6 Conclusion

6.1 General commentary

6.1.1 Project commentary

We were especially glad to have spent a long time brainstorming basic game ideas before finally committing to Space Trouble, as it stands now. We have observed other groups who were even more eager than us to start coding, but who failed to put enough thought into their game's gameplay, and had to keep changing core mechanics. We are happy to say that most changes to Space Trouble were made due to either aesthetic concerns, or because they were designed to be quickly removable, exchangeable or adaptable as part of our quickly-prototyping, agile design process.

Unfortunately, we weren't able to profit very much from the physical prototype phase. This might be due to the fact that our game concept was set out to be casual, and mainly motivated by the quick pace of a multiplayer game, which is very hard to emulate in a board game. We do however see the benefit of the concept, for some very specific types of games, especially more strategically motivated ones.

Instead, one of the phases we would have been interested to spend more time on was playtesting. Due to the strict time constraints, we were only able to receive one iteration of feedback, whereas two iterations could facilitate more immediate improvements to our game. As it is, we must trust that we have successfully implemented all changes as indicated by our testers, instead of being able to verify it.

Besides this small issue, the general course schedule was feasible, mainly because our three team members were able to have meetings two times a week. For larger teams or more unfortunate student timetables, this could prove to be much more of a challenge.

Our initial design ideas translated very well into our finished game. That is, the ones we were absolutely adamant about. In many cases, such as the planet chunks, we were unsure about their success from the beginning, and as such, designed the issues in such a way that we would have several alternatives at hand very quickly. In the case of the planet chunks, this meant problems such as collision damage, initial velocities, size, collision shape, and others. This open design was, in our opinion, one of the reasons for our success.

We are glad to say that we could mainly keep to our development schedule as planned. The biggest obstacles were some performance issues, and meta-work such as completing this development notebook as well as the fortnightly presentations and demos. In many parts, we were in fact able to dynamically adjust our schedule to external timing issues such as holidays, other deadlines and sickness.

Overall, all three of us are proud of our creation - we were able to keep the basic gameplay we devised, but constantly improve on it based on internal as well as external feedback. At the end of the 14-week development cycle we can look back on both a teaching and a business experience which we will keep in mind for years to come.

6.1.2 Course commentary

The course theme of “attraction” was very helpful in many regards: During brainstorming, it helped manifest a broad spectrum of ideas. When implementing game functionality, it guided us in not losing focus of the core gameplay. In playtesting, we were able to twiddle with constants to emphasize or tone down the influence of gravity, and find a good balance between fun and challenging gameplay.

The impression of the course in regards to engineering was a clearly positive one. As engineering is only one aspect of game development, however, it is our opinion that the game lab could profit greatly from a inter-disciplinary, inter-institutional cooperation which uses outsourcing to bring the development process closer to the game industry. Splitting the process into

- Audio Design, Modelling, and Media Design (trailer, web page...) (all of which are more suitable to a non-engineering course such as at ZHdK or SAE Zurich)
- Game Design, in cooperation between all parties, and
- Engineering, as performed to date at ETH

could boost this course to a more global scale, putting it on par with the great schools of Game Design and Engineering such as Digipen.

With a strong industry partner such as Disney behind it, and with the collected experience from distributed courses such as Distributed and Outsourced Software Engineering (who by the way, also design simple games), this next step is not so much of a challenge, but more of a logical continuation of the game lab concept.

6.1.3 Technical challenges

Even though our team had some prior experience with working with XNA, a project of this scale brought on a few issues, especially in memory management. The XNA garbage collector served as adversary throughout our development process, forcing us to design code not for readability, reusability or flexibility, but performance.

In most other regards, though, XNA proved to be quite a boon, all but removing the pain of resource loading and other such undesirables.

As a complete aside from that, the 10MB upload limit of the provided TWiki was quite a hindrance, especially as we were asked to provide screenshots and photographs quite often. To provide images in any reasonable quality and resolution, this forced us to always carefully split our document into chapters, and in some cases, even remove files to fit into the upload limit. As this whole document has been collaboratively edited on Google Docs, it would have been easier to just privately link to that shared resource.

