A Framework for Branched Storytelling and Matchmaking in Multiplayer Games

Vitor Pêgas[1(✉)], Pedro Santana[1,2], and Pedro Faria Lopes[1,3]

[1] ISCTE-Instituto Universitário de Lisboa, Lisboa, Portugal
{vitor_pegas,pedro.santana,pedro.lopes}@iscte-iul.pt
[2] Instituto de Telecomunicações, Lisboa, Portugal
[3] ISTAR-Instituto Universitário de Lisboa, Lisboa, Portugal

Abstract. Video games often either have good single player campaign modes or good multi-player campaign-less modes. This paper presents a framework aimed at the full game development pipeline, from designers to programmers, to aid in creating multiplayer campaigns by providing components that help singleplayer story modes to be used in multiplayer interaction settings. We also propose a custom matchmaking system capable of matching players so as to intertwine their individual stories. The proposed framework has been validated in a case study. A set of experimental results show that the framework is capable of producing valuable story crossings and proper matchmaking.

Keywords: Video games · Multiplayer · Campaign · Framework

1 Introduction

Currently, there are two main types of commercial video games, SinglePlayer (SP) and MultiPlayer (MP) games. In SP games, a player plays against Non-Player Character (NPC) controlled by Artificial Intelligence (AI), whereas in MP games players can play against each other (Human vs Human) or in cooperative modes (Humans vs NPC).

In 2017 there were five SP games in the top ten sales, with millions of copies sold worldwide [15]. On the other hand, MP games are not only strong in sales, but the revenue of such games surpasses SP revenue, mainly due to micro-transactions. For instance, games like Grand Theft Auto Online (GTA V Online) and League of Legends (LoL) made more than 500 Million USD [13] and 2.1 Billion USD [3] in in-game item sales alone. This shows that although MP games dominate the market in revenue, SP games continue to be bought and played by millions.

A problem that SP games may suffer is that their quality is closely linked to the quality of the AI controlling the NPC. An improper AI may render a game either too easy or too hard to overcome. On the other hand, players in MP games may find themselves immersed in what is known as toxic communities, in which players engage on hostile behavior against each other, a phenomenon known

A. L. Brooks et al. (Eds.): ArtsIT 2018/DLI 2018, LNICST 265, pp. 91–100, 2019.
https://doi.org/10.1007/978-3-030-06134-0_10

to naturally occur in competitive gaming scenarios, such as MP online games [10]. Popular online MP games such as Call of Duty and League of Legends are well known for their community toxicity [7]. To overcome these limitations we propose a generic framework agnostic to the game genre in order to create campaigns with multiplayer interaction that prevent the use of bad AI controlled characters and a matchmaking system that provides a more balanced matching between players to prevent toxicity and ease the learning process of new players.

This paper is organized as follows. A literature review is provided in Sect. 2. In Sect. 3 the proposed framework is described. Then, in Sect. 4, a set of results are presented, followed by conclusions and future work avenues in Sect. 5.

2 Literature Review

2.1 Single Player and Multi-player Crossing

A player engagement study [8] found that players show higher engagement levels when facing humans in a competitive scenario instead of AI controlled characters. This may also be influenced by the fact that AI-based characters quite often produce either too easy or too hard game mechanics. Nevertheless, players do not refrain from playing and enjoying SP games as this type of games often features a well written story (e.g., Until Dawn [4]). For this reason, our framework aims at allowing players to enjoy the flow of a storyline while playing in a MP setting, that is, making use of NPC only when no online players are available.

To improve the lack of engagement or challenge imposed by AI, game developers blurred the line between SP and MP, by having the SP game with its well written and designed story and levels, and occasional MP interactions, such as in Dark Souls and Journey. For instance, in Dark Souls players can be invaded by other human players, which will control NPC's inside the first player story, whereas in Journey players can help each other. Other games, like Absolver, each player's game instance can also host other players to fight, co-operate or ignore one another. There are games with MP campaigns, such as Borderlands and Left4Dead, but these are simply a set of co-op missions that try to create a sense of narrative for each player (friends or random online players).

