

Exploiting traditional gameplay characteristics to enhance digital board games

Fulvio Frapolli*, Apostolos Malatras* and Béat Hirsbrunner*

*Department of Informatics
University of Fribourg, Switzerland
Email: {name.surname}@unifr.ch

Abstract—Computer enhanced board and card games constitute a highly engaging and entertaining activity as attested by their widespread popularity and the large amount of dedicated players. Nonetheless, when considering these digital counterparts of traditional board games it becomes evident that certain features of the latter, such as the flexibility of games and the inherent social interactions that regard the player as an active participant and not merely as the end-user of a product, have been in general neglected. In particular, the ability to customize and adapt games according to the players' needs is one of the key factors of their success and should thus not be ignored when porting them to the digital environment. In this paper we present our work on a holistic framework titled FLEXIBLERULES that addresses these limitations by bringing the intrinsic flexibility of board games played in the traditional environment into the digital one, while additionally aiming to increase the level of involvement of end-users, the players, with basic programming skills, in the game development process. A multi-faceted user evaluation study validates the effectiveness of our approach and provides encouraging results for further research.

I. INTRODUCTION

Recent advances in the computer science domain have undeniably influenced (to a lesser or greater degree) the way in which social activities are being conducted. Since playing board and card games is one of the longest-living social activities of human beings which, for centuries, has taken place only in the traditional physical environment it is not surprising that it has also recently “invaded” the digital world [1], [2]. In so doing this type of games have gained interesting functionalities which improve the gaming experience (performing of mundane tasks, calculating intricate winning conditions, adding sound and dynamic effects to the gameplay, etc.) but, at the same time, they have dramatically diminished their degree of flexibility in terms of customization (house rules, balancing of the complexity for certain players, balancing of the game duration, etc.) [3] and have lessened the degree of social interactions that occur when playing or formulating games [4].

The flexibility offered by board games played in the traditional environment, in terms of customization of rules and variants, is not only one of their distinguishing characteristics [5] but also plays an important role in the success that board games have experienced throughout history and should thus be also supported in the digital environment [6]. The significance of this feature has also emerged as a key requirement, which is currently missing in existing computer enhanced board games, in a recent survey that we conducted amongst a large

population of board game players and human-computer interaction experts [7]. Additionally, the key role that the players have in the physical environment in modifying game aspects to increase their level of enjoyment, as highlighted in [8], should not be neglected. In this respect, when shifting to the digital environment end-user involvement, namely the ability of players with basic programming skills (i.e. understanding of fundamental concepts of algorithmics, such as *if ... then ... else* constructs or *for* loops) to customize a board game, should be strongly promoted.

In order to achieve these goals, we present here the FLEXIBLERULES framework which enables runtime modification of all aspects of a game in a straightforward manner. The core of this framework is a novel approach for modeling board games, which replicates the one used to explain a game to human users and is thus perceived as natural by players. An important part of our work involves an architecture capable of handling all run-time aspects of gameplay, allowing for rapid game prototyping. Furthermore, to increase the involvement of end-users, a high-level development environment is presented which allows handling of the different aspects of board game design (graphical representation, initialization, rules, etc.) in a visual and thus simpler manner. To assess the fulfillment of our research objectives we conducted a qualitative user evaluation study demonstrating the benefits of our approach for increasing end-user involvement in game development through our intuitive model. The results of this evaluation also indicated that the development environment partially satisfies its requirements because of its prototype nature, which still presents usability issues that are the basis for future enhancements. Additionally, a second evaluation in a real-world setting gave very positive results in terms of bringing the flexibility of the traditional environment to the digital one, as expressed by the levels of fun and satisfaction experienced by the users.

The rest of this paper is organized as follows. Section II presents an overview of related work on digital board games and the potentials of improving them by taking into account features of traditional gameplay. An overview of our proposed approach to model digital board games is given in Section III, while Section IV introduces the FLEXIBLERULES framework that enables flexible and cross-platform gameplay. In Section V we describe in detail the user evaluation experiments that we conducted in order to establish the viability of our approach

and we report on the corresponding findings. Finally, Section VI summarizes the work presented in the paper and provides insights to future research directions.

II. RELATED WORK

One of the most intriguing characteristics of traditional board games is the fact that their rules are not rigid and are subject to negotiation between the players [5]. This flexibility in modifying game rules provides the players with a degree of freedom concerning their gaming sessions, allows for novel and potentially more engaging game variants to be conceived and enables games to become more personalized towards the preferences of the players [8]. While this level of flexibility is inherent to traditional board gameplay and is responsible for the success and fun of these games [9], [10], it has been by and large neglected when porting these games to the digital environment where the modification of the rules is cumbersome if not impossible (at least not without requiring advanced IT skills). Supporting such flexibility in the digital environment would thus in terms of social interactions assist in providing a comparable gaming experience to that of the traditional one.

