

DRMBM (1994) remake

Christopher Ashford

September 2022

Contents

1	Analysis	1
1.1	Introduction and Background	1
1.2	Alternative Solutions	1
1.2.1	Emulation	2
1.2.2	B Puyo	2
1.2.3	Project GelaVolt	3
1.2.4	Puyo Puyo Tetris 2	4
1.3	End User Input	5
1.4	Input, Data Processing, Output	11
1.5	Goals	12

1 Analysis

1.1 Introduction and Background

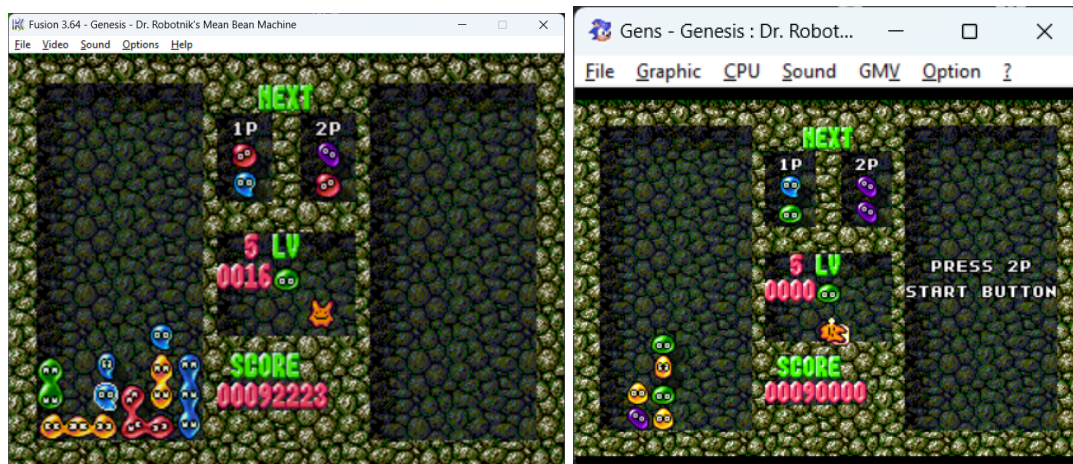
Dr Robotnik's Mean Bean Machine is a 1994 Westernised port of Puyo Puyo for the Sega Genesis/Mega Drive. It is a game that I have enjoyed throughout my childhood on many different forms – cheap emulation consoles, the Sega Mega Drive Collection for Xbox 360, using Fusion emulator on PC among other forms. However, all of these present glaring issues that directly affect the enjoyment of the player – emulation consoles usually are slow with clunky controllers and are not good for much else and thus are not practical to use permanently; the Sega Mega Drive collection on Xbox 360 suffers with a noticeable input lag problem, with inputs sometimes taking hundreds of milliseconds to be processed, directly affecting how fast you can play; PC emulation either results in a small or blurry image and makes it difficult to play with others or share your scores and achievements.

The goal of this project is to solve these problems by creating a superior, native PC remake of the game. Everything in the original game shall work exactly as in the original, including reconstructing the algorithms used for the playstyles of the various AI opponents. I also intend to include many quality of life improvements to solve the problems listed about: multiple customisable input method and handling will be supported, many algorithmic optimisations shall be made to improve performance, graphics shall be upscaled in a way that remains a crisp pixel look instead of introducing blur, an SQL web server will allow score and time leaderboards to exist and a replay file system shall be introduced to allow players to easily share gameplay. This project exists to create a superior version of DRMBM for a new generation to enjoy, as well as offering a way for modern Puyo Puyo players to enjoy the OPP rule set on modern devices.

If the goals above are reached, further extension goals include the introduction of my own custom AI opponents with algorithms designed for optimal, “perfect” gameplay and the use of web sockets to facilitate real-time online matches between two remote players.

1.2 Alternative Solutions

In this section I shall present my research on other Puyo Puyo games, compare the advantages and disadvantages of different versions from the perspective of the end user and take inspiration for my own project.



