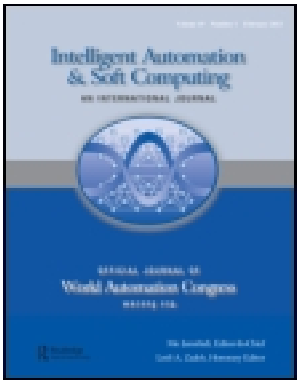

Intelligent Automation & Soft Computing
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tasj20

Evolutionary Search For Entertainment In Computer Games

Zahid Halim [a], A. Rauf Baig [b] & Mujtaba Hasan [b]

[a] Faculty of Computer Science and Engineering, Ghulam Ishaq Khan Institute of Engineering Sciences and Technology (GIKI), Topi, Pakistan

[b] FAST National University of Computer and Emerging Sciences Islamabad, Pakistan, 07360

To cite this article: Zahid Halim, A. Rauf Baig & Mujtaba Hasan (2012): Evolutionary Search For Entertainment In Computer Games, Intelligent Automation & Soft Computing, 18:1, 33-47

To link to this article: http://dx.doi.org/10.1080/10798587.2012.10643225

Intelligent Automation and Soft Computing, Vol. 18, No. 1, pp. 33-47, 2012

EVOLUTIONARY SEARCH FOR ENTERTAINMENT IN COMPUTER GAMES

ZAHID HALIM, A. RAUF BAIG*, AND HASAN MUJTABA*
Faculty of Computer Science and Engineering, Ghulam Ishaq Khan Institute of Engineering Sciences and Technology (GIKI), Topi, Pakistan
zahid.halim@giki.edu.pk

**FAST National University of Computer and Emerging Sciences Islamabad, Pakistan*
rauf.baig@nu.edu.pk, and hasan.mujtaba@nu.edu.pk

ABSTRACT—Games have always been of interest to all age groups. With the advancement in technology and increase in number of users of personal computers, increased number of games is introduced in market. This is resulting in efforts, both for the developers in writing scripts for games and for the end users to select a game which is more entertaining. In this work we present a solution to both the issues. Initially a quantitative measure is devised, which calculates the entertainment value of games. Based upon the proposed measure we use evolutionary algorithm to generate games for different genres on the fly. The evolutionary algorithm needs to be given an initial set of games which it optimizes for entertainment using the proposed entertainment measure as the fitness criteria. In order to compare the entertainment value of the new games generated with the human's entertainment value we conduct a human user survey.

Key words: Measuring entertainment, Automatic Game Creation, Computational Intelligence and Games, Evolutionary Algorithms

1. INTRODUCTION

With the advancement in technology and decrease in prices of electronic items, Personal Computers (PCs) are becoming common in all walks of life. This has resulted in PCs replacing many other electronic gadgets like televisions, radios and many others, as people are inclined to use these in their PCs through software. Games are not an exception from this electronic advancement. The number of games for PCs has increased today as it was some 20 years back [11]. There are many new genre of games introduced like predator/prey, real time strategy and platformer to name a few. All this has increased number of choice in computer games for the users, at the same time the quality of entertainment provided by these games has also decreased due to abundance of games in the market for PCs. On the other hand the task of game development for the developers is becoming tiresome, which requires scripting the game, modeling its contents and other such activities. Still after lots of efforts, by the developers, they cannot know how much the developed game is entertaining for the end users. Reason being, the fact that, entertainment is a subjective term. What might be entertaining for one user may not be entertaining for others. Another issue from the point of view of game developers in this context is the constant need of writing new games, requiring investment both in terms of time and resources.

In order to address afore mentioned challenges, to the computer gaming industry, the first task can be to device some metrics that can quantify the entertainment value that a game carries. Although creating a single quantitative measure for all genres of games is not trivial, still a separate metrics for each genre of game can be devised. Based upon the entertainment metrics some computational intelligence technique can be used to create new and entertaining games automatically.

1.1 Our Contribution

In this work we create two sets of metrics to measure the entertainment value of the two different genre of games, which include a) board based games and b) predator/prey type of games. The metrics devised are based on different theories of entertainment in specifically related to computer games, taken from literature. Further we use Evolutionary Algorithm (EA) to generate new and entertaining games using the proposed entertainment metrics as fitness function. The EA starts with the randomly initialized set of population and using genetic operators (guided by the proposed entertainment metrics) it reaches a final set of population that is optimized against entertainment. For the purpose of verifying the entertainment value of the evolved games with that of the human we conduct a human user survey.

1.2 Paper Organization

Rest of the paper is arranged as follows: Section 2 covers the previous work done in context of measuring entertainment and automatic generation of games, Section 3 lists the entertainment theories in computer games that serves as a basis of our entertainment metrics, Section 4 covers the search space and the fitness function (entertainment metrics) for both board based and predator/prey genre of games. Section 5 is dedicated to the experimentations and results which cover EA's setup, evolved games analysis and user survey and Section 6 concludes the paper.