Our framework was created to allow the possibility for players' individual stories to cross each other at story intersections. In those intersections (hereafter story nodes), players can be on the same or opposite sides. This is possible due to the existence of a list of ideal player profiles for each story node, and the framework uses this to allocate players to nodes in a coherent way.

2.2 Branched Storytelling and Narrative

Replay value is a term used in video game industry that states whether a game is good to play more than once. Usually, due to their linear design, SP games do not hold much replay value. This means that if players complete the story and then try it again, the story (and ending) will be the same. However, there

are SP games that try to branch their narrative influenced by player input, thus increasing their replay value. Bearing replay value in mind, our framework handles both linear and non-linear stories.

MP games are quite often repetitive. For instance, First Person Shooters (FPS) usually only have as goals killing the opposite team or capturing a given item or position. In spite of this, players keep playing FPS nonetheless, motivated by [16]: competition, as seen in games like DOTA 2 and CS: GO; socialization, as seen in games like Second Life and Habbo Hotel; and role-playing as seen in games like GTA V, and World of Warcraft.

In SP games, developers must find a way of showing the player that their actions matter and can change the outcome of the game, pushing the player to play the same game several times while producing different outcomes [12]. Procedural Content Generation (PCG) is popular choice to provide the so required in-game diversity (e.g., random levels, fauna, flora). PCG use has increased lately due to the skyrocketing production cost of AAA games [9], for which is difficult to scale up asset production (e.g., 3D models) [6].

Other way to increase replay value is by branching the narrative in non-linear ways. For this to happen, there must be more than one choice of action for the player to perform, and that action must somehow impact the story unfolding. Many games have exploited this ability widely, such as Façade, Witcher 3, and Undertale. These games require some narrative mediation system to ensure the player experiences a coherent story, by creating a path that the player travels according to one choices and the system automatically re-writes the path if the player decides to take a different direction.

Our framework uses multi-dimensional descriptors (hereafter *coherences*) at each story node so as to allow comparing nodes and, as a result, to determine which story node should be visited by the player as soon as it leaves the current one. This implements a form of non-linear procedural story generation whose plot is a result of the story nodes' associated descriptors.

2.3 Matchmaking

A matchmaking system is used in games to connect players in a given match. When a player wants to play online, its game instance (client) sends a request to the Matchmaking Server (MMS), receiving a response with necessary information to establish a connection [1]. A bad matchmaking service is known to be harmful to both skillful and casual players [11].

One of the most popular types of matchmaking, Elo based systems, uses a numeric value, called a Matchmaking Rating (MMR), to match players according to their skill levels and was created for chess competitions, in which the MMR is determined by the results of the player in past games [5]. Matchmaking usually picks players with close MMR values so as to ensure fair and, so, interesting, matches [14]. Match fairness and uniformity can be further enforced by ensuring that the players within a given team have an uniform MMR [2]. By using custom matchmakers, instead of general-purposed matchmaking systems as Elo, games can match players more accurately as a function of the game's mechanics.

This paper proposes a novel matchmaking system that uses multi-dimensional descriptors associated to players (hereafter *Stats*) to more accurately match players and story nodes and, thus, not only by their skill, but for their experience, class, and other in-game statistics.

3 Proposed Framework

The framework has been devised to be agnostic to game genre. Taking into account the typical game development pipeline, the framework is composed of a set of core game components to be handled by designers and artists during the conceptualization phase of the game. Designers and Artists can use a Graphical User Interface (GUI) application to create and manage these components and then pass that information to the programmers which will implement the game in-engine.

For the sake of clarity, let us illustrate an application scenario of the framework to a well-known commercial game, namely, BattleField 1. Battlefield 1 is a game that has a good SP campaign, that suffers from a linear narrative and a good MP part of the game with several modes where players face real human players online. When analyzing the SP campaign, we can identify common areas where the proposed framework could be used to offer aid in a non-linear storyline. In one of the campaign stories, the player is a tank driver whose mission is to conquer key operation posts. While aided by other NPC tanks and infantry, the player fights NPC enemy tanks and infantry. With our framework there could be different roles in this campaign, like the tank driver, the tank gunner and several infantrymen on both sides, each with their back story leading to that particular moment (story node). The tank driver could be playing that battle because he managed to win past missions, or his tank crew got wounded (seen by our framework as a partial mission success) and he found a new crew. To determine which player to allocate to each role in the story node, the players' descriptors (Stats), such as player health, level, experience, weapon information, are matched against the story node's player profiles.