In this line of thought, gaming platforms such as Vassal Game Engine [11] and ZunTzu [12] provide players with a virtual gaming board exactly emulating the graphical appearance of the corresponding physical board game, while letting the players collectively decide on the rules by means of communication channels such as e-mail or chat. Nonetheless, such an approach almost completely negates the benefits that the digital environment has to offer, such as record keeping, performing of mundane tasks, rules enforcing, etc. At the other end, intricate game development environments such as Microsoft's XNA [13] not only provide utilities for the design of games, but also enable programming of advanced functionalities and of the entire game logic. The limitation of tools such as the latter is the requirement to be proficient in programming languages (C# in the case of XNA) in order to be able to harness their advantageous features and hence they lack support for end-users, namely the players. This hindrance is addressed by projects such as Games Factory [14] and Game Maker [15] that provide an easy-to-use visual programming environment for end-users to create their own games. On the downside, these projects are mostly focused on arcade games (which typically have a simpler rule set than board games) and they do not support runtime modification of game rules. This feature is not only required for providing the players with the same degree of flexibility as in the traditional environment but also for promoting their involvement in the process of designing and implementing board games as was shown by the Kodu [16] project that nonetheless mainly focuses on 3D arcade games.

III. MODELING APPROACH

The previous discussion has highlighted the limitations of existing digital board games regarding flexibility, a notion which is prominent in the traditional gaming environment and

on which the popularity of board games and the enjoyment experienced by players is dependent. In this direction, our research work involves a holistic framework, called FLEXIBLERULES, aimed at facilitating the development of computer enhanced tabletop/board games targeting the game designer as well as the end-user, namely the players since it is according to their needs and preferences that the game will be adapted. The core of the FLEXIBLERULES framework is a conceptual model to design board games, the main goal of which is to promote modularity and clarity: the user should be able to quickly identify what is to be modified and where in order to change something in the game.

It is evident that the simpler and most straightforward way to model and implement a board game from an end-user perspective would involve replicating the approach used to explain it to a human player. The latter traditionally consists in describing the behavior of each game element along with the rules it follows. When playing on a traditional physical setting, knowing the role of each object, or group of objects in terms of what behavior it is able to perform (e.g. all the knights in chess follow the same template behavior), which conditions it has to meet in order to perform the chosen behavior and what secondary outcomes this behavior can produce, is equivalent to knowing the game. The game definition is thus naturally distributed within the objects composing the game. Nonetheless, the game objects themselves do not carry any information or awareness of the game logic, but are a mere interface allowing the player to recognize the abstraction that they represent and interact with it. The whole game logic is "stored" within the players' brains, which collectively are the one and only referee of game play, enforcing the game rules.

The main purpose of the FLEXIBLERULES model is to port this way of thinking about a game and its rules from the physical into the digital environment. The decoupling between the graphical appearance and the logic of the game objects is preserved by splitting their modeling in two parts: logic and representation. While the representation part will model the interface with which players can interact with the game objects, the logic part will define the behavior that the object is able to perform. The logic of the game will no longer only be "stored" within the players, but will also be distributed within the logical part of the objects which will be aware of their behavior. Moreover, both the logic and representation modeling will occur at an abstract level; this degree of abstraction can be reached by means of reusable object templates (e.g. all pawns in chess will have the same logic behavior and all white pawns the same representation). However, the game rules, in terms of pre-conditions on the behaviors (e.g. a pawn cannot move if the next forward square is occupied) and secondary outcomes of the game (e.g. if any piece is captured it is removed from the game), are modeled as separate aspects and are thus not part of the objects themselves. Instead, game rules can be applied to different objects and can concern different behaviors providing therefore full flexibility regarding when and how they will be applied.

The proposed game model (in depth presented in our

previous work [17]) is composed of a set of entities with precise properties and behaviors that collectively define the game and enable players to engage in the gameplay. All behaviors are triggered by the reception of messages, which are information containers that are exchanged between entities during execution and are the only way to perform coordination and communication tasks. In this respect, the execution flow always starts with a message sent to an entity to trigger a specific behavior. The different aspects of the game, such as the logical behavior of the different game objects, their graphical representation, and rules limiting the allowed behaviors as well as defining the outcome of each action are modeled separately and can be freely modified during gameplay. We believe that such a separation of concerns allows for a more natural way to define the logic behind a board game as well as its graphical aspect and is thus more easily understandable by typical board game players with limited IT skills. Furthermore, this separation enables the much desired flexibility by allowing rules to be plugged or unplugged without having to modify the logic or graphical representation of game entities.

