

Stranded Away: Implementation and User Experience Evaluation of An Indie Platformer Game Developed Using Unity Engine

Luka Blašković, Alesandro Žužić and Tihomir Orehovački
Juraj Dobrila University of Pula, Faculty of Informatics, Pula, Croatia
{luka.blaskovic, alesandro.zuzic}@student.unipu.hr
tihomir.orehovacki@unipu.hr

Abstract - Platformer games have been a staple in the video game industry for decades, dating back to the early days of arcade games such as Donkey Kong and Super Mario Bros. These games continue to be popular today and have evolved to include a wide range of subgenres and gameplay styles. This paper introduces Stranded Away - an indie platformer game developed in the Unity engine. The implemented game retains most of the main aspects of platformers, such as collectibles, moving platforms, different types of enemies, and obstacles, along with puzzle video game elements such as solving riddles to progress. However, the emphasis is placed on core game mechanics, namely player movement and puzzles, as well as tools we have developed to design game levels more efficiently. An empirical study was conducted to examine the user experience of the introduced game. The sample of study respondents consisted mainly of students who first played the game and then completed the post-use questionnaire. The reported findings uncovered to which extent Stranded Away has met the requirements of relevant user experience dimensions.

Keywords – Videogames; Unity Engine; Indie Game; User Experience; Human-Computer Interaction; 2D Platformer; Puzzle Games; Pixel-Art; Post-Use Questionnaire; Empirical Study

I. INTRODUCTION

Platformer games are a video game genre that involves navigating a character through a series of obstacles and enemies. Originating in the early 1980s, they played a key part in the evolution of video games and even helped game consoles to thrive. Typically, they are defined by their side-scrolling gameplay, where the player controls a character who must navigate through levels filled with obstacles, enemies, and puzzles. The usual objectives include reaching the end of each level, defeating bosses, or collecting certain items or power-ups along the way. These games often rely on precise controls, timing, and strategy to successfully navigate through each level.

In recent years, indie game developers have been able to create and distribute video games on their own, through various channels, including digital distribution platforms such as “Steam” and “App Store”. These indie platformer games often offer unique and creative takes on the classic platformer formula while introducing new mechanics, art styles, and storylines [3].

In this paper, we introduce „Stranded Away” – an indie 2D platformer game developed using the Unity engine. Unity is a versatile and user-friendly engine that enables developers to create cross-platform games more easily since it provides software components commonly used in game development, such as the physics and rendering engine. Stranded Away combines classic platformer elements, such as collectibles, moving platforms, enemies, and obstacles, with puzzle elements to create a unique and challenging gameplay experience.

To evaluate the user experience of Stranded Away, an empirical study was conducted with a sample of mainly student participants who played the game and completed the post-use questionnaire. The results of this study will provide valuable insights for future indie game developers and contribute to the understating of how platformer games can effectively engage and challenge players.

The remainder of this paper is structured as follows. The second section briefly describes the Unity game engine. Details on the main aspects of the game and its implementation are offered in the third and fourth sections, respectively. Findings of an empirical study are reported in the fifth section while the conclusion and future work directions are provided in the last section.

II. UNITY GAME ENGINE

Unity is a cross-platform development environment for video games, developed by Unity Technologies. The environment provides fundamental functionalities, including a rendering engine, sound importation and utilization, physics simulation, animation capabilities, and networking support. Unity provides tools for designing 2D and 3D games for a variety of devices and operating systems. The extensive availability of learning resources and pre-written libraries for frequently performed tasks is one of the reasons for Unity's popularity among beginner programmers. Unity utilizes the object-oriented programming language C#. All languages supported by Unity are object-oriented scripting languages. The object referred to as “GameObject” in Unity serves as the primary building block within the Unity environment. It allows game developers to encompass various elements such as a *character*, *object*, *camera*, *light source*, *terrain*, *image*, *text*, etc. To endow an object with attributes like objects in

the real world, such as light, window, or camera, it is necessary to add components to the object. Different combinations of components are added to the object depending on the type of object desired to be created.

