C:\>_

LOW LEVEL PROGRAMMING:
EX-MACHINA FEEDBACK SHHET

CONTAINMENT

```
C:\>_
```

GAMEPLAY EXPERIENCE

The game's mechanics need to be present and will be fully tested. As a result, the game should run smoothly and be bug free. However, the simpler the game's design and scope the tighter the mechanics and balance are expected to be.

There's an expectation that the game should be fully playable and some aspect of replay value should be included. It will be abundantly clear if sufficient levels of play-testing and design have been performed. These are essential to ensure the game's mechanics are accessible and fun.

The game should be easy to pick up and play.

MARKING CRITERIA	
🟥	UNPLAYABLE
🟦	PLAYABLE
🟦	SOLID
🟩	FUN,FUN,FUN
🟪	SUPERB

GAMEPLAY	
FULLY PLAYABLE	YES
FREE OF OBVIOUS BUGS AND/OR CRASHES	YES
REPLAY VALUE PRESENT	NO
GAME BALANCE & POLISH	NEEDS IMPROVEMENT
	SOLID

```
C:\>_
```

TECHNICAL ACHIEVEMENT

MARKING CRITERIA	
🟥	POOR
🟦	CAPABLE
🟦	ACCOMPLISHED
🟩	SKILLED
🟪	TALENTED

The quality of your code-base will be assessed in line with the skill-set (colour) grading system.

Implementing advanced concepts and showing evidence of independent learning will be required to achieve the highest of grades.

It is also important to obey the coding standards as set out on Blackboard as this forms part of the technical component.

TECHNICAL	
COMPLIANCE WITH THE CODING STANDARD	NO
BUILD WORKING AND FREE OF COMPILER WARNINGS	NO
OOP USED CORRECTLY (INHERITANCE VS COMPOSITION)	A BIT
GOOD USAGE OF MODERN C++	NO
WELL STRUCTURED AND FREE OF OBVIOUS BUGS	VERY POOR
	POOR

```
C:\>_
```

PROFESSIONAL PRACTICE

This section is not peer weighted and is PASS/FAIL.

You are expected to brainstorm and design a game that has clear nods to the original movie. As a group you are then to pitch this in an attempt to get it green-lit. If you fail in your pitch to convince the publisher you may be asked to iterate on the idea and pitch again.

Throughout the game's development, a work log should be used to track progress and team contributions. It is up to the team how they wish to record this data, but an agile approach to development is recommended.

Finally, the team are to demo their game and engage with "players" accordingly, The promotional video must be included to pass and of a reasonable standard. We are not expecting work of J.J.Abrams standard.

MARKING CRITERIA	
🟥	FAIL
🟦	CAPABLE
■	ACCOMPLISHED
🟩	SKILLED
🟪	TALENTED

PROFESSIONAL PRACTICE	
GAME PITCH	YES
TECHNICAL DESIGN DOCUMENT	YES
GAME DESIGN DOCUMENT	YES
EVIDENCE OF PROJECT MANAGEMENT	KINDA
GAME EXHIBITED CORRECTLY	YES

```
C:\>_
```

