Proposing an Automatic System for Generating Super Mario Bros Datasets

Matheus Prado Prandini Faria[1], **Etienne Silva Julia**[1], **Rodrigo Zamboni Silva**[1], **Marcelo Zanchetta do Nascimento**[1], **Rita Maria Silva Julia**[1]

[1]*Computer Science Department*
Federal University of Uberlândia
Uberlândia, Brazil
{matheusprandini.96, etienne16sj, rodrigozamboni2, marcelo.zanchetta, ritasilvajulia}@gmail.com

Abstract. *Video games have become a critical part of the entertainment industry. In addition to being designed for fun, games can also be used in education, health, and many other areas. For example, some methods from the Procedural Content Generation (PCG) approach are used to create content for Super Mario Bros (SMB) algorithmically seeking to develop personalized levels based on players' mechanics. This work aims to provide an automatic system for generating datasets composed of game logs and video frames in the Mario AI Framework, which is expected to speed up the development and validation of PCG, player modeling/profiling, and knowledge tracing methods. This also helps in the reproducibility of researches using the SMB game.*

Keywords— *Super Mario Bros, Game Log, Dataset, Procedural Content Generation*

1. Introduction

Video games have become a critical part of the entertainment industry, impacting the lives of billions of people every day [Wijman 2019]. They exert wide influence among the younger generation, whether for the appealing side of their content or for the messages and ideas they disseminate [Moosa et al. 2020]. In addition to being designed for fun and entertainment, games can also be used in education, health and many other areas [Sharifzadeh et al. 2020, Ullah et al. 2022]. For example, to understand underlying psychological and behavioral implications of the interaction of players with this medium. This is because video games provide extremely appropriate domains for the occurrence of rich affective experiences measurable through in-game behavior [Azadvar 2021].

Procedural content generation (PCG) methods are used to create content for games algorithmically aiming to satisfy the needs of designers or players [Shu et al. 2021]. Among the games used to study such an approach, the Super Mario Bros (SMB) Framework - an open-source clone of the original SMB game - stands out. In addition to having simple visual aspects, it stores and renders levels from text files, where each symbol represents a tile to be rendered on the screen [Hauck and de Castro Aranha 2020]. However, few works make their game datasets available to the community, which makes research reproducibility a difficult task.

For this purpose, the present paper proposes the automation of data generation in the SMB Framework through video frames and game logs, which register what the players are doing, as well as what is happening in the environment. These logs correspond to game information that efficiently represents the players' experiences, as already validated in recent works [Luo et al. 2018, Green et al. 2020, Faria et al. 2022]. The authors here expect that this will speed up the development and validation of future PCG methods, for example, to create personalized levels based on players' mechanics [Green et al. 2020].

Thus, the main contributions here are: 1) implementing an automatic system to capture the interactions of the players (humans or automatic agents) in the SMB Framework through game logs; 2) building a complete dataset composed of SMB gameplay data. Finally, the present work is open to the community (both the automatic system and the dataset)[1] and may serve as a basis for future works involving personalized learning in the SMB (in areas of PCG, player modeling/profiling and knowledge tracing).

The next sections are structured as following: section 2 resumes the background; sections 3 and 4 describes, respectively, the automatic system for generating game logs and the dataset proposed herein; finally, section 5 shows the conclusions and future works.

2. Background

The Mario AI Framework [Karakovskiy and Togelius 2012] is based on Infinite Mario Bros, an open-source version of the original SMB game. In this framework, players control Mario through a level until it reaches a flag on the right hand side or until it loses (if Mario falls down a gap or if Mario collides with an enemy). Players can walk, run, jump and shoot fireballs at enemies. Game events (described in section 4) are also detected during gameplay, such as when Mario collects a coin or bumps a block. This framework is popularly used in the PCG field, because it provides an interesting challenge in a controlled environment that at the same time is very simple to manipulate and test (as it generates and stores levels from text files).

PCG via Machine Learning (PCGML) [Summerville et al. 2018] methods have become increasingly important in recent years due to the ability of automatically extracting relevant information from existing data for level generation in SMB, such as [Green et al. 2020] [Hauck and de Castro Aranha 2020], [Shu et al. 2021]. However, in the context of SMB, this approach is very limited by the lack of available data [Shu et al. 2021], which is addressed by the present paper.