The user interface of Unity plays a crucial role in the utilization and functioning of the software. The Unity interface is highly flexible and user-friendly. For instance, it allows developers to further modify and even create custom new inspectors. That can be very useful for adding commonly used functionalities that are frequently needed to be accessible directly from the editor user interface (UI) [6].

III. GAME INTRODUCTION

Stranded Away is a 2D platformer/side-scrolling game with puzzle elements. The player controls a mysterious space traveler who lands on planet „Athion” with his spaceship. He is in search of the human species after the galactical apocalypse created by mad scientist dr. Hone - the main antagonist in the game. The player is introduced to the game in the first out of three game levels. Figure 1 represents the opening scene in the first level of the game. All game scenes and visuals are produced in the pixel-art style of graphics. Pixel art is a form of digital art drawn with graphical software where images are built using pixels as the only building block [5].

The story builds up as the player finds data files located on each level. Accordingly, the user unlocks different weapons and power-ups to successfully pass the upcoming levels. Core game mechanics, including player movement, shooting, object interaction, inventory system, and puzzle solving, will be described in the following subsections.



Figure 1. The opening scene in-game

A. Player character movement

Player character movement is an important aspect of every game. Platformers are based on the player traversing through levels and overcoming obstacles thus making character movement one of the core gameplay mechanics. Player movement in Stranded Away consists of two main parts:

- a. Finding the target velocity and then gradually changing towards it over time. The vector is smoothed out by a spring-damper-like function. A spring damper refers to a mathematical model used to simulate physics behavior. It is commonly used in game physics to create realistic motion, such as character jumping, as well as for smoothing out a follow camera [6].

- b. Attaching several *colliders* to the player, as shown in Figure 2. In unity, colliders are components that define the shape of an object for the purpose of physical collisions. For instance, the rounded circle-cast collider

will indicate that the player is grounded if it hits anything designated as ground.



Figure 2. Different colliders attached to the main character

B. Shooting system

The shooting system in platformers allows players to attack enemies or obstacles in the game world. This is implemented by creating an *arm pivot* object, a common mechanism that allows players to shoot projectiles more realistically and intuitively. It works by using an animation of an arm pivoting at the shoulder joint, along with a *ray cast*, to determine the direction of the shot.

C. Object interaction

Object interaction refers to how the player character interacts with various objects within the game world. In Stranded Away, this includes opening doors, pushing, and picking up boxes, and stepping on the pressure plates or jump pads. Object interaction plays a key element in the game since it consists of many riddles that must be solved to progress throughout the game. An example of interacting with an object is presented in Figure 3.



Figure 3. Highlighting doors with white outline

D. Inventory system

Inventory systems in video games allow players to manage and organize the various items they collect throughout the game. Accompanied by crafting elements, the player may collect certain raw materials, as well as finished products that can be equipped and sold. Crafting mechanics add an extra layer of depth and complexity to the gameplay, as the player has to manage his resources and strategize which items to craft to progress through the game more easily. For instance, the player primarily picks up the raw materials for crafting ammunition. Figure 4 shows the player's inventory and crafting grid in the game.



Figure 4. Inventory (right) and crafting grid (left)

E. Gameplay elements

Gameplay in video games refers to the interaction between the player and the game. Stranded Away includes typical platforming gameplay elements such as jumping, platforming, coin pickup, puzzle solving, power-ups, and different enemies. The main user interface is located at the bottom of the screen. As presented in Figure 5 (from left to right), the user interface consists of ten different parts: 1 - *ammunition*, 2 - *selected weapon*, 3 - *health bar*, 4 - *shield bar*, 5 - *oxygen bar*, 6 - *energy bar*, 7 - *total coins*, 8 - *settings*, 9 - *stats*, and 10 - *inventory*.

In the beginning, the player will only have a health bar unlocked and 4 points in it, which represent his/her maximum health. His/her default weapon is a blaster gun which has unlimited ammunition, but it is the weakest. The player will gradually unlock other weapons and power bars as he/she progresses through the game. Additionally, power-ups such as *health increment powerup* can be purchased with collectible coins. The far-right section of the interface has three buttons: *settings*, *stats*, and *inventory*, although they are also accessible using keyboard shortcuts.