4 Framework Components

Figure 1 depicts the major components involved in the proposed framework. The central component is denominated of *Node*.

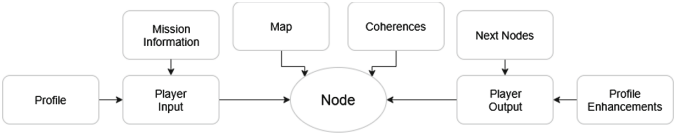


Fig. 1. Overview of the proposed framework's basic components.

Node. The Node is a point in a campaign story, which is a set of linked nodes. It contains engine information such as the assets needed for that node such as a Map (i.e., in Unity it would be associated with a Scene object to be loaded), contains information regarding the ideal profile for players inside the Node and the Node's coherence values, which define that Node in terms of location, appearance, time and story placement. These coherences are used for procedural story generation. In each node there is information for Player Inputs and Player Outputs. The Inputs have the desired player profiles and their roles in that Node, the Outputs have information regarding how the players leave that Node. In a linear story, our framework provides the possibility of near-success missions, this means that there is not a win and lose node, but a list of possible nodes for possible partial successes. Also in the Outputs we have profile enhancements that are used to change the players profiles (increase/decrease stats), for instance a player that wins a Node, deserves to be rewarded with some experience.

Player Information. Each player has a profile defined as a set of Stats. Stats are scalar values that define that player, like *Health*, *Class*, *Experience*, *Ranking*. In each Node, there are multiple ideal Player Profiles which are matched against potential players by the matchmaking system, thus attempting to find the most suited players for that specific node. In a Node, these Profiles have more information about the Objective of the matched player like its role and goal in that Node.

Coherences. Our framework defines Coherences as an one dimensional spectrum. Coherences can be used to define how semantically close are Nodes supposed to be. We defined some example Coherence types, such as Spatial (physical space, i.e., City vs Desert), Temporal (i.e., 1920 vs 2018), Story (Story line flow), and Random (to ensure some stochastic story line flow). These can be added to any Node and defined with a scalar value from 0 to 1. Coherences are used to generate procedural stories (branched storytelling), based on players previous nodes. Each Coherence i has a procedural story generation weight w_i that is used when finding for possible next nodes using a weighted Euclidean Distance. Concretely, if the player is leaving node $\mathbf{N_1}$, the cost of going to node $\mathbf{N_2}$ is

$$\Psi(\mathbf{N_1}, \mathbf{N_2}) = \sqrt{\sum_i w^i (N_2^i - N_1^i)^2}, \tag{1}$$

where $\mathbf{N_k} = (N_k^1, ..., N_k^z)$ corresponds to the z-dimensional coherence vector of node k, and w^i corresponds to the weight of the i-th component of the coherence vector. Weights w_i can be used by the game designer to generate a procedural story. The system compares the Node's coherences vectors to find the one with the least cost. Given that we are at node N_i, we pick, among the set L of the node pool, the node with lowest coherence distance.

$$l = \arg\min_{\mathbf{l^*} \in L} \Psi(\mathbf{N_i}, \mathbf{N_{l^*}}). \tag{2}$$

4.1 SP and MP Crossing

The way our framework crosses the two game genres, SP and MP, is due to its architecture for MP campaigns. Designers can use our framework to create tailored linear or non-linear stories for strict player types or create a pool (See Fig. 2 (A)) of unconnected nodes and let the procedural story generator create the story for each individual player as it plays.

