|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Week Beginning | Submission | Timing Guidance | Total Guide |  |
| 12th Sep | Analysis Draft 1 | 1 h of launch session + 1 h over summer break | 2 |  |
| 19th Sep | Analysis Draft 2 | Approx. 4 h in lessons + 1.5 h homework 1 | 7.5 |  |
| 26th Sep | Analysis - final | Approx. 1 h lesson + 1.5 h homework 2 | 2.5 |  |
|  |  | Total for Analysis section |  | 12 hours |
| 10th Oct | Design Draft 1 | Approx. 4 h lesson time + 1 h homework 4 | 5 |  |
| 1st Nov | Design Draft 2 | Approx. 4h lesson time + 2 h homework 6 | 6 |  |
|  |  | Total for Design Drafts |  | 11 hours |
| 28th Nov | Technical Sol Draft 1 | Approx. 4h lesson time + 2 h homework 8 | 12 |  |
| 2rd Jan | Technical Sol Draft 2 | Approx. 8h lesson time + 2 h homework 13 or 14 | 10 |  |
|  |  | Total for Technical Solution Drafts |  | 22 hours |
| 6th Feb | Testing & Evaluation | Approx 6 h + 2.5 h Homework 18 | 8.5 |  |
|  |  | Total for Testing & Evaluation |  | 8.5 hours |
| Mar (tbc) | Final Report | 2.5 homework 25 |  | 2.5 hours |
|  |  | Total guided lesson & homework time |  | 56 hours |

Project proposal form

**Project title** An attempt recreate Sega’s Dr Robotnik’s Mean Bean Machine on a modern platform

**Outline of project:**

|  |
| --- |
| Dr Robotnik’s Mean Bean Machine is a game for the Sega Genesis/Sega Mega Drive game console released in 1993 and acts as an American port of the traditionally Japanese tile matching game Puyo Puyo. It’s a game that I’ve played throughout my childhood, however it is currently only playable through the use of emulation and lacks many modern features and additions. The objective of this project is to remake this game for PC using pygame, staying faithful to the original while offering the user the choice of many additional quality of life features such as remappable controls, bug fixes, performance optimisations and storing of replay data in files. Some challenges will include recreating the unique playstyles of the original AI within the game and mimicking the original control system, while allowing the user to customise the handling of the game (changing values such as DAS, ARR and SDF, something not available in the original or any official Puyo Puyo release since).  The project meets complexity requirements for the top bands of marks. Algorithmic complexity can be found in many places, such as the optimisation of updating the board. Puyo Puyo works on a colour system: 4 or more beans of the same colour disappear, and all other beans then fall down. It will be challenging to implement this in an efficient way, without iterating over ever bean on the board. Furthermore, the implementation of computer players will require various algorithms to control their method of play and difficulty. The entire program will be object oriented, with a base “bean” class and different types of beans (player placed beans, refugee/clear beans, etc.) inheriting from this class, being created in real time as they are placed on the board. This is just one example of how the project will implement complex OOP. Complex SQL will be used to store user data, leaderboard and replay information in the base model, as well as information about replays and matches if online play is implemented; of course online leaderboards and online play will involve a client-server model. A custom file type shall be made to efficiently store replace, manually altering bits to allow hours of gameplay in under a megabyte.  I believe a project like this has a lot of potential to expand given the time to do so. For example, a major feature missing from the original is the ability to easily play with other people: the nature of emulation makes this difficult. Thus given the time I intend to create a fully functioning web server to host online multiplayer matches and an online leaderboard/user data structure. Another potential expansion would be the inclusion of a “perfect” bot: either done using some kind of minmax algorithm, or if necessary the training of a Tensorflow AI model.  In conclusion DRBMB 1993 is a game that I am passionate about and would like to port and upgrade for players in the 21st century to enjoy, together. |

