Evolved Weapons for RPG Drop Systems

Joseph Alexander Brown
School of Computer Science, University of Guelph, Guelph, Ontario, Canada, N1G 2W1 Email: jb03hf@gmail.com

***Abstract*—Presented is an overview of an evolutionary algorithm with interactive fitness evaluation as a method for development of video game weaponry, with an emphasis on games with role playing game (RPG) elements. After a short survey of current industry practises for video game weapons, an evaluation of the novel evolutionary method forms the body of this monograph. The method uses the crossover of weapon data structures of similar weapon classes. The player attempting this crossover is shown two possible weapons and is allowed to save only one. This acts as an interactive fitness/selection operator. Such a process, over the course of a game with many item drops, approximates the results of an evolutionary search. The proposed method allows for an increased engagement on the part of the player in their weapons, armour, and gear. Finally, areas for both industry applications of this technique and potential academic research topics for game balance are speculated upon.**

I. Introduction

Weapon design should fit a player's needs, and this evaluation meets with the requirements for a interactive evaluation. What a player in a game desires for their weapon is subjective, especially when there are factors which could not be taken into account with a simple multiple objective optimization such as spread, scope type, one/two-handed, and other such non-quantitative factors. It is the contention of the author that such a framework would provide a novel gameplay experience based on current practises in industry and hence deserves the attention for both academics as a potential area for research on both quantitative factors such as game balance, as well as, qualitative factors including replay value and player enjoyment. For industrial practise it presents a novel feature for games with a number of interesting enhancements for player involvement.

Already, procedurally generated content using evolution has been employed in the creation of weapon systems for players. Hastings et. al. [1] uses what they called a *collaborative content model* which takes the play experiences of all players. Players are presented with a number of weapon types, and the game keeps track of player selections over time. These selected weapons were given a higher chance to remain in the game and form the selection for future generations of weapons. This collaborative method provides an innovative method for the production of player approved weapon types. Recently, the EvoTanks game has been developing weapons which balance criteria of their weapons via a co-evolution [2]. What these methods lack is a gameplay mechanic or in-game motivation for such a weapon creation cycle. Further, these techniques do not allow for a specialization to player requirements as they draw on the experiences of the entire set of players and not a single player's preferences. Pantaleev's examination of various powers in an RPG and their selection by players presents a different direction, the goal was to understand a single player's selections [3].

In this poster paper, a potential for using interactive evolution as part of a weapon drop framework for video games with role playing game (RPG) elements is presented. A short overview of current weapon types in video games is presented with the potential benefits and drawbacks examined. The proposed data structure and framework for genetic weapons for RPGs is presented, along with potentials for their application to current games. Finally, benefits as well as the drawbacks for such a method are examined with ample areas for future developments.

II. Current Weapon Types

The default weapon has set properties, either in visible statistics to the player, or hidden statistics known only to the game. This can be improved by adding the player's skill with the weapon. As the player gains experience, this allows for the unlocking of better damage, a larger clip size, or explosive rounds. This presents no ability for customization on the part of the player as the attribute is just unlocked. These attributes, however, had no drawbacks in their application, only benefits. Without a trade-off, there is no need for the player to not take the upgrade. Player interaction with such upgrades is providing a list of possibilities for which upgrade they would like at the time, such as the *power to the people* vending machines in *BioShock* [4]. There could also be a game currency expenditure, such as the upgrades in *Dead Island* [5].

Weapon drops are the first area in RPGs which moves beyond simple player buffs, and creates a optimization problem presented to the player as to which weapons set would they like to use. This is commonly a multiple criteria optimization, usually combined with a knapsack problem due to the limited space for dropped weapons. *Diablo II* [6], the *Fallout* series [7], [8], and *Borderlands* [9], all present either drops or looting which requires the player to solve such problems. The benefit to the player is clearly seen, they now have a number of weapons to choose from which work well in different situations. The weapons can upgrade with the character, so later levels will have more powerful weapons. The class one rouge may only have a butter knife, whereas, a high level rouge carries an envenomed vorpal kris. The drawbacks are the inverse on the part of developer cost. They must now create a wide variety of weapons, and ensure that these trade-offs are balanced among the weapon classes.

Slotted weapons, such as those presented in *Diablo II* [6] allow for gems which act as buffs to weapon statistics or add

978-1-4673-5311-3/13/$31.00

properties for the loss of the gem slotted, and the inability to remove this from the weapon. This acts as a trade-off for the player in that the gem is lost and the slot is used. A player must make this irrevocable choice or wait hoping for a better gem or a better base weapon. *Fallout: New Vegas* [8] presents a slot like system in mods which add properties to weapons, however, these are removable and limited to a single weapon type. Slots allow for a limited and usually deterministic change to the weapon properties.

