can be described as being shaped by utility. Briefly stated, utility captures how much an expected outcome of a decision is worth to the decision maker versus the expected cost and risk of attaining that outcome. Humans typically attempt to optimize the utility gained from a decision which is then considered rational action. Research in decision theory has shown that the nature of this optimization process is shaped by the utility expected from the decision, meaning that the assignment of cognitive resources and the balance between heuristic and analytic reasoning is based on the perceived importance of the decision [8]. The decision making process happens under *bounded rationality* [9]. Generally, utility is considered idiosyncratic and decision theory does not try to explain why a given outcome has utility to the decision maker (though other directions in psychology such as personality psychology [10] or motivational psychology [11] might be helpful in explaining this). Instead, it looks at the decision maker's tendency to take risks to attain particular outcomes. This means that for real world decision making problems, the possible sources of utility are practically infinite though often context can be used to identify *probable* sources of utility.

Games, and certain computer games in particular, can be considered special, limited cases of decision making problems, when the game's decision space is delineated by its rules and mechanics. As a game becomes more complex this decision space of course expands and complexifies rapidly. However, knowing the rules and mechanics of a game provides a well-defined context for making assumptions about possible sources of utility in the game. A game's stated goals and the possibilities inherent in the game's mechanics constitute affordances [12] which are likely to be of utility to the player,

since they are typically the very reason for playing the game in the first place. By analyzing the mechanics of a game we should be able to detect likely sources of utility, though due to the idiosyncratic nature of utilities we can never be certain to have covered all cases. This is relevant to our purpose of developing decision making procedural personas, as the analysis of the affordances in a game can provide us with a list of possible goals to direct the behaviors of procedural personas. Once a hierarchy of goals, representing sources of utility, has been established for a persona, we can effectively use this as a representation or metaphor for a decision making style.

In the following section, we describe how this approach can be used to enable a form of player modeling which seems to be relatively underexplored in the academic literature, though perhaps more common for ad hoc industry purposes [13].

B. Player Modeling

Since this work aims to represent archetypical decision making styles in our test-bed game, each resulting persona can be considered an individual player model. Smith et al. [13] provide a useful inclusive taxonomy of player modeling methodologies. The work categorizes player modeling techniques via four different facets: the *scope*, *purpose*, *domain*, and *source* of the player model. **Scope** determines the generalizability of the player model. For this work, the scope of each model is limited to the game in question, since the decisions and utilities are contingent on the particular game, in this case MiniDungeons. **Purpose** refers to the intended use of the model. Our method is generative in the sense that the final intent is to express decision making styles in games, either styles defined a priori by designers or styles adapted to match human styles observed across groups or from individual playthroughs. **Domain** refers to what the model generates, in this case player decisions expressed through in-game actions at the same level that human players would. **Source** refers to the motivation or substrate from which the player model is derived. The models generated from our approach are hybrids in the taxonomy. They are initially *interpreted* in the sense that the a priori personas are developed by the game's designers based on expert knowledge about typical decision making styles of human players, but aim to grow empirically *induced* in the sense that they are partly evaluated on how well they express the decision making styles of actual human players and ultimately should evolve to adapt to these. As such, the method attempts to achieve player modeling by evolving from game designer interpreted personas to player data induced personas, bridging the designer's expert knowledge and empirical play data. Approaching the problem from an alternative framework by Yannakakis et al. [14] our method combines a way to move iteratively from a *model-based* (designer centric) player model to a *model-free* (data centric) player model, creating a *hybrid* player model.

C. Procedural Content Generation

As stated above, the main goal of the procedural persona method is to provide low-cost, low-fidelity mock playtesting in a manner that is useful in supporting game content creation. This may be useful to a human designer manually creating a piece of content in an editor, but human designers are, to some

extent, capable of informally mentally simulating different decision making styles their content might enable. We would argue that the procedural persona method could potentially be of greater use to search based procedural content generation processes [15] that are either wholly procedural or based on mixed-initiative co-creative processes where a human designer and an AI-driven support tool collaboratively produce content [16]. Human designers might use procedural personas as input to a co-creative process, controlling the AI’s search for novel content by asking it to generate content that fits certain decision making styles.

III. PREVIOUS WORK

In this section we briefly present the previous work this paper builds on, and the testbed game on which the experiments were performed.

A. *MiniDungeons*

The test-bed game used, *MiniDungeons*, implements the fundamental mechanics of a roguelike dungeon exploration game. The turn-based game puts the player in a top-down viewed tile-based dungeon (of 12 by 12 tiles) containing monsters, potions, and treasures, as displayed in Fig. 3. Impassable tiles constitute the walls of the dungeon, while passable tiles may contain enemies or items for the player. 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 40 *hit point* (HP) health counter and dies if this drops to zero. Monsters randomly deal between 5 and 14 HP of damage while potions heal 10 HP up to the maximum value of 40 HP. Treasures have no game mechanical effect other than adding to a counter of collected treasures which is displayed to the player. The game contains 10 levels (see Fig. 2) and a tutorial level. Excepting the tutorial level, all levels are generated using the multi-genre mixed-initiative co-creation tool Sentient Sketchbook [17]. For further details on the test-bed game and discussion of its properties, we refer to our previous work [6].

