Post-Mortem Anastasis

Daniel Dworski, Aaron Wagner, David Dworski; Villach 23.06.2025

Daniel Dworski is the only one attending “Computer Games”. The collaboration with David Dworski and Aaron Wagner was based on having a fun little project to work on, when there was time. The main goal of the game was not to have a working finished game until the time of the deadline (23.06.2025), but to create a start for this game in order for it to become big and to be worked on occasionally, especially in the free time of the summer holidays.
The target of this game is to be a survival RPG with a focus on combat, collecting items and always increasing content; this is why a modular and strong structure of the project is essential in order to save a lot of future refactoring work.

Development Process:
The first stage was defining, what the game should be, and subsequently how the gameplay and the unique selling point of this game was supposed to be. After much talk with the developers and creation of the game concept, the basics were clear.
The framework on how to organize everything in Godot was researched and uploaded on GitHub, where everybody was able to work on it in their versatile free time.
No clear scheduling timetable or goals were set, so everybody did what they wanted to do and had fun doing so.
While working on aspects of the game, frequent talks about, what was worked on, and sharing thoughts united the spirit and goals of the developers.
Technologies used are of course ChatGPT to get insights on how Godot works, just like many YouTube tutorials. In order to create custom Sprites, “Piskel” was used to create simple animations (HP-Bar). But when increasing the ambition of good animations, a variety of software was considered, but ultimately “PixelOver” was used, but sadly never finished.
The main problem arising from the development was creating custom sprites, in order for the game to feel authentic, but because of the limited artistic potential of the developers up to this date only sub perfect sprites are used. And other problems arose due to the learning curve of the software being used leading to extended unproductive periods of time.

Discipline-Specific Insights:
What went Wrong?

- When first starting with the development an unexpected shortage of time lead to one of the developers (David Dworski) leaving the development almost completely.
- The aim of custom sprites despite the lack of artistic capabilities of the developers had lead to much time loss and reduced efficiency.
- The goal of the game was too great to be possible in the given time frame of the development, so only a small demo could be finished until the deadline (23.06.2025)
- The inexperience of Godot and lack of exercise in games lead to more time being spent thinking and researching, instead of actually working and implementing features.
- A lack of planning the development phase and directing tasks ended in the demo being small and a slow development process.

What went Well?

- Although the game is still in its early stages, each aspect implemented seems to be working quite well and satisfactory with its experience.
- The general idea of the game comes to live with this first demo and shows to be a good start for the idea at the start of this development.
- The detailed thoughts of every implementation in the game indicates that a high performing game can emerge in the future.

Lessons Learned:

Creating new sprites, especially animations is not worth doing, when there is a lack of talent, experience and motivation. Just downloading some images would greatly improve performance and size of the game.

Too much thinking about every single line of code can create a good project, but is way too inefficient and frustrating, when something isn´t working properly or easily. Just focusing on a light demo and refactoring the code afterwards would be a better development flow.

A clear development plan should be made in advance, leading to some fast working demo and distribution of work.

Action Items:

- Avoid making everything by themselves, like importing sprites and animations instead of learning how to make them from scratch.
- Instead of trying to make everything perfect, aim for fast development, like adding many features and focus on making the project clean afterwards.