GIMME: Group Interactions Manager for Multiplayer sErious games

Samuel Gomes
Intelligent Agents and Synthetic Characters Group INESC-ID & Instituto Superior University of Lisbon, Portugal
samuel.gomes@tecnico.ulisboa.pt

João Dias
Intelligent Agents and Synthetic Characters Group INESC-ID & Instituto Superior University of Lisbon, Portugal
joao.dias@gaips.inesc-id.pt

Carlos Martinho
Intelligent Agents and Synthetic Characters Group INESC-ID & Instituto Superior University of Lisbon, Portugal
carlos.martinho@tecnico.ulisboa.pt

***Abstract*—Serious games are moving towards multiplayer environments which aim to concurrently train and educate people. Therefore, we believe there is a growing need to construct systems which account for this tendency in order to ease learning and training. We here propose a model named Group Interactions Manager for Multiplayer sErious games (GIMME) which aims to improve the collective ability of players interacting in environments such as these. Our method organizes the players in groups and computes the types of interactions that should be promoted for each group. These interactions are then promoted by generating adequate game mechanics. We validate the most important aspects of the model by performing several agent-based simulations. The simulations suggest that learning can be improved when applying our strategy, as the average ability of the agents rapidly converged to high, near optimal values, as opposed to a random baseline (a strategy possibly implemented by real teachers when they do not know the students) which maintained low values. Moreover, unlike the random strategy, GIMME considerably approximated the promoted interactions profiles to the agents' computed preferences.**

***Index Terms*—Multiplayer, Serious Games, Group Management, Player Interactions**

I. Introduction

Since games have revealed extensive benefits in respect to cognitive and social processes [1], several methods have been developed to apply games and game elements to serious contexts [2]. From those methods, we highlight the Adaptive Learning Systems (ALS), which are Learning Support Systems (LSS) that tailor the content being taught to the students' needs [3, 4, 5]. Most researched ALS applied in conjunction with serious games focus on the learning aspects of individuals. However, we are witnessing a rapid evolution of serious games into multiplayer environments which approach collective learning and training [6, 7]. Examples of multiplayer serious games include *Escape From Wilson Island* and the serious game for teamwork workshops *TeamUp* [6]. However, multiplayer environments are generally harder to conceive because developers need to take into account not only the individual progress of each player, but also social factors such as the types of interactions between the players and how to select teams, and design aspects such as the players' persistency. As such, we believe processes must be created so that this type of games better adapt the content being taught to the collective needs. As the main learning and training goals are to develop people's overall ability, we aim to approach the following research problem:

How can the collective ability of players be improved in a multiplayer serious game?

To help solve this problem, we here propose a model named Group Interactions Manager for Multiplayer sErious games (GIMME) which functions upon two basis: first, the players are organized into groups and for each group, a Group Interactions Profile (GIP) is estimated. A GIP represents the types of interactions that need to be promoted so that group learning is improved (collective ability is increased); then, to promote those interactions, several game mechanics which incentivize specific interaction types are combined.

The next sections detail our model and present several agent-based simulations which were conducted to validate its most important aspects and inform us about additional refinements and improvements. In order to ease the understanding of our explanations, we will present some illustrative examples of its applicability to the domain of *a music learning game which allows students to learn to play instruments*.

II. Related Work

A. Studying the interactions and characteristics of students

In order to understand how to meaningfully manage and interact with a group of students, we first analyzed some work delineating different types of interaction and their influence in the students' learning. Multiplayer serious games research divides the types of possible multiplayer interactions as competitive, cooperative or collaborative [6]. In fact, as presented next, these are dimensions which appear in both game-oriented and education-oriented studies.

Burguillo described how friendly competition was used in order to improve the learning experience of students of a programming course [8]. After analyzing course evaluation questionnaires and the students' overall feedback over 10 years, the author concluded that the average scores were higher when competition was applied and that no major increase in

978-1-7281-1884-0/19/$31.00

the students' workload was seen. More recently, ALS also resorted to competition in order to facilitate learning [9].

Additionally, collaboration has been analyzed in conjunction with competition. Firstly, tasks implying collaboration and competition were acknowledged when studying group organization processes oriented to games [10]. Secondly, it was shown that, while playing a serious game which implied both collaboration and competition, divergent strategies are preferred by students with different levels of ability [11]: there is a tendency for students with below-average prior knowledge to benefit more from collaboration and less from competition than students with above-average prior knowledge.

Besides ability, another characteristic frequently approached in learning and games research is engagement. Studies in automatic adaptation usually aim at measuring engagement objectively through metrics like re-playability and retention [10, 12]. Other more learning-oriented metrics are also identified in studies about student progression [13, 14], namely the on-task and off-task times (the time students spend accessing or not the task at hand).