B. *Previous Experiments with Q-learning*

Our previous work demonstrated a proof-of-concept for the idea of training procedural personas to express decision making styles using temporal difference-based (td-based) reinforcement learning, specifically Q-learning.

A data set of 380 human play traces from *MiniDungeons* was collected and used as a reference for determining to which extent the defined personas expressed actual human decision making styles. The resulting personas matched the human players’ decisions with an average precision of 78%, a result that would not have been feasible with any one persona alone.

The specification and logic of the personas was straightforward and intuitive, as the reinforcing rewards given to the agents during training worked as a direct metaphor for the personas’ respective utilities. For instance a “Monster Killer” type persona was given a large reward for every monster killed in a given level and a smaller reward for reaching the exit. Personas trained on this reward configuration consistently



Fig. 3: The *MiniDungeons* interface while a game is played.

expressed decisions that we would argue are subjectively interpretable as a “Monster Killer” style.

Unfortunately, the technique of using td-based reinforcement learning suffers from a number of issues that precludes it from being useful for the practical purpose of on-line interactive content creation. Firstly, the method is computationally expensive as it requires the Q-learning agent to run through a significant number of simulations to learn the appropriate Q-table, amounting to hours worth of training time on a modern desktop computer to train a single persona for a single level. Secondly, given the applied technique of Q-learning, where agents were trained on an only slightly abstracted version of the state space, the personas do not generalize across levels. This necessitates retraining of the personas whenever the level content is changed, further exacerbating the problem of practical applicability. Thirdly, the Q-learning agents would not provide useful starting points for adapting the personas to observed human behavior as it would most likely be more efficient to train new agents using inverse reinforcement learning to represent groups of human players’ or individual human players’ decision making styles. This would again make it impractical to implement the desired iterative hybridization of the designer specified interpreted model and the observation based induced model. Though optimizations, such as applying active learning to the Q-learning process, might possibly mediate these drawbacks, the sum of the concerns listed above motivates us to attempt to replicate the results of the td-based reinforcement learning technique with faster, evolution-based methods.

IV. METHODS

In the following section we describe how we achieve this by evolving linear perceptrons selecting which of the currently available affordances to act on. While these agents are structured very differently, they enact decision making styles equivalent to those of the Q-learning agents trained on observation of the game’s state space.

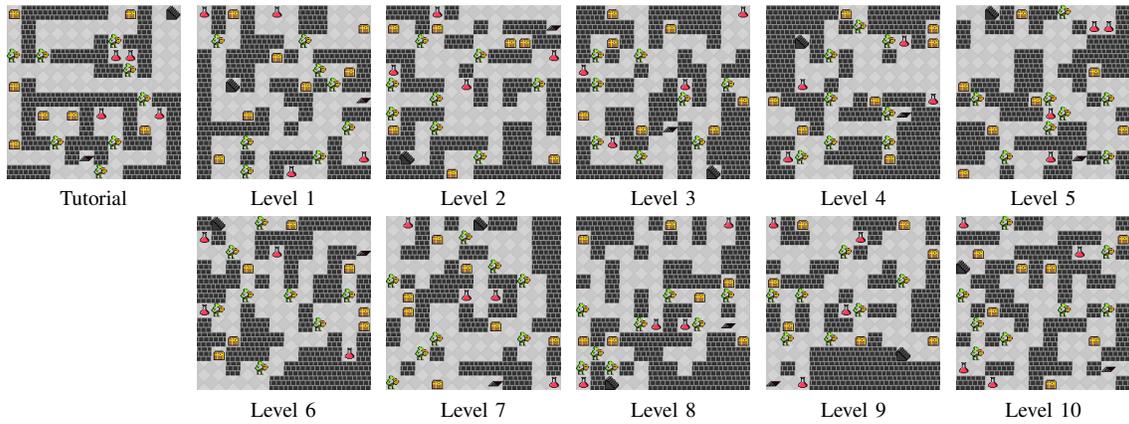


Fig. 2: The levels included in MiniDungeons. The tutorial level is hand-crafted, and could be played multiple times. The “real” levels (1-10) were played only once (no retries if the hero died) and were created in a mixed-initiative fashion.

