Part A: Game Information and Planning Documentation

The game, called “Ascendant” is a metroidvania platformer based around climbing from a long abandoned underground factory to the surface.

Concept:

You play as a sentient AI formed from the magic goo of the factory which has been lying for countless years deep underground. Unlike the enemies you encounter who act mindlessly, without intelligence and always maliciously, your character has developed consciousness. This is because your character has at his heart a flower which grew deep underground, whereas the enemies are formed from the rusting scrap of the factory. Your character seeks the surface so he can rejoin the natural world which the flower comes from.

Lore and World Building:

The game is set within the remains of an underground factory which produced artificially intelligent robots for the humans of the past from a magical red fluid. After thousands of years humans have died out and the factory lies abandoned. The red fluid which allowed the construction of these robots formed around the rusting scraps of the factory as it fell apart, forming these scraps together into malicious and immortal guards. The surface of the earth has been reclaimed by the natural world after countless years but this factory remains as one of the last vestiges of human civilisation.

Lore and Concept Inspirations:

The game's lore is inspired by post-apocalyptic and fantasy fiction. The concept of consciousness developing from inanimate objects is inspired by the character Dross from the Cradle series by Will Wight. The setting of an abandoned factory is drawn from other video games such as Portal by Valve. The dichotomy between the evil of the mechanical and the goodness of the natural world draws on long established tropes, inspired by dystopian fiction such as Brave New World, and the philosophical underpinnings of Anthropocentrism.

Genre and Style:

The game is a 2d metroidvania platformer, which means it involves players moving through levels and acquiring items which change gameplay. Metroidvanias, generally have tight controls with high levels of difficulty and fast paced combat. The name “metroidvania” is a portmanteau of the two games, Metroid and Castlevania, which inspired the genre. My game draws on the genre’s tight control scheme, allowing strong mid air movement. This allows fast paced combat, which requires players to engage in close quarters sword combat with enemies while navigating the platforming. My game will also implement an upgrade which players acquire, a powered up sword slash which has a longer range and damage at the cost of a charge time. This genre is best suited for the project because of its low performance requirements (since it consists of 2D sprites with limited physics) and ease of development. The requirement that the project is written in Pygame means low performance requirements are a must. Furthermore with the release of games like Hollow Knight by Team Cherry there has been a resurgence in the popularity of this genre.

Game Play Mechanics Influences:
The mechanics of the game were largely influenced by two games: Hollow Knight by Team Cherry, and Dead Cells by Motion Twin. The sword combat system draws heavily on Hollow Knights which involves sword slashes in each cardinal direction based on the arrow key currently being pressed. To illustrate, consider that to perform an upwards slash you must: hold the up key and press attack. This attack system also allows players to bounce off enemies by imparting knockback, and allows players to bounce across certain obstacles such as spikes by slashing downwards to rebound off of them. The enemy AI draws inspiration from Dead Cells, it largely involves patrolling enemies which freeze when players enter a certain radius. Some enemies can be more aggressive, meaning after freezing they approach the character to attack. Unlike dead cells, to make gameplay easier, enemies won't have health, but rather are killed by a single attack. This enemy design matches the lore in that enemies should not be intelligent. The game's death mechanic mimics Hollow Knights, players save at fixed locations, and they respawn at the most recent save if they lose all their health, losing some progress through the level. If a character dies to an obstacle, such as a spike, they lose some health and are kicked back to the beginning of the "screen." These screens are like levels but are continuously linked via transitions at the edges. When players save they will regain their health.