(a) Fusion emulator

(b) Gens emulator

Figure 1: Some examples of emulators running the game

1.2.1 Emulation

Link: cannot be provided due to specialised hardware being required to dump the ROM. Yet another disadvantage.

Many different emulators exist for the Sega Mega Drive, such as Fusion or Gens shown above, or the official Sega emulator that can be found on Steam. These are programs that accept a binary ROM dump of the original cartridge and attempt to emulate the code.

Advantages:

- Convenient for mass production and distribution. Sega can create one Mega Drive emulator and release an entire of library of games that use the same program
- True to the original experience. Since you are playing a copy of the original game, you can be sure you are getting an authentic experience
- While clunky, save states allow you to save high scores and progress through the story, as well as letting you manipulate sequences of beans

Disadvantages:

- Resolution is locked at the console's original and upscaling is blurry and unappealing
- Very static and not customisable. It is incredibly difficult to edit a ROM if you wanted to play with, for example, different handling or textures
- Saving progress is difficult
- Emulators are difficult to run and can easily lag on lighter hardware, running the game at higher levels can struggle on older processors
- It is impossible to play with friends remotely (or if it is possible, then it's too difficult for the average user to achieve)

1.2.2 B Puyo

Link: <http://bx1.digick.jp/puyo/dl.php>


B puyo is a popular online Puyo-clone recommended to me by the Japanese community.

Advantages:

- Custom textures, custom AI, custom rules, custom anything really
- Easy to use online multiplayer
- Great performance as a native PC program



Figure 2: A screenshot of B Puyo. Some text is broken running on an English computer.



phoernian_jp 08/09/2022


Hello I'm Japanese Original Puyo Puyo Player.

We Japanese OPP players use B Puyo, this is a clone software that can play by OPP rule.

B Puyo has many useful systems: for example online match (free match & ranked match), generating replay data and B Puyo contains AI as CPU.

If you wanna create your own AI, you can do it. Some JP player have create them own AI. (Surprisingly we can use AI instead of player at online match and we can also make match of AI vs. AI)

In addition B Puyo has more systems: around 20 special rules, practice mode (comfortable and multi-functional than official's one), changable of Puyo skin... B puyo is often said like that "Puyo Puyo, which can do everything except countering."

When using B Puyo, the system will work correctly, although the text will not display correctly if Windows is not compatible with Japanese systems. You may want to install B Puyo to try it out. 

B Puyo DL site: <http://bx1.digick.jp/puyo/dl.php>

B Puyo guide: <https://seesaawiki.jp/phoernian/d/Bpuyo%20quick%20start%20guide>

Playing movie example: <https://youtu.be/q2QI-yYTLDm>

Figure 3: Information about B Puyo from a well known Japanese player.

Disadvantages:

- Will only run on Windows, excluding Mac and Linux users
- The entire thing is in Japanese, with no translation options. Furthermore, servers are in Japan, creating ping issues for non-Japanese players. This is great for the Japanese community, but unfortunately disadvantages me as a Western player
- The resolution is locked to being a small window, making it uncomfortable to use on high-resolution displays

1.2.3 Project GelaVolt

Link: <https://gelavolt.io/>

To quote the game's creator, "Project GelaVolt is a modern, techno-themed pixel art fangame of SEGA's Puyo Puyo series, one of Japan's most successful puzzle fighter franchises. Currently, GelaVolt is focused on the competitive aspects of the game and it's intended purpose is to help introduce people and help people get better at Puyo Puyo. However, if all goes well, GelaVolt will become a free alternative that plans to solve some of the communities problems: lack of players, lack of crossplay and lack of general quality netcode." It is a Puyo-clone written in Haxe that runs in browsers.