B. Studying the emergence and promotion of social behaviors in games using game mechanics

Several different definitions for game mechanics have been proposed over time [15, 16]. Our model focuses on a special type of mechanics: those which lead players to incur in certain types of social interaction. We identify this type of mechanics as *social interaction mechanics*. In a recent study already following this definition [17], groups of two players interacted with a game using key combinations which were secretly attributed such that the game actions could either be performed by the players independently or required their concurrent input. An interesting finding was that when both players had to interact to perform a game action, they implicitly perceived that they were playing cooperatively.

Additionally, mechanics tailored to collaboration were proposed by Oksanen and Hämäläinen [18]. They included the use of shared space and objects between the players, complementary actions (actions which can only be made by a group) and indirect actions (situations where a player has a task which requires other player's action).

Moreover, Consalvo presents some strategies aimed to promote competition like generating challenges and allowing player-vs-player matches [19].

C. Discussion

The mentioned research presented two main types of interaction: *Collaboration* and *Competition*. Cooperation was also mentioned, but we will not distinguish it from collaboration, as in the scope of our work, we define the *Collaboration* dimension more generally as: *between-players interactions with a mutual benefit in mind*. This definition contrasts with our *Competition* definition: *between-players interactions with an exclusive benefit in mind*. Moreover, although the goal of our system is to promote tasks in groups, we believe several grouped elements can also concurrently train resorting to individual tasks. As such, we add another dimension of interaction besides *Collaboration* and *Competition*: *Self-improvement*.

Considering the example of music learning, a collaborative mechanic could lead players to perform a group rehearsal where every element is rewarded, a competitive mechanic could lead only the best player to be rewarded and a *Self-improvement* mechanic could lead players to practice a chord alone.

Besides underlining the different types of interaction, the related studies also referred two metrics which can be applied to characterize serious game players: ability and engagement. To compute these attributes, we need to extract some measures from the interaction between the players and the game. On the one hand, we believe we can measure ability simply by providing scores for the game tasks. In a music learning game, this can be reflected as the amount of correctly played chords while training a song. On the other hand, we can use the on/ off task times, player retention and flow theory applied to games [20][1] to objectively compute engagement. We can think of a player's task engagement as the ratio of time he/she spends accessing the task and completing it. This is applicable when each task can only be completed or restarted if the player fails it. An engaged player will have high on-task time and completion rate. A high on-task time but low completion rate indicates that the player is often repeating the task, completing a low portion. A low on-task time but high completion rate indicates that the player quickly completed the current task. The previous two scenarios are associated with low engagement according to the flow theory applied to games. A low on-task time and low completion rate can also mean that the player does not like this particular type of task and is not even trying to access it, and so a low engagement value should also be considered for this situation. Using another music learning example, a player which selected a song to learn and either spends most of the time stuck practicing a small part of a song, completes it very fast or is not even trying to complete it should have low engagement in that task.

III. GIMME DESCRIPTION

While introducing this work, we referred that the goal of our model is to improve the collective ability of the players of a multiplayer serious game. In order to do so, we divided its execution in three steps (as presented in Figure 1):

1) Periodically check and update the players' charcteristics (their state);
2) Use the players' states to organize them into groups, computing Group Interactions Profiles (GIPs) for each group representing the interaction types that should be promoted;

[1] Flow theory applied to games refers that game tasks should be challenging enough for the players to not feel bored, but not so challenging that they feel overwhelmed. In other words, a balance has to be kept so that engagement is maintained.

3) Combine game mechanics capable of promoting the GIP of each group, acknowledging also the state of the players in the group.

In order to dynamically characterize a player, we introduce the notion of Player Learning State (PLS), consisting of values for the players' overall ability and current task engagement, normalized in the range [0, 1]. From a PLS, we can define Group Learning State (GLS) as the average of the values computed for all of the players' PLSs in a group, and a Configuration Learning State (CLS) as the average of all of the GLSs in a collection of groups (we name a collection of groups a configuration).

A. Defining a Group Interactions Profile (GIP)

To define a GIP, we will use what we believe to be the three main dimensions of interaction: *Collaboration*, *Competition* and *Self-improvement*. As such, we can profile a task by the ratios of the *Collaboration*, *Competition* and *Self-improvement* it implies. If we assume a learning profile to be represented by a tuple $< Coll, Comp, Self >$ such that each component is defined in the range $[0, 1]$ and $Coll + Comp + Self = 1$, we can draw the space of possible learning profiles as an unitary triangle where each vertex represents purely *Collaborative*, *Competitive* or *Self-improvement* profiles (Figure 2). A profile which promotes a mixture of these behaviors can be seen as a point contained inside the triangle. As such, the values <0.27; 0.23; 0.50> would represent a learning profile promoting 27% *Collaboration*, 23% *Competition* and 50% *Self-improvement*.