2. LITERATURE SURVEY

Iida in [1] has proposed a measure of entertainment for games which he used to analyze the evolution of game of chess. His measure is the pioneer in quantification of entertainment. Iida's work is limited to chess variants but it can be applied to other board games. According to him, the entertainment value of a game is equal to the length of the game divided by the average number of moves taken by a player. The game tends to be entertaining if the value of this measure decreases. In [2] the authors introduce the uncertainty of game outcome as a metric of entertainment. If the outcome is known at an early stage then there is not much interest in playing it. Similarly if it is found at the last move then it is probably probabilistic. The outcome should be unknown for a large duration of the game and should become known in the last few moves of the game. Authors state that it is easy to create new board games and variants of classical games but to make them attractive to the human user, is challenging. In [2] a simple technique based on synchronism and stochastic elements is used to refine the game of Hex. Authors proof that the game's attraction has increased by conducting experiments to show an increment of the outcome uncertainty. In [3] Symeon uses board games for e-learning. He proposed an e-learning board game that adopts the basic elements of a racing board game but cultivate in students skills like creativity, problem-solving, and imagination, as students are trying to reach the end by improving their performance in a variety of learning activities. In the work done in [3] the issues of measuring entertainment value of the games and its automatic generation of game contents are not addressed.

Togelius [4] has evolved entertaining car racing tracks. The tracks were represented as b-splines and the fitness is evaluated using the performance of a neural network based controller on the track. The objectives were for the car to have made maximum progress in a limited number of time steps, high maximum speed, and high variability in performance between trials. The game model used for experimentations in [4] is simple both graphically and physically (being 2D). In [5] three metrics have been proposed for measuring the entertainment value of predator/prey games. The first metric is called appropriate level of challenge (T), which is calculated as the difference between the maximum of a player's lifetime and his average lifetime over N games. The second metric is behavior diversity metric (S). It is standard deviation of a player's lifetime over N games. The third metric is spatial diversity metric E {Hn}. It is the average entropy of grid-cell visits by the opponents over N games.

In [6] the author identifies how the Super Mario game works. It defines the rules and highlights the goals of the game that the user must reach. More importantly, it shows how the content is generated in the game. According to work of Nicole in [7], people play games to change or structure their internal experiences. Adults in this study, enjoy filling their heads with thoughts and emotions unrelated to work or school, others enjoy the challenge and chance to test their abilities. Games offer an efficiency and order in playing that they want in life. Chris in [8] modified the level generator to create new level on the bases of four parameters three of which deals with the performing different operations with holes and last parameter deals with the direction of movement of Mario. In [8] several statistical features are noted during the playing of game these include completion time, time spent on various tasks (e.g. jumping), killed enemies (e.g. way of killing) and information on how the player died. Chris used neuroevolutionary preference learning of simple of non linear perceptron to predict certain player emotions from game play features.

In [9] Kate believes that level designs in platform games rely on rhythm. When player is in rhythm of game making jumps requires not only correct distance calculation but also timing. When obstacles are placed in rhythm they make the movement of player rhythmic hence making each jump easier. In Mario by shuffling some of the elements like holes, pipes, blocks designer can construct long and interesting level.

The work done by Yannakakis in [10] an approach for capturing and modeling individual entertainment preferences is applied Playware[11] platform. The aim is to construct, using representative statistics computed from children's physiological signals, an estimator of the degree to which games provided by the playground engage the players. For this purpose child's heart rate (HR) signals, and their expressed preferences of how much "fun" particular game variants are, are obtained from experiments using games implemented on the Playware playground. Previously work has been done on some force sensing tiles to form an environment called "Smart Floor" [12]. The system was developed for user identification and tracking based on the features derived from the force of a person's footsteps. In [12] authors created user footstep models using a set of footstep profile features and have been able to achieve a recognition rate of 93%. However, it was not meant for any entertainment purpose. There is some work done in [19-21] somewhat related to the one already mentioned above.

3. THEORIES OF ENTERTAINMENT

Rauterberg [13, 14] in his work as introduced the concept of incongruity as a measure of interest in a task. Given any task, humans make a mental model about its difficulty. Incongruity is defined to be the difference between the actual complexity of the task and the mental model of the complexity that a person has. Congruity is positive for positive difference and negative for negative difference. For negative incongruity the task at hand is considered to be easy. Entertainment of a task is highest when the incongruity is neither too positive nor negative. In case of large positive incongruity the humans have a tendency to avoid the task being it very hard. Thus requirement of right amount of incongruity is same as right amount of challenge, as defined by Csikszentmihalyi's theory. According to Csikszentmihalyi's theory of flow [15, 16], the optimal experience for a person is when he is in a state of flow. In this state the person is fully concentrated on the task that he is performing and has a sense of full control. The state of flow can only be reached if the task is neither too easy nor too hard. In other words the task should pose the right amount of challenge.

In addition to the right amount of challenge, Malone [18] proposes two more factors that make games engaging: fantasy and curiosity. If a game has the capability of evoking the player's fantasy and makes him feel that he is somewhere else or doing something exotic then that game is more enjoyable than a game which does not do so. Curiosity refers to the game environment. The game environment should have the right amount of informational complexity: novel but not incomprehensible.

Koster's theory of fun [17] states that the main source of enjoyment while playing a game is the act of mastering it. If a game is such that it is mastered easily and the player does not learn anything new while playing then the enjoyment value of that game is low.