A. Evolved Controllers

In order to control the personas in MiniDungeons and express the desired decision making styles, 7 linear perceptrons are combined into an evolvable controller. The perceptrons take 8 inputs in addition to the bias, and through weighted sums produce the 7 outputs. The 8 inputs consist of the hero’s current hit points (1 to 40) and 7 distance measures derived from A* path finding (with Manhattan Distance heuristic) in the maze: the distance to the nearest monster, the distance to the nearest treasure, the distance to the nearest treasure while avoiding monsters, the distance to the nearest potion, the distance to the nearest potion while avoiding monsters, the distance to the level exit, and the distance to the exit while avoiding monsters. The inputs are chosen under the assumption that human players will typically survey the whole play area and pick from the available paths to the various affordances in the level. The distance to each affordance type, accepting and avoiding the risk of fighting a monster, are then considered an acceptable abstraction of the game state into a number of utility providing options that the player can choose from. In the same vein, each of the linear perceptrons are mapped to represent one of the available strategies, with or without risk taking: pursuing the nearest monster, pursuing the nearest treasure, pursuing the nearest treasure while avoiding monsters, pursuing the nearest potion, pursuing the nearest potion while avoiding monsters, pursuing the exit, or pursuing the exit while avoiding monsters. The controller re-evaluates the state of the game for each step. The network is fed forward, and the linear perceptron with the highest activation value is identified and from the corresponding affordance, the next step in the path is selected. In the case that an affordance is unavailable, e.g. if all paths to the nearest treasure, avoiding monsters, is blocked, the next ranked affordance, based on activation, is selected, and so on, until the controller ultimately picks the risky path to the exit as a final fall-back affordance.

B. Evolutionary algorithm

We use a $(\mu + \alpha)$ evolution strategy without self-adaptation. This is a truncation-based evolutionary algorithm which for each generation retains the 50% best performing individuals, discards the lowest performing half, and produces single-

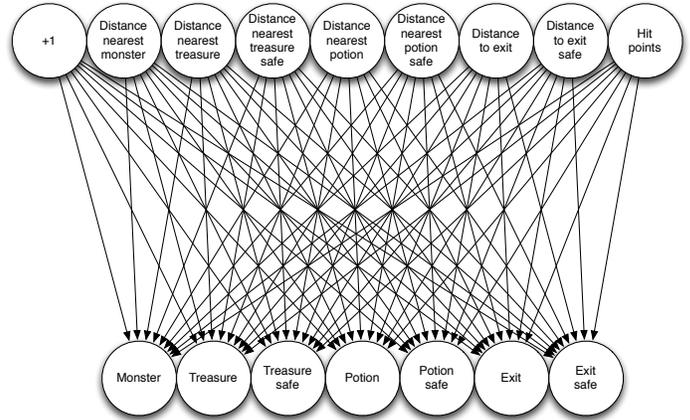


Fig. 4: The controller network which was evolved to generate the five personas.

parent offspring from the remaining individuals to maintain the population size. Finally all population individuals are mutated, except for members of an elite group, consisting of the top performing 2% of the population, which remains unchanged. 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 deviation of 0.3. All experiments are done using a population size of 100 individuals, trained for 100 generations. One set of personas are evolved for one level specifically for 100 generations. Another, generalized, set of personas are evolved by playing on 9 of the 10 levels, keeping the 10th level unseen. As such, each generation of these generalized agents is exposed to 9 times the level content compared to the level specific personas.

C. Fitness Functions for Evolving Personas

As noted above, the procedural personas are evolved to represent archetypical decision making styles, motivated by utilities. In an effort to represent this in the personas, the fitness functions used to evolve the linear perceptrons are constructed as compounds of the relative importance of each potential source of utility in the game and the persona’s ability

to achieve these utilities. Five different potential sources of positive or negative utility were identified in the MiniDungeons game, based on an analysis of the game’s mechanics: making a *move*, fighting a *monster*, collecting a *treasure*, *dying*, and reaching the *exit* of the level. Collecting an HP restoring potion could have been considered a source of utility in itself, but was not included, as it was considered subsumed under the other sources of utility. This resulted in five distinct personas being defined: The *Exit* persona simply tries to reach the exit of the level, representing a player who mainly cares about progressing through the levels of the game. The *Runner* persona attempts to reach the exit in the fewest steps possible, representing a step-optimizing “speed runner”. The *Survivalist* avoids damage to the largest extent possible, while moving toward the exit, representing a conservative player who does not like to lose. The *Monster Killer* represents an aggressive player by seeking out and fighting every monster in the level, while attempting to reach the exit without dying. The *Treasure Collector* represents a completionist player who cares about collecting every treasure, before progressing to the next level. The specific utilities of the personas were set to mirror the rewards given to our Q-learning agents in our previous work [6] as closely as possible. The individual values are presented in Table I. Collecting treasure and killing monsters are associated with positive utility, while moving or dying are associated with negative utility for the relevant personas. All personas derive a slight amount of negative utility from each move made, in order to ensure progression through the level. The value subtracted is doubled for the Runner persona.

