Balancing Act: Leveraging 'PARAMETERS' Serious Game as a Tool for Mastering Game Design in Higher Education

Jens-Martin Loebel
Department of Economics
Magdeburg-Stendal University of Applied Sciences
Magdeburg, Germany
jens-martin.loebel@h2.de
https://orcid.org/0000-0002-8960-1042

Jochen Koubek
Department of Media Studies
University of Bayreuth
Bayreuth, Germany
jochen.koubek@uni-bayreuth.de

2024 IEEE Gaming, Entertainment, and Media Conference (GEM) | 979-8-3503-7453-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/GEM61861.2024.10585553

***Abstract*—This paper introduces "PARAMETERS", a serious game implementation, as a novel pedagogical tool for enhancing game design and balancing education within higher education settings, specifically in game development courses. Recognizing the complexity and educational value of game balancing—a process crucial for ensuring fairness, engagement, and replayability in games—this research aims to bridge the gap between theoretical knowledge and practical application in academic curricula. By integrating PARAMETERS into a game design curriculum, we examine its effectiveness in facilitating active learning and mastery of game design principles among students. Our methodology leverages PARAMETERS' simplified RPG mechanics, focusing on numerical interactions and stat management, to allow students hands-on experience with core game development tasks without the distractions of complex graphics or narratives. The outcomes of this integration, evaluated through a detailed case study, indicate a significant enhancement in students' comprehension of game balancing concepts and their application in real-world scenarios. Our study contributes to the discourse on game-based learning in technical education by demonstrating the pedagogical benefits of using serious games like PARAMETERS. It offers insights into effective strategies for embedding practical game design exercises into higher education, underscoring the potential of serious games to complement and enrich traditional learning approaches in this field of game development.**

Keywords—game balancing, game design, serious games

I. Introduction

Game balancing is recognized as a crucial yet complex aspect of game design, characterized by the lack of a universally accepted definition. It involves the tuning of a game's rules, levels, difficulty, numbers, and algorithms to achieve specific goals like making a game winnable, fair for all players, challenging, and replayable. Despite the critical role of game balancing in creating successful games, no two authors share an identical understanding of the concept [1].

Video games can be balanced in several aspects, such as presentation, narration, game experience and game mechanics [2]. Sigman [3] defines game mechanics from a mathematical perspective: "Mathematically, a game mechanic is usually just a function. A function is a mathematical 'black box'. Given a certain input, the black box (game mechanic) creates an output." The balancing of game mechanics is therefore designing the relationship of the mathematical output with the input. To do this, either the input values or the function processing this input can be changed. Nevertheless, numerous parameters within a game are interrelated.

A. Teaching Game Balancing

Game balancing consists of the following steps: (1) the determination of balancing goals, (2) the identification of the parameters relevant to these goals and (3) the appropriate selection of their functional linkage, which reflect skills and learning outcomes that should be mastered by students.

Teaching game balancing in a university curriculum is a particular didactical challenge [4]. Although competencies can be formulated as learning objectives and translated into teaching units, an understanding of the game balancing principles requires working on a specific project. The basic principle of situated learning also applies here: Balancing is only learned by balancing a project.

In the sense of constructive alignment [5] it is helpful if the teaching content and assessment are aligned with the competencies that correspond to the desired structure of observed learning outcomes.

Since student projects regularly focus on the design and development of a game prototype in the available project time, aspects of game balance are pushed back or a systematic approach to the development process is neglected entirely.

II. Methodology

A. The PARAMETERS Serious Game Reimplementation

To ensure that the principles and skills of game balancing are not only offered in theory, we created a serious game as a learning prototype. For this purpose, we adapted and reimplemented the short Flash web game PARAMETERS, created by Japanese developer Yoshio Ishii of Nekogames Studio [6], in C# using the popular Unity game engine, and applied it as a teaching tool in game design classes.

PARAMETERS is an abstract Role-Playing Game (RPG) that simplifies the traditional game experience to a series of numerical interactions, with visual aesthetics resembling an Excel spreadsheet. With no significant graphics or narrative, the game concentrates on elements/tropes often found in RPGs: grinding, farming, and "stat managing". A statistic (or stat) in RPGs is a piece of data that represents a particular aspect of a fictional character However, it's this simplicity and focus on numbers that make PARAMETERS an engaging tool for education in game design and balancing.

B. Game Description and Gameplay Mechanics

The game interface presents a grid of rectangular percentage bars of different width and height that symbolize rooms with quests and enemies. Players interact with these bars through mouse clicks, aiming to complete tasks or defeat foes by filling the bar. This interaction reduces the player's hit points (HP), necessitating a balance between attack and recovery. Strategic gameplay is encouraged through the distribution of keys for unlocking new challenges and the need to manage resources such as cash, experience points (XP), and action points (ACT) for progression.

