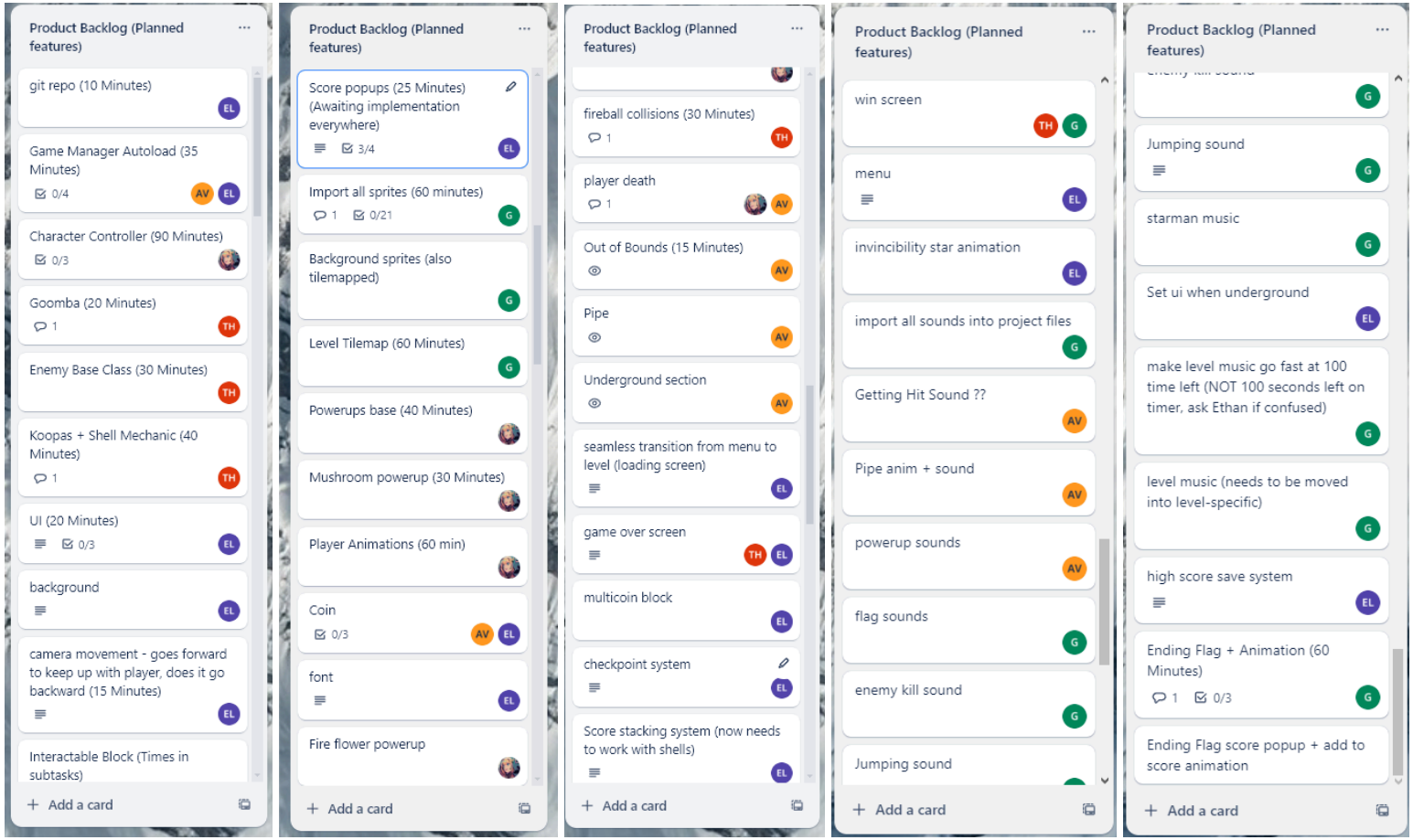
12PM Group 3 - Lab 2 Documentation