IV. FLEXIBLERULES DEVELOPMENT ENVIRONMENT

In order to prove the suitability and validate the benefits of our modeling approach, presented in section III, we developed a language which reflects the fundamental principles of our proposed model. The need to introduce a novel language is spurred from the observation that current board game development efforts make use of languages that have either a scope that is too general (e.g. Unity [18], XNA [13]), too restricted (e.g. Game Maker [15], Games Factory [14]) or avoid providing abstractions dealing with the game logic and merely focus on simplifying the management of the graphics hardware (e.g. Pygame [19], Lua, Ruby [20]). The FLEXIBLERULES language is specifically targeted at board game design and is presented in detail in our previous work [17]. It is noteworthy to mention that this language is integrated and implemented within a broader architecture capable of handling all run-time aspects of the game execution and allowing for game prototyping. The implementation of the runtime architecture has been driven, similarly to the modeling part, by the pursuit of the maximum degree of flexibility and end-user involvement. Specifically, the architecture supports flexibility during the execution of the game, allowing for runtime modification of all game aspects (game elements, rules, etc.) which are automatically and dynamically integrated and immediately taken into account. It should also be noted that the whole architecture has been implemented in Java in order to benefit from its inherent portability and thus enforcing a further level of flexibility, that of the choice of the playing environment (interactive tables, smart-phones, desktop computer, etc.). The architecture enables the compilation and execution of FLEXIBLERULES games that have been coded using the aforementioned language.

Figure 1 illustrates the main components of the architecture, i.e. the *execution manager*, the *code provider*, the *compiler*, the *rule manager* and the *virtual machine*, as well as their runtime

interactions. In line with our modeling approach, all behaviors are triggered by the exchange of messages. Every message is processed by the execution manager, which first retrieves the current state of the messages' recipient by querying the game situation, which stores all the data regarding the game entities (e.g. current state, values of properties). Once the execution manager has all the needed information, i.e. message label and entity's state, it forwards this information to the code provider and requests the compiled version of the corresponding behavior to execute (list of instructions). In order to obtain an up-to-date list of instructions, the code provider exploits the operations of both the compiler and rule manager modules, which recompile the requested behavior subject to any runtime modifications. Moreover, the execution manager launches a new virtual machine to execute the generated list of instructions that could potentially lead to the creation of new messages, the rendering of the representation layer or the modification of the game situation. It should be noted that this execution flow is a closed loop process: every newly sent message will create a new virtual machine thus eventually building a hierarchy of virtual machines each executing a specific behavior.

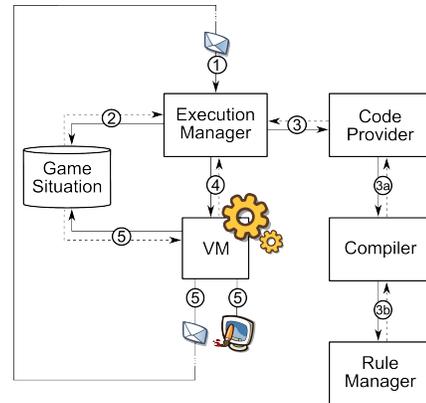


Fig. 1. FLEXIBLERULES architecture

Despite the merits of having an advanced runtime architecture and a domain-specific language both targeted at computer enhanced board games, it is nonetheless evident that their applicability remains mostly limited to users with an IT background. In order to further simplify the game prototyping process and thus promote end-user involvement, we present here the FLEXIBLERULES game development environment that has been implemented on top of the presented architecture. The purpose of this environment is to provide a high level graphical user interface for managing a game project in all its facets and hiding from the user the low level details of the game project structure, while additionally providing access to high level tools dealing with the specification of the different aspects of a game.

The main entry point of the framework is the FLEXIBLERULES Game Manager, which allows for creating, modifying and executing board games by means of a graphical user interface. The goal of the Game Manager is twofold; on one

side it hides the details of the game projects' structure at the filesystem level by automatically generating and handling the different files of the game, and on the other side it provides access to a set of visual tools, each developed for handling a specific game aspect, such as the definition of the game behavior and the graphical appearance of the game as well as the game rules. We focus here on the tools targeted at improving end-user involvement and rule flexibility, the Visual Code Editor and the Rule Editor respectively.

While visual tools are available to facilitate the design of the game's user interface and the initialization of its entities, advanced functionalities such as specifying the game logic (i.e. behaviors of entities) and the rules of the game require coding using our domain-specific language. Confronted with the necessity of satisfying the needs of both advanced developers and end-users with basic programming skills, we developed a visual environment, referred to as the Visual Code Editor (Figure 2), which aims at simplifying the editing of the FLEXIBLERULES code. All the functionalities of the language are displayed in the upper panel as blocks and are grouped in thematic categories. Each functionality can be used by dragging it within the selected behavior and dropping it in the desired area. Highlighting of the possible and valid areas within the code where a command block can be dropped is enabled in order to guide the user and hinder the generation of erroneous instructions. It is also possible to request the documentation for a specific functionality by dragging the corresponding block onto the help panel located on the right side. The Visual Code Editor draws inspiration from the Alice project [21], a 3D programming environment using a drag-and-drop interface aiming at facilitating the programming experience. This leads to a reduction in the frustration for users with basic programming skills, who are then able to focus on the concepts of objects and encapsulation, rather than dealing with the details of punctuation (parentheses, commas, etc.) [22].