4. SEARCH SPACE AND ENTERTAINMENT METRICS

For the purpose of generating new games we need to define a search space that will be used by the evolutionary algorithm to evolve new games, for these games to be entertaining the evolutionary algorithm will be guided by a fitness function, which will be our proposed entertainment metrics. As we are addressing two different genres of games we need to have separate search space and fitness functions for both.

4.1 Board Based Games

For defining the search space for board based games we use the search space of the popular board games of chess and checkers as a super set. Figure 1 summarizes the search space, whose details follow.

Search Space Dimension	Values
Play Area	Both white & black squares are used
Types of Pieces	6
Number of pieces/type	variable but at maximum 24
Initial position	First 3 rows & Both white & black
Movement direction	All directions, straight forward, straight forward and backward, L shaped, diagonal forward
Step Size	One Step, Multiple Steps
Capturing Logic	Step over, step into
Game ending logic	No moves, no king
Conversion Logic	Depends upon rules of the game
Mandatory killed	Depends upon rules of the game
Turn passing allowed	No

Figure 1. Board based game search space dimensions.

1. The size of play area in our search space is a grid of 8x8 squares, alternating white and black, all squares can be used.
2. Combining the rule space of chess and checkers we have total six types of pieces in our search space. Each type can have a minimum of 0 and a maximum of 16 pieces. However, total pieces should not be zero nor exceed 16. The initial positions of the pieces

are the nearest three rows of a player. A cell can have one piece of type 0 to 6, type 0 means no piece is present in that cell.
3. The search space to evolve new games consists of only those six movement logics as in both chess and checkers. Which include diagonal forward, diagonal forward and backward, movement in all directions, L shaped movement, straight forward and straight forward and backward.
4. The step size for a piece can either be one or up to an occupied cell.
5. Capturing is done by jumping over or moving into the opponent's cell. The result of capturing is death of captured piece.
6. The game ends if there are no more pieces left of a specific type. We call this type piece of honor. There can be zero or one piece type declared to be a piece of honor. A game ends if a piece of honor of any player is dead or the player without moves is the loser.
7. A game can have a maximum of 100 moves.
8. A piece may or may not convert to another type after reaching last row. Evolution decides which type is convertible. Each piece has a conversion logic which decides which type it will convert to when last row is reached.
9. Turn passing is not allowed.

4.2 Predator/Prey Games

The predator/prey genre consists of one or more predators, predators may be homogeneous or heterogeneous in their behavior, obstacles (which may or may not be for both predator and prey) and some objective for the prey to achieve. Pac Man [22] is a very popular game of this genre. Keeping the above constrained in mind and inspired by its closeness to the rule space defined by J. Togelius in his work [23], we have defined our rule space as follows.

1. Play area consists of 20 X 20 cells.
2. There are *N* predators of type *M*; each type is represented by a different color. We have selected *N* to be 0-20 and *M* in range 0-3, where colors are red, green and blue.
3. Each type of predator moves around the play area according to any of the following three schemes. Still, turn clockwise upon encountering an obstacle and turn counter clockwise upon encountering an obstacle.
4. The predators may collide with each other and the prey. As different predators may have different behavior so the response to collision needs to be different for each type. The possible types of responses to a collision are as: death of the prey and/or predator, random change in current location of the prey and/or predator, no effect on the prey and/or predator.
5. The score is calculated for the prey only, which is one of +1, 0, or -1 upon collision with a predator. Decrease in prey's score due to collision between predators shows the uncertain nature of the game which adds some level of entertainment in the game as well.
6. The time for which the game will be played vary from 1-100 time steps. And the maximum score that a prey can achieve vary from 1-2000.
7. The game will stop if any of the following is true: the time exceeds its maximum limit, prey has died, or the prey score exceeds the maximum score.

Figure 2 displays a typical environment of one such game created based on the rule space defined. The yellow is the prey and remaining are the predators of different types.

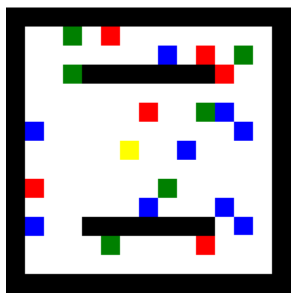

Figure 2. The play area of the predator-prey game.

4.3 Fitness Function (Entertainment Metrics) for Board Based Games

Our proposed fitness metrics consists of four components a) duration of the game, b) intelligence, c) dynamics and d) usability. These are explained below.

I. Duration of the game: In general, a game should not be too short or too long, as both are uninteresting. The duration of play (D) of a game is calculated by playing the game n times and taking the average number of moves over these n games. The average value of D is taken because if the game is played multiple times with a different strategy each time then we do not get the same value of D every time. For averaging, the game is played n = 20 times in our experiments. Equation (1) shows the mathematical representation of D.

$$D = \frac{\sum_{k=1}^{n} L_K}{n} \tag{1}$$

where L_K is the life of the game playing agent in game K. In order to reward games neither too short nor too long value of raw D is scaled in range 0-1. The boundaries for scaled value of D are shown in figure 3.