B. Organizing students according to their Player Learning States

We approach this part of our model in two steps as represented by the areas A and B of Figure 3:

- First, we generate n random group configurations (ways to group players) and for each group a random GIP, as represented in Figure 3 A[2];
- Next, from the previously generated n configurations, we choose the one which maximizes both the ability and engagement of the players. This is depicted on Figure 3 B. We do that by predicting, for each configuration c, a CLS_c. Because CLS_c represents the average of all of c's *GLS*s and one *GLS* represents the average of all of the players' *PLS*s in one group, a prediction has to be computed for the students' *PLS*s.

  To predict a player's *PLS* inserted in a group g according to c, we use *KNN Regression* [21]. First, from the past *GIP*s experienced by the player, we select the k nearest *GIP*s to the GIP_g: $\{GIPnear_0, ..., GIPnear_k\}$. Next, we observe the *PLS*s obtained by the player when experiencing each of the k nearest *GIP*s: $\{PLSnear_0, ..., PLSnear_k\}$. Finally, we calculate the *PLS* prediction based on the average of the $\{PLSnear_0, ..., PLSnear_k\}$ weighted by the distances between the GIP_g and the $\{GIPnear_0, ..., GIPnear_k\}$.

  After predicting the *CLS* for all configurations, we can resort to two weights aW and eW to obtain the best configuration by calculating:

$$argmax_c\ (aW \times CLS_c.ability + eW \times CLS_c.engagement)$$

C. Translating a behavior profile to game mechanics

The last stage of our model consists in translating a profile to concrete game mechanics. In order to solve this problem, we consider a mechanic to be a combination of n sub-mechanics tailored to our basic types of interaction: *Collaboration*, *Competition* and *Self-improvement*. As each dimension in a behavior profile $< Coll, Comp, Self >$ represents a ratio such that $Coll + Comp + Self = 1$, we can use $Coll$, $Comp$, $Self$ to check how many of the n mechanics promoting *Collaboration*, *Competition* or *Self-improvement* should be included. In the related work, we already presented and discussed mechanics connected to our basic interaction types, like using shared objects to promote collaboration or challenges to promote competition (subsection II-B).

Besides the incentivized behaviors, the difficulty level of the mechanics promoted for a group g should also be taken into account and adapted to the average ability and engagement values of the group (the GLS_g). As such, we can divide a mechanic into several *mechanic instances* leading players to perform tasks with different difficulty levels. To define a mechanic instance, we simply have to ensure that the tasks are parameterizable, meaning that the tasks' objectives can be changed, but not the actions that need to be performed to solve the tasks. Considering other example of music learning, if a mechanic leads players to perform a task "learn to construct a chord", we can think of mechanic instances leading players to perform easier tasks such as "learn to construct a C chord", intermediate tasks: "learn to construct a C2 chord" or harder tasks: "learn to construct a C9 7maj (aug 4) chord" [3]. Figure 4 depicts our mechanics combination process.

IV. Validating GIMME through multi-agent simulations

In order to validate the most important aspects of our model, we performed several agent-based simulations. This approach also aimed to inform us about additional refinements and improvements, helping us to determine adequate parameterizations for our model's future deployments.

We focus in validating the proposed method of predicting the PLSs, which is crucial for the group organization process. In particular, we verify if, under a set of assumptions, the predictions are able to make our model converge to an optimal group organization that would occur if the player's exact learning states were known.

[2] We could improve the efficiency of this first step by generating GIPs using an optimization process accounting for the previous players' states when facing different GIPs. However, as a first approach, we are still not considering such approach so that we better understand the dynamics and parameterizations which work better for our problem.

[3] For more information about chords notation, music learning books such as [22] can be consulted.

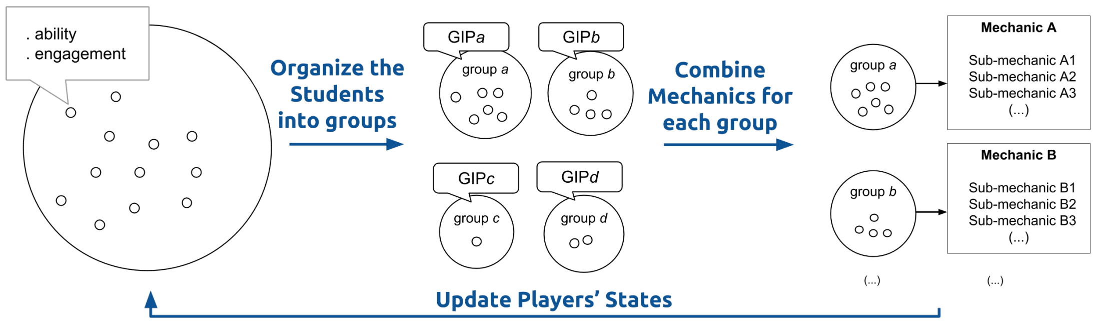

Fig. 1. Scheme representing the steps of our method.