Fig. 1 shows the gameplay screen with stats and log sections at the top and the main level with its rooms, containing quests, enemies (yellow), shops (with symbols) and secrets (green chests), some of them locked, some already open.

Additionally, the game introduces a minimalistic yet captivating effect when collecting items and bonuses, enhancing its charm. Money, items, and bonuses "erupt" from completed quests (i.e. the bars) in a fountain-like animation and need to be collected by the player by navigating the mouse cursor over them in a timely manner. With the exception of keys, items disappear shortly after reaching their final landing position.

C. Development Process and Design Goals

The reimplementation of the game PARAMETERS was undertaken with the explicit aim of utilizing its core mechanics and intuitive accessibility to novices in an educational setting, particularly for teaching game design and game development. Our objective was to transform this prototypical RPG into a significant resource for game design classes. However, the original version of the game was developed in Adobe Flash with Action Script, necessitating a complete re-implementation of the core game loop and functionality for adaptation and use in our curriculum. The original source code was graciously provided to us by Mr. Ishii.

The decision to redesign the game in Unity using C# was motivated by Unity's status as a modern, widely adopted game engine that is well-suited for educational purposes in game development. Game objects were designed and programmed to encapsulate and separate functionality, thus serving as a best practice model from which students could learn. This approach not only facilitated the understanding of game development concepts but also ensured that the game code could act as a learning tool in itself.

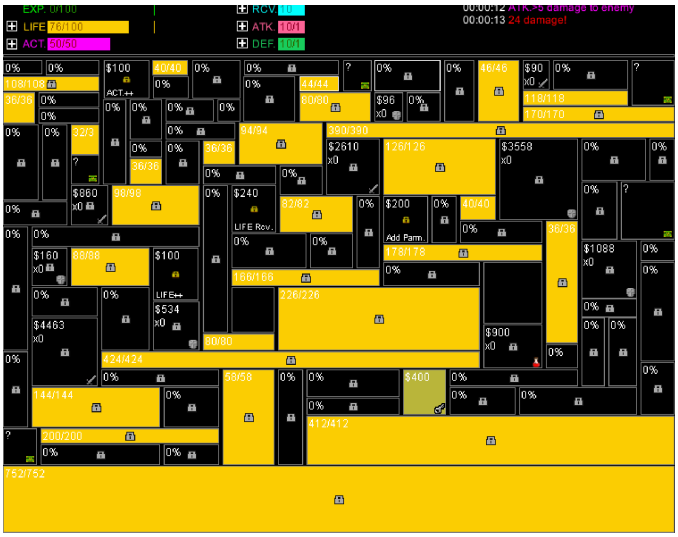

Fig. 1. Interface and game space of PARAMETERS.

In aligning with modern game design methodologies and positioning PARAMETERS as an educational tool, our main design goals focused on creating a data-driven and modular design. This design philosophy was implemented through best practice game design patterns and object-oriented programming patterns, including inheritance, interfaces, decorator, and delegation patterns [7]. A significant aspect of this approach was the implementation of game logic and balancing algorithms, called "strategies," through a standardized interface. This allowed for easy modification and swapping between gameplay implementations, making the game highly adaptable for educational purposes.

One of the pivotal design decisions was to separate game stats and level design from the core code, organizing them into data files, notably in the form of comma-separated values (CSV) spreadsheets. This approach not only mirrors industry practices but also simplifies the testing and management of balancing elements, allowing for intuitive adjustments without direct engagement with game code. It effectively shortens the development pipeline by facilitating rapid changes in mechanics to directly observe their effects on gameplay. Moreover, spreadsheets enable the separate simulation and testing of algorithms, enhancing the educational utility of the game.

To further streamline the development and testing process, we developed a Unity Editor plugin. This tool aids in the quick loading of balancing sheets and algorithm changes, as well as in previewing loaded level designs and balance stats without the necessity to run the game. Consequently, changes in gameplay can be applied in real-time, enabling immediate feedback through modifications in gameplay, thus reflecting a responsive development environment conducive to educational settings.

Beyond the scope of this paper, our game implementation incorporates several other teaching elements, including internationalization, cultural-specific balancing, debug logs, and analytics. These features not only facilitate the development and testing processes but also enrich the educational value of the game, demonstrating the depth and breadth of game design and

development concepts that can be explored by using PARAMETERS as a serious game.

III. Case Study

In our case study, we detail the integration of the serious game PARAMETERS into the Game Design II course, as part of the Master of Arts (M.A.) program in Computer Game Studies at the University of Bayreuth, Germany. This program synthesizes theoretical knowledge and practical application to teach media studies and computer science concepts and methods, aiming to provide a holistic understanding of video games from both sociocultural and technical perspectives. Students are thus prepared to critically analyze and actively engage with the video game industry, delving into the historical, aesthetic, and technical dimensions of video games to foster the creation of innovative game concepts and assess the medium's future developmental potential.