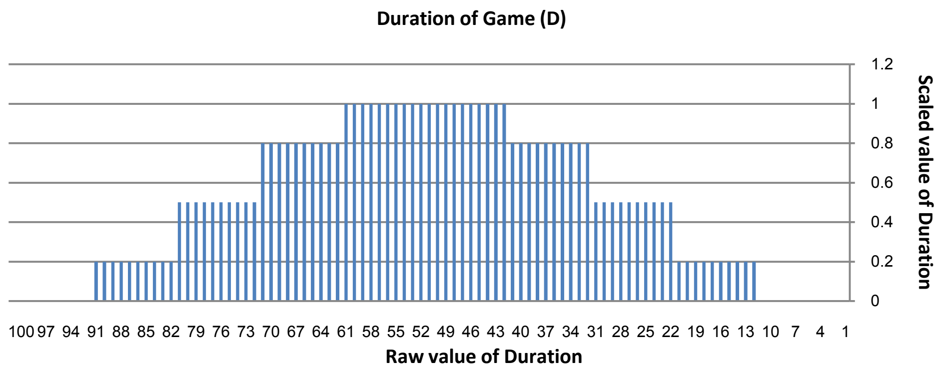

Figure 3. Scaling ranges for raw value for duration of game.

II. Intelligence for playing the game: A game is interesting if the rules of the game are such that the player having more intelligence should be able to win. The intelligence (I) is defined as the number of wins of an intelligent controller over a controller making random (but legal) moves. For this purpose the game is played *n* times (*n* = 20 times in our experiments). Higher number of wins against the random controller means that the game requires intelligence to be played and does not have too many frustrating dead ends. Intelligence I is calculated using equation (2).

$$I = \frac{\sum_{k=1}^{n} I_K}{n} \tag{2}$$

where, I_K is 1 if intelligent controller wins the game otherwise its 0.

III. Dynamism: This aspect assumes that a game whose rules encourage greater dynamism of movement in its pieces would be more entertaining than a game in which many pieces remain stuck in their cells for the entire duration of the game. The dynamism is captured by the following fitness function given in equation (3).

$$Dyn = \frac{\sum_{i=1}^{n}\left(\frac{\frac{\sum_{i=1}^{m}(c_i)}{Li}}{m}\right)}{n} \tag{3}$$

where,

C_i is the Number of cell changes made by piece i during a game
L_i is life of the piece i
And m is the total number of pieces specified in a chromosome.
The dynamism is averaged by calculating it for 20 games for the same chromosome. This fitness function has a higher value if the pieces show a more dynamic behavior.
IV. Usability: It is interesting to have the play area maximally utilized. If most of the moving pieces remain in a certain region of the play area then the resulting game may seem strange. The usability is captured using equation (4).

$$U = \frac{\sum_{i=1}^{n}\left(\frac{\sum_{k=1}^{m}(C_k)}{|C_u|}\right)}{n} \tag{4}$$

where,
C_k is usability counter value for a cell k.
$|C_u|$ is the total number of usable cells.
n is 20 as explained previously.
V. Final fitness: The above four metrics are combined as: all chromosomes in a population are evaluated separately according to each fitness functions. Then the population is sorted on duration and a rank based fitness is assigned to each chromosome. The best chromosome of the sorted population is assigned the highest fitness i.e. 20 because we have pool of 10 parents and 10 offspring; the second best chromosome is assigned the second best fitness i.e. 19, and so on. The population is sorted again in the same manner for rest of the three metrics. The four rank based fitness values obtained for each chromosome are multiplied by corresponding weights and then added to get its final fitness.

$$FF = aD + bI + cDyn + dU \tag{5}$$

where a, b, c, and d are constants. We fix the values of weights to 1. The calculation of rank based fitness gets rid of the problem of one factor having higher possible value than another factor.

4.4 Fitness Function (Entertainment Metrics) for Predator/Prey Games

For the predator/prey genre of games fitness function proposed consist of following four aspects: Duration, appropriate level of challenge, diversity and usability. Some of these are same as those for board based games but as the genre of game is different we will need different equations to measure these.
I. Duration of the game: This is same as mentioned above for board based games.
II. Appropriate level of challenge: The game is interesting if the rules of the game are such that the player has to display intelligence to obtain a reasonable score. The challenge C is converted into a fitness function using equation 6:

$$c = e^{\left(\frac{-|S_m - S_a|}{S_m}\right)} \tag{6}$$

where,

$$S_a = \frac{\sum_{k=1}^{n} S_K}{n}$$

S_K is score of the agent in game Kth time it plays a game.
n is 20 as explained previously.
Since the value of S_a can also be negative, hence we use the following processing:

$$S_a = \begin{cases} S_a, & \text{if } S_a \geq 0 \\ |S_a| + 20, & \text{otherwise.} \end{cases} \tag{7}$$

III. Diversity: The behavior of the moving artifacts of the game should be sufficiently diverse so that it cannot be easily predicted.

$$\mathrm{Div} = \frac{\sum_{i=1}^{n}\left(\sum_{k=1}^{m}(\partial_k)\right)}{n} \tag{8}$$

where,
m is the total number of pieces (all three types) specified in a chromosome.
∂_k Number of cell changes made by piece k during a game.
n is 20 as explained previously.
IV. Usability: It is interesting to have the play area maximally utilized during the game. If most of the moving artifacts remain in a certain region of the play area then the resulting game may seem strange. Usability is captured by equation (9).