Figure 5. User interface in-game

One of the core game mechanics is the interaction and manipulation of *energy boxes*. The player can pick them up and arbitrarily move them around the game world. The boxes are primarily used for emitting signals by activating the *pressure plate* – a type of switch that activates when something is on top of it, more precisely player or energy box. When activated, pressure plates emit a visible signal to certain objects such as *laser doors* and *moving platforms*. However, it is not always the case that a single pressure plate will trigger a single object. Instead, there are four different possibilities: (1) One pressure plate → Single object, (2) One pressure plate → Two or more objects, (3) Two or more pressure plates → Single object, (4) Two and more pressure plates → Two or more objects. This allows for the design of more complex and challenging riddles for the player.

IV. IMPLEMENTATION OF MORE ADVANCED TOOLS

Video game development, as well as development in general, often consists of much repetitive code writing which is replaced in later stages with automated tools and systems. This helps streamline the processes and make them more efficient, reducing the amount of manual work that needs to be done. Following subsections present various tools that have been created during the development process.

A. Enemy AI

Enemy artificial intelligence (AI) is a crucial aspect in the creation of engaging and challenging virtual opponents in video games. In the Unity game engine, one commonly utilized approach to implementing enemy AI is using *ray*

casting – a technique that involves emitting a line or "ray" from a starting point in each direction and determining the first object that intersects with the ray [6]. The implementation of *ray casting* in the "Enemy AI system" enables the detection of the player character. This allows for dynamic adjustments of the AI behavior based on the main character's actions and position, as shown in Figure 6.

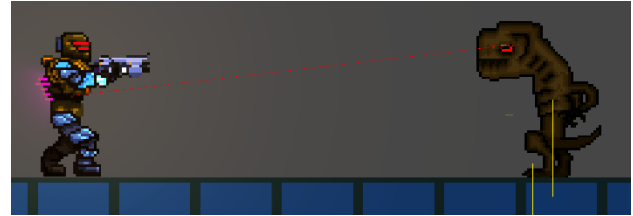


Figure 6. Enemy raycasting

The *Enemy AI* script utilizes a state machine to dynamically control the behavior of enemy characters in the game. The state machine has multiple states including *Idle*, *Roaming*, *Chasing*, *Attacking*, *Spitting*, and *Flying*. Each state represents a distinct behavior or action that the enemy can perform.

B. Tilemap painter

The process of designing levels in a 2D video game can be quite repetitive as it often involves the manual placement of individual *tiles* (small, regular-shaped images). However, this approach is time-consuming and requires the placement of each *tile*, decoration, and lighting element individually. A two-dimensional grid of *tiles* is usually referred to as a *tilemap* [6]. Since the solution provided by Unity for the automatic generation of *tilemap* textures did not suffice the game's requirements, a custom system for automatic terrain and *tilemap* generation was developed instead. In that respect, the "Tilemap Advanced Ruleset" tool was developed to streamline the process of *automatic terrain and tilemap* generation within the Unity game engine. The tool provides a user interface customized to fulfill specific requirements. It enables the level designer to utilize a prototype tile palette to paint the prototype tilemap which represents the environment where the player will navigate. The designer then selects one or more of the pre-established tilemap rulesets. A ruleset refers to a set of defined rules that dictate the behavior of tile placement within a specific context. These rules are based on the relationship between the current tile and its surrounding tiles, effectively creating a system of logic similar to cellular automata, which is commonly used in modeling and simulating complex systems. Ruleset functions represent a sort of "game of life" for tile painting, where the state of each tile is determined by the state of its neighbors and the predefined set of rules [1].

Finally, the level designer "paints" the *tilemap* using template blocks, and when done, the tool automatically creates tilemaps by utilizing the appropriate tiles from pre-defined palettes using rulesets. This process, along with the custom editor for that tool, is presented in Figure 7. Furthermore, additional tools were developed for

generating decorative elements to further enhance the visual appearance of the levels. One such tool is “Tilemap Advanced Generation” which also uses predefined rules for generating pillar structures in caves.