The course structure was designed to facilitate an interplay between theoretical instruction and practical exercises utilizing PARAMETERS. The course curriculum was arranged to alternate between theoretical background lectures and hands-on exercises, guiding students through the development of serious games and providing best practice reference implementations for different concepts of PARAMETERS. The course commenced with an introductory session on mathematical game theory, game design patterns, focusing on both game-specific patterns and technical object-oriented patterns such as interfaces, decorators, and delegate patterns. This theoretical foundation was immediately applied in practical exercises involving PARAMETERS, allowing students to understand game design concepts through direct engagement.

The practical exercises began with students being provided three balance sheets for PARAMETERS in CSV format: regular.csv, detailing the game's original stats and level design; hard.csv, which significantly reduced player stats and increased enemy strength, requiring more strategic planning for success; and cheat.csv, making the game extremely easy by vastly enhancing player resources (life, gold, keys) and weakening enemies. Editing CSV files required no coding experience and mimicked industry practices, allowing students to focus solely on the gameplay balancing mechanics and providing an easy introduction to the game development cycle.

Following this introduction, students were tasked with testing the game play, familiarizing themselves with Unity Editor tools and the contents of the balance sheets, and then discussing and reporting their play experiences. This exercise led to the development of their own balance sheets, which were tested in group play sessions. Theoretical lessons on Unity Editor Analytics tools were interspersed, culminating in a deep dive into game strategy implementation [7]. Students explored provided C# strategies through the IParametersStrategy interface, experimenting with DefaultStrategy.cs, Experimental-Strategy.cs, and EmptyStrategy.cs files to understand and manipulate core gameplay balancing mechanics by changing the underlying mathematical functions. This segment of the course emphasized the technical skills required for game development and allowed students to directly influence gameplay outcomes through code.

The course further delved into psychological game balancing and level design, prompting students to create and test their own strategy programs based on interface guidelines. This hands-on approach extended to planning, organizing, conducting, and evaluating user playtests, offering students practical experience in designing for specific audiences and refining game balance, strategies, mechanics, and level designs based on user feedback. The transfer of the lessons was expected in game projects, which are developed by the students a part of the curriculum where they must apply at least one of the shown balancing methods demonstrated in PARAMETERS.

The class's effectiveness was assessed through standardized university feedback forms. Feedback was incorporated into subsequent iterations of the course.

IV. Results and Discussion

The standardized university assessment forms revealed overwhelmingly positive student feedback. Participants particularly valued the knowledge transfer into the practical domain and the experience gained from conducting playtests. Some concerns were raised regarding the complexity of learning to read and modify underlying game code. This feedback underscores the importance of practical exercises in enhancing theoretical understanding and the essential role of user-centered design in game development education.

A. Design Benefits in Educational Settings

The reimplementation of PARAMETERS in an educational context highlights several pedagogical benefits that make it suitable as a teaching tool:

- **Core Game Mechanics**: By reducing the RPG experience to its numerical foundations, students can focus on understanding and manipulating core mechanics without the distraction of elaborate graphics or storylines. The game's emphasis on stat management, resource allocation (cash, XP, keys), and strategic selection of challenges promotes a deeper understanding of the strategic elements vital to game design. This facilitates strategic thinking about game resource management.
- **Game Balancing**: PARAMETERS offers a varied choice of challenges with varying difficulties, contrasting with the more linear and level-appropriate challenges of traditional RPGs. This allows students to engage in critical thinking about game balance and progression strategies by adjusting level configurations, stats, and challenge difficulty to the audience.
- **Interactive Learning**: The tactile and engaging nature of collecting rewards as well as the direct feedback provided by switching CSV balancing sheets, strategies, and level designs and observing effects on gameplay makes the educational process both enjoyable and effective.

PARAMETERS has proven to be successful in the teaching context on the one hand because it offers the opportunity to design practical teaching units that can always be linked back to a specific game, which makes the principles of game balancing

easier to understand and discuss. However, although small, it is a complete game with a considerable code base size. For students who are simultaneously designing and, in some cases, already developing their own game projects, it would involve a disproportionate amount of effort to familiarize themselves with other people's code to understand the implementation details of the concepts mentioned. For the purpose of didactic balancing, PARAMETERS was therefore not used to its full potential in our study but to demonstrate and test possibilities and concepts that were not analyzed in detail by students.