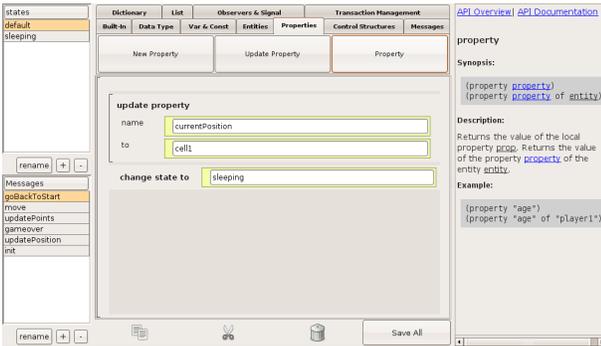


Fig. 2. Visual Code Editor

In the context of the FLEXIBLERULES framework the managing, even at runtime, of the rule aspects of the game is of paramount importance, and thus should be facilitated by a high level graphical interface that can easily be manipulated by end-users, i.e. board game players. This interface is provided

by the Rules Editor as illustrated in Figure 3 for the case of the Snakes and Ladders game that has been used throughout our evaluation as will be shown in the next section. The Rules Editor allows managing the creation and modification of rules (either disallow a particular behavior or make it produce some secondary outcome), their grouping into high level rules and variants, as well as their application during the game. The players can thus at any moment, before or during gameplay, decide on the variant they prefer by simply enabling or disabling rules accordingly. Moreover, by means of the Rules Editor the players can define their own rules and thus play the desired game variant and even change it at runtime, experiencing the full degree of flexibility that is normally provided only in the physical environment.



Fig. 3. Rules Editor

V. USER EVALUATION

In the previous sections we presented the FLEXIBLERULES integrated environment for implementing and playing computer enhanced board games that better reflect the advantages of traditional gameplay, i.e. the one that takes place in physical environments. To confirm that our work effectively satisfies the identified challenges, we conducted a preliminary evaluation targeting all of the issues raised. In particular, our evaluation aims at assessing the FLEXIBLERULES framework in terms of facilitating the development of computer enhanced board games, promoting player involvement in their design and implementation and also improving the level of satisfaction concerning social interactions that emerge through playing games. In this respect, as a preliminary validation of the expressiveness of our proposed approach we implemented several games (Awele, Go, Connect Four, Tic Tac Toe, Turn the Tide™, Himalaya™ and Snakes and Ladders) exposing different characteristics in terms of complexity, game mechanics and management of the user interface [17]. The selection of these games was made based on their orthogonal characteristics and thus they serve as a validation of how FLEXIBLERULES can facilitate the development of a wide range of computer enhanced board games.

Furthermore, we conducted a user evaluation to obtain proper feedback on the usefulness as well as the usability of our approach and the integrated environment in a scenario where the user is asked to play the role of a game designer

who wishes to modify existing games or create new ones from scratch. The analysis of the corresponding results highlighted the usefulness of the proposed approach which, combined with the identified clarity of the underlying model contributes to higher players’ involvement in game design despite the presence of usability issues. We also carried out a qualitative evaluation of the playing environment in a multiple user scenario where players participate in a game session in which the rules can be changed dynamically, in order to assess the benefits of allowing for rule flexibility. The results illustrated that providing this level of flexibility, which is generally missing in existing digital games, can result in a significant enhancement in player enjoyment to a level comparable to the one of traditional physical settings.

A. Player Involvement

The first user evaluation that we conducted tried to involve the players in the development process of a board game by asking them to perform some tasks of incremental complexity and aimed at gathering information about the usability of the development environment. As suggested in [23], the notion of usability encompasses efficiency (assessed here by means of task completion time), effectiveness (assessed by means of successfully completing a task or not and its perceived complexity) and user satisfaction (assessed by means of the fun level experienced by the users). Moreover, we also evaluated the usefulness of our modeling approach, in terms of the ability of the users to clearly identify different game aspects.

In accordance to [24], who recommend using between three and five users per test as a way of simplifying user evaluation while gaining almost the same benefits as one would get from more elaborate tests with large numbers of subjects, we conducted our experiments with 5 users. The users were first individually introduced to FLEXIBLERULES by means of a 20 minute presentation of both the adopted modeling approach and the development environment and were then asked to complete a tutorial which explained the basic functionalities. This activity allowed them to get familiar with the different tools available in the FLEXIBLERULES development environment. Once the users had successfully completed the tutorial, they were asked to visit our labs to perform the evaluation that took place on a dedicated machine.