Advantages:

- Appealing design
- Is lightweight and capable of running well in browsers
- Supports many different control schemes out of the box (controller, keyboard, etc.)
- Only version I've played that has hard drop

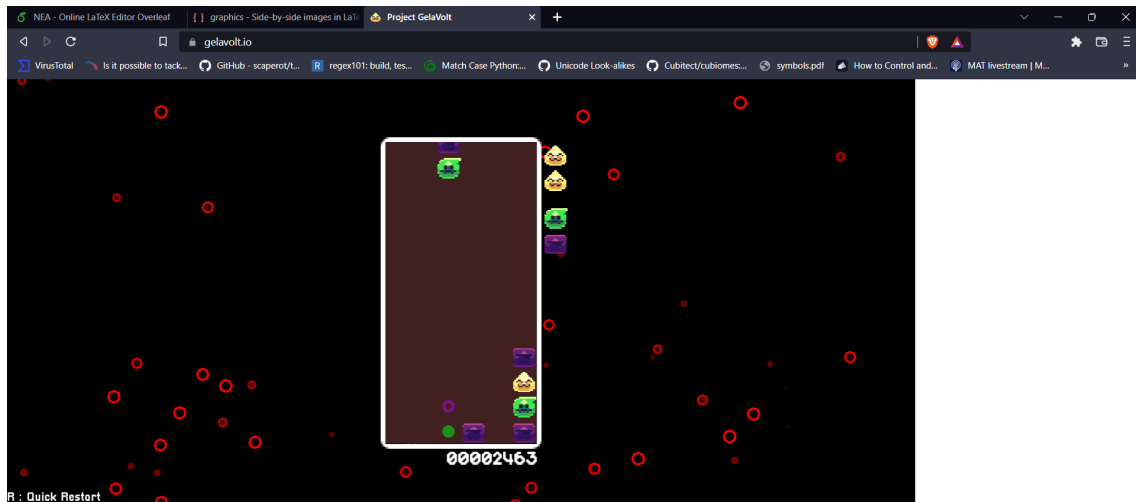


Figure 4: A screenshot of GelaVolt running in a chromium-based web browser.

Disadvantages:

- Multiplayer is in the works but is currently not supported at the time of writing
- Things such as textures are not customisable
- Is unstable and crashes regularly

1.2.4 Puyo Puyo Tetris 2



Figure 5: A screenshot of a versus battle, I'm playing Tetris and the CPU is playing Puyo Puyo.

Link: https://store.steampowered.com/app/1259790/Puyo_Puyo_Tetris_2/

Puyo Puyo Tetris 2 is the latest Puyo Puyo game released by Sega and combines Puyo Puyo gameplay with Tetris, allowing players of both games to seamlessly play against one another. It has a full story and online mode.

Advantages:

- Cutesy art style is appealing to many, but can be swapped out with unlockable designs
- Being an official release, it is very stable with a consistent online multiplayer
- CPU opponents
- Fully voice-acted story with unique and creative characters
- Active modding community

Disadvantages:

- Ranked multiplayer is fundamentally flawed as leaving matches is not punished
- CPU opponents fail to provide a challenge
- The game is very expensive, whereas all other options listed above are free
- Tsu ruleset, unable to be changed

1.3 End User Input

Being a popular game, many people enjoy the Puyo Puyo franchise, but the best people to survey for this project were the people who were most familiar with DRMBM specifically - speedrunners. A lot of the research in this document was greatly helped by the members of the "Puyo Speedrun" Discord server, and the contributors to the DRMBM-specific channel they have there. In order to efficiently collect statistical end user input, a form was created using Microsoft Forms, a PDF version of which can be found here: <https://github.com/Kris-0605/nea/blob/master/documentation/NEA.pdf>

Question 1: Have you played Dr Robotnik's Mean Bean Machine before?

- Yes
- No
- Other Puyo Puyo game

Only one answer was permitted.

1. Have you played Dr Robotnik's Mean Bean Machine before?

[More Details](#)

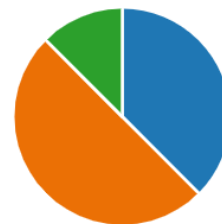
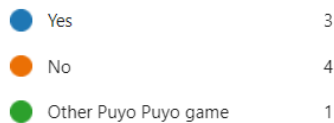


Figure 6: Results to the first survey question.

The only notable thing about this question is that anyone who answered "No" was taken to the end of the form and was unable to answer any other questions. Thus, only 4 people continued to fill out the rest of the form.