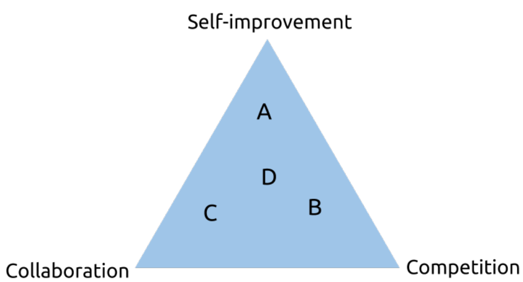

Fig. 2. Representation of our Group Interactions Profiles (GIPs) space. The letters represent examples of GIPs. The GIP represented by A majorly promotes *Self-improvement*. When valuing either *Competition* or *Collaboration*, we get GIPs such as B and C. When the weights given to all dimensions are the same, we get the GIP represented by D.

The simulations were executed on a computer using an Intel(R) Core(TM) i7-8700 CPU with 3.20GHz clock speed and 8Gb of RAM.

For simplicity, we considered our agents as students interacting with a multiplayer serious game in a secondary school class. As such, while performing our tests, we assumed the class size $C = 23$, which according to the Organisation for Economic Co-operation and Development (OECD)[4] is the average size of lower primary and secondary education classes. The assumptions considered to compute the student's progression in each iteration were:

- Some students learn faster than others and the base learning rate of the students follows a normal distribution. Moreover, the rate at which students increase ability while performing a task (task learning rate) follows another normal distribution centered on their base learning rate;
- The students have an inherent preference for a specific GIP and their engagement increase on a task is proportional to the difference between such preference and the task profile. The students' preferences are uniformly distributed and do not vary along the execution of the game[5];
- In each simulation step, the increase of the students' ability is proportional to their level of engagement.

TABLE I
LIST OF THE VARIABLES USED BY THE SIMULATIONS.

Symbol	**Meaning**	**Description**	**Values considered in the simulations**
blr	base learning rate	how much the student ability grows (speed to which the student learns)	$\mathbb{U}(0.2, 0.6)$[6]
tlr	task learning rate	variations of blr that may occur when the student faces similar tasks	$\mathbb{N}(blr, 0.05)$[6]
ip	inherent preference	GIP preferred by a student	$\mathbb{U}(0, 1)$ for all dimensions
C	class size	the number of students in the class	23
ppw	player profile window	the number of the past profiles stored by each player	30
mgs	max. group size	the maximum number os students per generated group	5
k	-	number of neighbors used by KNN in the group organization process	$1, \mathbf{5}, 24, 30$
gs	generated samples	number of generated samples considered by the organization process	$10, \mathbf{100}, 1000, 2000$
aW	ability Weight	importance given to the predicted ability for the quality of a configuration	0.5
eW	engagement Weight	importance given to the predicted engagement for the quality of a configuration	0.5

Table I presents all of the variables acknowledged by our simulations. An important note is that $\mathbb{U}(min, max)$ represents a uniform distribution function in the range $[min, max[$ and $\mathbb{N}(\mu, \sigma)$ a normal distribution function. The values of our simulations' variables mgs, aW and eW were selected resorting to empirical experimentation.

The general implementation of our simulations is presented in Algorithm 1. Before executing the group organization algorithm and simulating the students' progress, we computed the learning states of all of the students when considering

[4]https://stats.oecd.org/Index.aspx?DataSetCode=EDU_CLASS (visited May 16, 2019)

[5]We acknowledge that in real scenarios this is not always the case, specially if we target a college class where the preferences can present similarities. However we believe this assumption is adequate for a simulation context such as the one presented here.