$$U = \frac{\displaystyle\sum_{i=1}^{n}\left(\frac{\sum_{k=1}^{m}(C_k)}{|C_u|}\right)}{n} \tag{9}$$

where,
C_k usability counter value for a cell k.
$|C_u|$ is the total number of usable cells.
n is 20 as explained previously.
5. Final fitness: The method of calculating the final fitness is same as that of board based games. Equation 10 shows the final fitness.

$$FF = aD + bC + cDiv + dU \tag{10}$$

where, *a, b, c,* and *d* are constants and are set to 1 in our experiments.

4.5 Chromosome Encoding

Each individual in the GA population will be representing one complete set of rules for the game. The genes of the chromosome will represent one rule of the game. Below we explain the chromosome encoding for each genre of games separately.
I. Board based games: Based upon the search space defined above the structure of the chromosome we have used is listed in figure 4. The chromosome consists of a total 50 genes. First 24 genes may contain values from 0 to 6 were 1 represents a piece of type 1, 2 for piece of type 2 and so on. 0 is interpreted as no piece. The piece type represented by gene 1 is placed at the cell 1 of the game board; piece type represented by gene 2 is placed at the cell 2 of the game board and so on.

Gene	Title	Value
1		
:	Placement of gene of each type	0-6
24		
25		
:	Movement logic of each type	1-6
30		
31-36	Step Size	0/1
37		
:	Capturing logic move into cell or jump over 0/1	0/1
42		
43	Piece of honor	0-6
44		
:	Conversion Logic 0-6	0-6
49		
50	Mandatory to capture or not	0/1

Figure 4. Structure of the chromosome for board based games

Gene 25 to 30 represents movement logic for each piece type respectively, where 1 is for diagonal forward, 2 for diagonal forward and backward, 3 for all directions, 4 for L shaped movement, 5 for straight forward and backward and 6 for straight forward. Genes 31 to 36 are used for step size of each type, where 0 is used to indicate single step size and 1 for multiple step size. Genes 37 to 42 are used for capturing logic of each type, where 0 is used to indicate step into and 1 for step over. Gene number 43 indicates the type of piece that will be the piece of honor, possible values include 0-6, where 1-6 indicate the piece type and 0 represents that there is no piece of honor in the game. Genes 44-49 represents the conversion logic, of piece type 1 to 6 respectively, when they reach the last row of the game board. Where 0 represents the piece will not be converted to any type and 1-6 represents the types of pieces. The last gene represents if it is mandatory in the game to capture the opponent piece in case it could be, 0 represents no and 1 represents yes.

II. Predator/prey games: We have 30 genes in a chromosome. The rule/feature of the game they represent and the allele are as follows: (1) number of red type predators (2) number of green type predators (3) number of blue type predators (4) movement logic of red type predators (5) movement logic of green type predators (6) movement logic of blue type predators (7) collision logic of red with red (8) collision logic of red with green (9) collision logic of red with blue (10) collision logic of red with agent (11) collision logic of green with red (12) collision logic of green with green (13) collision logic of green with blue (14) collision logic of green with agent (15) collision logic of blue with red (16) collision logic of blue with green (17) collision logic of blue with blue (18) collision logic of blue with agent (19) collision logic of agent with red (20) collision logic of agent with green (21) collision logic of agent with blue (22) score logic of red with red (23) score logic of green with green (24) score logic of blue with blue (25) score logic of agent with red (26) score logic of agent with green (27) score logic of agent with blue (28) score logic of green with red (29) score logic of blue with red (30) score logic of green with blue.

The allele for gene 1-3 is 0-20, for gene 4-6 its 0-4 where they represent still, clockwise, counter-clockwise, random short and random long movement respectively. For gene 7-21 allele is 0-2 where 0 means nothing happens against collision, 1 means the predator or prey will die and 2 means the predator or pray will be moved to a new randomly chosen location. Allele for gene 22-30 is between -1 to +1. The chromosome encoding is shown in figure 5.

Number of predators	Red	0-20	Collision logic	Blue-Green	0-2
	Green	0-20		Blue-Blue	0-2
	Blue	0-20		Blue-Agent	0-2
Movement logic	Red	0-4		Agent-Red	0-2
	Green	0-4		Agent-Green	0-2
	Blue	0-4		Agent-Blue	0-2
Collision logic	Red- Red	0-2	Score logic	Red- Red	-1 - +1
	Red- Green	0-2		Green-Green	-1 - +1
	Red-Blue	0-2		Blue-Blue	-1 - +1
	Red- Agent	0-2		Agent-Red	-1 - +1
	Green-Red	0-2		Agent Green	-1 - +1
	Green-Green	0-2		Agent-Blue	-1 - +1
	Green-Blue	0-2		Green-Red	-1 - +1
	Green-Agent	0-2		Blue-Red	-1 - +1
	Blue-Red	0-2		Blue-Green	-1 - +1

Figure 5. Chromosome encoding for predator/prey game.