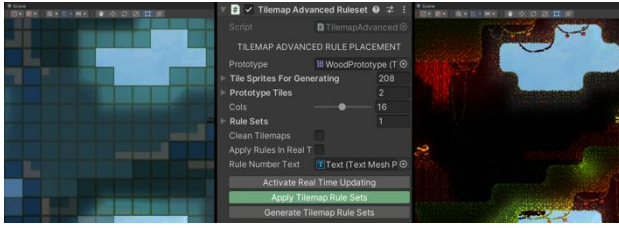


Figure 7. Before and after applying the tilemap generation tool

C. Level generation

Besides the main campaign, the game also contains a “challenge” level called “The floor is lava”. In brief, the player must climb up infinitely while avoiding the rising lava. He/she must jump from platform to platform, avoiding obstacles and collecting power-ups. The level implementation posed a challenge, as the terrain must be generated in a way that is both random and fair to the player. This was achieved through the implementation of the “Level generation” tool. Instead of randomly placing tiles, which could potentially cause obstacles for the player, a form of procedural generation algorithm was designed. This method uses predefined sets of tiles, referred to as “platforms”, and generates them randomly as shown in Figure 8, ensuring that each section is well-designed and fair to the player. Additionally, the tool is designed to keep track of the player's maximum height, which allows for the generation of additional elements like “money barriers” which prevent the player from climbing if he/she did not collect enough coins, as well as the application of effects on the player based on the height reached.

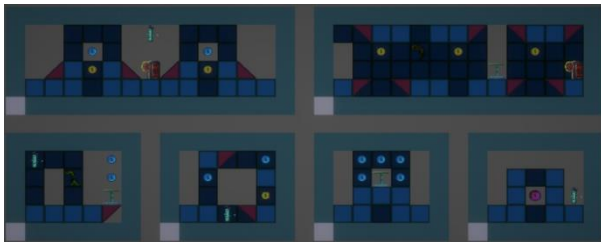


Figure 8. Predefined sets of tiles

This approach in combination with “Tilemap Advanced Generation” not only keeps the level interesting and fresh for the player but also ensures that the player's path is not blocked.

V. USER EXPERIENCE EVALUATION

A. Participants

To gather an understanding of to which extent Stranded Away has met the requirements of relevant user experience dimensions, an empirical study was carried out. A total of 75 participants, mostly students from the Juraj Dobrila University of Pula, Faculty of Informatics took part in the study. The sample was composed of 72% male and 25.3%

female respondents while the remaining 2.7% declined to provide an answer in that respect. The majority (44%) of participants were enrolled in the first year of undergraduate study, 24% were enrolled in the third year of undergraduate study, 20% were enrolled in the second year of graduate study while the remaining 8% percents were enrolled in the second year of undergraduate and first year of graduate study. Three participants (4%) were not students. The average age of respondents was 22.15 years ($SD = 3.86$).

Regarding previous gaming experience and gaming habits, the participants were rather diverse, ranging from casual gamers to avid gamers, with different preferred genres. Participants' responses related to the most important game feature are summarized in Figure 9. The majority of participants (40%) agreed that enjoyment is the most important aspect of a video game.

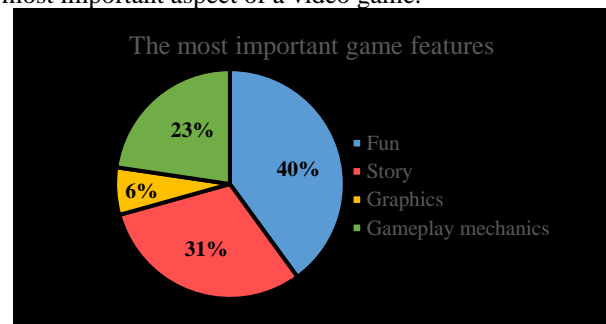


Figure 9. The most important game features

Platformers might pose a challenge for players with little to no experience for several reasons, such as precision jumping, timing-based obstacles, challenging enemies, and increasing difficulty. As shown in Figure 10, a majority (59%) of players had no to limited previous experience with platformers, while others were moderate to very experienced in such games.