Agents’ fitness scores are calculated by dividing the amount of utility obtained by the individual during the playthrough by the maximally attainable utility for the level in question. The only exception to this rule is the number of moves made, which is not normalized. The sources of utility depend on the persona and are shown in Table I along with their utility weights. For personas evolved on a single level, the fitness is computed for each playthrough of the level. For personas evolved across multiple levels, the fitness for each level played is first computed after which the mean across all 9 played levels is computed and used as the fitness score for the individual in that generation.

D. Decision-level playtrace comparison

To determine the degree to which the decision making styles of the evolved personas match the previous reinforcement learning trained personas, as well as actual human decision making styles, a simple measure of agreement is used. For each human play trace, we replay the whole game and at each decision point (in this case every action), we input the state description to all of our artificial agents, and compare the player’s decision to the decisions of the different agents. Essentially, we query each agent “What would you do, given this situation?”. The resulting metric is a simple count of agreements, normalized to a ratio from 0 to 1 by the number of decisions in the human play trace.

E. Baseline Agents

In order to allow for a fair comparison of the performance of the reinforcement learning trained agents and the evolved agents, a number of baseline agents are constructed. The

TABLE I: All personas tested in the experiments.

Persona	Code	Utility weight of tiles or events				
		Moved	Monster	Treasure	Death	Exit
Q-learning						
Exit	q_E					0.5
Runner	q_R	-0.01				0.5
Survivalist	q_S				-1	0.5
Monster Killer	q_M		1			0.5
Treasure Coll.	q_T			1		0.5
Evolution						
Exit	e_E	-0.01				0.5
Runner	e_R	-0.02				0.5
Survivalist	e_S	-0.01			-1	0.5
Monster Killer	e_M	-0.01	1			0.5
Treasure Coll.	e_T	-0.01		1		0.5

TABLE II: Baseline agents used in experiments.

Baseline Agent	Code	Method	Primary Objective
Monster Killer	a_M	A* ignoring monsters	Nearest monster
Runner	a_R	A* ignoring monsters	Exit
Treasure Collector	a_T	A* ignoring monsters	Nearest treasure
Runner Safe	$a_{R,s}$	A* avoiding monsters	Exit
Treasure Collector Safe	$a_{T,s}$	A* avoiding monsters	Nearest treasure
Random Controller	Z	Random legal move	None

simplest one of these is a random controller that every step picks a legal (i.e. leading to a passable tile) decision in the level. Additionally, five more advanced baseline agents are constructed using a finite state machine on top of the A* algorithm for pathfinding. These agents act single-mindedly by always making a decision for following the shortest path toward one primary objective until this objective is exhausted, after which they follow the shortest path to the exit. Their objectives are to pursue either monsters, treasure, or the exit. The treasure and exit objectives are implemented in two modes: one where the baseline agent tries to avoid monsters along the path if possible and one where monsters are ignored and fought if they are present along the shortest path. The baseline agents are not considered personas as such, since their behaviors are too simple, but are included in an attempt to provide baselines for simple random moves, and for single-mindedly following paths to classes of affordances in the game. The complete list of personas tested in our experiments is provided in Table I along with the parameters for their training or evolution.

V. RESULTS

To verify that the developed personas actually exhibit decision making styles in accordance with their intended persona identity, Table III presents the performance of each individual agent across all levels. This verification is necessarily a process of subjective interpretation. From Table III we can identify that both Q-learning based and evolved Exit personas engage in little combat. Additionally they exhibit a relatively low degree of exploration. The Monster Killer personas engage with the most monsters across methods, exhibiting the desired decision making style. Interestingly, the evolved Monster Killer personas tend to collect more potions than the Q-learning trained Monster Killer and also succeed in killing more monsters. The Runner persona exhibits a relatively low exploration value. In the case of the Q-learning agent, the exploration is lower than most other personas, but significantly higher than the

TABLE III: Average statistics of each persona’s play traces across all levels in MiniDungeons. M stands for monsters killed, T for treasures collected, P for potions drunk, Ex for tiles explored, and D for times died. With the exception of D, values are averaged across 20 test runs; D includes all playthroughs tested.

Persona or Agent	M	T	P	Ex	D
q_E Exit	22.8	9.4	2.1	237.9	8
q_M Monster Killer	54.4	9.0	15.9	300.3	66
q_R Runner	22.4	7.7	2.0	231.8	19
q_S Survivalist	4.0	5.0	1.0	135.0	0
q_T Treasure Collector	48.6	49.4	3.9	335.7	167
e_E Exit	22.1	5.5	1.6	217.0	10
e_M Monster Killer	69.5	10.3	30.9	413.9	89
e_R Runner	23.1	6.5	1.4	230.2	15
e_S Survivalist	21.6	7.2	1.2	227.2	2
e_T Treasure Collector	50.1	57.5	6.5	413.9	151
Σe_E Exit	24.2	4.5	1.4	217.7	30
Σe_M Monster Killer	75.5	10.4	37.0	454.8	102
Σe_R Runner	24.1	4.7	1.7	216.8	27
Σe_S Survivalist	24.1	5.0	1.9	221.0	23
Σe_T Treasure Collector	58.9	60.4	23.1	493.4	93
a_M Monster Killer	49.2	3.8	1.2	200.6	200
a_R Runner	24.3	4.8	1.7	218.9	25
$a_{R,s}$ Runner Safe	19.0	4.5	1.4	225.4	31
a_T Treasure Collector	48.2	49.1	1.6	329.9	190
$a_{T,s}$ Treasure Collector Safe	44.7	57.5	3.0	421.9	122
Z Random Controller	58.5	38.4	19.9	521.2	123