Fully configurable weapons have systems similar to slots; however, it approaches closer to actual weapons manufacture in that the parts can be swapped in terms of their types. This is commonly presented in first person shooter games such as *Team Fortress* [10], or the *Call of Duty* [11] series. The receiver of the weapon or layout of the player weapon has a cost in slots associated with the different parts. These parts or weapons are commonly unlockable, given good performace. *Dead Rising* [12] and *Fallout 3* [7] present the weapon model closest to the proposed method. They allow for the combination of weapon parts to create new weapons.

III. Evolving Weapons

In order to define an EA, there needs to be a data structure, a set of evolutionary operators, and a fitness evaluation and selection process. We present the framework in this section. The data structure for such weapons could either be based about the parameters of the weapons; or the weapon could be made up for the slotted parts as examined above. These parts would have their own attributes which would be taken *in toto* and summed to produce the final weapon.

Crossover would be applied to the attributes or to the parts. The best crossover operation to simulate a player's avatar putting together a new weapon out of all the parts of the two older ones is for each attribute or part to be given a random chance to be swapped between the weapons. Mutations would be applied to weapons on a parameter-by-parameter basis or to parts. The mutation would make a change to the single attributes with small probability of an effect. This could be explained as the player repairing broken parts or causing damage to the parts traded between the weapons.

Finally, the player must destroy one of the created weapons as a selection operation. This allows for the evolutionary search to be implemented by the human. Further, this operation prevents a player from constantly applying the crossover on two weapons in order to create a super weapon without drawback. Each breeding step costs a weapon drop. As an in-game explanation, the tearing apart and building of the new weapon from parts of the old has required some of the fittings, brackets, casings, and metals such that only one can be made back from parts of the two. This allows for a story driven reasoning behind the evolution of a single player's actions. This process could also create *scrap* as a side effect. These scraps could be used as a currency or could be used to purchase mods with known positive attributes.

Elitism and other selection processes are already implemented through normal game mechanics such as limiting inventory size. The limit causes the player to sell/drop/scrap weapons which are not to their liking (selection) and carry perhaps a number of special/situational weapons (elitism).

The shortcomings of such a procedure must also be expanded upon. There is an issue with the game balance problems which such a process entails. Games such as *Borderlands* [9] or *Diablo II* [6] control balance by putting level requirements on weapons based on a proprietary formula. How such a formula would be applied to evolved weapon types should be examined. It would not be beneficial if the player could modify their weapon too quickly to a super weapon state as this would remove the challenge of the opponents for the player. Conversely, if this process happens too slowly, the weapons found in the world will become better than the evolved weapons — thus giving the player no incentive.

The number of generations within an evolutionary method is usually quite large; the number of drops, required to form a weapon, must be examined in the application of the system. Additionally, evolutionary operations will not necessarily create better weapons on each application. The author actually sees this as a feature of the process and not a bug. The player will have a risk when applying the operation; a mini-game of risk v. reward. Further, this could lead to an interesting change to the game mechanics, allowing for an attribute (such as repair skill) or a perk (jury-rigger) which would bias the genetic operations to produce weapons which are on the tail-ends of the operators, re-attempt the operators, merge two weapon properties of dissimilar classes, or keeping both the weapons.

IV. Conclusions

Presented was a prospective on using interactive evolution to inform the creation of weapons for games with RPG elements. The proposed method uses a crossover operation which is player driven. In future, this system should be tested for game balance effects.

Acknowledgements

The first author would like to thank the National Science and Engineering Research Council (NSERC) of Canada.

References

[1] E.J. Hastings, R.K. Guha, and K.O. Stanley, "Automatic content generation in the galactic arms race video game", *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 1, no. 4, pp. 245–263, 2009.

[2] Adam Pace and Tommy Thompson, "Procedural content generation and evolution within the evotanks domain", in *Foundations of Digital Games*, 2013, pp. 439–440.

[3] Alex Pantaleev, "In search of patterns: Disrupting rpg classes through procedural content generation", in *The third workshop on Procedural Content Generation in Games*, 2012.

[4] 2K Games, "BioShock", 2007.

[5] Techland, "Dead Island", 2011.

[6] Blizzard Entertainment, "Diablo II", 2000.

[7] Bethesda Game Studios, "Fallout 3", 2008.

[8] Bethesda Game Studios, "Fallout: New Vegas", 2010.

[9] 2K Games, "Borderlands", 2009.

[10] Valve, "Team Fortress 2", 2007.

[11] Activision, "Call of Duty: Modern Warefare 2. Black Ops", 2011.

[12] Capcom, "Dead Rising", 2006.