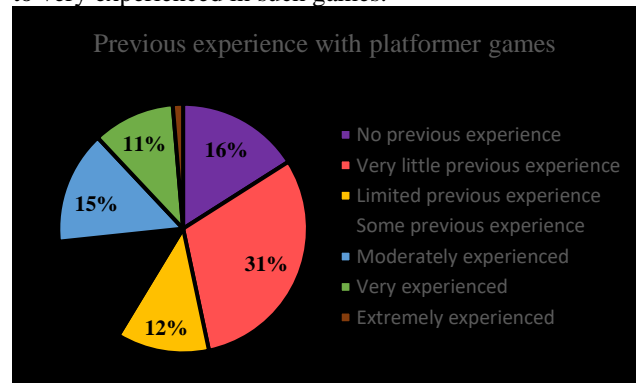


Figure 10. Previous experience with platformer games

Finally, a bar chart presented in Figure 11 illustrates the most popular video game genres. The respondents chose preferred genres of video games from a multiple-choice question list. Results indicate that action and shooter games are by far the most popular game genres, as chosen by half of the respondents. Around a third of respondents play survival, strategy, and RPG games, while 25% of respondents prefer genres such as horror, racing, sports, fighting, etc. However, only 12% of participants play platformer-type games.

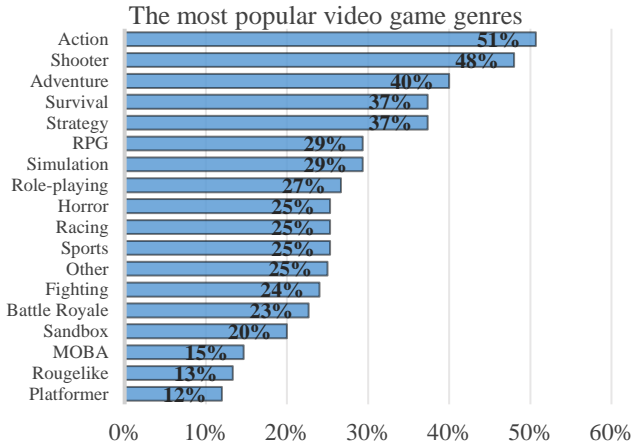


Figure 11. The most popular video game genres

B. Apparatus

The user experience in *Stranded Away* was examined by the means of a post-use questionnaire. It was composed of forty (40) items meant for measuring the following relevant user experience dimensions:

- Gameplay mechanics (*GMH*) – refers to the rules and systems that govern player interaction and actions in the game,
- Learnability (*LRN*) – the degree to which the game is easy to learn and master,
- User interface sensibility (*UIS*) – refers to how well-designed, organized, and intuitive the game’s user interface is,
- Visual aesthetics (*VIS*) – the visual design of the game and its impact on the player’s overall experience
- Audio aesthetics (*AUD*) – refers to the overall feel, mood, and ambiance that in-game sound creates for the player,
- Engagement (*ENG*) - state of focus, concentration, and immersion experienced by players,
- Enjoyment (*ENY*) - the degree to which the player experiences pleasure from playing the game,
- Behavioral intentions (*BEH*) – the likelihood of the player recommending the game, playing again, or purchasing similar games in the future.

Since the game is slow-paced at the beginning, participants were asked to play the first two levels to explore and unlock most of the game’s features and fine details. Afterward, participants were asked to complete an online questionnaire containing items on the Likert scale ranging from 1 (strongly disagree) to 5 (strongly agree). The user experience evaluation procedure was adopted from [2,4] and tailored to the context of this study.