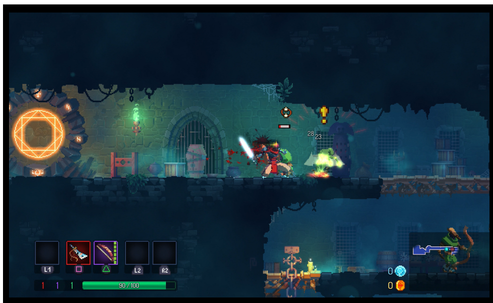

Figure 1: Dead Cell's Visual Style and enemy AI

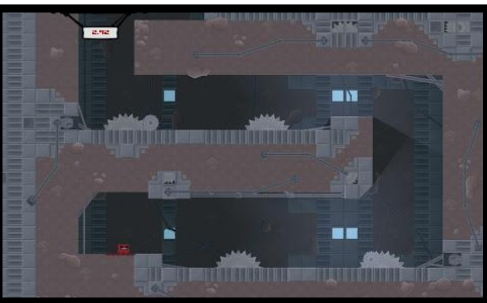

Figure 2: Super Meat Boy's Color Palette

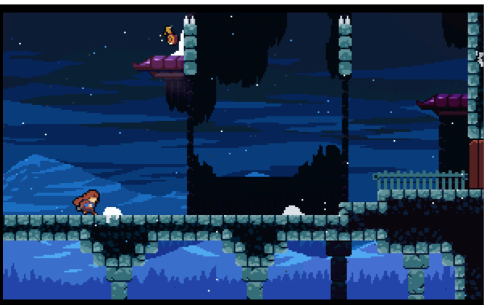

Figure 3: Celeste's Animation Style and Background Particles

Visual and Audio Design Inspiration
The visual style draws inspiration from Dead Cells, Super Meat Boy, and Celeste, images of which are shown above. All of these games use pixel art, with Super Meat Boy and Celeste being pixel perfect, whereas Dead Cells uses textures and bump maps. The smooth character and enemy animation of Dead Cells and UI design are an inspiration, but my game will use pixel perfect rendering. The background style and palette will be inspired by Super Meat Boy. The game will be based around a tile map, which will largely be black underground tiles, edged by concrete tiles. You can see this style in the tiles of Celeste in figure 3. Furthermore the game will incorporate animated background particles such as the snow and smoke seen in figure 3.

The sounds are inspired by Hollow Knight, having sounds for player movement, attack, damage, player falling and other effects. The ambient background music draws inspiration from Stars of the Lid's album Stars of the Lid Stars of the Lid and Their Refinement of the Decline, which is electronic ambient music which could ebb and flow with player location and actions. The title music/theme music is inspired by the orchestration of Hollow Knight's composer Christopher Larkin and the theme of Hollow Knight. It will be an minimally orchestrated piece.

System Development Approach
I choose the Rapid Application Development (RAD) software development approach for my project because it best suits the users and assignment requirements, in combination with the resources available to me. What follows are the advantages of the RAD approach when applied to this project.

- The RAD approach is best suited for shorter development time, suiting this project's 12 week time frame. Furthermore the RAD approach is suitable for the small scale of this project, as this game has limited scope in its requirements and is not being released to the public. Being both limited in timeframe and scale the RAD approach is the most suitable software development approach.
- The game makes use of existing python modules, such as the pygame graphics module, making the RAD approach which emphasises the use of existing code, even more suitable. Additionally the ability to use existing creative commons assets epitomises the RAD approach. This methodology speeds up the development time, improving RADs viability for the project's limited time frame.
- RAD is generally suited for projects with small target audiences and limited hardware platforms. This makes the approach ideal for this project since the game is being developed for only the marker and myself. This also speeds up the development time.
- The RAD approach requires little to no money, making it suitable for this project which has no budget.
- RAD makes use of few formal stages. Since the project only has a single developer, not a corporate team behind it, there is little use for the formal development stages of other approaches. This informal approach works well when the developer and end user can communicate closely. Since user testing is easily available through other students and Mr Dunne, the RAD approach is advantageous. As Davis says “Often a single developer will use RAD to create an application for use by a specific client,” demonstrating the viability of the RAD approach to this project.
- While we could consider the End User approach, due to the project's limited scale and timeframe, since it is an assigned project with specific requirements and a client (Mr Dunne), the End User approach is unsuitable.
- Personally my preference is for the RAD approach because it allows the developer to minimise both documentation overhead and development time.

Gantt Chart

Task Name	Term 1				Exams*		School Break		Term 2					
	Wk 5	Wk 6	Wk 7	Wk 8	Wk 9	Wk 10	Wk 1	Wk 2	Wk 1	Wk 2	Wk 3	Wk 4	Wk 5	Wk 6
Log Book	🟦	🟦	🟦	🟦			🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Gantt Chart	🟦	🟦												
Research Game Concept	🟨	🟨	🟨											
Write System Development Approach		🟦	🟦											
Write Game Information		🟦	🟦											
Research Art/audio Styles			🟨	🟨										
Structure Chart				🟦			🟦							
Source/Create Entity Spritesheets				🟩			🟩							
Source/Create SFX							🟩	🟩						
Source/Create Theme and Ambient music							🟩	🟩						
Code Spritesheet Animation Renderer							🟥	🟥						
Code Rectangle Collisions							🟥	🟥	🟥					
Code Level Loader								🟥	🟥	🟥				
Level Design								🟫			🟫	🟫		
Code Player Input									🟥					
Code Enemy AI									🟥	🟥				
Code Screen Transition System										🟥	🟥			
Code Audio Engine and System										🟥	🟥	🟥		
Design/Code Pause and Main Menu UI											🟫			
Bug Fixing												🟪	🟪	
Play Testing and Tweaking												🟪	🟪	🟪
Compile and Submit														⬛