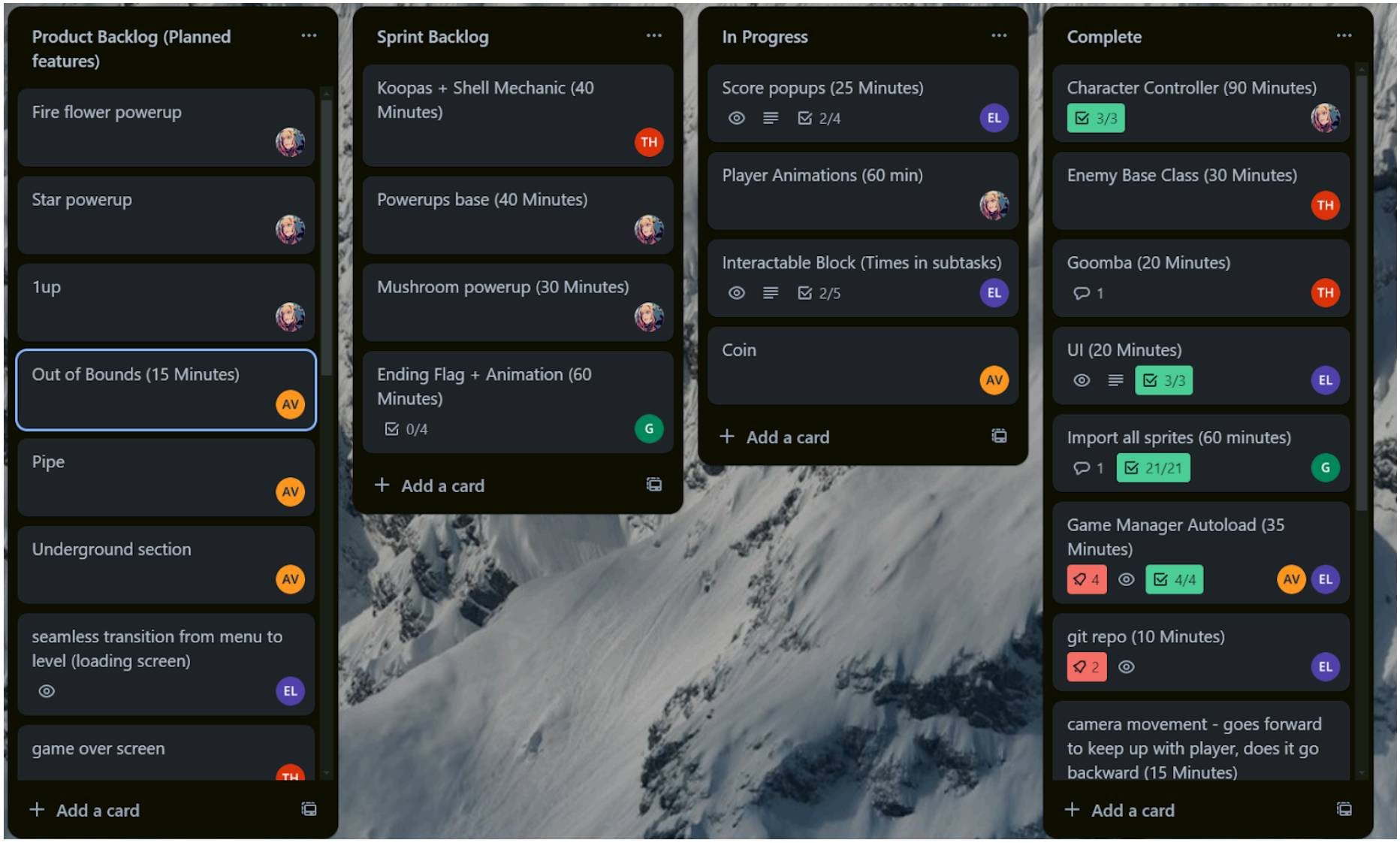
Post-Planning Meeting Trello Board (Pre-development phase)