6.2 Final results

6.2.1 Final changes

Besides the fixes and changes already enumerated in the Playtesting chapter, the changes we made were mostly of an aesthetic nature:

Credits screen

As some of the used material (mainly audio) requires attribution by license, we introduced a credits screen accessible through the options menu.

Additional models and model improvements

To both improve performance and visual experience, we improved the model quality of some of our 3D models, and introduced models where they were still missing (such as for relic pieces).

Particle effects

Inspired by the “respawning” particle effects already discussed in the playtesting chapter, we added additional visual improvements based on particle effects, such as effects for the appearing and picking up of collectibles, an effect when a laser beam is charged, one indicating space dust in the background, and so on.

Final missing visual and audio effects as well as music

With all game features frozen, we scoured various free-to-use sound databases for suitable effects, and introduced visual effects where the game background still seemed lackluster.

The same could be said about the background music. As we were simultaneously looking for four tracks (menu music, credit screen music, gameplay music as well as a track for the teaser trailer), we unified our efforts and secured all of these tracks with help from personal friends (for the gameplay music) and jamendo.com, a Creative Commons music collection, for the other three tracks.

Two victory fanfares (one for a clear win, another for a tied score) as well as some sound effects (such as engine roaring) were made by the team.

Meteor storms

To cause further mayhem as gameplay progresses, we introduced meteor storms which happen in intelligently placed randomized intervals, and cause planet as well as player destruction. The generated meteors are also coupled with particle effects to make them visually more appealing.

Relic pieces in all game modes & Uber gun

As the relic pieces in collect-the-relic proved to be very fun in internal playtesting, we

introduced the same mechanic to deathmatch and last-man-standing as well, with minimal code changes. Collecting three relic pieces now gives players the “Uber gun” as secondary weapon, which is extremely lethal. Collected pieces are shown by letting them orbit around the player, with associated particle effects. When being shot, a player drops all his collected relic pieces. With this, we add the game concept of rewarding accomplishments to our game, which was lacking with only random weapon pick-ups.

Orbiting chunks

To make the initial planet setup more engaging, planets now have a randomized number of asteroid chunks orbiting them.

“Game Over” screen and progress bar

We improved the simple “game over, continue?” screen with elaborate statistics for deathmatch: We list the numbers of kills and deaths, as well as their ratio (the “kill-death-ratio”, a.k.a. KDR) of all active players and sort them according to their KDR. Depending on if there is one clear winner, or a tie (especially if in last-man-standing, two players die at the same time, both with one life remaining), one of two fanfares plays.

To further elucidate the current ranking of players in deathmatch, a progress bar at the bottom of the screen shows the proportion of kills of each player compared to the total number of kills.

General design unifications

To avoid the player confusion we encountered in playtesting, certain collectibles were renamed, and the pop-up bubbles on collection are now formulated much more clearly.

