pguillot

(https://profile.intra.42.fr)

(https://profile.intra.42.fr/searches)

Scale for project Piscine CPP (/projects/piscine-cpp) / Rush00 (/projects/piscine-cpp-r

You should correct 2 students in this team

★

Git repository

```
vogsphere@vogsphere.42.fr:intra/2015/activities/piscine_cpp_rush00/pile
```

Introduction

Today, the subject is especially vague, and this grading scale is also somewhat vague. This is because there are many, man ways to do this project correctly. So in essence, you will validate that the basics of the subject have been respected, and the points for features you think are cool / well done / useful.

The usual obvious rules apply : Only grade what's on the git repository of the student, don't be a dick, and basically be the would like to have grading you.

Do NOT stop grading when a question is wrong.

Guidelines

The Makefile must compile with clang++, with -Wall -Wextra -Werror, or you must flag the project as "Invalid compilation"

Any of these means you must flag the projet as "Incomplete work"
- A function is implemented in a header (except in a template)
- A class is not in Coplien's form

Any of these means that you must flag the project as Cheat:
- Use of a "C" function (*alloc, *printf, free)
- Use of a function not allowed in the subject
- Use of "using namespace" or "friend" (Unless explictly allowed in the subject)
- Use of C++11 features (Unless explictly allowed in the subject)

Also remember that this is a C++ project. Therefore, please take some time to look at the actual code of the project, and if it is just "C+", meaning that it's essentially a C program with a few classes "just for show" and std::cout instead of printf (you know what I mean), just mark the project as "Incomplete work" and be done with it. As the subject says, to be relevant in this project, you have to actually make an effort to use the object-oriented properties of the language.

Ratings

Define the type of error (if there is an error), which ended the correction.

✓ Ok Empty work Incomplete work No author file Invalid compilation Norme Cheat

Attachments

Attachments

Subject (https://cdn.intra.42.fr/pdf/pdf/130/rush00.en.pdf)

Sections

Basic requirements

These are requirements to be graded here, if they are not met, mark this project as "Incomplete work" and stop grading: - Th the terminal using the ncurses library or something close to it. - The program actually is a sort of shoot-em-up game : That m player displayed on screen, you can use some kind of input to control it, and there is some kind of notion of time passing in th now, the quality of these mechanisms is not relevant.

Basic game

The player can move.
There are enemies that spawn either randomly or in a preset pattern
Collisions are handled (If the player collides with an enemy, he dies, or at least takes damage)

✓ Yes ✕ No

Ever-so-slightly less basic game

The player can shoot.
There is some kind of scrolling system (that is, the area represented by the screen "moves" in the world, very much like in R-T

✓ Yes ✕ No

Additional requirements

If the "Basic requirements" section is not completely validated, please do not grade this part.

Info

There is information displayed on screen about lives and/or a score and/or other interesting data.

✓ Yes ✕ No

Timing

The notion of time in the game is linked to some kind of clock and not just to how many frames have been rendered. Whateve facility or library is used for this is mostly OK.

✓ Yes ✕ No

Entities

Entities can occupy multiple squares and not be a single character on screen.

✓ Yes ✕ No

Shooting

Enemies can also shoot

✓ Yes ✕ No

Scenery

There are pieces of scenery on the screen (Asteroids, cavern walls, whatever, use your judgement)

✓ Yes ✕ No

Bonuses

Grade these only if the "Additional requirements" part is fully (or almost fully) validated.

Lots of bonuses !

Rate from 0 to 5 depending on how many cool / useful / interesting bonuses are present and work well. Here are some examples of these bonuses (from the subject), but others can also be valid, use your own judgement:
- Multiplayer games
- Scripted game worlds, with pre-determined batches of enemies, things like that
- Enemies have a scripted behaviour and/or scripted shooting patterns
- Large and hard-to-beat boss enemies

Rate it from 0 (failed) through 5 (excellent)

○

Conclusion

Leave a comment on this correction

*** (required) Comment**

Finish correction