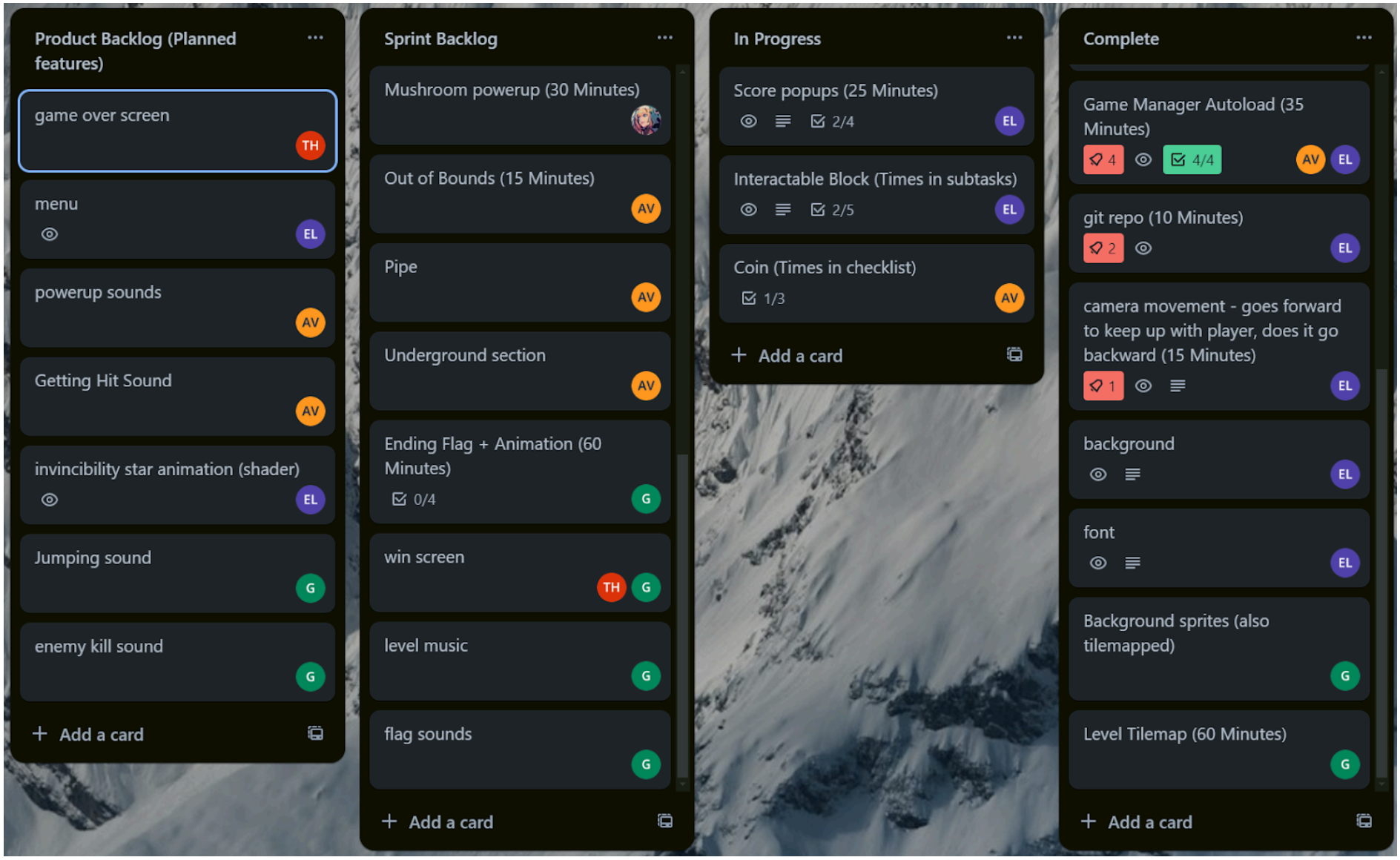
SCRUM 1 Meeting

Who	What was completed?	What will you be doing in the next sprint?	What is stopping you from completing tasks?
Ethan	UI, Game Manager autoload, setup Git Repo, background, font	Finish score popups, finish blocks, menu, loading, checkpoint	• Refactoring code that didn't need to be refactored. • Dependency on player implementation when waiting to implement blocks. • Time due to work (non-negotiable)
Xander	Character controller, player animations, animation logic	Powerup base class, powerup logic (for mushroom, flower, star, 1up)	• "Region-based" animation setup was time consuming • Needed to plan what needs to change with each powerup before any logic was implemented
Alex	GameManager, Learning the basics of Godot, Underground coin	Coins from block, Coin Hud, Death (Straight up)	• Learning Godot • Also 3D animation lab wk 1
Tom	Base enemy class, goombas, goomba dying animation, goomba movement	Koopa and shell mechanic, then pull things off the product backlog	• Incomplete player node
Lauren	Imported all asset sprites, set up tilemap for game	Ending flagpole animation and code, import all sound effects	• Took awhile to find the correct assets to import since different emulators use a slightly different colour palette and i wanted everything to be consistent • Still learning parts of the game engine