[6]These values were picked so that the engagement values was approximately defined in the range [0, 1]

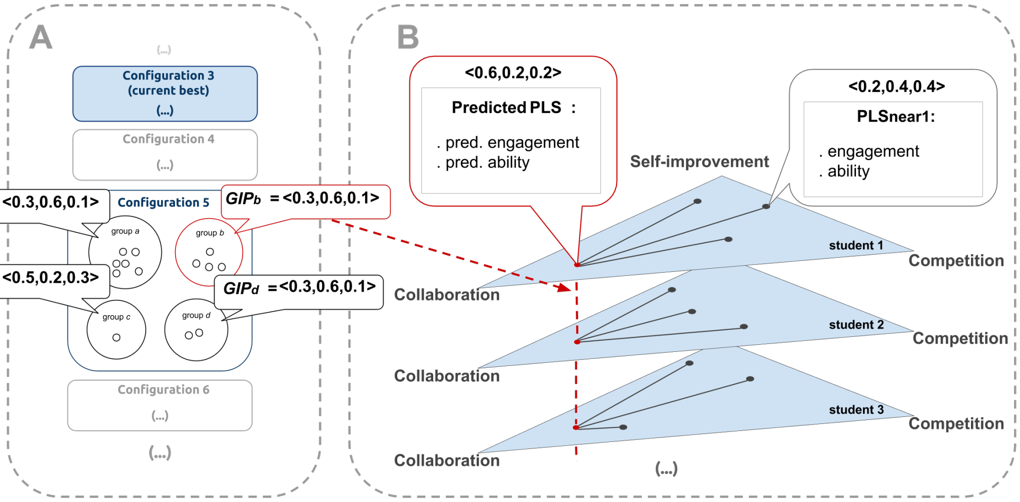

Fig. 3. Scheme representing the student organization process.

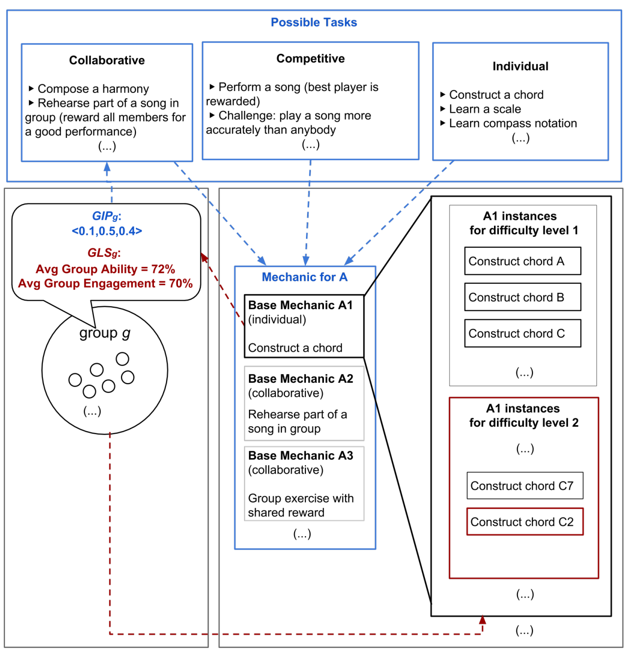

Fig. 4. Scheme representing the mechanics generation process for a group g. Firstly, several mechanics are combined according to the GIP_g (elements colored in blue). Then, mechanic instances are picked according to the GLS_g (elements colored in red).

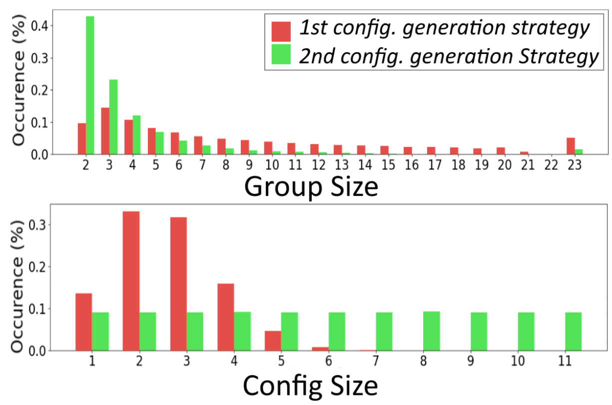

Fig. 5. Plots showing the distribution of group sizes (amount of students in each group) and the distribution of configuration sizes (amount of groups in each configuration) for each of the developed configuration generation strategies.

30 random configurations. This created sufficient data for the estimation of the quality of the configurations generated in the first iteration.

A. Simulating the students' progression and creating the students' Player Learning States

In each iteration, the engagement was computed by an exponential moving average [23] of the distances between the student's ip and the promoted profiles (with a degree of weighting decrease $\alpha = 0.5$). The ability was then incremented according to the expression: $tlr \times engagement$. The distances were computed as euclidean distances. Euclidean distances were chosen for this case due to computation simplicity, as spatial relations between points are maintained: points close/distant in euclidean space will also be close/distant in our adaptation space.