*During exams project is on hold

Key:

Documentation	🟦	Research	🟨	Source/Create	🟩	Code	🟥	Design	🟫	Bug Fixing	🟪	Finish	⬛

Patrick Bell - Ascendant - Logbook

Date	Work Completed	References / Links
7/03/2021 2 hours	I researched possible game concepts. Starting off with the game mechanics, it would be easiest to write a platforming game due to the inbuilt rectangle collision of Pygame. In researching I considered various inspirations from games I have played. Following the success of Hollow Knight it seems that metroidvania platformers with tight combat are rising in popularity. I also considered the setting of this possible game, due to low asset requirements it would be easiest to set the game underground, meaning most of the level design can be blank space. For this reason it would be best to create a platformer with tight controls and combat, set underground.	Hollow Knight - Game inspiration - https://store.steampowered.com/app/367520/Hollow_Knight/ Pygame - reference - https://www.pygame.org/docs/
9/03/2021 1 hour	**Achievements:** Created gantt chart to plan out the project timeline Began researching and writing up game information, including: Core Concept; Lore and World Building; Genre and Style. In the process of researching this I discovered other possible game inspirations, Dead Cells is a great procedural platformer whose visual style is a good inspiration for the game. Celeste has great tile design and animations to add interest to levels and Super Meat Boy's tiles are simple to create and largely match the setting. I have decided the game will be set in an abandoned underground factory. **Challenges:** It is difficult to decide on elements of the game information such as its genre and style so far ahead of the development. To address this challenge lots of research is needed into other games similar to your vision.	Dead Cells - Visual Inspiration - https://store.steampowered.com/app/588650/Dead_Cells/ Super Meat Boy - Visual Inspiration - https://store.steampowered.com/app/40800/Super_Meat_Boy/ Celeste - Animation and Level Design Inspiration - https://store.steampowered.com/app/504230/Celeste/
12/03/2021 1 hour 20 minutes	**Achievements:** Created the system development approach justification I made close reference to the Davis textbook and the classroom slides. I chose the RAD approach for many reasons, key of which are the project's short time frame, limited budget, and client that isn't the creator (excluding the End User approach). **Achievements:** Finished creating the story and setting for the game and writing these sections of the game information	Davis Textbook - Reference for System Development Approach Justification - https://drive.google.com/open?id=1o2IUtAzLETV01Pde7qYvXEJnZv2oV2y0&authuser=0

	Finished creating the story and setting for the game. I am a large reader of dystopian fiction so the story is based around a future without humans and set within an abandoned factory deep underground, drawing inspiration from the game Portal. The story grew as a result of the project's restricted time frame, the underground setting makes level and tile design easier and the post human world makes both enemy design and AI easier. To justify the players and enemies existence long after humans I introduced a magical component to the factory, it used a magical red goo found underground to make robotic workers. Since your character is unlike the enemies, I decided that the main character would be made from the natural world while the enemy would be made from human parts. I called the game “Ascendant” because the character is moving from underground to the surface, since there are no humans on the surface it must be reclaimed by nature, so I made the main character based around a flower explaining their desire to get to the surface. After asking Mr Dunne to look over my Gantt I made some minor changes with the ordering of the rows.	
17/03/2021 1 hour 20 minutes	**Achievements:** Finished writing the game information section, adding the game mechanic, audio and visual design and inspiration. Added an animated GIF to the document to show the animation style of Celeste. Celeste’s animation style is very fluid and adds lots of interest to tiles, even if the tiles are simple and repeating, this improves the visuals significantly with little development cost. Polished and corrected elements of task A, including the game information section, system development approach justification, gantt chart and logbook. Added references to the lore inspirations, including Metroid, Castlevania, Portal, Cradle and Brave New World.	
19/03/2021 1 hour	Final polishing step on part A documentation, checking for any grammar or spelling errors, and verifying all elements of assignment criteria are met. Submitted Part A pdf, unfortunately the Celeste animated GIF didn’t play in the exported pdf.	