B. *Further Use Cases and Applications*

Lessons learned from using PARAMETERS as a learning tool in game design classes showed the applicability of the concept for various topics in the area of balancing:

- **Balancing of parameters**: This can be done in various degrees of abstraction. From magic numbers in the code, to entries in the declaration part of classes, to the inclusion of external files in .csv or json format. The game PARAMETERS uses externalization, which has the great advantage that it can also be used in a compiled build at runtime. Students can experience the advantages of this separation of code and value balancing as well as use the code examples as samples for their own projects.
- **Balancing of methods**: This pertains to changing the mathematical function in which the values are processed, made possible in PARAMETERS with the help of the strategy pattern. It allows different functions for selected mechanics to be changed at runtime, e.g. to replace a polynomial increase with a piecewise linear increase in a progression system. In combination with the possibility of varying the values without changing the code, considerations about possible balance settings can be tested flexibly and the conventions proposed by Schreiber and Romero [8] can be critically examined.
- **Balancing of randomized mechanics**: The game PARAMETERS contains a casino in which sometimes considerable winnings can be made for a small amount of in-game currency. Accompanied by a teaching unit on stochastics, the aim is to balance mechanics with random variables on the one hand and to understand that this can also only be achieved with the variation of variables and methods on the other.
- **Cultural balancing**: This refers to internationalization and localization. In PARAMETERS, cultural balancing is implemented with the help of a localization class that not only translates texts, but also provides alternative file paths for assets and can thus exchange the game's graphics at runtime. Therefore it provides an example of the extent to which the semiotic layer, characterized by symbols and cultural conventions, influences gameplay in combination with the game mechanics, cf. [9].
- **Psychological balancing**: A game in mathematical equilibrium is not necessarily an engaging game. One example is the handling of probabilities: Humans are notoriously bad at dealing with chance [10]. Mechanics balanced by expected value and variance can feel unbalanced or even unfair. An important lesson is therefore not to use the balancing techniques shown as an end in themselves, but as a means of specifically enhancing the game experience.

It would be possible to design a one-semester course based on this educational game. In addition to balancing, topics could include pattern-oriented software development (PARAMETERS uses numerous ‘Gang of Four’ patterns [7]), debugging and bug hunting (there is a logger that outputs filterable HTML code) or level design (levels can be created visually in the Unity editor and output as .csv files). Students can be given the task of creating balancing sheets to either change individual aspects or design the entire game experience.

V. Conclusions

In conclusion, PARAMETERS as an educational game effectively bridges the gap between theoretical understanding and practical application of game design principles in own projects, particularly game balancing. Our study highlights its potential to serve as a versatile educational resource and in general underscores the value of incorporating serious games into educational settings, particularly for complex subjects, showcasing their role in improving student engagement and learning efficacy. The ability to modify game parameters, implement different balancing strategies, and explore the effects of these changes in real-time offers an invaluable hands-on experience.

The source code along with technical and teaching instructions are openly available on GitHub [11]. We are committed to the continuous improvement of the project and warmly welcome feedback and contributions from the community to further enhance the project's utility and impact.

References

[1] A. Becker and D. Görlich, "What is Game Balancing? An Examination of Concepts," ParadigmPlus, vol. 1, no. 1, pp. 22-41, 2020.

[2] J. Schell, "The Art of Game Design: A book of lenses," 3rd ed.: CRC press, 2019, p. 212 et seqq.

[3] T. Sigman. "Anatomy of a Game Mechanic." gamedeveloper.com. https://www.gamedeveloper.com/design/anatomy-of-a-game-mechanic (accessed Mar. 22, 2024).

[4] J. J. Refai, S. Bateman, and M. W. Fleming, "External assistance techniques that target core game tasks for balancing game difficulty," Frontiers in Computer Science, vol. 2, p. 17, 2020.

[5] J. Biggs, C. Tang, and G. Kennedy, Teaching for Quality Learning at University, 5th ed. McGraw-Hill Education (UK), 2022.

[6] Y. Ishii. "Parameters Flash Game." Nekogames. http://nekogames.jp/g2.html?gid=PRM (accessed Mar. 22, 2024).

[7] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, "Design Patterns: Elements of Reusable Object-Oriented Software," Addison-Wesley, 1995, ch. Behavioral Patterns, p. 315 et seqq.

[8] I. Schreiber and B. Romero, Game balance. CRC Press, 2021.

[9] E. Aarseth, "Define real, Moron! Some remarks on game ontologies.," DIGAREC Series, no. 6, pp. 50-69, 2011 2011. [Online]. Available: https://publishup.uni-potsdam.de/frontdoor/index/index/docId/5044

[10] D. Kahneman, Thinking, Fast and Slow. Farrar, Straus and Giroux, 2011.

[11] J.-M. Loebel. "PARAMETERS Repository." github.com. https://github.com/jmloebel/PARAMETERS (accessed Mar. 22, 2024).