6.2.2 Final screenshots

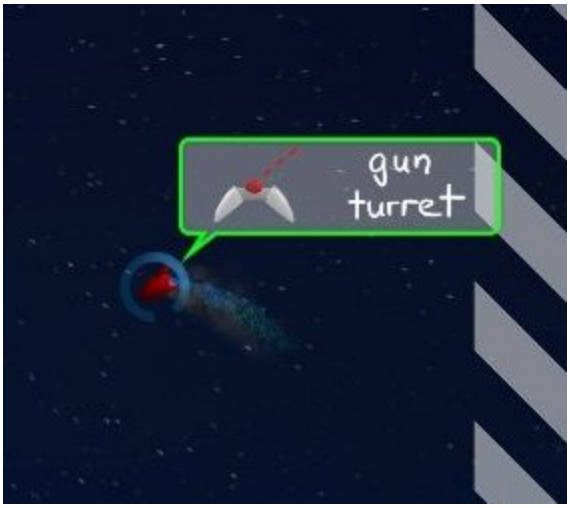

Fig 6.1: Pickup messages now use hand-drawn text and images

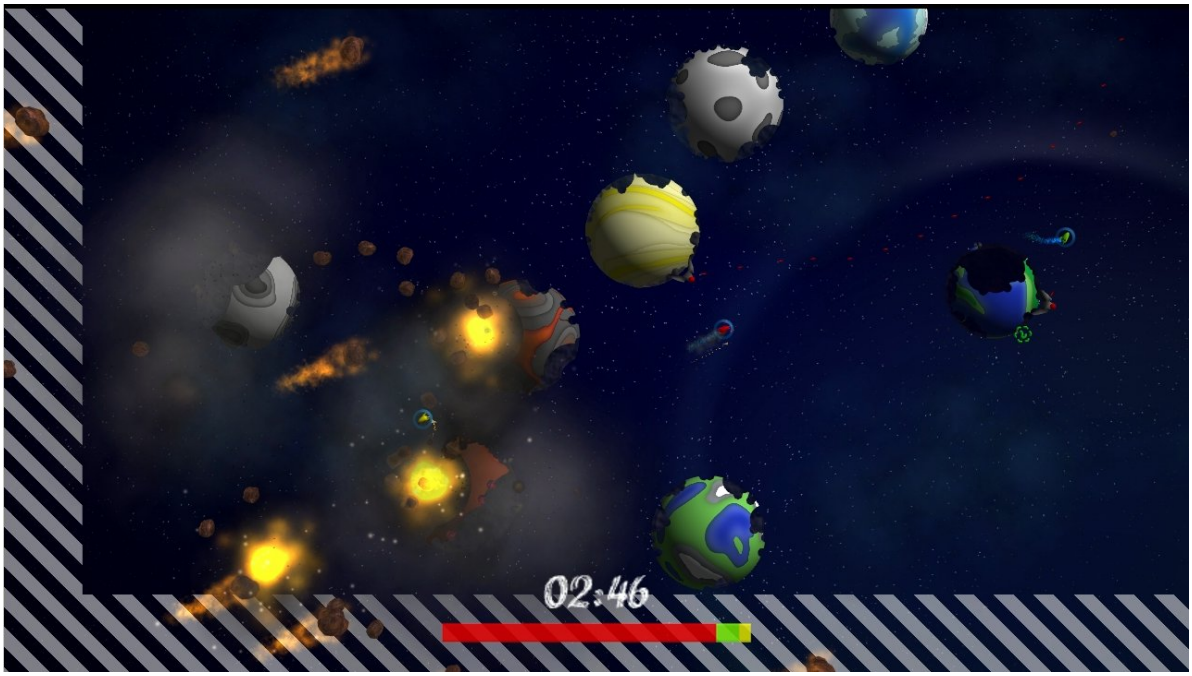

Fig 6.2: Meteor storm and progress bar. It becomes clear that red is the dominating player here. Green and yellow should start cooperating to oust him from that position...

Fig 6.3: Automatically scrolling credits sequence. Not pictured is the excellent music playing in the background.