To create a procedural story, our framework uses previously defined Coherences to get the closest possible node to a player's current node. Once a close Node has been picked from the node pool (see Fig. 2 (B)), the player will move to that Node and the matchmaking process begins. The system now has to either find players that have a close and similar profile to our player and are set to play in the picked Node or to find any open sessions (a match already on-going) for this player to join (see Fig. 2 (C)). While the system attempts to find a match, a time-fill mission can be played, if previously defined by designers. An example for a time-fill mission could be the an objective to move towards to or the gathering of certain items that can be useful later. For both options, player profiles are compared and matched against each other to allow for balanced matches. In case of failure to find any open sessions, the player starts a game session on the current Node and waits for other players to join later.

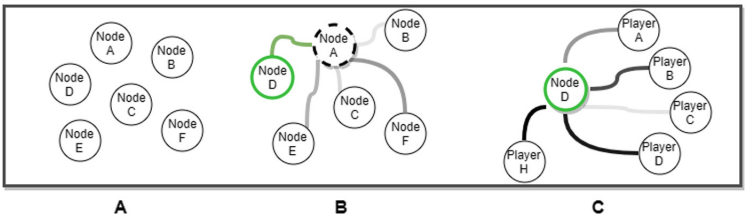


Fig. 2. Framework procedural story generation

4.2 Matchmaking

Usually competitive games base their matchmaking on the skill rating of each player, and the system attempts to match players with a close rating value. Alternatively, we propose a matchmaking system that weighs the several *Stats* included in the player's profile, promoting a more balanced matchmaking. This is attained by computing the Euclidean distance between the multi-dimensional *Stats* vectors composing the profile of the player, $\mathbf{P_p}$, and the required profile set for the Node in question, $\mathbf{P_n}$:

$$d(\mathbf{P_p}, \mathbf{P_n}) = ||\mathbf{P_p} - \mathbf{P_n}||_2, \tag{3}$$

where $\mathbf{P_k} = (p_1^k, \ldots, p_n^k)$ represents a n-dimensional *Stats* vector. The distance between the players Stats is then used to obtain the cost of matching the player

and the node requested profile, which favors matches with players waiting longer for a match:

$$\Phi(\mathbf{P_p}, \mathbf{P_n}) = \alpha \cdot d(\mathbf{P_p}, \mathbf{P_n}) + (1 - \alpha) \cdot \gamma e^{-\beta t}, \tag{4}$$

where α and β are empirically defined scalars the game designer can tune to favor, or not, players that are waiting longer for a match, and t represents time. Then, the system computes the cost of matching the player with all available players and uses a threshold to limit the difference between players' Stats.

If a good-enough match is found, then the system sends all the connection information the player needs to join a MP match. While this happens, the player that sent the request can either wait for a match, or play a time fill in mission which is associated to the node in question. When the connection information arrives, the player connects and the matchmaking process ends. If immediate matchmaking is not possible, the server will insert the request on the list for future requests and will set a timeout. When the timeout ends, and no matches are possible, the client or the server can inform the other about giving up the matchmaking process and play with NPCs instead.

4.3 Case Study

To test the implementation of our framework we designed and developed a 2D online web game to be used as case study and for user testings. To fully use the framework we decided to create a Role Playing Game (RPG) as this genre of game usually focus on individual player stories. We designed a story composed of Nodes that would describe a journey across a border of two nations in war, with different missions opposing both countries. Nodes can be played in a linear fashion like most games, but we can also create procedural stories by using Node Coherence components to assign the player's next node. These Nodes were defined by three types of Coherences: *Location*, *Story*, and *Appearance*; for instance, a Node that is physically located in the North of the world would have a *Location* value of $N^{location} = 0$, while a Node physically located in the South would have the value of $N^{location} = 1$.

5 Results and Discussion

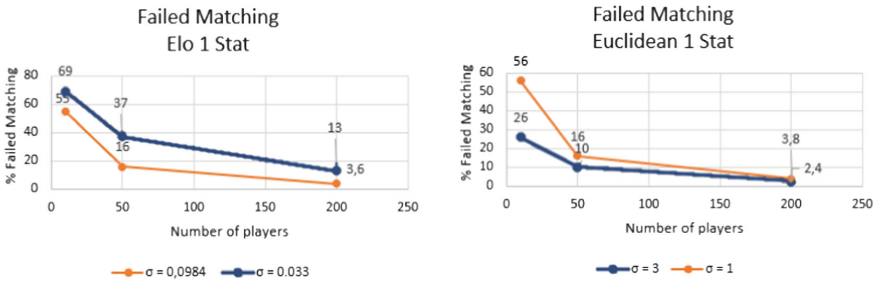
Our framework was successfully implemented in the case study game with every component being used like predicted. We used this game to test the matchmaking system by running simulations of concurrent NPCs.