survivalist persona, while in the case of the evolved personas it is practically tied with the survivalist. The special case of the Q-learning survivalist can be attributed to the fact that in some cases that persona opts to stop progressing when faced with monsters blocking its path, as is also evident from the low number of monsters killed. The evolved survivalist on the other hand proceeds to fight monsters when this is the only available course of action to reach the exit. Given the restricted nature of the play environment the Runner and Survivalist generally overlap in their performance statistics, as the safest and fastest paths to the exit typically deviate with only a few steps. The Treasure Collector personas unsurprisingly consistently exhibit the greatest collection of treasures as well as the greatest exploration of the game levels. The Q-learning trained Treasure Collector and the level specifically evolved Treasure Collector seem relatively comparable. Notably, the generalized, evolved Treasure Collector performs better than both of them and picks up more potions on its way through the level. The A* based baseline agents generally perform worse than their persona counterparts, but following comparative strategies, while the random controller has the largest exploration ratio of all personas and agents, since the controller just roams the map until it randomly reaches the exit, runs out of allocated testing actions, or dies. A set of indicative heatmaps from level 7 is included in Fig. 5, showing the varied behaviors of the personas and agents.

A. Agreements between Q-learning personas and evolved personas

As mentioned above, the evolved personas’ fitness functions are designed in an attempt to make them emulate the decision making styles expressed by the Q-learning trained personas. To test whether this is accomplished, the decision-level playtrace comparison method described in Sec. IV-D is applied between similar personas, across all 380 collected playtraces, using the human decisions as the baseline. This comparison

TABLE IV: Agreements between individual personas based on human players’ play traces.

Q-learning Persona	Evolved Persona	Agreement
q_E Exit	e_E Exit	0.71
q_M Monster Killer	e_M Monster Killer	0.69
q_R Runner	e_R Runner	0.72
q_S Survivalist	e_S Survivalist	0.63
q_T Treasure Collector	e_T Treasure Collector	0.70

TABLE V: Agreements between personas/baseline agents and human players. Evolved personas which were generalized by holding out the test level are marked with a Σ .

Q-learning Persona	Agreement	Evolved Persona	Agreement
q_E Exit	0.52	e_E Exit	0.55
q_R Runner	0.53	e_R Runner	0.56
q_S Survivalist	0.49	e_S Survivalist	0.57
q_M Monster Killer	0.54	e_M Monster Killer	0.59
q_T Treasure Collector	0.62	e_T Treasure Collector	0.71
Baseline Agent	Agreement	Generalized persona	Agreement
a_M Monster Killer	0.53	Σe_E Exit	0.55
a_R Runner	0.55	Σe_R Runner	0.55
$a_{R,s}$ Runner Safe	0.55	Σe_S Survivalist	0.56
a_T Treasure Collector	0.71	Σe_M Monster Killer	0.60
$a_{T,s}$ Treasure Collector Safe	0.70	Σe_T Treasure Collector	0.74
Z Random Controller	0.43		

is not made between Q-learning personas and generalized, evolved personas, since these would have to be tested on levels unseen to them, but seen to the Q-learning personas. For each step in each human play trace, the game is advanced to the game state from which the human play trace was collected. The game state is then input to the comparable pair of personas, and both are queried for their next action. If they report the same action, even if in disagreement with the human player, this is counted as an inter-persona agreement. If they report different actions, irregardless of the human choice, this is counted as an inter-persona disagreement. The results are presented in Table IV. As is evident, all types of personas seem to exhibit agreements at levels ranging from approximately 60% to 70%. Though better than random performance, this indicates that the weightings of utilities cannot be naively transferred from one method to the other.

B. Agreements between personas and human players

To ascertain to which extent the individual personas express actual human player decision making styles all personas (as well as baseline agents) are compared to the human play data, again using the decision-level playtrace comparison method. In this configuration, agent reports are only counted as persona-human agreements if they report exactly the same action that is present in the human play trace, in response to the game state. The results for each agent, averaged across all human play traces on all levels, are presented in Table V.