Pre-meeting Trello board



Post-meeting Trello board

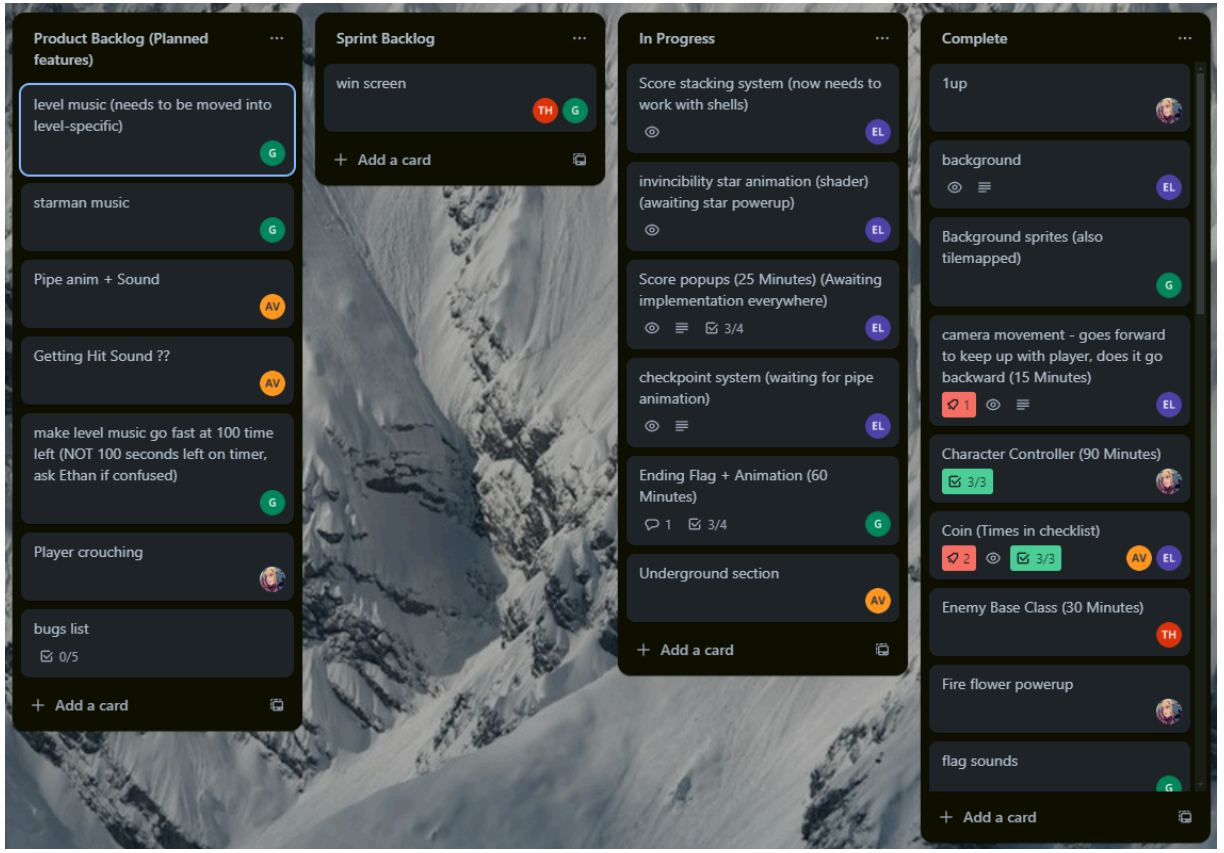


SCRUM 2 Meeting

Who	What was completed?	What will you be doing in the next sprint?	What is stopping you from completing tasks?
Ethan	• High score system • General bug fixing • SFX work • All block types complete	Externally dependent tasks: • Score stacking system • Score popups on all required objects • Spawning player when leaving underground Additional: • Bug fixing • Complete tasks from sprint backlog	• Dependencies on other tasks • Bloated player controller script is difficult to read and interface with • Lack of documentation in scripts (all group members) • Collision layers not being named
Xander	Powerup base class, powerup logic (for mushroom, flower, star, 1up), player hurt + flashing invincibility state	Player crouch animation and logic	• Git commit conflicts • Needing to refactor player script so other team members can work with it -- something to be improved in future projects
Alex	Pipe transport services are now available. Death controller for the player + death animation	Rest of the underground (background, coins, music) Pipe animation	• Still learning Godot (still spend more time than needed looking though Godot docs)
Tom	Koopas and shell implemented, shell kills goombas and player when moving, star invincibility implemented	fireball collisions and any other enemy bugs that come up	• Difficulty with collision layers
Lauren	Imported all the sound effects and music into the game, defined correct flag regions for scoring, created ending flag animation +	Add the correct score when the player touches the flag, add in all other sound effects that are not present in-game, the flag itself needs to animate down the pole with the	• Needed to learn how to set up the animations correctly • Had a bit of trouble interfacing with existing code to get it to happen at

	added sound effects + ending music	player.	the correct time (lots of debugging) • Messed up a merge commit so i had to redo some work