The evaluation comprised 10 tasks of incremental complexity grouped into three phases. During the first phase the user was asked to add some rules to a simple dice game (i.e. add a winning condition, add a new rule). In the second phase a partial implementation of the game Snakes and Ladders was given to the user, who was then asked to modify it or add some missing features (i.e. extend number of players, modify graphical appearance, add snakes and ladders functionalities). Finally, in the third phase the user was faced with the task of creating a simple game from scratch (i.e. create two entities, model their logic and representation, implement and modify their behavior). It is worth mentioning that the tasks were not of equivalent complexity, but were rather aimed at putting the user in different situations and thus forcing her to make use of

different functionalities in the framework. Before every task the users were asked to express an estimation of its expected difficulty level. After having completed the task or given up on it, we gathered information on completion time, complexity and fun, in order to evaluate the usability of the framework as previously discussed.

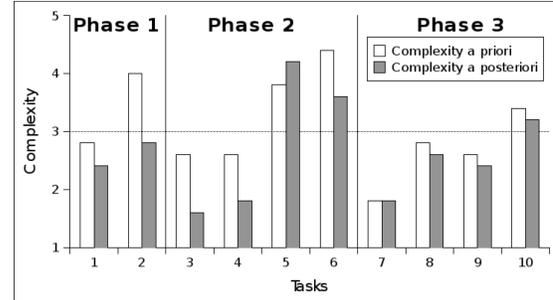


Fig. 4. Tasks’ complexity (a priori vs. a posteriori)

The modeling approach that distinguishes the different aspects of the game was very positively evaluated in terms of clarity. Users never had problems in finding the parts of the game projects that were to be modified in order to achieve the specific task goals. Moreover, the tasks revealed themselves as being easier to perform than expected, as shown in Figure 4 (using a Likert scale from 1 to 5 where 1 denotes a very easy task and 5 a very hard one; the results represent average values with a standard deviation of 0.43 for both cases), the latter being attributed to the clarity of the modeling approach. Nonetheless, the users pinpointed issues with the usability of the development environment (ranked as poor in terms of usability of the tools), which in their opinion remains difficult to get familiar with in a short amount of time. The latter statement is not surprising, taking into account the prototype nature of the development environment and the inherent complexity of learning the functionalities of a suite of tools as well as a new language. Despite the aforementioned issues, it should be pointed out that the users managed to achieve all but one of the tasks. Additionally, an encouraging result involves the users’ assessment of the usefulness of the framework, which has been judged as being between good and very good overall.

It is noteworthy to emphasize the assessment of Task 4 in Phase 2, since it involved the introduction of a new rule in an existing game and thus closely relates to the driving motivation of this research on supporting rule flexibility. All the users were able to integrate the rule within the game, although the task was considered as a hard one (but it emerged as being not as hard as expected). Being successful in such a hard task made it fun to perform although reaching the goal required a significant amount of time (1 hour on average, with the users pointing out design issues with our prototype framework as detailed later). This apparently paradoxical result is due to the fact that users thought that what they were doing was very useful in terms of gameplay and thus reached a high degree of satisfaction despite the large amount of time that they spent

trying to fulfill the task.

In terms of the design of the FLEXIBLERULES development framework, most of the users stated a lack of interoperability and integration between the different editors, in particular between the Visual Code Editor and the editors regarding the graphics and the initialization. This lack of coordination between tools was due to the adopted incremental development process. Each editor was conceived as a standalone tool aimed at facilitating a particular task (entities' initialization, graphical appearance, etc.), which, while being in line with our view of clearly separating each concern, evidently created the problems highlighted by the users. Resolving the latter is the basis for future work on the development environment.

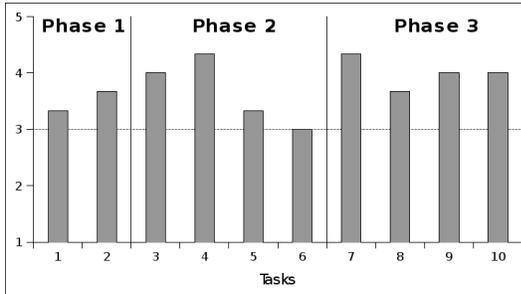


Fig. 5. Visual Code Editor evaluation

Furthermore, users had problems getting familiar with the different editors with the exception of the Visual Code Editor which was the only one which had an integrated help functionality. For the other editors users had to rely only on the information contained within the tutorial. In this respect, a promising observation can be seen in Figure 5 (using a Likert scale from 1 to 5 denoting increasing level of comfort), which illustrates the learning phase concerning the Visual Code Editor. Users reached a satisfactory level of comfort after performing the first four tasks. The decrease in the comfort level that can be seen regarding tasks 3 and 4 of Phase 2 can be attributed to the aforementioned lack of integration with other tools that were to be used during these tasks in conjunction with the Visual Code Editor.