Question 2: Which modes in DRMBM are you experienced with and enjoy using?

- Scenario mode
- 1P VS. 2P mode
- Exercise mode
- Other

2. Which modes in DRMBM are you experienced with and enjoy using?

Scenario mode	0
1P VS. 2P mode	0
Exercise mode	0
Other	0

Figure 7: Results to the second survey question.

Any number of answers were permitted.

No-one answered this question, thus nothing meaningful is gained from it.

Structure "This project is intended to both remake the original game in it's purest form, apply enhancements to it, thus the game will be split into two modes, that will from now on be referred to as "Classic mode" and "Enhanced mode". Classic mode is intended to be an exact recreation of the original game, and Enhanced mode should contain any additions and improvements."

A message explaining some of the games structure that is important to understand when considering survey questions, which will be discussed further in the documented design section.

Question 3: Consider scenario mode's password feature. Enhanced mode will allow the player to use save files that store additional data such as score, times and replays. What do you believe is the best way for the password menu to be implemented?

- Classic mode will use the same passwords from the original game in their original form, taking you to a level but not restoring data such as score
- Classic mode will generate new unique password that stores a hidden save file, so that the user is still required to use a password, but this password restores data such as score when used
- The password menu should be entirely replaced by save files in both modes
- Other

Only one answer was permitted.

3. Consider scenario mode's password feature. Enhanced mode will allow the player to use save files that store additional data such as score, times and replays. What do you believe is the best way for the password menu to be implemented?

[More Details](#)

- Classic mode will use the same ... 2
- Classic mode will generate new ... 0
- The password menu should be ... 2
- Other 0



Figure 8: Results to the third survey question.

The results to this question were an exact 50/50 split, thus I shall stick to my original plan of having Classic mode use passwords in their original form without any additional data, and using save files for enhanced mode.

Question 4: What is your opinion on scenario mode's difficulty?

- Harder modes should be added to challenge more difficult players

- Easier modes should be added to help new players
- The difficulty options should remain the same in scenario mode, more customisable opponents should be available in a separate "training mode" in enhanced mode
- I don't believe any changes should be made
- Other

Any number of answers were permitted.

4. What is your opinion on scenario mode's difficulty?

[More Details](#)

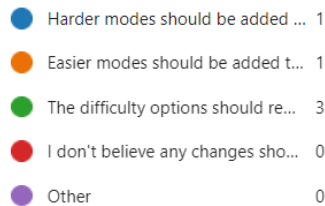


Figure 9: Results to the fourth survey question.

The majority vote represented the solution that I believe would fit best and already planned on implementing: in both classic and enhanced mode, difficulty shall remain the same as in the original. However, in enhanced mode, you can play against customisable opponents, such as the same algorithms from scenario mode with different speeds, as well as new AI altogether.

Question 5: The original game uses the OPP ruleset for scenario mode, the main difference being that garbage cannot be cancelled. What do you believe is the best configuration of rulesets?

- Classic mode scenario mode should use the OPP ruleset to recreate the original game and Enhanced mode should allow the user to choose before starting a save file
- Force OPP for scenario mode in both modes and allow players to choose Tsu when creating custom games
- Other

Only one answer was permitted.

5. The original game uses the OPP ruleset for scenario mode, the main difference being that garbage cannot be cancelled. What do you believe is the best configuration of rulesets?

[More Details](#)

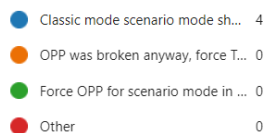


Figure 10: Results to the fifth survey question.

The only unanimous result in the entire survey, as well as the solution I was planning on implementing. I will talk more about rulesets in the documented design section.

Question 6: Do you have any other additions or comments regarding scenario mode? This question permitted a text answer.

Response ID 2 makes a very valid point. In newer versions of puyo puyo, difficulty settings change the number of colours that appear in play between 3, 4 and 5, whereas being an older game DRMBM uses 5 puyo colours in all difficulty modes. I will be sure to include the suggestion in enhanced mode.