[Karpouzis et al. 2015] and [Svoren et al. 2020] proposed ready-made SMB datasets for the community both providing game log information and also visual data of players' facial expressions during matches. Although the present paper does not yet include visual player data (which is its main limitation compared to the aforementioned works), its main novelty is to allow the creation of new custom datasets based on human or automatic agents through the proposed system herein (presented in the section 3).

3. Automatic Log Generation System

This section presents the first contribution of this paper: the implementation of a new feature in the Mario AI Framework that automatically stores all player interactions with the game. As already mentioned in the 2 section, the Mario AI Framework is open-source for the community. This allowed the authors of the present work to develop all the modules necessary to capture relevant information from the player's interaction with the

[1]https://github.com/matheusprandini/Mario-AI-Framework-Generate-Dataset

SMB, such as the actions performed by the player (for example, pressing the button up, down, to the right, or to the left) and the game events that are happening (e.g. killing an enemy, collecting a coin, winning the level). It is important to note that this information must be stored after games are over, which is also covered in this work. The subsection 3.1 and the subsection 3.2 describe, respectively, all the components of the system created to capture these interactions and the detail of their storage flow.

3.1. System Architecture

Figure 1 presents the architecture of the automatic system proposed in this paper, which is formed by input parameters, an execution queue, the Mario AI Framework engine, and a data storage module. This system is described in detail in the following.

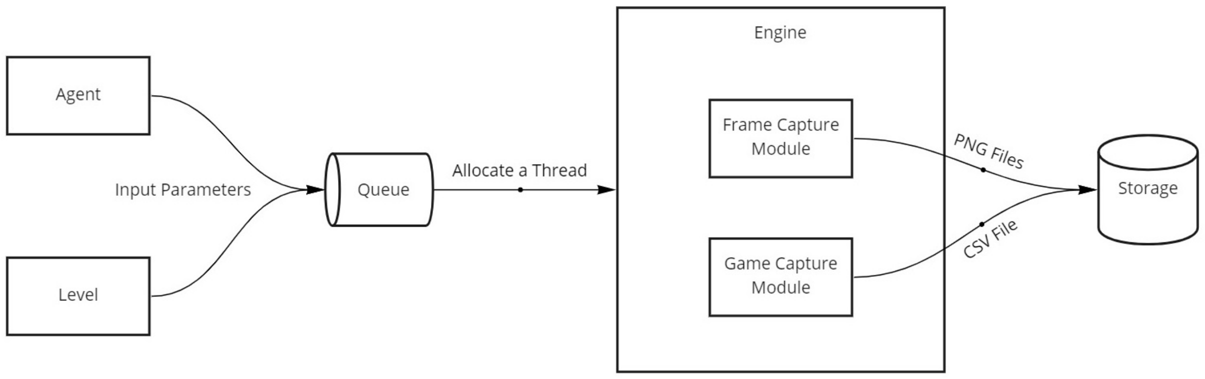

Figure 1. Architecture of the Proposed System

It receives as input parameters the agent and the level in which is desired to collect the data. It is important to note that the agent can be a human player or an automatic player. To facilitate the generation of a dataset, the system was adapted to receive a set of automatic agents and a set of levels as well, where the combination of each agent and level - representing a game - is executed and the information of each one is stored. To do so, each agent/level combination is sent to an execution queue.

With such parameters received, the queue is responsible for allocating a thread for execution in the Mario AI Framework engine. Instead of sequentially executing each agent/level combination passed as a parameter to the system (that is, one combination at a time), up to 4 threads can run at the same time (this value was empirically defined). This allows more data to be collected in the same period of time. Each thread is responsible for executing a game.

In the engine, the authors implemented the following modules: Frame Capture and Game Capture. The former captures game frames (screenshots) in Portable Network Graphics (PNG) image format. While the latter retrieves information about the actions performed by the agent and the changes that occur in the level and saves them in a Comma Separated Values (CSV) file. All this information is saved in a local directory (that is, on the machine where the system runs). The storage flow is described in the next subsection.