**Proposed third party:** “Puyo Speedrun” and “English Puyo Puyo Community” discord servers

**Qualifications for role:**

|  |
| --- |
| The public discord servers listed above contain thousands of players that are as passionate about Puyo Puyo as I am, and many players have many years of experience with the game. This user base will allow me to interview players regarding what features they would like in such as game as well as providing me with a large user base to test the project. If they like it, it gives the project purpose as a game that people can enjoy. |

**What table A (at end of document) skills will you be using?**

|  |
| --- |
| Complex data model – Multiple linked SQL tables will be used to store user data and information about games such as replay metadata in order to create a fast and efficient leaderboard system for things like time and scores, as well as the hope of using said user data to implement an online play system.  Many complex algorithms are scattered throughout. Algorithms involving an optimal way to detect groups of identical objects and update graphics on screen without unnecessary iteration, unique algorithms to match the playstyle of each CPU, compression algorithms for creating custom replay files that take advantage of individual bits, the list goes on.  Complex OOP model for general running of the game, dynamically generated objects representing the beans the player sees on screen, inheritance from a base bean class to represent different types of beans, classes for things like buttons, classes owning other classes such as a “grid” class containing all the beans for easy rendering.  Client/server model – base project will include basic requests to web server for leaderboard and replay storage. Extension goal will include web sockets to allow real-time gameplay to be streamed for online matches. |

**What are the specific questions you will address in your initial research?**

|  |
| --- |
| Will sprites be represented as vector graphics or traditional image files?  How often does the grid need to be checked for matching groups, where needs to be checked? How will the game treat different kinds of matches (computer, local multiplayer, arcade, etc.)?  How will inputs from different devices (controller, keyboard, etc.) be handled?  Should the UI be updated or kept to look as close to the original as possible? How can it be adapted to suit keyboard and mouse users?  How is the performance of a player assessed, i.e. how does the game know when to use different reactions (angry, happy) for your computer opponent?  What handling settings (DAS, ARR, SDF, etc.) does the game use?  How is scoring calculated? How is the amount of garbage/refugee beans sent to your opponent calculated?  What is the most efficient way to store inputs as a replay file?  How does each unique computer player’s algorithm/playstyle function? |

dfgfd



2021-22 Project log

A-level Computer Science (7517) Computing Practical Project (7517/C)

|  |  |  |  |
| --- | --- | --- | --- |
| **Centre number**       54208 |  | **Centre name**  Exeter Mathematics | |
| **Candidate number**    **FILL IN** |  | **Candidate’s full name** |

**FILL IN**

Project title **FILL IN**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Project type |  |  | problem ~~investigation~~ |

x

**FILL IN**

**Outline description**

To be completed by the teacher:

From the given description the project is at a standard required for A-level Yes/No