6. Do you have any other additions or comments regarding scenario mode?

2 Responses

ID ↑	Name	Responses
1	anonymous	cheese actually tastes amazing,
2	anonymous	If for beginners, number of Puyo color must be 4 sorts. 5 colors rule is not only difficult but also increasing RNG. By the way later Puyo series have 3 colors in easiest difficult.

Figure 11: Results to the sixth survey question.

Question 7: While ambitious, the plan is to eventually include online multiplayer in the game for enhanced mode. Which of the following modes would you be interested in using?

- Customisable private rooms that you can invite other players to
- Customisable public rooms, given in a listing that anyone can join
- Ranked multiplayer, with a rating system
- A super lobby (i.e. 20+ players)
- Other

Any number of answers were permitted.

7. While ambitious, the plan is to eventually include online multiplayer in the game for enhanced mode. Which of the following modes would you be interested in using?

[More Details](#)

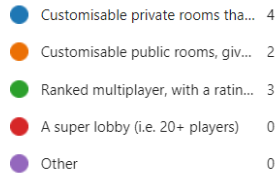


Figure 12: Results to the seventh survey question.

All of the above are planned to be implemented, but the distribution of votes gives me a timeline with which to work on each feature.

Question 8: When considering the Has Bean and Big Bean bonuses in exercise mode, which of the following statements do you agree with?

- Has Bean and Big Bean should be toggleable when playing exercise mode in enhanced mode
- Exercise mode attempts using Has Bean and Big Bean should use a separate leaderboard
- Has Bean and Big Bean should always be forced in exercise mode since they are part of the game mode, and should be toggleable when playing custom games
- Other

Any number of answers were permitted.

The majority of people would like leaderboards to be split between runs that use Has Bean/Big Bean and runs that do not. This is surprising to me, but not particularly difficult to implement thus shall be included. This overrides the one person's comment about always forcing them.

Question 9: In DRMBM, the score counter is capped at 99,999,999, and the puyo counter is capped at 9,999. In the original game, these counters froze on the event of a max out. How do you think a max out should be handled?

8. When considering the Has Bean and Big Bean bonuses in exercise mode, which of the following statements do you agree with?

[More Details](#)

- Has Bean and Big Bean should b... 2
- Exercise mode attempts using H... 3
- Has Bean and Big Bean should a... 1



Figure 13: Results to the eighth survey question.

- In Classic mode, the counter should freeze, in Enhanced mode the counter should physically expand to accommodate more digits
- The counter should always freeze
- The counter should always expand
- Other

Only one answer was permitted.

9. In DRMBM, the score counter is capped at 99,999,999, and the puyo counter is capped at 9,999. In the original game, these counters froze on the event of a max out. How do you think a max out should be handled?

[More Details](#)

- In Classic mode, the counter sh... 2
- The counter should always freeze 1
- The counter should always expa... 1
- Other 0



Figure 14: Results to the ninth survey question.

A majority of people would like to implement the solution that I personally had in mind: freezing the counters in classic mode and allowing them to physically expand in enhanced mode, so this is what I shall implement. I understood that allowing the counters to freeze in classic mode would be important to keep because a popular speedrun of the game is trying to max out the score counter in the least possible time, and removing this bug would take away from one of the ways people enjoy the game.

Question 10: As part of the project's requirements, I am going to include an online leaderboard. What stats do you think should be available as a leaderboard? This question permitted a text answer.

10. As part of the project's requirements, I am going to include an online leaderboard. What stats do you think should be available as a leaderboard?

3 Responses

ID ↑	Name	Responses
1	anonymous	Total Score, Time
2	anonymous	score, time alive, games played, matches won, win rate %.
3	anonymous	Winning percentage in ranked match For example: http://bx0.digick.jp/puyo/rate_rank.php

Figure 15: Results to the tenth survey question.

These are all fairly generic examples, but I do appreciate the link provided to a Japanese ranked BPuyo leaderboard to use as an example.

