realworld one

C++ Programming Test

realworld-one.com

TASK BACKGROUND

A programmer, who shall remain nameless, was asked to implement a Space Raiders game to a given rule-set. However, he was sloppy and not very thorough, and so the task has been passed on to you.

PROGRAMMING TASK

1. Go through the provided source code and make sure that all features are implemented and working properly according to the rule-set.
2. Refactor the code however you feel is necessary. Feel free to use your initiative!
3. Fix any bugs you come across, as well as any code that looks unsafe or wrong.
4. Implement the additional features listed under “Extensions”.
5. Optimize the overall performance of the game, along with any code that looks unnecessarily inefficient. (Hint: not only game logic performance is important)
6. Feel free to go ahead and implement additional features to impress us (but make sure to keep the core rule-set working as outlined.)

TEST INSTRUCTIONS

In a separate document, please keep a record of what you implemented, fixed and optimized and any key decisions that you made.

You should spend no more than 10-12 hours on this assignment. (The 10-12 hours do not need to be consecutive and could be spread over a few days – we would like to have the test results submitted within 5 business days) If you spend notably more or less than this time, then please include a record of that in your documentation.

Verify that your application runs properly before zipping up the project (along with your documentation) and sending it to us. Include all files that are necessary to build your project but try to keep the file size as small as possible by including only necessary files. If your file is larger than 5mb, please send us a download link.

Please treat this like a real programming assignment. Make sure that the work you submit to us is something that you would be happy to submit in a real programming environment.

SPACE RAIDERS: RULE SET

Outline:

The game should run inside the Windows console. To start with, the game will run in a random test mode, and end after 30 seconds.

On start, the application spawns a fixed number of alien ships and exactly one player ship.

The player ship and alien ships behave as described below. Every relevant game object has a unique visual representation.

If an alien ship gets destroyed the score count will go up by 10 for normal aliens and 20 for “better aliens”.

If the player dies, the game should freeze in its current state and display a game over message including the reached score. This screen should stay displayed until dismissed by the player.

Alien Ship:

Moves with a constant speed in a given direction (mainly horizontal, slightly vertical). If it reaches the edge of the screen, it will change its direction.

If an alien ship reaches the bottom of the screen, the player dies (game over).

An alien ship randomly shoots laser shots depending on its fire rate and can transform into a “Better Alien” ship” after reaching a certain amount of energy. The amount of energy increases randomly per frame. After reaching a certain energy level the alien ship can transform.

Collisions between alien ships and the player will result in both objects being destroyed (game over).

“Better Alien” Ship:

This is an alien ship with a higher fire rate and faster movement speed. It also takes two laser shots to destroy is, compared to one shot for a normal alien.

Player Ship:

The player ship is like an alien ship, except that it moves and fires lasers based on random control inputs.

Laser Shot:

Laser shots are spawned by the player ship or the alien ship. They are spawned in front of the spaceships.

The laser shots from the player move vertically upwards, while the laser shots from the aliens move in the opposite direction.

On collision, they will destroy any spaceship or other laser shot with which they collide. This will result in the destruction of both. An explosion will be spawned at the hit point.

Explosion:

Collisions between lasers and/or spaceships are visualized by an explosion. Explosions will automatically remove themselves after a short amount of time.

SPACE RAIDERS: EXTENSION

1. Add a second mode that allows the player ship to be controlled via input keys. This mode should end only when the player ship is destroyed.
2. Spawn alien ships in continuous waves. (The number of aliens can increase per wave.)
3. Blocking Walls (procedurally spawned blocking walls per wave, can be destroyed by multiple laser shot hits).
4. Destroyed aliens have a 10% chance of dropping a Power-Up object. Power-ups move towards the bottom of the screen. If the power-up reaches the bottom of the screen then it will be removed. If the player collects a power-up (by-colliding with it), they will gain one of the following enhancements for a certain amount of time:

Speed boost - Player has faster movement speed.
Fire-rate boost - Player shoots laser shots faster.
Triple fire - Player shoots 3 laser shots at once, moving in different directions: straight, 45° left and 45° right

Thanks for your general interest in realworld one and for your engagement in spending your valuable time on the process with us. This is highly appreciated.

Good luck and all the best!

Your realworld one team

IMPORTANT NOTE:

Please do not share the C++ Programming Test as it's covered by our NDA. The results cannot be published on any website. The Assessment should only be created for the evaluation process with regards to any position at realworld one. Any other information regarding this art assessment (additional material and this document itself) must not be shared with any other person under any circumstances.

We really appreciate your understanding regarding this matter. If you have any questions, please do not hesitate to contact me. Thanks!