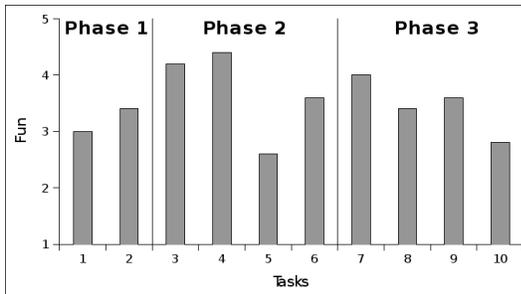


Fig. 6. Fun experienced while performing tasks

The levels of enjoyability that the users exhibited throughout the evaluation were almost always above average (more than 3), as depicted in Figure 6, with the sole exception of task

3 of Phase 2 which 40% of the users did not manage to complete. The latter was assessed as being the least fun task, with 4 users giving it a rank of 2 and one user ranking it with 5; the users attributed this ranking to the need to use the graphics editor of the FLEXIBLERULES framework in this particular task. Whereas tasks involving the logic of the game (balancing the game and making it more appealing in terms of gameplay) were deemed to be fun, the modification of graphical aspects was not appreciated by the users, mainly due to the prototype nature of our framework and the associated usability drawbacks.

Summarizing the key findings of the single-user evaluation, we underline the identified usefulness of our novel modeling approach and the overall positive impressions obtained by the users, who also provided us with useful feedback in particular in terms of the usability of the FLEXIBLERULES framework and its development tools. We assert that player involvement will become even more facilitated by further enhancing the latter features.

B. Rule Flexibility

The main purpose of the second set of experiments was on one hand to evaluate the usefulness of the flexibility provided by our framework and on the other hand to assess the experienced user satisfaction as reflected by the amount of fun had when playing a game with flexible rules. The multi-player user evaluation involved a real-world scenario where users sat around an interactive table (MERL DiamondTouch™) and played a board game, as shown in Figure 7, for instance Snakes and Ladders, developed with the FLEXIBLERULES framework. We conducted our experiments on 6 groups of three users each, for a total of 18 individual users (13 men and 5 women) in the age group ranging between 25 and 35. We identified four categories of users based on how frequently they used to play board or card games, and hence their experience with them. In particular, we distinguish between users that play rarely, occasionally (i.e. at least once per month), at least once per week and hard-core players (i.e. playing every day). Groups were randomly composed of players of different categories. The experiments took place in our lab during the course of one week. The mechanics of the game, its basic rules as well as a set of proposed optional rules aimed at rendering the game more interesting were first explained to every group by means of an introductory game session. After this introduction to the gaming environment the actual evaluation took place. Users were asked to play the game three times with different settings: *basic*, *custom* and *dynamic*.

- The *basic* game session consisted of playing Snakes and Ladders with the traditional rule set. In order to limit the duration of the experiment we placed an upper bound on the number of game turns. To avoid completing the game without a winner, an additional winning condition was added: after the completion of the last turn, the players nearest to the final cell were declared the winners.
- In the *custom* session the users were asked to discuss among themselves and build their own game variant



Fig. 7. Typical multi-user evaluation setting

by choosing their preferred optional rules among the proposed ones. The session was then played up until the last turn without the ability to further change the rules. All the modifications were performed by an external game master not participating in the game.

- During the *dynamic* session the users, besides being able to customize the game prior to the session, were also given the possibility to convene and agree on further modifications of the game rules during the game session when asked to do so by the external game master.

To assist users in deciding on their customization, we provided them with a list of all possible optional rules. In order to ensure that the experiment was independent of the order in which the sessions were played, each of the six groups of users followed a different order out of the six possible ones. Upon completion of the experiment users were asked to fill out a questionnaire to capture their feedback concerning their gaming experience, their preferred session, the appreciation of the FLEXIBLERULES framework and the options that it provides, their perceived fun, etc.

One of the most important outcomes that the analysis of the results yielded is that, when compared to the traditional physical environment, the experience with the FLEXIBLERULES framework deployed on an interactive surface was rated between good and very good by all the users. Users pointed out that the typical social interactions that occur before and during a traditional board game session have been preserved throughout the experiment and attribute this to the tabletop setting, which preserves the face to face interaction of the traditional environment (“*adds fun and conviviality*”), and to the ability to discuss and decide on which rules to integrate in the game (“*very cool to play and decide rules together*”). Additionally at the game play level, users appreciated the rapidity of preparing the game settings, as well as the added value of sound and animations that in their opinion improve the game experience when compared to the traditional environment.