B-puyo							
<div> <div>menu</div> <ul style="list-style-type: none"> 遊び方 ダウンロード 動作環境 よくある質問 IRC(チャット) Link <div>rate</div> <ul style="list-style-type: none"> レート戦って？ ユーザー登録 登録情報修正 ランキング <div>skin</div> <ul style="list-style-type: none"> スキンって？ 使い方 作り方 作り方(サウンド編) キャラスキン 仕様 <div>server</div> </div>							
レート戦ランキング							
順位	レート	名前	総本数	勝ち	負け	勝率	最終対戦日
1	2717	TAK (TAK)	158194	96595	61430	61.1%	2022-10-06
2	2609	PHENY (pheny)	22498	12238	10252	54.4%	2022-10-07
3	2539	KUYO (KUYO)	71680	42698	28928	59.6%	2022-09-23
4	2483	AVIGMI (Avigmi)	9546	4649	4894	48.7%	2022-10-06
5	2397	TERURU (TERURU)	20452	11380	9058	55.6%	2022-09-13
6	2242	924IGO (924igo)	46624	25838	20707	55.4%	2022-09-24
7	2199	GOLGO (golgo)	97647	48458	49118	49.6%	2022-10-09
8	2146	XENO (xeno)	317978	166250	151512	52.3%	2022-10-07
9	2044	YATUHASI (yathasi)	119391	54552	64751	45.7%	2022-10-07
10	2022	RICK (rick'seye)	124454	65923	58399	53.0%	2022-10-03
11	2003	ARMAVUL (armavul)	10271	4423	5841	43.1%	2022-08-28
12	1977	NUTMEG (Nutome167)	76481	37608	38782	49.2%	2022-09-23
13	1883	KAIRO (kairo)	49406	24195	25101	49.0%	2022-10-06
14	1833	ROI (roi)	11981	5250	6717	43.8%	2022-09-16
15	1827	KUMANNNU (kumannnu)	125428	58821	66491	46.9%	2022-10-05
16	1803	MACARON (macaron)	169801	83997	85440	49.5%	2022-09-08
17	1713	ENRA (enra)	45542	20343	25149	44.7%	2022-09-25
18	1688	AEROZAWA (aerозawa)	52208	22920	29243	43.9%	2022-09-06

Figure 16: The ranked leaderboard from the Bpuyo website.

Question 11: Enhanced mode will allow the game to support a 16:9 aspect ratio. What do you believe should be used to fill the space? This question permitted a text answer.

11. Enhanced mode will allow the game to support a 16:9 aspect ratio. What do you believe should be used to fill the space?

3 Responses

ID ↑	Name	Responses
1	anonymous	a larger border, maybe a wider playing field.
2	anonymous	Nothing
3	anonymous	i dont know, so imma say cheese

Figure 17: Results to the eleventh survey question.

The solution to this problem remains to be determined, so I will probably fill it with empty space for now and see if I figure out something convenient later.

Question 12: Below are other features that I plan to implement into Enhanced mode. Rate their importance. The options contained within rows were:

- Custom texture support
- Custom handling settings
- Custom resolutions (any aspect ratio)
- Allow for custom AI and bots
- Simple modding API, mod installation built-in to the game

The options contained within columns were:

- I actively dislike this
- Would be nice to have, but not needed

- Should be included in final release
- Critical, prioritise this!

12. Below are other features that I plan to implement into Enhanced mode. Rate their importance.

[More Details](#)

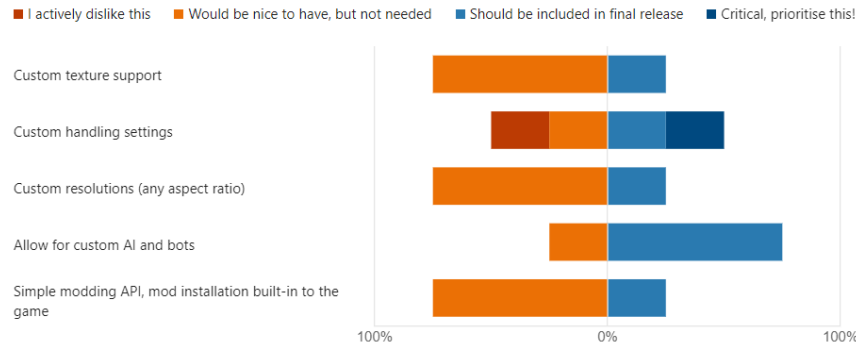


Figure 18: Results to the twelfth survey question.