The matchmaking system was tested using simulated players $N = \{10, 50, 200\}$ which were trying to find a match for a given node. We tested our euclidean based system against an elo based system with: one, two, and three Stats with values ranging from 0 to 10. As the elo system only is capable of handling one stat (the rating of the player or MMR), for the two and three stat testing we used the average of the stats as the one defining criterion for matching. We also tested restricting or relaxing the matching threshold (see

Sect. 2.3) to observe how it affects matching. For each test we ran three times and averaged the results.

In each test we used our procedural story generator to generate a story for each NPC based on its current node. We used three Coherences, namely, Story ($w_{story} = 0.8$), Location ($w_{location} = 0.05$), Appearance($w_{appearance} = 0.05$) and Random($w_{random} = 0.1$), so the chance of players going different ways is higher than with a linear story.

Our results for one stat matchmaking are shown in Fig. 3. For N=10, which is a significantly low amount of players, considering our procedural story line and multiple values for one stat, the euclidean based system fails 56% of times for a $\sigma = 3$, meaning that 56% of the total requests for matches, ended up playing with NPCs. Relaxing the value of the matching threshold (σ), we can observe that the system is more successful, as it hits 26% with the same 10 players. It is not ideal, but in a game where there is the possibility of this much deviation, it is no surprise. If we increase the number of players to N=200, the failed matching lowers to 3.8% with the strict threshold and 2.4% with a relaxed σ value. The results show a tendency to lower the failed matching the more players we have. The elo system shows high failed percentage as well for lower player bases but manages to show the same decrease as player base increases. The σ values used for the Elo system were determined to be the same in proportion to the ones used in the Euclidean System, meaning that they limit the same amount of difference between players.



(a) Elo based system Matchmaking Results

(b) Non-Weighted Euclidean based system Matchmaking Results

Fig. 3. Results for one Stat matchmaking. As shown both systems are more successful in matching when the player base increases.

For three Stats however the non-weighted euclidean based system shows more failed matching (14% for $N = 200$ and $\sigma = 3$) due to the increasing difference between possible players, but the weighted system maintains low failed percentages (3.1% for $N = 200$ and $\sigma = 3$ and 80/10/10% weight distribution). The elo based system maintains a low percentage due to the averaging of the stat values with 2.5% for $N = 200$ and $\sigma = 0.0984$.

In terms of failed percentage the elo appears to be able to match players more easily in diverse situations; however this does not prove its total efficacy. Let us imagine an example of a three criterion matching with Stat A,B, and C. Each player can have one of the stats (A,B,C) with value ten and the rest of the values are 0. Player X would have $A = 10, B = 0, C = 0$ and player Y would have $A = 0, B = 10, C = 0$. Now let us try and match Player X and Y. Following the euclidean distance (non-weighted), the system would give a matching value of $m = 14.1$, showing that the players are indeed different. But when we switch over to the elo system, as it only handles one criterion, we have to average the stats giving us a total match value of $m = 0.5$ because the average of the three stats is the same $10/3$. So in conclusion the matching efficacy is not the whole picture as accurate matching is paramount for a good player experience, and Elo based systems cannot guarantee that balance with more than one stat.

The elo works as shown by games such as Rocket League, Overwatch, CS:GO, League of Legends and others, but is limited when given the possibility to match with more than one criterion than just skill, and as demonstrated in the previous example, it can match players but it does not guarantee a balanced match in every situation. However, our framework is not bound to the euclidean based systems, thus being possible to use and implement elo based systems or any more suited system that designers see fit for their game.