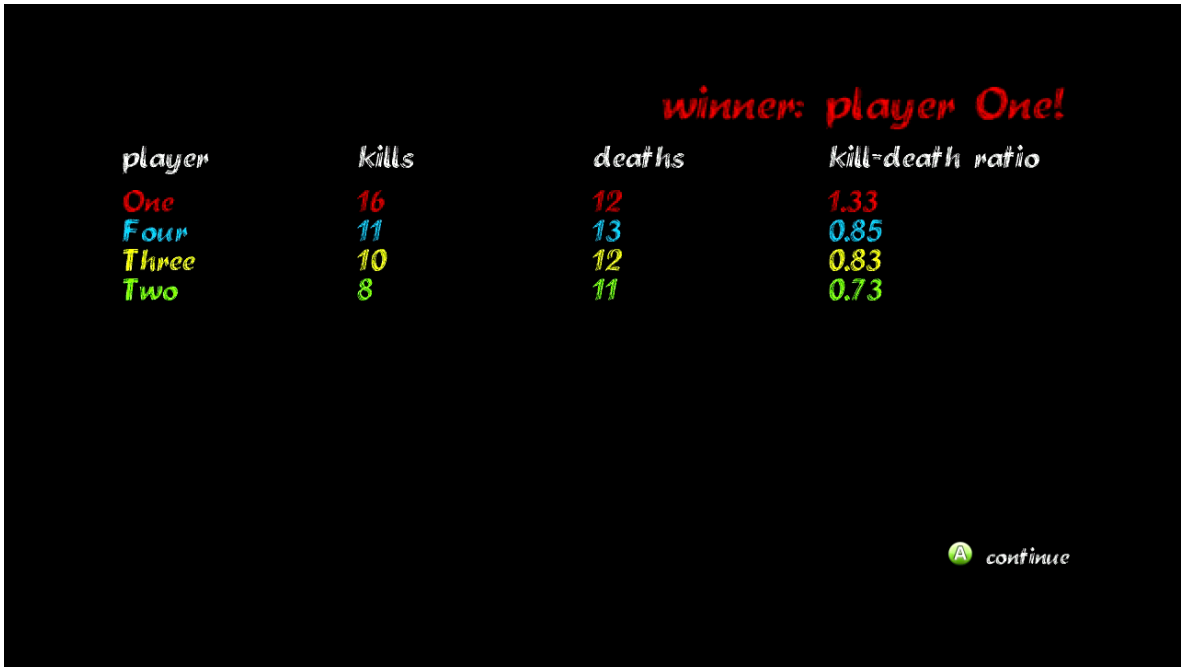

Fig 6.4: "Game Over" screen statistics after a round of deathmatch. This clearly shows that player one dominated, but the fight for second place was a very hard-fought one.

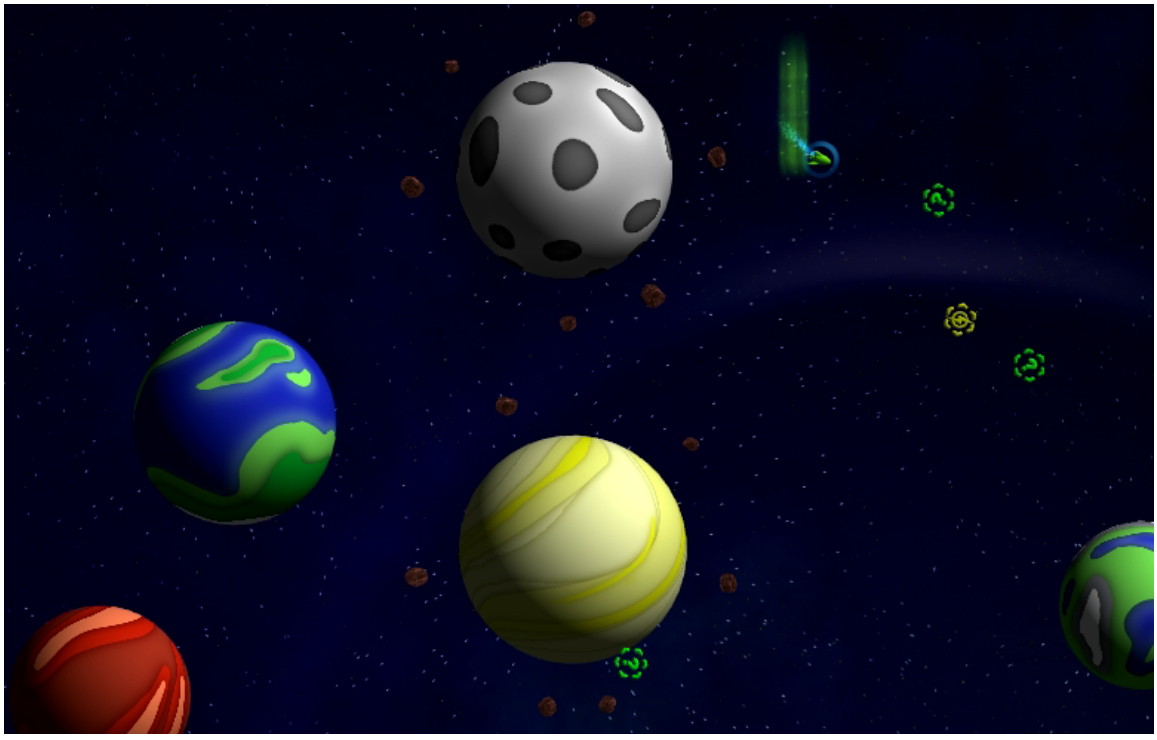

Fig 6.5: In early game, the orbiting chunks are still clearly visible around the yellow and grey planets.