Pre-meeting Trello board

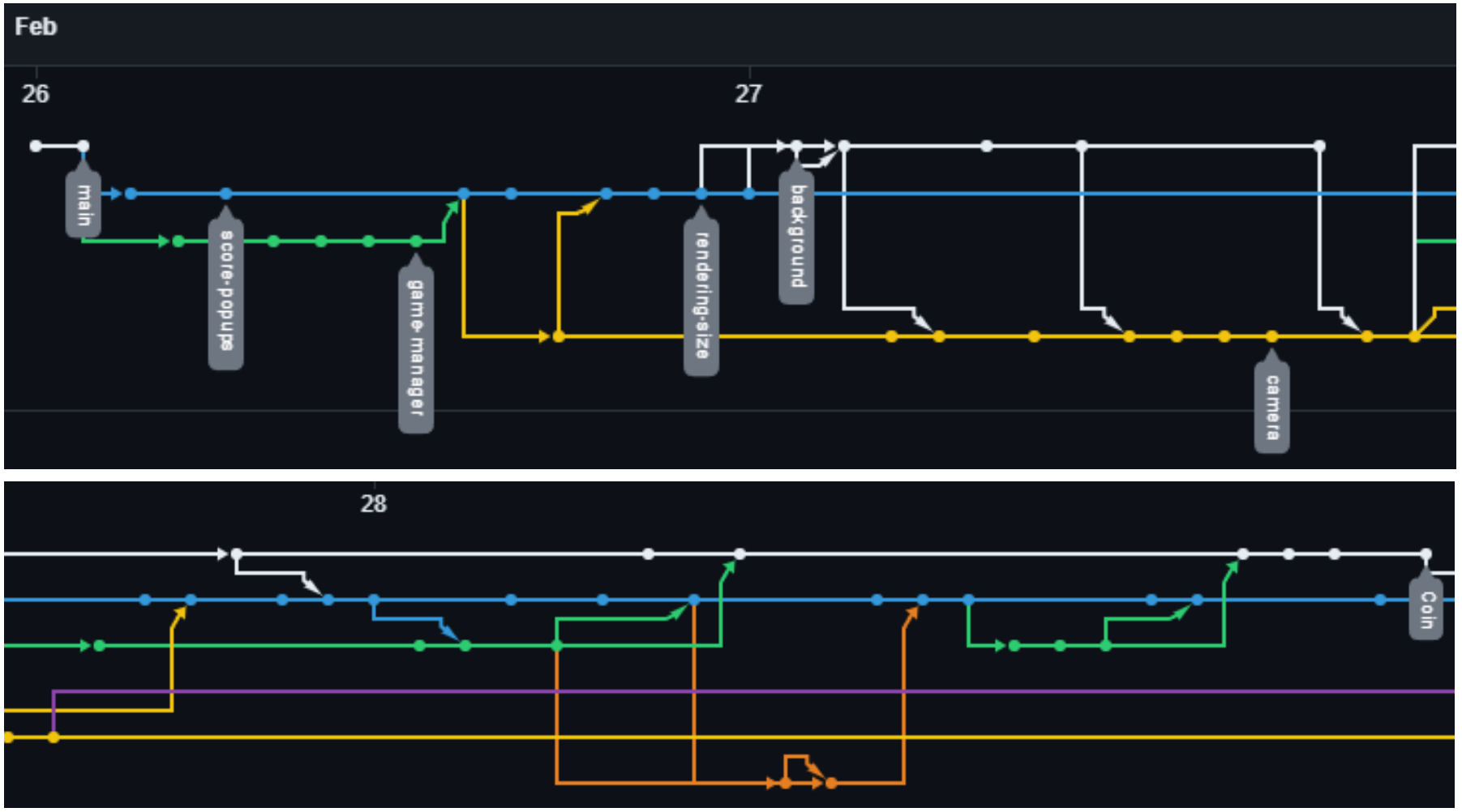


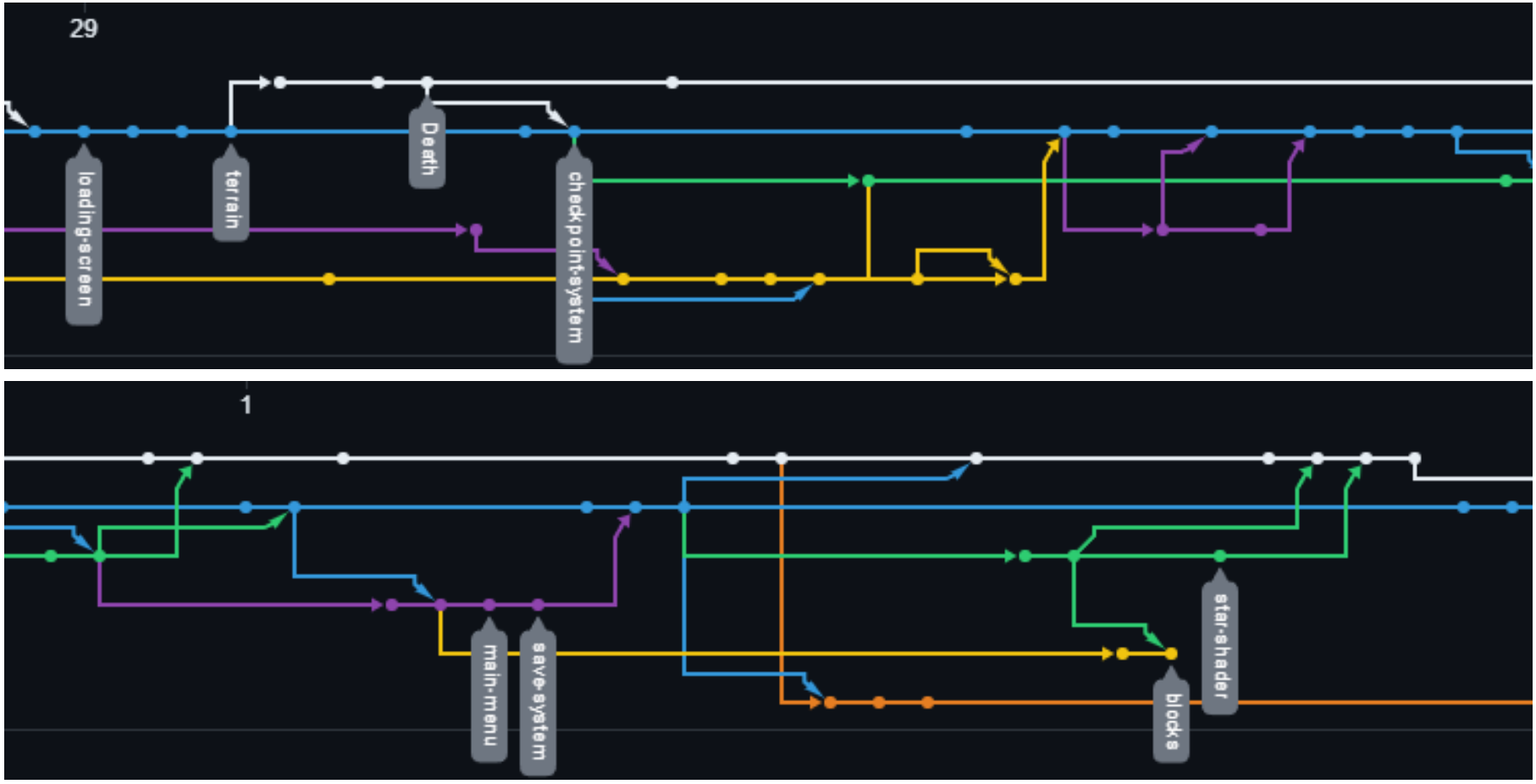
Post-meeting Trello board

Unfortunately, no screenshot was taken, but it mirrored the “What will you be doing in the next sprint?” column

Evidence of source control use

Github link: https://github.com/Eefschmeef2310/GDS1-Lab-2




29
loading-screen
terrain
Death
checkpoint-system
1
main-menu
save-system
blocks
star-shader

Reflection

What went right?

Our version control practices were sufficient, as we have completed a few projects before using Godot and have previously had issues with merge conflicts due to poor practices. For this project, we utilised a full main, dev and features branches setup, ensuring that minimal merge conflicts occurred.

The use of a Trello board was also beneficial. Team members ensured that once a feature was completed, the Trello board was updated and the rest of the group was kept informed. This practice made it easy to monitor the progress of development, and also made it so group members could take on additional tasks in the product backlog if needed.