From Table V it is clear that the average action agreement ratio across all players seems limited, perhaps with the exception of the Treasure Collector personas and baseline agents. This is, however, to be expected, as each persona is should express a distinct decision style which should only be displayed by some human players. Therefore, averaging the agreements across all players on all levels obscures the human decision making style expressiveness of each persona. To

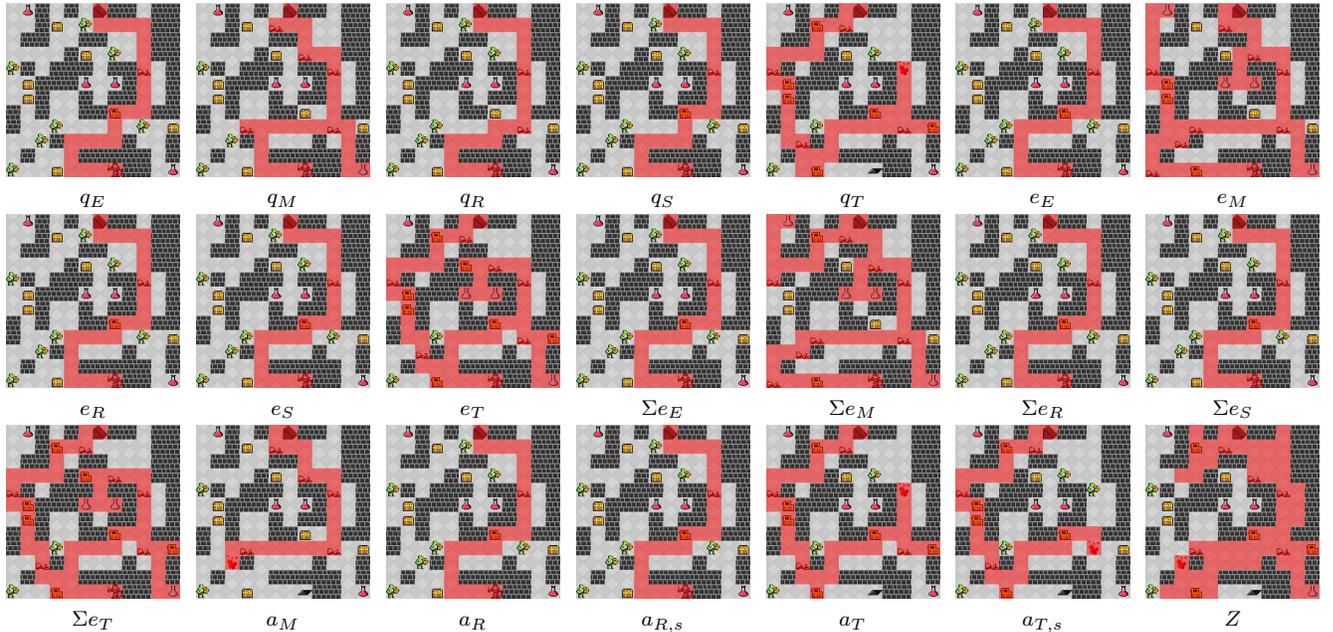


Fig. 5: Heatmaps of all personas and baseline agents on level 7 of MiniDungeons.

mitigate this problem, we instead map each human play trace to the persona which agrees the most with it, or put differently, the *best matching persona*. Table VI shows the frequency with which each persona or agent is the best match for the human players. The results indicate that treasure seeking personas dominate the covering of the human play traces, followed by monster killing, and finally the exit seeking strategies expressed; Survivalist and Runner personas only cover a few human play traces. The baseline agents also cover some human playtraces. This could be attributed to the limited decision space of the game and the fact that treasure collecting is a common culturally reinforced affordance for roguelike games; without further information about the players, however, this remains speculation. Two notable characteristics of the results, however, is that the evolved personas generally cover more of the human play trace sample than the other methods, and that the generalized, evolved agents tend to attain better coverage of the human play traces than the level specific ones. This could speculatively be attributed to the fact that the generalized evolved agents have learned from a broader range of examples.

While the frequency table provides insight into how much of the human sample each persona/agent covers, it is also relevant to investigate the quality of each persona/agent’s coverage. Table VII displays the mean agreement for each persona or agent. The mean is calculated over all human playtraces for which the persona or agent was the best match, for every level. From the results it is clear that the best matching personas agree with human players on between 60% and 88% of decisions, with a great deal of variation across levels. In order to better capture each persona’s agreement with humans, Table VIII lists the mean agreement values for each persona across all play traces on all levels. Generally, the personas display mean performances ranging from 0.7 to 0.8, which is comparable to the baseline agents in the relatively few cases that they attain a best match, but still leaves room

TABLE VI: Frequencies of personas and baseline agents being the best match for individual human play traces.