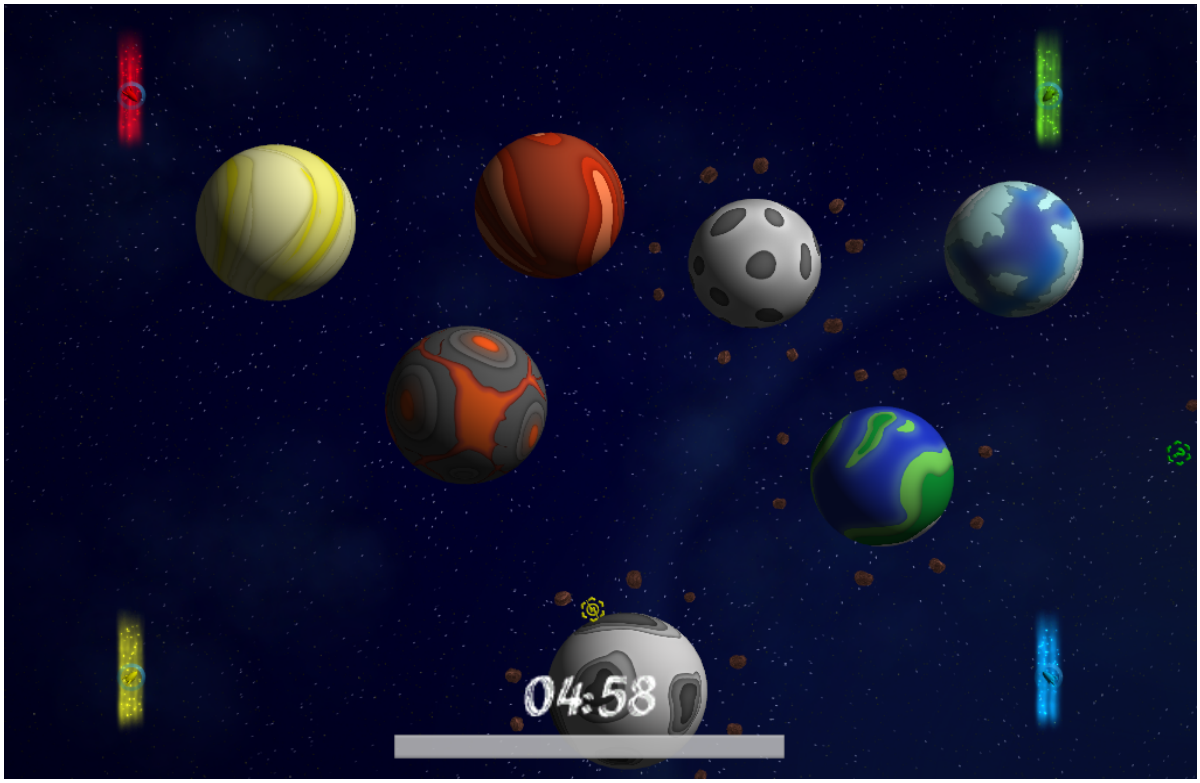

Fig 6.6: A beam particle effect makes it clear where each player starts the game.