5. EXPERIMENTATIONS AND RESULTS

5.1 Setup

For both genres of games we use a population of 10 chromosomes, randomly initialized, with legal values. In each generation one offspring is created for each chromosome by duplicating it and then mutating any one of its gene. All genes have equal probability of being selected. The mutation is done by replacing the existing value with some other permissible value. All permissible values have equal probability of being selected. We have kept different policy of promotion to next generation for each genre: for the predator/prey genre of games the parents and offspring form a pool of 20 chromosomes from which 10 best are selected for the next generation. Whereas for the board based games we compare the fitness of parent and its child using equation (11). If the result is greater than 4, which means that the child is half times stronger that the parent, we promote the child to next population otherwise we continue with the parent.

$$\text{Fitness Difference} = \sum\nolimits_{\text{for all metrics}} \left(1 - \frac{\text{fitness}_\text{p} - \text{fitness}_\text{c}}{\text{fitness}_\text{p}}\right) \tag{11}$$

where,

fitness_p is the fitness value of parent for current metrics

fitness_c is the fitness value of child for current metrics

This evolutionary process is continued for 100 generations. The fitness of a chromosome, in our case, is based on data obtained by playing the game according to the rules encoded in the chromosome. This is achieved using software agents. We have a total of three agents 1 for predator/prey genre of games and two for the board based games. The software agent used for calculation of fitness of predator/prey games is a rule based controller. The rules of the controller are listed in Figure 6. According to the game rules, at each simulation step the agent must take exactly one step. The agent looks up, down, left and right. It notes the nearest artifact (if any) in each of the four directions, and then it simply moves one step towards the nearest score increasing artifact.

For the purpose of evaluating the fitness values of board based games we need two controllers (a) a random controller, the random game playing agent plays the game by randomly selecting a legal move at each step. The agent follows the following algorithm listed in figure 7 and (b) an intelligent controller we have implemented this as it generates all the possible one ply depth game boards using a min-max algorithm. Each of the resulting game board is evaluated using a rule based evaluation function and the one with the highest evaluation is selected as a next move. Figure 8 lists the algorithm for the evaluation function we use.

1. *Move in the direction which is completely empty (there is only the wall at the end). If more than one directions are empty move towards the farthest wall (in the hope that subsequent position changes would show it a score increasing artifact)*
2. *Move in the direction which contains a score neutral artifact. The farthest, the better.*
3. *Move in the direction which contains a score decreasing artifact. The farthest, the better.*
4. *Move in the direction which contains a death causing artifact. The farthest, the better.*

Figure 6. Rules of the agent for predator/prey games

```
1.  Generate all legal moves
2.  Store the moves in a queue
3.  Shuffle the queue
4.  If Not mandatory to kill
5.      Randomly select a move from the queue.
6.  Else
7.      Select a move that captures an opponent's piece, if such move exists
8.      Otherwise, randomly select a move from the queue.
9.  Return next move to take
```

Figure 7. Algorithm for the random playing agent

```
1.  For each piece
2.      priority=0
3.  For each piece
4.      if is piece of honor
5.          priority = priority +1 000
6.      if movement logic all directions
7.          priority = priority + 8
8.      if movement logic diagonal Forward and Backward
9.          priority = priority + 7
10.     if movement logic Straight Forward and Backward
11.         priority = priority + 7
12.     if movement logic diagonal Forward
13.         priority = priority + 6
14.     if movement logic Straight Forward
15.         priority = priority + 6
16.     if movement logic L shaped
17.         priority = priority + 5
18.     if capturing logic step into
19.         priority = priority + 4
20.     if capturing logic step over
21.         priority = priority + 3
22. Count the number of pieces of Player A
23. Multiply the number of pieces of a type with its relevant priority
24. Count the number of pieces of Player B
25. Multiply the number of pieces of a type with its relevant priority
26. Calculate boardValue = WeightSumofA-WeightSumofB
27. Check if the Piece of Honour is dead add -1000 to boardValue
28. Check if the Piece of Honour is NOT dead add +1000 to boardValue
29. return boardValue
```

Figure 8. Algorithm for evaluation of board positions

The evaluation function assigns priorities (weights) to piece-type according to whether its disappearance would cause the game to end, flexibility of movement (more directions and multiple step sizes are better), and capturing logic (capturing by moving into opponent's cell is better). Once the priority of a piece is calculated we multiply each piece with its corresponding weight and calculate weighted summation for self and opponent. The board evaluation is the self weighted summation minus opponents weighted summation.

For the purpose of selecting the best evolved games, only in case of board based games, we keep and archive of four slots. These are used to place the best chromosomes based on each of the four fitness criteria. For the predator/prey games the final set of population serves as the best evolved population. Archiving is used for board based games because we are using Evolutionary Strategy (ES), 1+1 ES, in that case.

5.2 Evolved games

In this section we list the final set of evolved games for both the genres. To keep the decision brief we take the best games (according to fitness rank) for the predator prey genre of games, from the last population, whereas from the board based games we calculate the diversity, based on the games in archive, on each of the four metrics for all pair of games using equation (12).

$$Game\ diversity\ = \frac{Game\ Column\ Fitness - Game\ Row\ Fitness}{Selected\ Metrics\ Maximum\ Value} \tag{12}$$

Based upon the above criteria the rules of the evolved predator/prey and board based game are listed in figure 9 and 10 respectively.