Persona or Agent	1	2	3	4	5	6	7	8	9	10	Total
q_E Exit	0	0	0	0	0	0	1	0	3	1	5
q_R Runner	0	0	2	0	0	1	1	0	0	0	4
q_S Survivalist	0	0	0	1	0	0	0	0	0	0	1
q_T Treasure Collector	1	2	1	1	4	1	2	1	0	2	15
e_E Exit	0	0	0	0	0	0	1	0	0	1	2
e_R Runner	0	0	0	0	0	0	2	0	0	0	2
e_S Survivalist	0	0	0	0	0	0	0	1	0	0	1
e_M Monster Killer	8	0	0	1	0	0	1	0	3	0	13
e_T Treasure Collector	0	15	19	13	10	12	1	11	7	3	91
Σe_M Monster Killer	0	0	0	1	1	0	6	1	0	0	9
Σe_T Treasure Collector	28	7	13	20	19	22	21	23	18	23	194
$a_{E,s}$ Exit Safe	0	1	0	0	0	0	0	0	0	0	1
a_T Treasure Collector	1	7	1	0	4	2	0	0	1	1	17
$a_{T,s}$ Treasure Collector Safe	0	6	2	1	0	0	2	1	6	7	25
Z Random Controller	0	0	0	0	0	0	0	0	0	0	0
Total	38	38	38	38	38	38	38	38	38	38	380

for improvement at matching human decision making styles.

Finally, Table IX compares the mean agreements for best matches between Q-learning and evolved personas designed to express the same decision making style. For most personas, not enough human play traces were matched for robust comparisons. The Treasure Collectors, however, show a borderline significant difference between q_T and e_T using Wilcoxon’s Rank Sum test (α -level 0.05). Comparisons are not made between baseline agents and personas, since the baseline agents are not based on utilities.

VI. DISCUSSION

It could be argued that the testbed game we employ is too simple and not representative of actual games. However, we argue that the game is similar in complexity to many of a new wave of roguelike games that have recently become popular on hand-held devices - the likes of *868-hack*, *Hoplite* and

TABLE VII: Mean agreement ratios for personas and baseline agents when they were selected as best matches.

Persona or Agent	1	2	3	4	5	6	7	8	9	10
q_E Exit								0.78	0.86	0.86
q_R Runner			0.70			0.78	0.74			
q_S Survivalist				0.60						
q_T Treasure Collector	0.65	0.71	0.77	0.78	0.79	0.62	0.66	0.74		0.85
e_E Exit								0.88		0.85
e_R Runner								0.72		
e_S Survivalist									0.82	
e_M Monster Killer	0.72			0.79			0.69		0.74	
e_T Treasure Collector		0.79	0.80	0.73	0.76	0.81	0.66	0.79	0.74	0.86
Σe_M Monster Killer				0.70	0.72		0.75	0.80		
Σe_T Treasure Collector	0.75	0.71	0.83	0.77	0.75	0.78	0.72	0.83	0.76	0.82
$a_{E,s}$ Exit Safe		0.71								
$a_{T,s}$ Treasure Collector Safe	0.69	0.76	0.86		0.73	0.77			0.72	0.87
$a_{T,s}$ Treasure Collector Safe		0.75	0.79	0.78			0.66	0.79	0.81	0.83

TABLE VIII: Mean agreement ratio across all traces from all levels where personas or agents were selected as best matches.

Persona or Agent	n	Mean	SD	Min	Max
q_E Exit	5	0.84	0.07	0.76	0.94
q_R Runner	4	0.73	0.06	0.64	0.78
q_S Survivalist	1	0.60		0.60	0.60
q_T Treasure Collector	15	0.74	0.07	0.62	0.85
e_E Exit	2	0.87	0.02	0.85	0.88
e_R Runner	2	0.72	0.16	0.61	0.83
e_S Survivalist	1	0.82		0.82	0.82
e_M Monster Killer	13	0.73	0.05	0.65	0.84
e_T Treasure Collector	91	0.78	0.07	0.56	0.95
Σe_M Monster Killer	9	0.75	0.06	0.64	0.82
Σe_T Treasure Collector	194	0.78	0.07	0.57	0.95
$a_{E,s}$ Exit Safe	1	0.71		0.71	0.71
$a_{T,s}$ Treasure Collector Safe	17	0.76	0.06	0.67	0.87
$a_{T,s}$ Treasure Collector Safe	25	0.79	0.07	0.63	0.90

Out There. Those games cannot be considered “toy problems” more than any other successful game. This is not to say that *MiniDungeons* is as entertaining or “deep” (in game design terms) as those games. Future work should include iterating over the *MiniDungeons* design to provide more satisfying gameplay, which will help us collect more and better player data.