B. Strategies for generating n random configurations

As defined in the beginning of section III, our group organization method starts by producing several random group configurations (collections of groups). Two strategies were developed for this process. One of them recursively divided the students until no un-grouped students remained. The second strategy picked a random number of groups to generate and distributed the students among them. Figure 5 plots the distributions produced for the amount of students in each group and the amount of groups in each configuration after executing the strategies 1000 times. The first strategy (represented by

noend 1 General Implementation of our Simulations

```
procedure SIMULATE(List<Student> S)
    Initialization ¬
    for each Student s in S do
        Compute progression of s using 30 random configurations

    for each GIMME iteration do
        Simulate Group Organization ¬
        List<Configuration> generatedConfigs ← Generate n Random Configurations
        bestConfigQuality ← 0
        for each Configuration c in generatedConfigs do
            if quality of c > bestConfigQuality then
                bestConfig ← c

        Simulate students' progression ¬
        for each Student s in S do
            Compute the progression of s when promoting bestConfig
```

the red bars) generated configurations with small amounts of diversely sized groups and the second one (represented by the green bars) produced uniformly sized configurations containing small groups. Since we want the sizes of configurations to be uniformly distributed so that more profile possibilities are tested, the second strategy seems more adequate. College professors with more than 10 years of experience in the fields of computer science and psychology supported this choice and informed us that such strategy can be a desirable way to group students. As such, for the rest of the tests we used this strategy.

Additionally, we can observe that the second strategy rarely produced groups with more than 5 students ($< 20\%$ of all of the generated groups). Therefore, the simulations acknowledged $mgs = 5$.

C. Dynamics of the our group organization implementation

Until now, we assumed $C = 23$ and $mgs = 5$. From now on, we consider $aW = eW = 0.5$ (to estimate the quality of each configuration) and $ppw = 30$ (we believe 30 samples are enough to ensure significant coverage of the profiles space).

In order to test the dynamics of our group organization implementation, we executed the algorithm varying the gs and k values. Exploring different k values allows us to check the best KNN parameterizations and exploring different gs values allows us to guarantee that the number of samples we randomly generate without any optimization process is sufficiently good to be used by our simulations. These analysis resorted to the average ability increases experienced by the students in each iteration.

The results of executing 100 runs of the algorithm with different gs values are plotted in Figure 6. The execution times for each parameterization are also included. The plot indicates that bigger gs values lead to bigger ability increases. This can be explained by the fact that more choices mean more accurate KNN predictions. However, the differences seen from $gs = 10$ to $gs = 100$ are much higher than the differences seen from $gs = 100$ to $gs = 1000$ and so on. Bigger gs values also mean

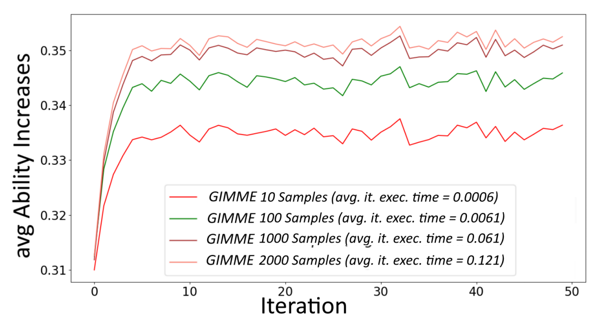

Fig. 6. Evolution of the collective ability of the students using GIMME group formation strategies with different amounts of generated samples (gs). As gs increases, the algorithm produces slightly better solutions. This comes at the cost of higher execution times.

higher execution times and so a balance has to be taken into account[7]. For simulation purposes, we considered $gs = 100$ to be enough to demonstrate the algorithm's capabilities and so used this value in subsequent tests.

The results of executing 100 runs of the algorithm with different k values are plotted in Figure 7. We can observe that GIMME converged to a higher ability value when $k = 5$. Due to this fact, this parameterization is going to be considered from now on. We believe the poor performance when high k values were used relates to the introduction of noise by acknowledging more distant samples. Oppositely, a low k is not enough to accurately predict the learning state, as we modeled the student progress to be partially based on past data.

From these results, we can also observe that our strategy rapidly converges (the values stabilize beyond approximately the 8th iteration).

[7] In a real scenario, if we consider a maximum delay of 0.1 seconds per iteration (so that the system would still give an instant response feel in the computer used for running the simulations [24, 25]), we could generate between 1000 and 2000 samples in each iteration using our hardware.