Number of predators	Red	11	Collision logic	Blue-Green	1
	Green	20		Blue-Blue	1
	Blue	4		Blue-Agent	2
Movement logic	Red	2		Agent-Red	0
	Green	4		Agent-Green	0
	Blue	2		Agent-Blue	2
Collision logic	Red-Red	2	Score logic	Red-Red	0
	Red-Green	2		Green-Green	1
	Red-Blue	0		Blue-Blue	1
	Red-Agent	2		Agent-Red	-1
	Green-Red	2		Agent Green	1
	Green-Green	2		Agent-Blue	0
	Green-Blue	0		Green-Red	-1
	Green-Agent	1		Blue-Red	1
	Blue-Red	1		Blue-Green	1

Figure 9. Rules of the evolved predator/prey game.

Piece No	Movement Logic	Step Size	Capturing Logic	Conversion Logic
1	L	Multiple	Step Into	6
2	Diagonal Forward & Backward	Single	Step Over	5
3	All Directions	Multiple	Step Into	Nil
4	Straight Forward	Multiple	Step Into	1
5	Straight Forward	Multiple	Step Over	2
6	All Directions	Multiple	Step Over	3
	Piece of Honor	5		
	Mandatory to Capture	No		

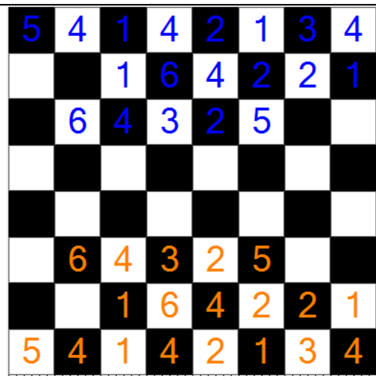

Figure 10. Rules of the evolved board game.

5.3 User Survey

To validate the results produced against human entertainment value we performed a human user survey on 20 subjects, chosen such that they have at least some aptitude towards playing computer games. We provided the users with a total of four games to be played. Two were the evolved games listed in the previous section and one each a random initialed game of both genres. Each subject was asked to play a game 2 times to reduce noise (thus making a total of 40 games being played); the rules of the game were already explained to the subjects and also displayed on the software they used. Subjects were asked to rank the game they play as 1- liked, 2-disliked and 3-neutral. Table I summarizes the results of the user survey. For visual purposes we use 1 for liked, -1 for disliked and 0 for neutral. The games given to the users were numbered as, game 1 was an evolved predator/prey game, game 2 was random predator/prey game, game 3 was an evolved board based game and game 4 was a random board based game. Subjects were unaware of which were evolved games.

Table I. results of the user survey

	Game 1		Game 2		Game 3		Game 4	
Subject	Run 1	Run 2	Run 1	Run 2	Run 1	Run 2	Run 1	Run 2
1	0	1	0	-1	0	1	-1	-1
2	1	1	-1	-1	1	1	-1	-1
3	0	1	0	0	0	1	0	-1
4	1	1	-1	-1	1	1	-1	-1
5	0	0	0	-1	0	1	-1	0
6	1	1	-1	-1	1	1	-1	-1
7	0	0	0	-1	1	1	0	-1
8	1	-1	0	1	-1	0	-1	-1
9	0	1	-1	-1	1	1	-1	-1
10	1	1	1	1	1	1	0	1
11	1	1	0	0	1	1	-1	-1
12	1	1	0	-1	1	1	0	-1
13	1	1	-1	-1	1	1	0	-1
14	0	1	-1	-1	0	0	-1	-1
15	0	1	0	-1	0	1	0	-1
16	1	1	-1	-1	0	1	-1	-1
17	0	0	0	-1	0	1	0	-1
18	0	1	0	0	-1	1	-1	-1
19	1	1	1	1	0	0	-1	0
20	0	-1	0	0	0	-1	0	0

The results of Table I suggest that on average 62% users have liked the evolved predator/prey game as against 12% liked the random one. In case of the board based game the evolved games are liked by 62.5% subjects and only 2.5% liked the randomly initialed game.

6. CONCLUSION AND FUTURE WORK

The idea of evolution of game rules to produce new and entertaining games seems to be very promising. In this work we have presented an experiment for evolution of computer games of two different genres of games. In the process we have presented some metrics for entertainment which are based on duration of game, level of challenge, diversity, intelligence, and usability. These metrics are combined in a fitness function to guide the search for evolving rules of the game. The result of our experiment is that games can be evolved in this manner. Even though we cannot claim that the best evolved game is the most entertaining of all, but still we are able to evolve a suite of equally interesting games which are much better than randomly generated games. Further work needs to be done to make the proposed entertainment metrics more generic so they can automatically cover more types of computer games. Another research direction could be validation of the results produced.