All of the items listed are planned to be included, it is simply a matter of prioritising what the end user considers important. For custom textures, 3 people said it would be nice to have and 1 said it should be in the final release. The inclusion of custom texture support itself is fairly trivial to implement due to the nature of having to import textures using the engine anyway, the time-consuming part would be writing documentation that explains how people can create their own texture packs that would be compatible with the game. I now know that this should not be prioritised.

Custom handling settings was the most divisive option, with each of the 4 applicants choosing separate options. I don't understand the rational behind actively disliking custom handling settings as the default will be the same as they are in the actual game, however it may be worth considering forcing certain handling settings in ranked matches or games that will be displayed on leaderboards; perhaps different leaderboards with enforced handling and custom handling? This will have to be considered.

Custom resolutions received the same reception as custom textures - it would be nice to have but isn't overly important. Different aspect ratios are actually especially challenging and non-trivial to implement. My original design for the engine involved scene data containing a background image variable, however this static image doesn't account for different resolutions and aspect ratios. Thus in order to account for future support for multiple aspect ratios, the engine must be coded to accept the background as a function that draws the background. Then when coding scenes, the specific scene can decide the solution that is most appropriate for drawing the background, whether that be a solid colour, stretching an image to fit a resolution, having multiple images to support multiple aspect ratios or some kind of tiling solution.

Custom AI and bots received a positive reception. I shall have to create an object that allows for the implementation of AI to create the CPUs in scenario mode, thus it shall be trivial to allow modders to run their own function within this class (with some kind of primitive virus protection by not allowing external modules to be accessed).

The described "modding API" will simply be an expanded version of what is described above - allowing users to modify their game by importing new objects written by other users that are compatible with the engine, at the user's own risk.

The survey was supposed to include a poll about replays, but unfortunately I forgot to include it. I can only assume it would be a desired feature.

1.4 Input, Data Processing, Output

The program is started with main.py. This script shall verify the integrity of local game assets using SHA hashing and querying a simple API on a web server, retrieving assets as necessary. Then, the script shall import Kris's Engine, an engine that shall be written and packaged by me with the game.

Kris's Engine is built upon the idea of two fundamental class templates Scene and Entity, which shall be further described within the Documented Design section of this report. The engine shall first initialise itself, loading assets and creating the networking thread. The engine shall then import the scene that is predefined by main.py, which will probably be title.py to load the title screen. From then, the engine must complete tasks every subframe (there are 60 frames and 600 subframes in a second) such as gathering keyboard/controller input information and running the update() task of all the entity objects that the engine is responsible for. These entity objects are then responsible for completing data processing and output (i.e. pygame rendering on screen) within their update function as appropriate.