6 Conclusion and Future Work

With MP games becoming more common due to their long-run revenue the need for good, story-driven SP games increases. This paper proposed a framework that aims at blurring the line between both genres by allowing developers to connect a good immersive story with the possibility of human vs human interaction. We presented not only a framework that aids the creation of SP games with MP interaction but also proposed a matchmaking system that matches players in a balanced way. Our simulations show that the matchmaking system is capable of balanced player matching. Our case study shows that the framework is generic enough to be used in a story driven game, game-genre independent. As future work, additional testing should be made to validate that with big budget commercial games's player base the efficiency of the matchmaking systems decreases as predicted.

References

1. Agarwal, S., Lorch, J.R.: Matchmaking for online games and other latency-sensitive P2P systems. In: Proceedings of the ACM SIGCOMM Computer Communication Review, vol. 39, pp. 315–326. ACM (2009)
2. Alman, J., McKay, D.: Theoretical foundations of team matchmaking. In: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, pp. 1073–1081. International Foundation for Autonomous Agents and Multiagent Systems (2017)

3. Altay, O.: Top free to play pc games by revenue 2017 - superdataresearch (2018). https://mmos.com/news/top-free-play-pc-games-revenue-2017-superdataresearch. Accessed 26 April 2018
4. Brew, S.: Until dawn, the interactive movie, and storytelling (2015). http://www.denofgeek.com/games/until-dawn/37259/until-dawn-the-interactive-movie-and-storytelling
5. Glickman, M.E., Jones, A.C.: Rating the chess rating system. Chance **12**, 21–28 (1999)
6. Hendrikx, M., Meijer, S., Van Der Velden, J., Iosup, A.: Procedural content generation for games: a survey. ACM Trans. Multimed. Comput. Commun. Appl. (TOMM) **9**(1), 1 (2013)
7. Kwak, H., Blackburn, J.: Linguistic analysis of toxic behavior in an online video game. In: Aiello, L.M., McFarland, D. (eds.) SocInfo 2014. LNCS, vol. 8852, pp. 209–217. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-15168-7_26
8. Lim, S., Reeves, B.: Computer agents versus avatars: responses to interactive game characters controlled by a computer or other player. Int. J. Hum.-Comput. Stud. **68**(1), 57–68 (2010). https://doi.org/10.1016/j.ijhcs.2009.09.008. http://www.sciencedirect.com/science/article/pii/S107158190900130X
9. McLaughlin, M.: New GTA V release tipped to rake in £1bn in sales (2013). https://www.scotsman.com/lifestyle/gadgets-gaming/new-gta-v-release-tipped-to-rake-in-1bn-in-sales-1-3081943
10. Märtens, M., Shen, S., Iosup, A., Kuipers, F.: Toxicity detection in multiplayer online games. In: Proceedings of the 2015 International Workshop on Network and Systems Support for Games (NetGames), pp. 1–6, Dec 2015. https://doi.org/10.1109/NetGames.2015.7382991
11. Myślak, M., Deja, D.: Developing game-structure sensitive matchmaking system for massive-multiplayer online games. In: Aiello, L.M., McFarland, D. (eds.) SocInfo 2014. LNCS, vol. 8852, pp. 200–208. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-15168-7_25
12. Roth, C., Vermeulen, I., Vorderer, P., Klimmt, C.: Exploring replay value: shifts and continuities in user experiences between first and second exposure to an interactive story. Cyberpsychol. Behav. Soc. Netw. **15**(7), 378–381 (2012)
13. Tassi, P.: GTA online's $500m in microtransactions could mean a very different 'GTA 6' (2016). https://www.forbes.com/sites/insertcoin/2016/04/14/gta-onlines-500m-in-microtransactions-could-mean-a-very-different-gta-6
14. Véron, M., Marin, O., Monnet, S.: Matchmaking in multi-player on-line games: studying user traces to improve the user experience. In: Proceedings of Network and Operating System Support on Digital Audio and Video Workshop, p. 7. ACM (2014)
15. VGChartz: Global yearly chart (2017). http://www.vgchartz.com/yearly/2017/Global/. Accessed 26 April 2018
16. Yee, N.: Motivations for play in online games. CyberPsychol. Behav. **9**(6), 772–775 (2006)