While the usefulness of the ability to modify every game aspect prior to the game session (custom) was rated between good and very good among every player category (Figure 8), the fact that the framework allows to modify rules and parameters even during the game session (dynamic) was

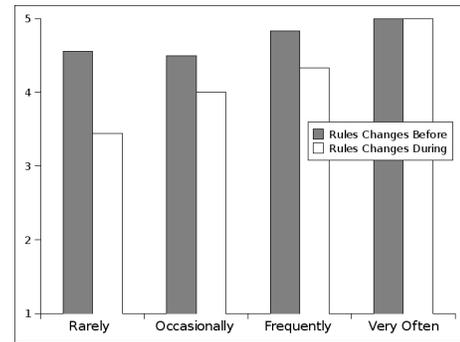


Fig. 8. Usefulness of support for flexible rules

mostly appreciated by users who play on a frequent basis. The latter observation is not surprising because occasional players have probably been confronted less frequently with game situations where it was necessary to modify a rule during the game. Nonetheless, even occasional players appreciated the ability to modify a simple parameter at any time, for example to restore an item used by mistake or increase the number of game turns.

An interesting observation regarding the driving motivation for rule changes that was revealed by our experiments is that in the beginning the users were mostly spurred by curiosity, i.e. being interested in how a certain rule would affect game play and eventually increase fun. Typical comments included: “*Collecting and stealing money sound interesting*”, “*I want to add something new (not seen yet)*” and “*Try out rules to see how it would affect the game*”. Once they got familiar with the game and understood the different implications of the optional rules, the driving motivation shifted to adding some strategy to a game that is normally based only on luck. In this respect, their comments were: “*Increase the possibility to play many different strategies*”, “*Increase the complexity of the game*” and “*I wanted to render the game more strategic*”. These results are in our opinion very encouraging concerning the need of players to have a system that allows them to modify a game to fit their preferences, which might significantly vary over time and thus increase the appeal and the lifetime of the game compared to a static version.

Figure 9 shows how the players evaluated the three different game sessions in terms of fun. In general, users had more fun while playing the custom session, which was consistently rated as being between fun and a lot of fun. There is an evident difference in the ranking of fun between inexperienced and experienced players, since the former prefer the custom session while the latter the dynamic one. The reason for this is that inexperienced players tend to prefer not to modify the game rules during the session, not wanting to be forced to modify their strategy during the game. It should be noted that, when asked whether they would like to play any of the three sessions again, nobody picked the basic one or even chose to stop playing. Rather 60% of the users chose to play a custom session, while 40% of them would prefer a dynamic one. It is also interesting to notice that the latter choice is neither

correlated with the fact that players have won or lost in a specific setting, nor with the order of the sessions, thus proving that the main motivation for playing remains having fun.

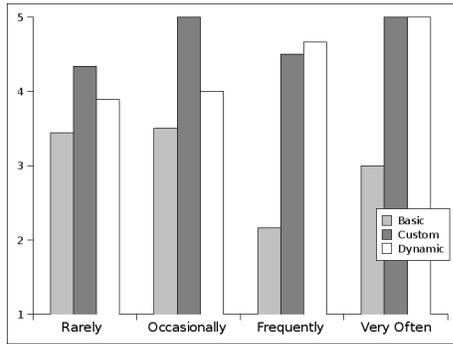


Fig. 9. Fun experienced during multi-user evaluation

Overall, the evaluation results were very positive concerning the need of players to have a system that allows them to adapt a game according to their preferences, thus increasing the fun and consequently the longevity of playing it.

VI. CONCLUSIONS

When examining the manner in which activities take place in either their traditional settings or with the improvements provided by digital media, it becomes evident that the numerous related benefits of the latter also come with shortcomings in reaching a comparable degree of enjoyment to that found in physical environments. This paper dealt with this issue in the representative field of board games and aimed at establishing solutions to effectively address the porting of the gameplay experience of the physical environment to the digital one by means of the proposed FLEXIBLERULES framework.

Our driving research hypothesis involved the inclusion of players in the design and development of board games and the provision of a full degree of freedom to modify game rules, a distinguishing characteristic of traditional board gameplay settings. Based on the user evaluation studies that we conducted, we assert that our work, seen within a broader perspective comprising interactive surfaces to emulate the actual physical experience, can profoundly ameliorate the enjoyment that players exhibit when playing digital board games and can lead to greater user satisfaction, both of which constitute main characteristics for the success and popularity of digital games. In this respect, we can effectively claim that our work can be deemed as a positive contribution towards achieving more realistic and convivial gaming experiences.

Our future work will mainly focus on further rendering the game development framework more easily accessible to end-users with very limited IT skills. According to the comments and suggestions that emerged during the user evaluation, the interoperability of the different tools comprising the game development framework should be improved. In addition, our future research will consider improving the gaming experience by taking into account the advances in the mixed use of tangi-

ble objects within a computer enhanced digital environment, which has found applicabilities in the game domain [2].