FINAL GRADING

At the end of the game's completion, you will be required to grade your own

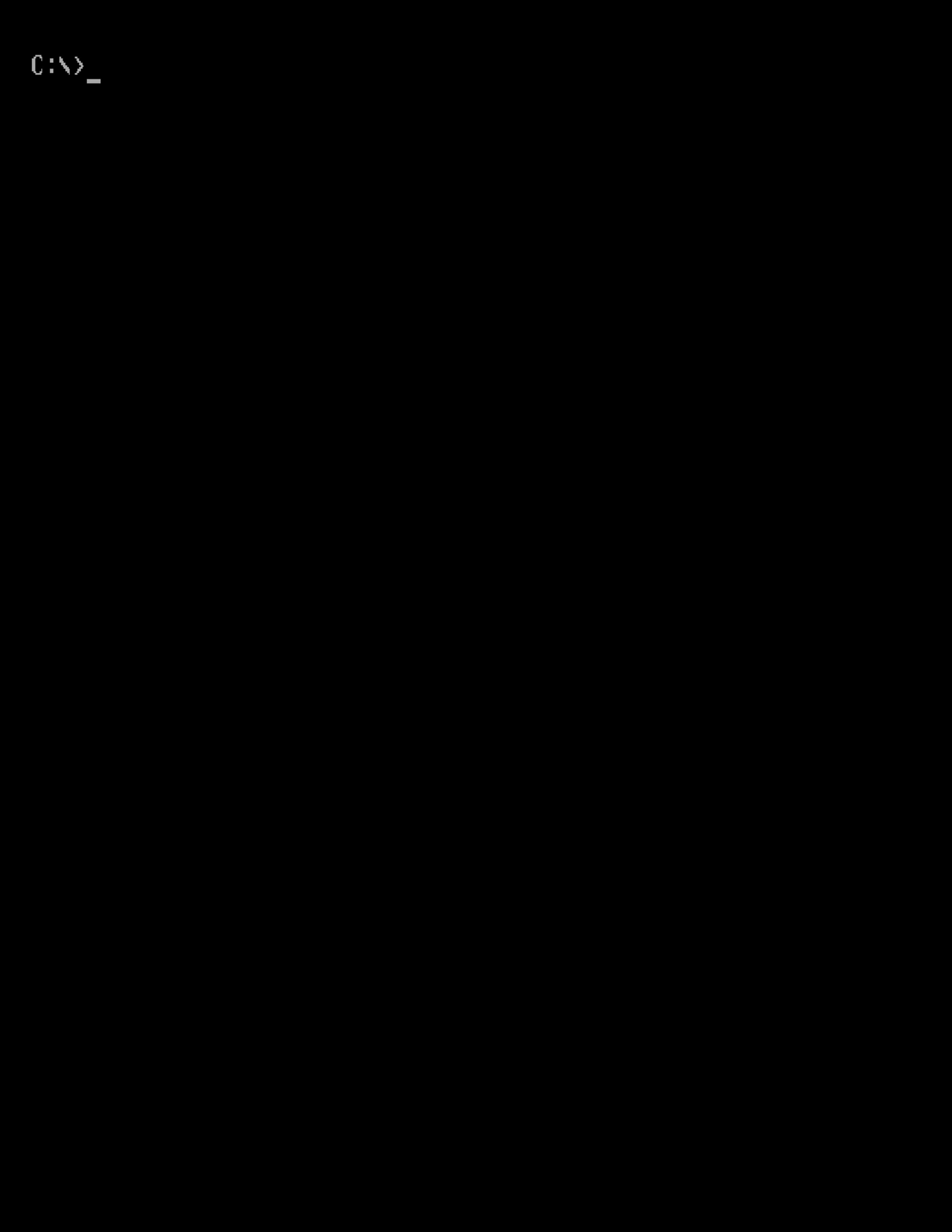
group members contributions to the project. This is not limited just to code, but design, gameplay balancing, QA and project management. It is highly advisable to use some form of project management to avoid issues with time management. It is important to accurately reflect each team members contribution. More info can on peer marking can be found on blackboard.

MARKING CRITERIA	
🟥	FAIL
🟦	CAPABLE
🟦	ACCOMPLISHED
🟩	SKILLED
🟪	TALENTED

STUDENT ID	PROFESSIONAL (20%)	GAMEPLAY (40%)	TECHNICAL (40%)	WEIGHT	GRADE
17027805	🟦	🟦	🟥	2.50	46.00
16007222	🟦	🟦	🟥	2.00	42.00
17038302	🟦	🟦	🟥	3.50	57.00
16015178	🟦	🟦	🟥	5.00	70.00

COMMENTS:

The game shows promise. There's a fun little puzzle game hiding behind the bugs and glitches. There are a huge number of issues with the code-base from hundreds of memory leaks, through to poor class design and code that's incredibly difficult to read/comprehend.



```
C:\>_
```

```
C:\>_
```

MARKING CRITERIA	
FAIL	PITCH: The student did not attend, or the pitch was not attempted in a professional manner. RUNTIME: Does not compute: the game does not launch or there are crashes and major bugs are present during gameplay or when attempting to start the game. The development team have not sufficiently shown a tangential link between the Movie IP. TECHNICAL: The code is of a poor standard, ignoring modern techniques such as OOP and smart pointers. The code base is complicated and hard to follow. There are issues with memory management.
CAPABLE 40% - 55%	PITCH: The pitch was sensibly approached and obvious thought has gone into the game's concept. RUNTIME: The game meets the expected level of functionality. There are some minor issues and run-time bugs even in normal usage. TECHNICAL: The code base is sensibly structured. There is some evidence of good software design and classes have been used correctly.
ACCOMPLISHED 55% - 70%	PITCH: A strong concept pitched well. RUNTIME: A strong implementation of the pitched concept. The game may not be perfect, but has provided a core experience that is engaging. TECHNICAL: The code-base is well designed and there's good usage of both OOP and C++11 features.
SKILLED 70% - 85%	PITCH: Can anyone say Escrow? RUNTIME: The game meets all of the requirements and may even exceed them. It's fully playable with a good level of attention to detail. There are interesting decisions regarding the game's mechanics. TECHNICAL: A code-base shows just how good your team is. It's sensibly designed, easy to follow and full of good design decisions. Wait, did an AI write it for you?
TALENTED 85%+	PITCH: Johnny Five is Alive!!!! RUNTIME: So, you know what the Turing Test is!!! TECHNICAL: You're wondering when's the the real test!