VII. CONCLUSION

We have addressed the problem of creating procedural personas, which are generalized generative player models that represent the behavior of a class of players with particular playing styles or decision making styles. Based on an analysis of the affordances in a simple roguelike game, we identified five different reward structures, which were used in the training of personas. A persona representation was devised based on an evolvable perceptron that selects which immediate goal to pursue based on knowledge of internal state and distances to various level features. This evolutionary persona representation was compared with a previously devised method based on Q-learning, and it was found that the evolutionary solution is better both at agreeing with human players and optimizing the rewards, while also being generalizable to unseen levels. These models are well-suited for e.g. simulation-based testing in procedural content generation.

ACKNOWLEDGEMENTS

We thank the players of the game. The research is supported by the FP7 ICT project C2Learn (project no: 318480).

TABLE IX: Wilcoxon Rank Sum Test for differences between persona pairs, with personas created either via Q-learning or evolution.

Persona	n	Persona	n	W	p
q_E Exit	5	e_E Exit	2	6.0	0.86
q_R Runner	4	e_R Runner	2	4.0	1.00
q_S Survivalist	1	e_S Survivalist	1	1.0	1.00
q_M Monster Killer	0	q_M Monster Killer	13		
q_T Treasure Collector	15	e_T Treasure Collector	91	880.5	0.07
q_E Exit	5	Σe_E Exit	0		
q_R Runner	4	Σe_R Runner	0		
q_S Survivalist	1	Σe_S Survivalist	0		
q_M Monster Killer	0	Σe_M Monster Killer	9		
q_T Treasure Collector	15	Σe_T Treasure Collector	194	1801.0	0.13
e_E Exit	2	Σe_E Exit	0		
e_R Runner	2	Σe_R Runner	0		
e_S Survivalist	1	Σe_S Survivalist	0		
e_M Monster Killer	13	Σe_M Monster Killer	9	67.0	0.60
e_T Treasure Collector	91	Σe_T Treasure Collector	194	8553.5	0.67

REFERENCES

- [1] R. Caillois, *Man, play, and games*. University of Illinois Press, 2001.
- [2] A. Rollings and D. Morris, *Game architecture and design: a new edition*. New Riders Indianapolis, 2004.
- [3] C. Holmgård, J. Togelius, and G. N. Yannakakis, “Decision Making Styles as Deviation from Rational Action. A Super Mario Case Study.” in *Ninth Annual AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2013.
- [4] G. N. Yannakakis, A. Liapis, and C. Alexopoulos, “Mixed-initiative co-creativity,” in *Proceedings of the 9th Conference on the Foundations of Digital Games*, 2014.
- [5] A. Liapis, G. N. Yannakakis, and J. Togelius, “Generating map sketches for strategy games,” in *Proceedings of Applications of Evolutionary Computation*, vol. 7835, LNCS. Springer, 2013, pp. 264–273.
- [6] C. Holmgård, A. Liapis, J. Togelius, and G. N. Yannakakis, “Generative Agents for Player Decision Modeling in Games,” in *Foundations of Digital Games*, 2014.
- [7] D. Kahneman and A. Tversky, “Prospect theory: An analysis of decision under risk,” *Econometrica: Journal of the Econometric Society*, pp. 263–291, 1979.
- [8] G. Gigerenzer and W. Gaissmaier, “Heuristic Decision Making,” *Annual Review of Psychology*, vol. 62, pp. 451–482, 2011.
- [9] A. Rubinstein, *Modeling Bounded Rationality*. MIT Press, 1998, vol. 1.
- [10] N. Yee, “Motivations for play in online games,” *CyberPsychology & Behavior*, vol. 9, no. 6, pp. 772–775, 2006.
- [11] A. Canossa, “Give me a reason to dig: qualitative associations between player behavior in minecraft and life motives,” in *Proceedings of the International Conference on the Foundations of Digital Games*. ACM, 2012, pp. 282–283.
- [12] J. Gibson, “The Concept of Affordances,” *Perceiving, Acting, and Knowing*, pp. 67–82, 1977.
- [13] A. M. Smith, C. Lewis, K. Hullett, G. Smith, and A. Sullivan, “An Inclusive Taxonomy of Player Modeling,” *University of California, Santa Cruz, Tech. Rep. UCSC-SOE-11-13*, 2011.
- [14] G. N. Yannakakis, P. Spronck, D. Loiacono, and E. André, “Player Modeling,” in *Artificial and Computational Intelligence in Games*. Saarbrücken/Wadern: Dagstuhl Publishing, 2013, pp. 45–55.
- [15] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, “Search-Based Procedural Content Generation: A Taxonomy and Survey,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 3, pp. 172–186, 2011.
- [16] A. Liapis, G. N. Yannakakis, and J. Togelius, “Designer modeling for personalized game content creation tools,” in *Proceedings of the AIIDE Workshop on Artificial Intelligence & Game Aesthetics*, 2013.
- [17] —, “Sentient sketchbook: Computer-aided game level authoring,” in *Proceedings of the 8th Conference on the Foundations of Digital Games*, 2013, pp. 213–220.