VII. ACKNOWLEDGEMENTS

The authors would like to thank Amos Brocco and Denis Lalanne for their support and useful comments. This research has been partially supported by the Swiss Hasler Foundation, project Nr. 09090-0630-011.

REFERENCES

- [1] E. Tse, S. Greenberg, C. Shen, and C. Forlines. Multimodal multiplayer tabletop gaming. *Computers in Entertainment*, 5(2):12, 2007.
- [2] J. Leitner, M. Haller, K. Yun, W. Woo, M. Sugimoto, M. Inami, A. D. Cheok, and H. D. Been-Lirn. Physical interfaces for tabletop games. *Computers in Entertainment*, 7(4):1–21, 2009.
- [3] F. Frapolli, B. Hirsbrunner, and D. Lalanne. Dynamic rules: Towards interactive games intelligence. In *Tangible Play: Research and Design for Tangible and Tabletop Games*, pages 29–32. IUI, January 2007.
- [4] C. Magerkurth, T. Engelke, and M. Memisoglu. Augmenting the virtual domain with physical and social elements: towards a paradigm shift in computer entertainment technology. *Computers in Entertainment*, 2(4):12–12, 2004.
- [5] K. Salen and E. Zimmerman. *Rules of Play: Game Design Fundamentals*. The MIT Press, 2003.
- [6] T. Bekker, J. Sturm, and B. Eggen. Designing playful interactions for social interaction and physical play. *Personal Ubiquitous Computing*, 14(5):385–396, 2010.
- [7] F. Frapolli, A. Brocco, A. Malatras, and B. Hirsbrunner. Flexiblerules: A player oriented board game development framework. In *3rd Intl Conf. on Advances in Computer-Human Interactions, ACHI, IEEE*, 2010.
- [8] C. Magerkurth, M. Memisoglu, T. Engelke, and N. Streitz. Towards the next generation of tabletop gaming experiences. In *GI '04: Proceedings of Graphics Interface 2004*, pages 73–80, 2004.
- [9] I. Soute, P. Markopoulos, and R. Magielse. Head up games: combining the best of both worlds by merging traditional and digital play. *Personal Ubiquitous Computing*, 14(5):435–444, 2010.
- [10] H. Iida, N. Takeshita, and J. Yoshimura. A metric for entertainment of boardgames: its implication for evolution of chess variants. In *Intl Workshop on Entertainment Computing*, pages 65–72. Springer, 2002.
- [11] The vassal game engine. <http://www.vassalengine.org/>, Sept. 2010.
- [12] Zuntzu: The online boardgaming platform. <http://www.zuntzu.com/>, Sept. 2010.
- [13] S. Cawood and P. McGee. *Microsoft XNA Game Studio Creator's Guide, Second Edition*. McGraw-Hill, Inc., New York, NY, USA, 2009.
- [14] Clickteam Development Group. The games factory 2. <http://www.clickteam.com/website/usa/the-games-factory-2.html>, Sept. 2010.
- [15] YoYo Games. Game maker. <http://www.yoyogames.com/make>, Sept. 2010.
- [16] Microsoft Research. Kodu Official homepage. <http://research.microsoft.com/en-us/projects/kodu/>, Sept. 2010.
- [17] F. Frapolli, A. Brocco, A. Malatras, and B. Hirsbrunner. Decoupling aspects in board games modeling. *International Journal of Gaming and Computer-Mediated Simulations (IJGMS)*, II(2):18–35, 2010.
- [18] Unity 3d game development tool. <http://unity3d.com/>, Sept. 2010.
- [19] W. McGugan. *Beginning Game Development with Python and Pygame: From Novice to Professional*. Apress, Berkely, CA, USA, 2007.
- [20] T. Gutschmidt. *Game Programming With Python, Lua, and Ruby*. Premier Press, 2003.
- [21] M. Conway, S. Audia, T. Burnette, D. Cosgrove, and K. Christiansen. Alice: lessons learned from building a 3d system for novices. In *CHI'00*, pages 486–493. ACM, 2000.
- [22] S. Cooper, W. Dann, and R. Pausch. Teaching objects-first in introductory computer science. In *34th SIGCSE technical symposium on Computer science education*, pages 191–195. ACM, 2003.
- [23] E. Frøkjær, M. Hertzum, and K. Hornbæk. Measuring usability: are effectiveness, efficiency, and satisfaction really correlated? In *CHI'00*, pages 345–352. ACM, 2000.
- [24] J. Nielsen. Guerrilla hci: using discount usability engineering to penetrate the intimidation barrier. *Cost-justifying usability*, pages 245–272, 1994.