Parallelized game "space defenders" documentation

Milica Puhalo

January 2026

Space defenders is a game that features a player at the bottom of the screen and asteroids falling from the top. The player shoots the asteroids so they don't "reach the ground". This was translated to three classes `Asteroid`, `Laser` and `Player` - all of them inheriting the abstract class `Entity` which contains all the shared variables (position on the screen and size of the game component) and methods of the three classes (draw and update position). Outside these three classes we need a container class that will bring all the components together - the `Game` class.
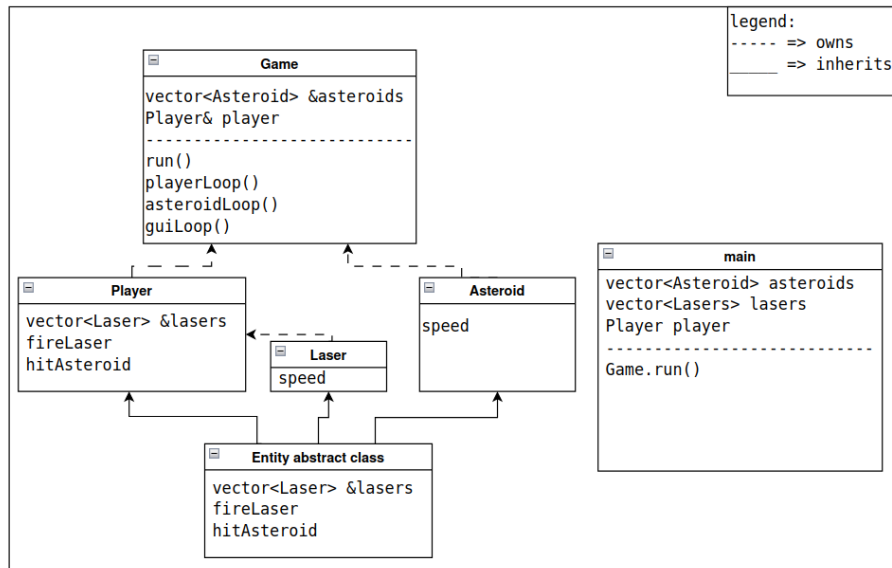

Figure 1: Component interconnection overview

All class instances are created within main, and passed by reference to their "owners". The player owns the lasers, while the game owns both the player and asteroids.

1

We call function `run()` to startup the three threads which implement the following functionalities:

- Player movement - `playerLoop()`

- Asteroid movement - `asteroidLoop()`

- Rendering the movements in the graphical interface - `guiLoop()`

The asteroid thread moreover manages asteroid movement by partitioning the workload so that each spawned thread is responsible for updating the movement of one active asteroid.
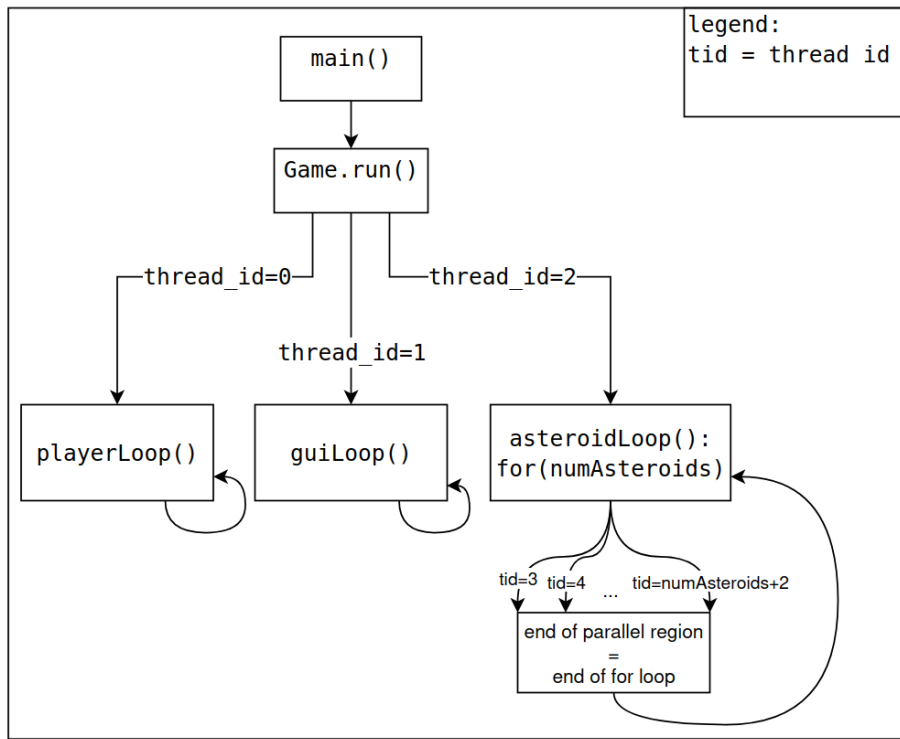

Figure 2: Thread creation view through function calls

These three main threads share certain variables. To make sure the variables are accessed correctly we need to safeguard them, making sure no variable is being read and written to or written to by two different threads at the same time. This is accomplished in the code by using locks.

Table 1: Shared resources across threads

Resource	playerLoop	asteroidLoop	guiLoop
player	x	x	x
laser array	x	x	x
asteroid array	/	x	x
player action	x	/	x
score	/	x	x
(won/lost)	/	x	x
running	x	x	x

From the number of shared resources shown in table 1 we can conclude that the Player class needs two locks (one for the player and another for the laser array) while the Game class needs five (for all the other resources).