3.2. Storage Flow

Figure 2 illustrates the storage flow of game logs that occurs in the execution of a thread. When playing a level, all the agents's interaction with the current game state (including not taking any action) and the changes on the game state are stored in the form of a log until the player wins or loses the level. In addition to the game logs, the frames are also saved (the combination of all frames corresponds to the gameplay video of the level). It is important to note that the system runs in real-time at 30 frames per second and each game log is associated with one frame. So, if the agent takes 10 seconds to complete or lose the level, a total of 300 video frames (PNG images) and 300 game logs (300 lines in the CSV file) are generated.

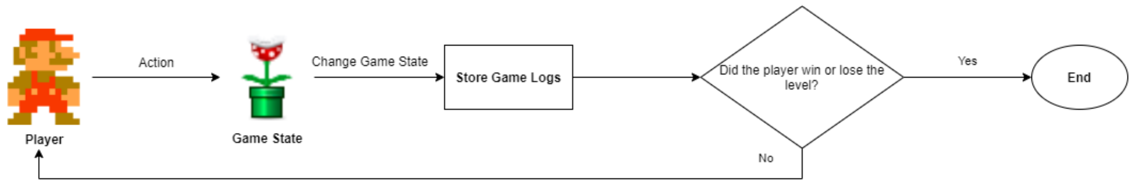

Figure 2. Storage Flow of a Thread

After completing the data generation, the directory that stores the resulting dataset has the following structure: *dataset* represents the root directory where all data is stored. Data is first grouped by agent and then by level. And for each level directory, there are the video frames and the CSV files generated at runtime. For example, in the tree below it is possible to see that there is data for *Agent 1*, *Agent 2* and possibly other agents. For *Agent 1*, it is noticed that some levels were executed, such as *level 1* and *level 2*. For each one, there are PNG files representing the frames and the CSV file (containing the game logs) representing the game information captured during the execution of the *Agent 1*. This occurs for each agent/level combination. Details of such stored data (video frames and game logs) will be described in the next section.

```
dataset
├─ Agent 1
│  ├─ lvl-1
│  │  ├─ frames
│  │  │  ├─ Frame1.png
│  │  │  ├─ Frame2.png
│  │  │  └─ ...
│  │  └─ lvl1.csv
│  ├─ lvl-2
│  └─ ...
├─ Agent 2
│  ├─ lvl-1
│  ├─ lvl-2
│  └─ ...
└─ ...
```

4. Dataset

This section presents the second contribution of this paper: building a complete dataset of SMB gameplay. The SMB dataset is composed primarily of two logfiles: frame log and game log. The former represents screenshots with image size (height and width) of 256 pixels x 256 pixels and a Red Green Blue (RGB) color model. The latter presents a complete record of interactions between the users and the game. It contains the main features extracted from the game engine, representing the history of the player's actions and in-game events. Each line in the game log (see Table 1) consists of the grid level, the coordinates of Mario's position, the buttons pressed by the player, and the game events:

Grid Level: 256 values forming the 16x16 grid level. Each value represents.
Position: coordinates of mario's position.
Actions: buttons pressed by the player (1 corresponds to active buttons).
Game Events: the Super Mario game events investigated in this work are defined by the Mario AI Framework itself as being fundamental to model the dynamics of a match (1 corresponds to the active game events). The game situations corresponding to these 12 game events are briefly described next:

- *EventBump:* Mario jumps having a block above its head;
- *EventCollect:* Mario collects a coin at the current level;
- *EventFallKill:* an enemy dies by falling out of the scene;
- *EventFireKill:* Mario kills an enemy by shooting fire at it;
- *EventHurt:* Mario takes damage by hitting an enemy;
- *EventJump:* Mario performs the jump action;
- *EventKick:* Mario kicks a Koopa shell;
- *EventLand:* Mario lands on the ground after having jumped;
- *EventLose:* Mario loses the game level (which happens either when this character dies or when the time to complete the current level is over);
- *EventShellKill:* Mario kills an enemy by throwing a Koopa shell at it;
- *EventStompKill:* Mario kills an enemy by jumping over it;
- *EventWin:* Mario wins the game (that is, the character successfully completed the current level).