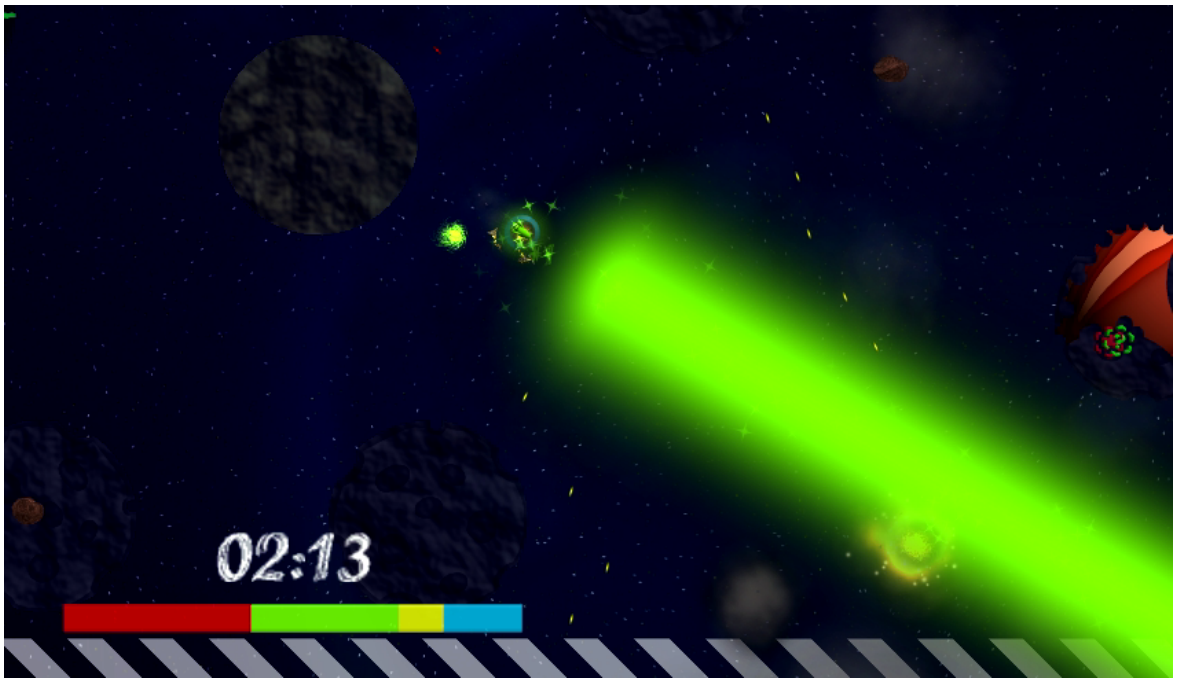

Fig 6.7: “Uber gun” is fired by the green player. The corresponding effect is affected by the player color, and quite destructive: the exploding ship of another player can be seen in the lower right.