REFERENCES

1. H. Iida, N. Takeshita, and J. Yoshimura, “A Metric for Entertainment of Board Games: Its Application for Evolution of Chess Variants”, In Nakatsu, R., and Hoshino, J., eds., Entertainment Computing: Technologies and Applications (Proceedings of IWEC 2002), pages 65-72. Boston, MA: Kluwer Academic Publishers. 2003.
2. A. Cincotti and H. Iida, “Outcome Uncertainty And Interestedness In Game-Playing: A Case Study Using Synchronized Hex”, In New Mathematics and Natural Computation (NMNC), vol. 02, issue 02, pages 173-181, 2006.
3. S Retalis, “Creating Adaptive e-Learning Board Games for School Settings Using the ELG Environment”, J. UCS 14(17): 2897-2908, 2008.
4. J. Togelius, R. D. Nardi, and S. M. Lucas, “Towards automatic personalised content creation for racing games”, in Proceedings of the IEEE Symposium on Computational Intelligence and Games, 2007.
5. G N. Yannakakis , J. Hallam, “Towards Optimizing Entertainment In Computer Games, Applied Artificial Intelligence”, v.21 n.10, p.933-971, 2007.
6. J. Togelius, S. Karakovskiy, J. Koutnik and J. Schmidhuber, “Super Mario Evolution”, IEEE Symposium on Computational Intelligence and Games, Milano, Italy, 2009.
7. N. Lazzaro, “Why We Play Games”, Four Keys to More Emotion Without Story, 2004.
8. C. Pedersen, J. Togelius, and G. N. Yannakakis, “Optimization of platform game levels for player experience',' in Proceedings of Artificial Intelligence and Interactive Digital Entertainment (AIIDE'09). AAAI Press, 2009.
9. K. Compton and M. Mateas, “Procedural Level Design for Platform Games”, Proc. 2nd Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE '06), Stanford, CA, 2006.
10. G. N. Yannakakis, and J. Hallam, "Entertainment Modeling through Physiology in Physical Play," International Journal of Human-Computer Studies, vol. 66, issue 10, pp. 741-755, 2008.
11. H.H. Lund and C. Jessen, “Playware – Intelligent technology for children’s play”, Technical Reports 2005, No. 1. ISSN No. 1601-4219, 2005.
12. R. J. Orr and G. D. Abowd , “The smart floor: a mechanism for natural user identification and tracking”, in CHI ’00: CHI ’00 extended abstracts on Human factors in computing systems, NY, USA: ACM Press, pp. 275-276, 2000.
13. M. Rauterberg, “Amme: An Automatic Mental Model Evaluation to Analyze User Behavior Traced in a Finite, Discrete State Space”, Ergonomics, 36 110: 1369-1380. 1993.

14. M. Rauterberg, "About a Framework for Information and Information Processing of Learning Systems", In Falkenberg, E.; Hesse, W.; and Olibve, A., eds., Information System Concepts, 54-69, IFIP Chapman & Hall. 1995.
15. M. Csikszentmihalyi, "Flow The Psychology of Optimal Experience", New York: Harper & Row, 1990.
16. M. Csikszentmihalyi and I. Csikszentmihalyi, "Introduction to Part IV in Optimal Experience", Psychological Studies of Flow in Consciousness, Cambridge, UK: Cambridge University Press. 1988.
17. R. Koster, "A Theory of Fun for Game Design", Paraglyph Press, 2005.
18. T. W. Malone, "What makes things fun to learn? Heuristics for designing instructional computer games", in Proceedings of the 3rd ACM SIGSMALL symposium and the 1st SIGPC symposium on Small systems, pp. 162-169, 1980.
19. G. N. Yannakakis, and J. Hallam, "Real-time Game Adaptation for Optimizing Player Satisfaction," IEEE Transactions on Computational Intelligence and AI in Games, vol. 1, issue 2, pp. 121-133, 2009.
20. G. N. Yannakakis, M. Maragoudakis, and J. Hallam, "Preference Learning for Cognitive Modeling: A Case Study on Entertainment Preferences," IEEE Systems, Man and Cybernetics; Part A: Systems and Humans, vol. 39, no. 6, pp. 1165-1175, 2009.
21. G. N. Yannakakis, "Game Adaptivity Impact on Affective Physical Interaction," in Proceedings of the Int. Conf. on Affective Computing and Intelligent Interaction (ACII'09), Amsterdam, The Netherlands, 2009.
22. Wikipedia Pac-Man, http://en.wikipedia.org/wiki/Pac-Man, 2009.
23. J. Togelius, J. Schmidhuber, "An Experiment in Automatic Game Design", Proceedings of the 2008 IEEE Symposium on Computational Intelligence in Games CIG-2008, Perth, Australia, 2008

ABOUT THE AUTHORS

Z. Halim is an Assistant Professor at Faculty of Computer Science and Engineering, GIK Institute. His research interests include Computational Intelligence in games, Content Evolution for games and improving replay value of games using Computational Intelligence based techniques.

R. Baig is a professor at FAST-National University. His areas of interest include Computational Intelligence (particularly Swarm Intelligence), Machine Learning, Intelligent Systems, Applications of CI in Games and Applications of CI in Data Mining.

H. Mujtaba is a HEC sponsored, PhD student at FAST-National University. His research interests include Computational Intelligence, Machine Learning and Intelligent Systems in Games.