.

Table 1. Game log sample

Sprite 1	...	Sprite 256	Mario X	Mario Y	Left Button	Right Button	Down Button	Speed Button	Jump Button	EventBump	...	EventCollect	...	EventWin
0	...	16	56.0	233.0	0	1	0	1	0	0	...	1	...	0

These logs provide information about the agent's interaction with the game and were defined mainly based on the work [Faria et al. 2022] in which the main objective was to identify game events in gameplay footage through deep learning methods. The authors of the present paper believe that this kind of dataset can be very useful to assist in the evolution of AI research carried out on the SMB game.

5. Conclusion and Future Works

This work proposed an automatic system for log generation in the Mario AI Framework to build datasets to speed up the development and the reproducibility of AI researches using the SMB game. This system was developed in order to automatically collect video frames and game logs (both contain all the agent's interactions with the game) from both human and automatic players. In future works, the authors intend: to gather visual data from players; to investigate the behavior of different players through their gameplay data (generated with the proposed system) and to implement new PCG methods through deep learning to design new personalized levels for these players.

References

Azadvar, A. (2021). Predictive psychological player profiling.

Faria, M. P. P., Julia, E. S., Nascimento, M. Z. d., and Julia, R. M. S. (2022). Investigating the performance of various deep neural networks-based approaches designed to identify game events in gameplay footage. *Proc. ACM Comput. Graph. Interact. Tech.*, 5(1).

Green, M. C., Mugrai, L., Khalifa, A., and Togelius, J. (2020). Mario level generation from mechanics using scene stitching. *2020 IEEE Conference on Games*, pages 49–56.

Hauck, E. and de Castro Aranha, C. (2020). Automatic generation of super mario levels via graph grammars. *2020 IEEE Conference on Games*, pages 297–304.

Karakovskiy, S. and Togelius, J. (2012). The mario ai benchmark and competitions. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):55–67.

Karpouzis, K., Yannakakis, G. N., Shaker, N., and Asteriadis, S. (2015). The platformer experience dataset. ACII '15, page 712–718, USA. IEEE Computer Society.

Luo, Z., Guzdial, M., Liao, N., and Riedl, M. (2018). Player experience extraction from gameplay video. *CoRR*, abs/1809.06201.

Moosa, A. M., Al-Maadeed, N., Saleh, M., Al-Maadeed, S. A., and Aljaam, J. M. (2020). Designing a mobile serious game for raising awareness of diabetic children. *IEEE Access*, 8:222876–222889.

Sharifzadeh, N., Kharrazi, H., Nazari, E., Tabesh, H., Khodabandeh, M. E., Heidari, S., Tara, M., et al. (2020). Health education serious games targeting health care providers, patients, and public health users: scoping review. *JMIR serious games*, 8(1):e13459.

Shu, T., Liu, J., and Yannakakis, G. N. (2021). Experience-driven pcg via reinforcement learning: A super mario bros study. *2021 IEEE Conference on Games*, pages 1–9.

Summerville, A., Snodgrass, S., Guzdial, M., Holmgård, C., Hoover, A. K., Isaksen, A., Nealen, A., and Togelius, J. (2018). Procedural content generation via machine learning (pcgml). *IEEE Transactions on Games*, 10(3):257–270.

Svoren, H., Thambawita, V., Halvorsen, P., Jakobsen, P., Garcia-Ceja, E., Noori, F. M., Hammer, H. L., Lux, M., Riegler, M. A., and Hicks, S. A. (2020). Toadstool: A dataset for training emotional intelligent machines playing super mario bros. In *Proceedings of the 11th ACM Multimedia Systems Conference*, MMSys '20, page 309–314.

Ullah, M., Amin, S. U., Munsif, M., Safaev, U., Khan, H., Khan, S., and Ullah, H. (2022). Serious games in science education. a systematic literature review. *Virtual Reality & Intelligent Hardware*, 4(3):189–209.

Wijman, T. (2019). The global games market will generate $152.1 billion in 2019 as the u.s. overtakes china as the biggest market.