C. Findings

The analysis of collected data uncovered that over 77% of participants agreed that game mechanics are well-designed, while 21% of them were neutral. A large majority (90.7%) of participants agreed that *Stranded Away* is easy to learn. Moreover, 96% of participants consider the user interface to be well organized. The visual aesthetics were appealing to 80% of participants, while

13.3% stayed neutral and 7% did not find them appealing. Regarding the audio experience, 76% of players expressed that the sound and music enhanced their overall gaming experience. 68% of players agreed on being engaged while playing the game and the majority (78.7%) found the game to be fun. Most participants (38.7%) stayed neutral expressing intentions of playing the game in the future. However, over half of them agreed that they would recommend the game to others in their gaming community and even try other games from the same development team. The reported findings are summarized in Figure 12 (see Appendix).

D. Interpretation of comments and suggestions

In addition to the demographic data and findings presented in subsections B. and C., respectively, respondents had a choice to write what they liked about the game, and what they did not, or to give constructive criticism or suggestion. A majority (56%) of respondents answered the question and provided various suggestions, praises, and criticism. Regarding gameplay mechanics, 16% of those respondents mentioned that game is overall well-made, while 19% pointed out some bugs such as “sprint jumping” and “glitchy audio”. 7% of respondents find the story well thought out. 11% of respondents praised the controls in the game but noted that dialogue text is hard to read. 7% of respondents did not like the crafting system and suggest further improvements. Visual elements such as color palette and contrast were praised by one-third of respondents (33%). Few respondents mentioned nice particle effects and animations, and two critiques were directed towards “flashlight” and low visibility caused by fog. In addition, 14% of people did not like the enemy design. 30% of participants liked the audio effects and the original soundtrack and 12% of them stated that audio design contributes to the overall atmosphere and ambiance. As much as 35% of respondents expressed enjoyment of the game, stating that it is “fun and simple” and “that it reminds them of their childhood”.

VI. CONCLUSION

Platformer games have been around for decades now, and recently gained popularity again thanks to indie game developers. *Stranded Away* is a 2D platformer game developed in the Unity engine. The versatility and user-friendly interface of Unity, make it a popular choice for indie game developers. The game was designed to combine classic platformer elements with puzzle elements, creating a unique and challenging gameplay experience. *Stranded Away* contains various gameplay elements, typical for the platformer genre, such as moving platforms, puzzles, enemies, and other obstacles. Various tools were built in the process of development to speed up common tasks, such as terrain generation, tile placement, and puzzle creation. An empirical study was conducted to evaluate the user experience in *Stranded Away*. Even though the previous experience and average gaming time vary vastly among respondents, the majority agree that the game aspects such as user interface, mechanics, audio and visuals were well designed. However, the majority of respondents stayed neutral when asked about playing the game in the future. A possible reason for this may be that only 12% of study

participants play platformers. Stranded Away serves as a demonstration of the potential of Unity as a game development engine and the potential of indie platformer games to offer unique and engaging gameplay experiences. Future work could involve expanding the game's content by adding more levels, enemies, power-ups, and boss fights. It is also important to acknowledge the limitations of the study, primarily a relatively small and homogenous sample, consisting of mainly students from the same university.

REFERENCES

[1] Antoniuk, I., & Rokita, P. (2016). Procedural Generation of Underground Systems with Terrain Features using Schematic Maps and L-systems. *Challenges of Modern Technology*, 7(3), 8–15. <https://doi.org/10.5604/01.3001.0009.5443>

[2] Bošnjak, M., & Orehovački, T. (2018). Measuring quality of an indie game developed using Unity framework. *Proceedings of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1574–1579. <https://doi.org/10.23919/MIPRO.2018.8400283>

[3] Dickson, P. E. (2015). Using Unity to Teach Game Development. *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, pp. 75–80. <https://doi.org/10.1145/2729094.2742591>

[4] Šajina, R., & Orehovački, T. (2018). User experience evaluation of 2D side-scrolling game developed using Overlap2D game editor and LibGDX game engine. *Proceedings of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1580–1585. <https://doi.org/10.23919/mipro.2018.8400284>

[5] Silber, D. (2017). *Pixel Art for Game Developers* (1st ed.). A K Peters/CRC Press.