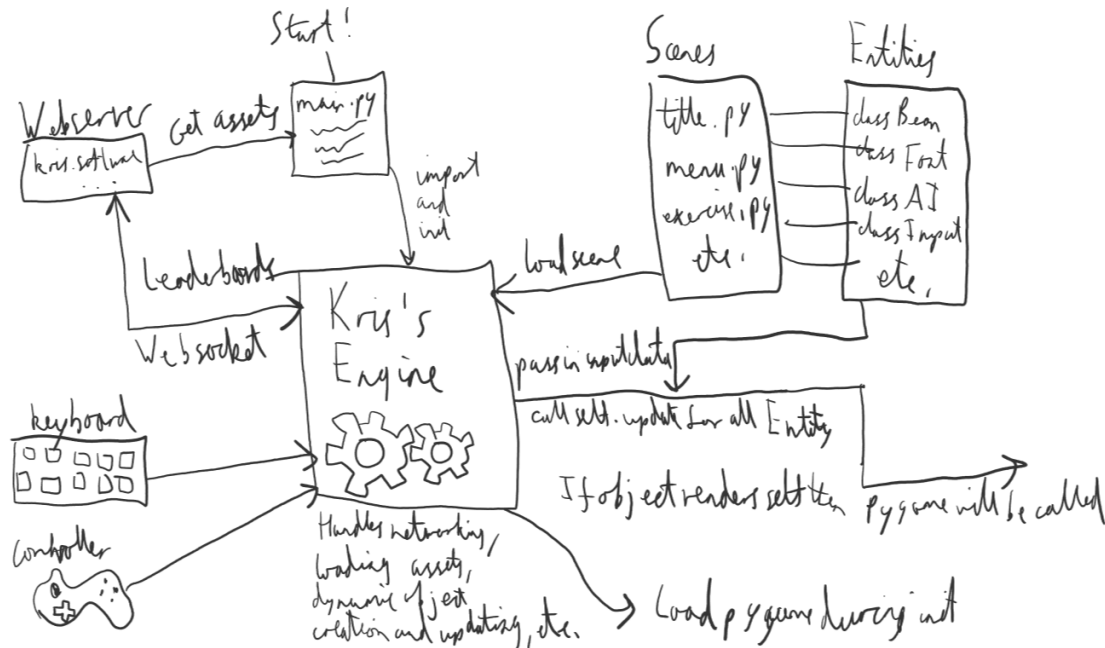


Figure 19: An Input, Data Processing, Output diagram.

1.5 Goals

The easiest way to demonstrate the various goal and features is to demonstrate what I intend to plan to add in each version. These versions will not necessarily be completed within the project's time-span as they may be extension goals and will be specified as such.

Version 0 engine CCreate an importable version of the engine as described in the documented design, implementing the Scene and Entity classes that game functionality is to be built off.

Version 0.1 alpha

The base version of the project. Should implement basic gameplay and functionality.

- Introduce basic start up script. Introduce basic repair and update functionality by hashing files and comparing against a simple HTTPS GET API.
- Create classes and methods required for basic classic mode gameplay. The easiest thing to create will be "exercise mode", a simple single-player "play forever" scoring mode. Things such as leaderboards will not be introduced yet, only the basic gameplay.
- Code all the algorithms required for basic gameplay, such as randomly generating bean pairs and identifying colour groups without excessive iteration. Establish important gameplay constants such as the formula for fall speed, scoring, garbage puyos and important handling settings such as DAS and ARR.

Version 0.2 alpha

Introduce the necessary code for multiplayer (handles code for games against an opponent i.e. CPU, NOT CODE FOR ONLINE MULTIPLAYER.) and code all AI opponents into the game.

- Code algorithms for all 13 AI opponents included in the story mode. This should be done in a modular fashion, such that it is easy for I, or someone attempting to modify the game, to create a new opponent with unique AI.

- Introduce code to accept different input methods.
- With this multiplayer code in place, it should not be difficult to add local 2 player mode. This will then mean the game has all modes from the original, thus completing the gameplay for classic mode.
- Iron out niche mechanics such as Has Bean and Big Bean.
- Create and begin adding minor enhancements to Enhanced mode, such as introducing a simple REST API based SQL score and time leaderboard for exercise mode

Version 0.3 beta

Introduce non-essential cosmetic features that will make the program useable and user friendly such as menus and transition animations. This version will be the first that will be released to a select few individuals for testing and ironing out of bugs. This version will meet all of the base requirements for the project, further updates shall exist as extension goals.

- Add menus and settings
- Fully animate sprites and transitions, allow the whole game to be accessible without the use of debug commands
- Create a simple C++ installation script that installs the game and it's pre-requisites for easy distribution and testing

Version 0.4 beta

This version will focus on the introduction of online multiplayer through web sockets and server programming. Being an extension goal, the method behind achieving this is more vague and shall be revealed if we get to that point.

Version 0.5 beta

If I have time I will attempt to create an ideal algorithm for playing the game itself, a perfect opponent to train against, and make this available to players. Bug fixes and finalisations in addition to taking requests from players about potentially adding more gamemodes. Bar the algorithm creation this is probably beyond the scope of this project.