Internal SWOTs:

Waterfall:

Strengths:

* Clear schedule.
* Allows for exact deadlines.
* Easy to tell where the project will be at any point in the future.
* Since the core game won’t be changed after the coding phase, there will be no need to re-test stuff.

Weaknesses:

* Any delays or unexpected problems break the entire plan.
* Doesn’t allow for adaptability.
* Stops the “Follow the fun” mentality that games need to be enjoyable to play.
* User feedback is pretty much non-existent.

Opportunities:

* By forcing yourself to meet strict deadlines, you’re guaranteed to have a completed game at the end of the cycle.
* Assets won’t become redundant as the project vision doesn’t change.

Threats:

* “A delayed game is eventually good but a rushed game is forever bad”. Similarly, due to the pre-existing indie game market saturation, if I force a game to be imperfectly completed, then runs the risk of not having any market-value.

Agile:

Strengths:

* Iterative development allows for unlimited user feedback.
* Less strict deadlines allow for tasks to be actively re-scheduled.
* Allows for developers to test whether features are necessary or enjoyable before implementing them into the final product.

Weaknesses:

* If a phase takes more iterations than expected, then the projects end deadline could be delayed.
* User feedback isn’t always useful or aligned with the project vision.

Opportunities:

* Extra feedback allows for new ideas and features to be implemented and removed dynamically.

Threats:

* Iterations never result in a perfect product, so it can be hard to determine when the game is “good enough” to transition to the next phase.

Kanban:

Strengths:

* Super dynamic, tasks can be re-assigned, delayed or focused on earlier than anticipated based on motivation, real-time opportunities etc.
* Effectively breaks down large goals into feasible tasks.
* Visually easy to tell what needs to be/has been completed.

Weaknesses:

* Lack of any clear deadlines can delay the project.
* Requires a lot of effort to break down all the large goals into bite-sized tasks.

Opportunities:

* Allowing the developers to work on tasks based on their mood or available time can speed up task completion.
* Stops developers from getting too overwhelmed by tasks and deadlines.

Threats:

* Difficult to notice if a project is struggling to reach deadlines until very close to them.

Based on these SWOTs, I’ve determined that the Kanban project management technique will work best for my game, as a dynamic and flexible workflow as well as user feedback is vital to creating a good game. As such, I will break down the project into small tasks, and then assign tasks to different sprints, each sprint lasting 2 weeks.

Sprint 1 (30/08/21 – 12/09/21):

Goals:

* Create scalable dialogue system.
* Write and implement story and combat tutorial.
* Build a basic combat prototype, as well as several variations of the movement and combat systems to trial after lockdown.
* Build a temporary movement system.
* Implement some basic enemies for combat testing.
* Add monster pillars and checkpoints.
* Add rock introduction cutscene.
* Create all the forest art and tile sets.
* Add real-time dialogue for the rock.
* Create a prototyped version of the side-character introduction.
* Start designing the first mini boss.
* Create the entrance to the underground city.
* Create the character creator.

Timeline

Description automatically generated with medium confidence