[6] Unity Technologies. (n.d.). *Unity Documentation*. Unity. Retrieved January 31, 2023, from <https://docs.unity.com/>

APPENDIX

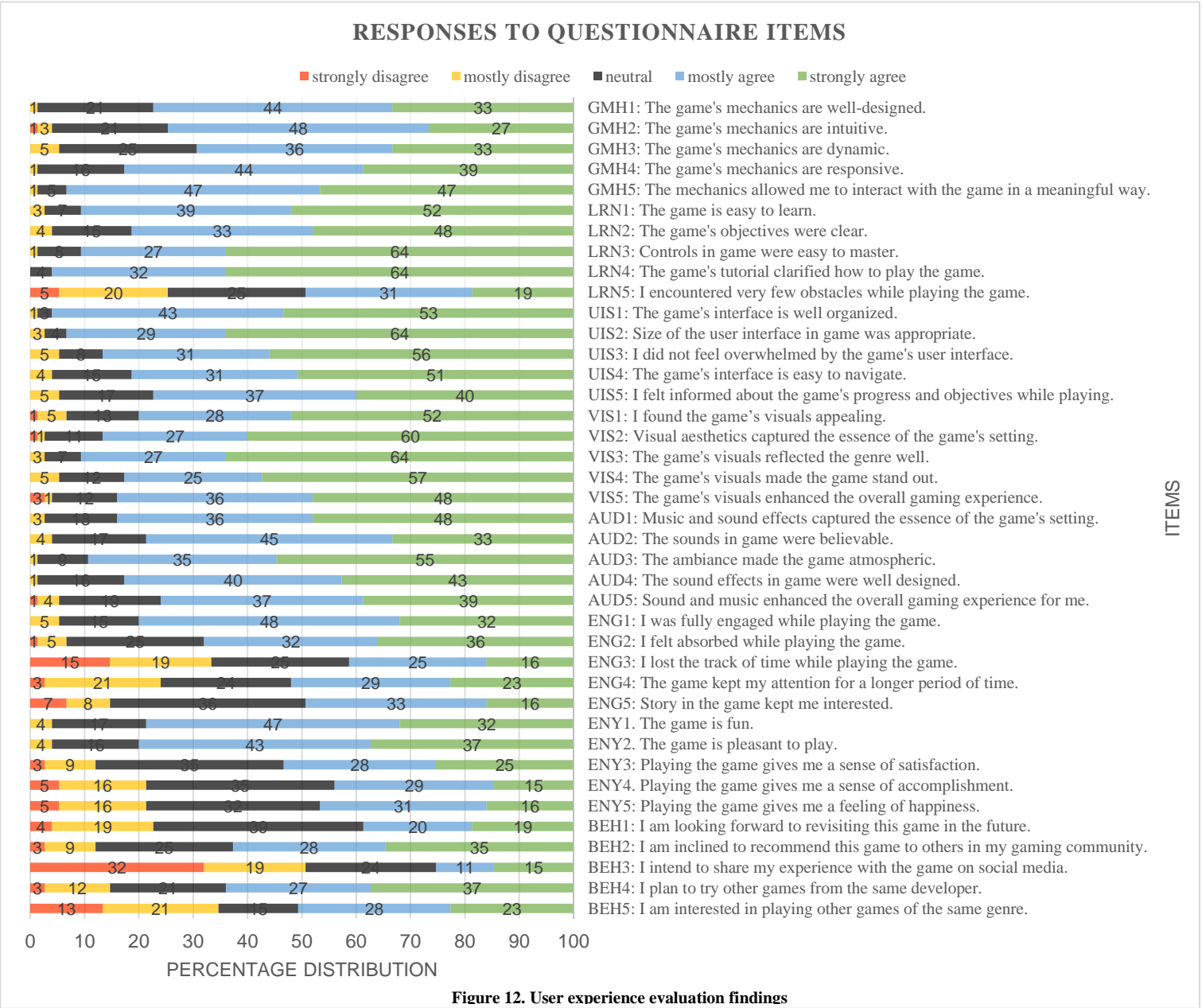


Figure 12. User experience evaluation findings