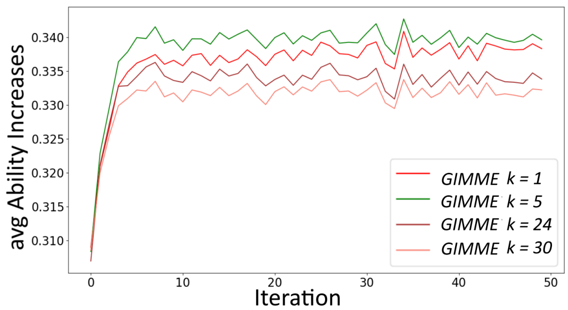

Fig. 7. Evolution of the collective ability of the students using GIMME group formation strategies with different values for k. We can observe that the average ability increase is higher when $k = 5$.

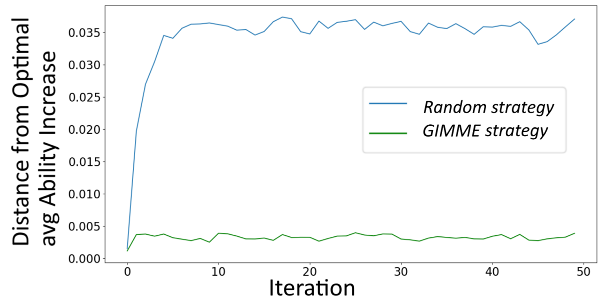

Fig. 8. Distance of the collective ability increase values between the optimal strategy and the random (blue line) and GIMME (green line) strategies . As we can observe, although both strategies start similarly, as iterations pass, unlike the random strategy, our strategy maintains values which are close to the optimal ones.

D. Validating GIMME's configuration quality estimation

Considering the analysis presented in the previous section, we wanted to test if the estimates calculated by our algorithm were able to make our method converge to an optimal group organization that would occur if the player's exact learning states were known. Therefore, two group organization strategies were compared to an *optimal group organization* that picked the group configurations which maximized the "real" students' ability increases[8] :

- The *GIMME group organization* that picked the group configurations with the higher estimated quality (using our group organization process);
- A *random group organization* that picked the groups randomly using a uniform distribution. This can be acknowledged as the simplest strategy implemented by teachers when they do not know the students.

The averaged results of executing 100 runs of the simulations applying these three strategies are plotted in Figures 8 and 9. Figure 8 plots the distance of the collective ability increase values between the optimal strategy and either our strategy or the random one. From the figure, we can observe that unlike the random strategy, our group formation method converges to a near optimal value. A similar result was observed when measuring engagement and so the plot was was omitted.

Finally, Figure 9 shows the average distances between the inherent preferences and the profiles promoted in each step. The distances vary a lot and are often high when applying the random strategy, which is expected since there are no relations between the preferences and promoted profiles when applying this strategy. The GIMME strategy on the other hand maintains low distances and remains close to the optimal strategy.

These results reveal the possible educational benefits that can be achieved using GIMME, namely in what respects to the students' gained ability and engagement.

V. Conclusion

In this document, we have proposed Group Interactions Manager for Multiplayer sErious games (GIMME), a model

[8]The increases computed by the algorithm described in section IV-A

Fig. 9. Average distances between the inherent preferences and the profiles promoted in each step. While the random strategy's distances vary a lot and are often high, the other strategies maintain smaller, more consistent distances.

which aims to improve the collective ability of players interacting with a multiplayer serious game. What sets our method apart is that we organize the players into groups, explicitly considering what interactions should be promoted for each group. After the groups are organized, game mechanics can be adapted to promote the interactions computed before.

A second contribution of the paper is the validation of several important aspects of the model using agent-based simulations. Results show that the group management system proposed has the potential for educational benefits, as it allowed high, near optimal average ability increases in contrast with a strategy organizing the students randomly (implemented by teachers when they do not know the students), which maintained low values. Furthermore, GIMME also managed to considerably approximate the promoted interactions profiles to the preferences of the virtual players.

The most important next step is to apply our model in a real context based on what we learned from our simulations. Besides, several directions of research can be taken to further improve the group organization phase. First off, we can use an optimization process instead of always generating random samples. This can further improve the efficacy of the algorithm in each iteration as only the most suitable configurations would be generated. We can also save the last n profiles in a more useful way using a grid acknowledging the most recent profiles per cell. This strategy can possibly achieve better KNN predictions when enough samples per grid are considered, as regions of the space can become overfilled when storing several learning profiles.

Despite the possibility of further improving the model using the above presented strategies, we believe in its potential as a strong basis to guide the design and improve the effectiveness of future multiplayer serious games.

Acknowledgments This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2019

References

[1] I. Granic, A. Lobel, and R. C. Engels, "The benefits of playing video games." *American psychologist*, vol. 69, no. 1, p. 66, 2014.