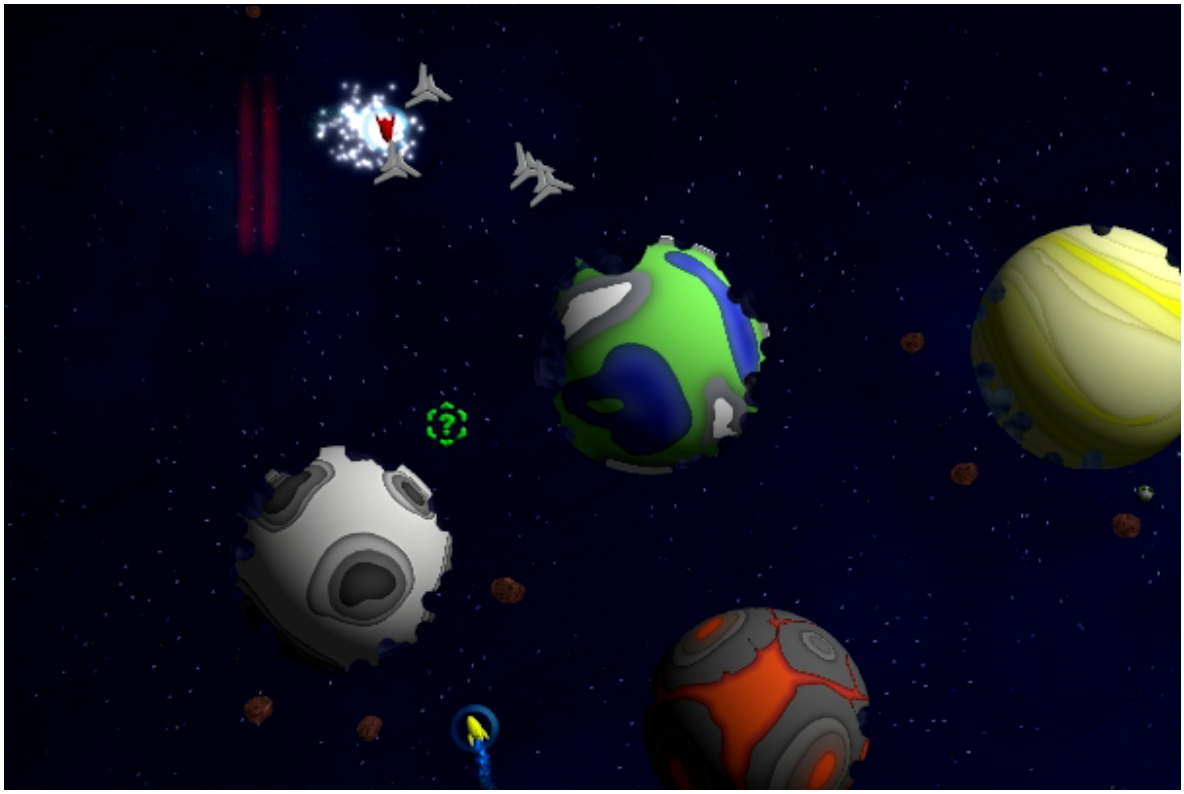

Fig 6.8: The red and yellow player head towards a bunch of relic pieces dropped by another player on his death. Also pictured is the particle effect that occurs when a player picks up a relic piece.

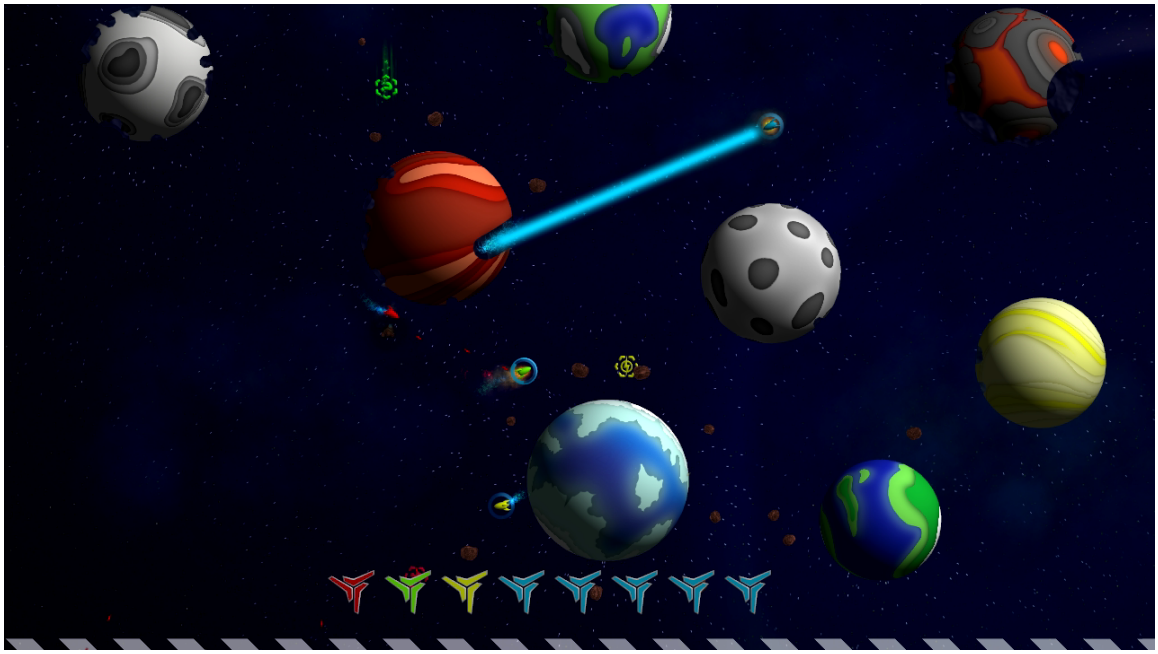

Fig 6.9: Full screen screenshot of the UI in collect-the-relic mode: the 8 pieces at the bottom are filled in their owner's respective color. In this case, blue is quite close to winning, but still has to kill all of the three remaining players to collect their pieces.