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Analysis** | | | | |
| **L** | **Criteria** | **page** | **Mark** | **Comments/evidence** |
| 3 | * Fully or nearly fully scoped analysis of a real problem, presented in a way that a third party can understand. Requirements fully documented in a set of measurable and appropriate specific objectives, covering all required functionality of the solution or areas of investigation. |  | 7-9 |  |
| * Requirements arrived at by considering, through dialogue, the needs of the intended users of the system, or recipients of the outcomes for investigative projects. |  |
| * Problem sufficiently well modelled to be of use in subsequent stages. |  |
| 2 | * Well scoped analysis (but with some omissions that are not serious enough to undermine later design) of a real problem. Most, but not all, requirements documented in a set of, in the main, measurable and appropriate specific objectives that cover most of the required functionality of a solution or areas of investigation. |  | 4-6 |
| * Requirements arrived at, in the main, by considering, through dialogue, the needs of the intended users of the system, or recipients of the outcomes for investigative projects. |  |
| * Problem sufficiently well modelled to be of use in subsequent stages. |  |
| 1 | * Partly scoped analysis of a problem. |  | 1-3 |
| * Requirements partly documented in a set of specific objectives, not all of which are measurable or appropriate for developing a solution. The required functionality or areas of investigation are only partly addressed. |  |  |
| * Some attempt to consider, through dialogue, the needs of the intended users of the system, or recipients of the outcomes for investigative projects. |  |  |
| * Problem partly modelled and of some use in subsequent stages. |  |  |
|  | No evidence presented |  | 0 | **Mark awarded:** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **Documented design** | | | |
| **L** | **Criteria** | | **page** | **Mark** | **Comments/evidence** |
| 4 | * Fully or nearly fully articulated design for a real problem, that describes how all or almost all of the key aspects of the solution/investigation are to be structured/are structured. | |  | 10-12 |  |
| 3 | * Adequately articulated design for a real problem that describes how most of the key aspects of the solution/investigation are to be structured/are structured. | |  | 7-9 |
| 2 | * Partially articulated design for a real problem that describes how some aspects of the solution/investigation are to be structured/are structured. | |  | 4-6 |
| 1 | * Inadequate articulation of the design of the solution so that it is difficult to obtain a picture of how the solution/investigation is to be structured/is structured without resorting to looking directly at the programmed solution. | |  | 1-3 |
|  | No evidence presented | |  | 0 | **Mark awarded:** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Technical solution – completeness** | | | | |
| **L** | **Criteria** | **page** | **mark** | **Comments/evidence** |
| 3 | * A system that meets almost all of the requirements of a solution/an investigation (ignoring any requirements that go beyond the demands of A-level). |  | 11-15 |  |
| 2 | * A system that achieves many of the requirements but not all. The marks at the top end of the band are for systems that include some of the most important requirements. |  | 6-10 |
| 1 | * A system that tackles some aspects of the problem or investigation. |  | 1-5 |
|  | No evidence presented |  | 0 | **Mark awarded:** |

# NOTES:

**Completeness is not only about how well a solution meets the objectives set by the student but also what an expected technical solution might perform for the solution of the problem.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Technical solution – techniques used** | | | | |
| **L** | **Criteria** | **page** | **mark** | **Comments/evidence** |
| 3 | * The techniques used are appropriate and demonstrate a level of technical skill equivalent to those listed in Group A in the **Table** at the end of this document. * Program(s) demonstrate(s) that the skill required for this level has been applied sufficiently to demonstrate proficiency. |  | 19-27 | **STUDENTS: make sure you highlight the techniques you have used in the table at the end of this document.** |
| 2 | * The techniques used are appropriate and demonstrate a level of technical skill equivalent to those listed in Group B in the **Table** at the end of this document. * Program(s) demonstrate(s) that the skill required for this level has been applied sufficiently to demonstrate proficiency. |  | 10-18 |
| 1 | * The techniques used demonstrate a level of technical skill equivalent to those listed in Group C in the **Table** at the end of this document. * Program(s) demonstrate(s) that the skill required for this level has been applied sufficiently to demonstrate proficiency. |  | 1-9 |
|  | No evidence presented |  | 0 | **Mark awarded:** |

# NOTES:

The mark to be awarded, within the level, should be decided upon using these factors:

1. The extent to which the criteria for the level have been achieved
2. The quality of the coding style that the student has demonstrated
3. The effectiveness of the solution.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Testing** | | | | |
| **L** | **Criteria** | **page** | **Mark** | **Comments/evidence** |
| 4 | * Clear evidence, in the form of carefully selected representative samples, that thorough testing has been carried out. This demonstrates the robustness of the complete or nearly complete solution/thoroughness of investigation and that the requirements of the solution/investigation have been achieved. |  | 7-8 |  |
| 3 | * Extensive testing has been carried out, but the evidence presented in the form of representative samples does not make clear that all of the core requirements of the solution/investigation have been achieved. This may be due to some key aspects not being tested or because the evidence is not always presented clearly. |  | 5-6 |
| 2 | * Evidence in the form of representative samples of moderately extensive testing, but falling short of demonstrating that the requirements of the solution/investigation have been achieved and the solution is robust/investigation thorough. |  | 3-4 |
| * The evidence presented is explained. |  |
| 1 | * A small number of tests have been carried out, which demonstrate that some parts of the solution work/some outcomes of the investigation are achieved. |  | 1-3  1-2 |  |
| * The evidence presented may not be entirely clear. |  |  |
|  | No evidence presented |  | 0 | **Mark awarded:** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Evaluation** | | | | |
| **L** | **Criteria** | **page** | **Mark** | **Comments/evidence** |
| 4 | * Full consideration given to how well the outcome meets all of its requirements. |  | 4 |  |
| * How the outcome could be improved if the problem was revisited is discussed and given detailed consideration. Independent feedback obtained of a useful and realistic nature, evaluated and discussed in a meaningful way. |  |  |
| 3 | * Full or nearly full consideration given to how well the outcome meets all of its requirements. |  | 3 |
| * How the outcome could be improved if the problem was revisited is discussed but consideration given is limited. Independent feedback obtained of a useful and realistic nature but is not evaluated and discussed in a meaningful way, if at all. |  |  |
| 2 | * The outcome is discussed but not all aspects are fully addressed either by omission or because some of the requirements have not been met and those requirements not met have been ignored in the evaluation. |  | 2 |
| * No independent feedback obtained or if obtained is not sufficiently useful or realistic to be evaluated in a meaningfully way even if attempted. |  |  |
| 1 | * Some of the outcomes are assessed but only in a superficial way. |  | 1 |
| * No independent feedback obtained or if obtained is so basic as to be not worthy of evaluation. |  |  |  |
|  | No evidence presented |  | 0 | **Mark awarded:** |

|  |
| --- |
| **Total mark /75** |
| **Concluding comments:** |
| **Signed: Date:** |

**Group A**

|  |  |  |
| --- | --- | --- |
| **Model** | **Algorithm** | **Why did you use this skill?** |
| Complex data model in database (eg several interlinked tables) | Cross-table parameterised SQL  Aggregate SQL functions  User/CASE-generated DDL script |  |
| Hash tables  Lists  Stacks  Queues  Graphs  Trees  Structures of equivalent standard      Files(s) organised for direct access | Graph/Tree Traversal    List operations        Linked list maintenance    Stack/Queue Operations Hashing |  |
| Complex scientific/ mathematical/ robotics/ control/ business model | Advanced matrix operations  Recursive algorithms    Complex user-defined algorithms (eg optimisation, minimisation, scheduling, pattern matching) or equivalent difficulty    Mergesort or similarly efficient sort |  |
| Complex user-defined use of object- orientated programming (OOP) model, eg **classes, inheritance, composition, polymorphism, interfaces** | Dynamic generation of objects based on complex user-defined use of OOP model |  |
| Complex client-server model | Server-side scripting using request and response objects and server-side extensions for a complex client-server model  Calling parameterised Web service APIs and parsing JSON/XML to service a complex client-server model |  |

**Group B**

|  |  |  |
| --- | --- | --- |
| **Model** | **Algorithm** | **Why did you use this skill?** |
| Simple data model in a database (eg. two or three interlinked tables) | Single table or non-parameterised SQL |  |
| Multi-dimensional arrays    Dictionaries    Records | Bubble sort    Binary search |  |
| Text files    File(s) organised for sequential access | Writing and reading from files |  |
| Simple scientific/mathematical /robotics/ control/business model | Simple user defined algorithms (eg a range of mathematical/statistical calculations) |  |
| Simple OOP model | Generation of objects based on simple OOP model |  |
| Simple client-server model | Server-side scripting using request and response objects and server-side extensions for a simple client-server model    Calling Web service APIs and parsing JSON/ XML to service a simple client-server model |  |

**Group C**

|  |  |  |
| --- | --- | --- |
| **Model** | **Algorithm** | **Why did you use this skill?** |
| Single-dimensional arrays | Linear search |  |
| Appropriate choice of simple data types | Simple mathematical calculations |  |
| Single table database | Non-SQL table access |  |