The timeline of development progressed smoothly and to a high standard. By the end of development, all tasks were completed to a very high standard.

What went wrong?

There were a few roadblocks when completing this project. For both sprints, some outstanding tasks were not completed, often due to relying on another feature's completion. This led to some tasks needing to be brought into the next sprint, increasing that sprint's workload.

Also, refactoring code became very difficult in the later stages of development, as minimal documentation was provided when groupmates created their core functions. This meant time was wasted studying another group member's code, which could have been spent polishing the game more.

Finally, additional tasks were often added to the Trello board after the SCRUM had already begun. This made it so an increased workload was introduced during development, though fortunately it didn't happen too often, and work could still be completed on time.

What should we continue to do as a team?

For future projects, we will continue to make good use of version control, as well as keeping the Trello board up to date. We will also keep other groupmates informed whenever a feature is implemented, or another feature is not working as intended. Through these practices, we can ensure that development throughout the sprint is consistent and on schedule.

What do we need to change as a team?

To document our codebase better, we need to establish better protocols for tracking whos in charge of each script and each part of the project and also adding a lot more functions and

documentation for our modules so they can be easily interfaced with other peoples sections without the other person having to mess around with (and potentially break) someone else's code. Additionally, we will more thoroughly determine the features the game requires before development begins, in order to better plan our workload to be more consistent throughout the sprint, instead of loading all the work to the end of the sprint.