

# SoundBullet: Design & Implementation

When designing my video game, I first thought about games I had played in the past which were particularly enjoyable. The flash game ‘Planet Racer’ came to mind as a game which had an interesting career mode within a universe setting. I had recently played a music themed video game called ‘Beat Hazard’ which used the game’s soundtrack to vary the environment, creating exciting explosions and colours. I carefully considered the features of these games which made them both fun to play and challenging, then came up with the idea for SoundBullet.

My game combines a career mode setting in a universe similar to that of ‘Planet Racer’ with a music-themed vertical shooter mode where a player attempts to conquer various planets.

I am particularly proud of the visuals in my game. A lot of time went into designing the game’s appearance and making the game environment look aesthetically pleasing, and I found that I had underestimated how much development time this feature would take. A lot of time was spent creating the graphics for the planets in photoshop, and then applying randomised tint, scale, and rotate effects in processing. These effects are generated per-game, making the universe unique every time you play. The background of the universe screen consists of generated stars, which shimmer during gameplay and add further depth to the appearance.

The second major screen in the game is the planet-attack mode, which is launched whenever a player clicks on a planet. This environment has an infinite-loop scrolling background, which is supposed to be representative of the chosen planet’s surface. The background has the same colour as the planet image in universe mode, giving consistency to the visuals. In gamemode, the enemy missiles are coloured in red and the friendly ones in green, making them both easy to identify.

My original plan was to create the graphics for friendly and enemy ships myself, however I decided that my time would be better spent implementing other features in the game. As a result, the ship graphics were sourced from

<http://millionthvector.blogspot.co.uk/p/free-sprites.html> licensed under a [Creative Commons Attribution 4.0 International License](#).

One of the most interesting features of SoundBullet is the integration of the soundtrack with game dynamics. Enemy ships fire at a rate determined by the root-mean-square amplitude of the music currently playing. This not only provides a variation of difficulty throughout each level and planet, but it also helps to create a more immersive player experience.

My initial proposal was based on the plan that planets would represent different music albums in the main universe screen. Whilst attacking a planet, the game would play a small portion of each song before skipping to the next one. However, in testing it became apparent

that the track skipping was rather annoying to play with. Likewise, it took too much time for the user to add seven albums to the game's music directory using the correct file system structure before the game was even playable. In response to this, I changed the game such that each planet represents a single music track instead of a full album. This made it much simpler to use a custom soundtrack, as the user simply places seven song mp3s into the music directory, and each song appears as a planet in the universe.

Time was a big limiting factor during the implementation phase. I made an effort to implement most of the planned features laid out in the design document, or at least a cut down version of them. An example of this is the workshop feature at the home planet. The original design specified five different types of ship upgrades which could be purchased at the workshop. In practice, only ship health and dual gun turrets were able to be implemented within the time allocated. While more upgrades would further improve the game's difficulty and playability, I chose to implement the two features which were most important and useful in gameplay.

Given more time I would implement the three additional ship upgrades to give the game a more complete feel. I would also investigate the possibility of an online leaderboard feature, but I feel this would require more modification to the game, since players using different soundtracks have an uneven playing field for competition. SoundBullet as it is now is more of a personal gaming experience, as it allows players to interact with their music collection in an interesting environment.

An online version of the game could integrate some kind of music streaming service, perhaps, for example playing a game using your personal music collection or playlists on Spotify. I think this would be a really interesting feature to explore, but it was beyond the scope of this coursework piece.

Overall I am pleased with my final game. I think that the environment is nicely designed, particularly in the universe screen, and the attack-planet mode is fun to play. Similar opinions were expressed by my testing group, giving me confidence that I have created a quality video game.