[2] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: defining gamification," in *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*. ACM, 2011, pp. 9–15.

[3] M. M. Bagheri, "Intelligent and adaptive tutoring systems: How to integrate learners," *International Journal of Education*, vol. 7, no. 2, pp. 1–16, 2015.

[4] H. Ai, R. Kumar, D. Nguyen, A. Nagasunder, and C. P. Rosé, "Exploring the effectiveness of social capabilities and goal alignment in computer supported collaborative learning," in *International Conference on Intelligent Tutoring Systems*. Springer, 2010, pp. 134–143.

[5] E. Butler, E. Andersen, A. M. Smith, S. Gulwani, and Z. Popović, "Automatic game progression design through analysis of solution features," in *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. ACM, 2015, pp. 2407–2416.

[6] V. Wendel and J. Konert, *Multiplayer Serious Games*. Cham: Springer International Publishing, 2016, pp. 211–241. [Online]. Available: https://doi.org/10.1007/978-3-319-40612-1_8

[7] M. Freire, Á. del Blanco, and B. Fernández-Manjón, "Serious games as edx mooc activities," in *Global Engineering Education Conference (EDUCON), 2014 IEEE*. IEEE, 2014, pp. 867–871.

[8] J. C. Burguillo, "Using game theory and competition-based learning to stimulate student motivation and performance," *Computers & Education*, vol. 55, no. 2, pp. 566–575, 2010.

[9] P. J. Muñoz-Merino, M. F. Molina, M. Muñoz-Organero, and C. D. Kloos, "An adaptive and innovative question-driven competition-based intelligent tutoring system for learning," *Expert Systems with Applications*, vol. 39, no. 8, pp. 6932–6948, 2012.

[10] G. Chan, A. Arya, and A. Whitehead, "Keeping players engaged in exergames: A personality matchmaking approach," in *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 2018, p. LBW064.

[11] J. ter Vrugte, T. de Jong, S. Vandercruysse, P. Wouters, H. van Oostendorp, and J. Elen, "How competition and heterogeneous collaboration interact in prevocational game-based mathematics education," *Computers & education*, vol. 89, pp. 42–52, 2015.

[12] A. Sarkar, M. Williams, S. Deterding, and S. Cooper, "Engagement effects of player rating system-based matchmaking for level ordering in human computation games," in *Proceedings of the 12th International Conference on the Foundations of Digital Games*. ACM, 2017, p. 22.

[13] S. Järvelä, M. Veermans, and P. Leinonen, "Investigating student engagement in computer-supported inquiry: a process-oriented analysis," *Social Psychology of Education*, vol. 11, no. 3, p. 299, 2008.

[14] E. Määttä, H. Järvenoja, and S. Järvelä, "Triggers of students' efficacious interaction in collaborative learning situations," *Small Group Research*, vol. 43, no. 4, pp. 497–522, 2012.

[15] M. Sicart, "Defining game mechanics," *Game Studies*, vol. 8, no. 2, 2008.

[16] C. Martinho, P. Santos, and R. Prada, *Design e Desenvolvimento de Jogos*, ser. Tecnologias de Informação. FCA, 2014.

[17] S. Gomes and C. Martinho, "Dynamic Interaction Mechanics CrossAnt," *arXiv e-prints*, p. arXiv:1811.07243, Nov 2018.

[18] K. Oksanen and R. Hämäläinen, "Game mechanics in the design of a collaborative 3d serious game," *Simulation & Gaming*, vol. 45, no. 2, pp. 255–278, 2014.

[19] M. Consalvo, "Using your friends: social mechanics in social games," in *Proceedings of the 6th International Conference on Foundations of Digital Games*. ACM, 2011, pp. 188–195.

[20] P. Sweetser and P. Wyeth, "Gameflow: a model for evaluating player enjoyment in games," *Computers in Entertainment (CIE)*, vol. 3, no. 3, pp. 3–3, 2005.

[21] O. Kramer, "K-nearest neighbors," in *Dimensionality reduction with unsupervised nearest neighbors*. Springer, 2013, pp. 13–23.

[22] S. Krenz, *Learn And Master Guitar*, ser. Tecnologias de Informação. FCA, 2010.

[23] R. Hyndman, A. Koehler, J. Ord, and R. Snyder, "Forecasting with exponential smoothing: The state space approach." 2008.

[24] S. K. Card, G. G. Robertson, and J. D. Mackinlay, "The information visualizer, an information workspace," in *Proceedings of the SIGCHI Conference on Human factors in computing systems*. ACM, 1991, pp. 181–186.

[25] R. B. Miller, "Response time in man-computer conversational transactions," in *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*. ACM, 1968, pp. 267–277.