Player ship

- Can be moved by player, up, down, left and right.
- Can't go outside the screen.
- Player can shoot bullets (2 at a time).
- Takes damage by colliding with enemy bullets or enemy ships.
- Doesn't take damage by colliding with own bullets.
- Is (visibly) immortal for a short time after taking damage.

Enemy Ship

- Moves diagonally down, "bouncing" on the vertical edges of the screen.
- Shoots bullets randomly, but never too soon after previous shot.
- Takes damage by colliding with player bullets or player ship.
- Doesn’t take damage by colliding with enemy bullets.
- Spawns at a random position, but always fully outside the screen at the top (not too far away).
- If it exits the bottom of the screen, it teleports back up, making it even more difficult.
- As time goes by, new enemies are spawned faster and faster, but never fast enough to become impossible (There are different approaches to this, pick one that you can argue makes the game fun).

Bullets

- Movement direction is based on the ship that fired it.
- Is destroyed when colliding with a differently allied ship.
- Both player and enemies shoot bullets of the same class.

Points

- Player gets score for each second that they stay alive.
- Score is displayed on screen.
- Is reset when the game is lost.

Explosions

Enemy ship explodes when damaged.

Colliding with explosions does not affect any ships or bullets.

Health

Players health is displayed on screen.

Player loses health when taking damage.

When health == 0, the game is lost. (You shouldn’t need to restart the game executable to play again.)

Graphics

All of the above game elements have a functional graphical representation.

Both textures and Fonts are used.

Optimization

The player should theoretically be able to play “forever” without the game breaking.
for example: make sure bullets are cleared from memory when they exit the screen.

Code

Should be structured appropriately and follow Object Oriented principles with reasonable coupling and cohesion.

Shouldn’t have unnecessary repetition of code, or unused code.

Should be clear and well-structured enough to understand for another programmer, in regards to both readability and usability.

It should be sufficiently easy to add new features or make changes to existing ones, as well as reuse different parts for other games.

Bonus Features (for grade VG)

Let the player enter a name.

A High-Score list that is persistent between sessions.

A main menu with (at least) the alternatives “New Game”, “High score” and “Quit”.

Use sound effect and/or music in the game.