

Menu

My projects

Holy Graph

List projects

Available Cursus

Your projects

Day 03

Day 04

Day 05

Day 06

Rush 00

CPP Module 00

Piscine Unity

SCALE FOR PROJECT

PISCINE UNITY / DAY 04

You should evaluate 1 student in this team

Git repository

git@vogosphere.kzn.21-school.ru:vogosphere/intra-uuid-18b65b41-7b9f-4b

Introduction

For the smooth running of this evaluation, please respect the following rules:

- Remain polite, kind, respectful and constructive whatever happens during this conversation. It's a matter of confidence between you and the 42 community.
- Highlight the potential problems you 've had with the work you're presented to the person or the group you're grading, and take the time to talk about and discuss those issues.
- Accept the fact that the exam subject or required functions might lead to different interpretations. Listen to your discussion partner's perspective with an open mind (are they right or wrong ?) and grade them as fairly as possible. 42's teaching methods can make sense only if peer-evaluation is taken seriously.

Guidelines

- You must only evaluate what you will find in the student's or group's GIT repository.
- You must stop grading when one exercise is not correct, even if the other ones are. However, you should keep discussing the other exercises with the evaluated student.

Attachments

subject.pdf

d04.tar.gz

Foreword

This section is dedicated to the evaluation start and the checking of prerequisites. It's not graded, but if something's wrong or a condition is not met, here or anytime during the evaluation, the grade is 0 and a flag can be ticked if necessary.

Observing the general instructions

- The repo contains the evaluated student's or group's work.
- The evaluated student or group can explain their work anytime during the evaluation.
- General and specific instructions are observed all through the evaluation.

Yes

No

Notes

General notes for the exercises

Remarques

- For each exercise, open the project in Unity and use the execute button in the editor (the little "play" button at the top of the window).
- Test each exercise thoroughly to make sure the turned-in work matches the subject's requirements.
- Make sure only the authorized functions were used to complete the exercise.

Yes

No

Ex00, Data Select!

Title and selection screens creation.

User profile

Playerprefs do store the levels the player has unlocked, the lives they lost, the number of rings they earned during all their plays and the best score for each level.

Yes

No

The data select

The DataSelect features a GUI and helps display all the infos stored in the playerprefs. It also allows the player to choose a level to launch.

Yes

No

Start and Reset

There's a scene with a title screen and a button to reset the user profile. You can pass the DataSelect scene pressing Enter.

Yes

No

Ex01, a basic level

The first level without any trap or enemy, but with a lot of rings and alternative paths.

A sketchy level design

Sonic appears at the beginning of the level and you can move him to the goal plate using different paths.

Yes

No

Time

A GUI displays the time passed since the beginning of the run following the required format (update every second and displaying only minutes and seconds).

Yes

No

Rings

The level is filled with rings you can collect. Each time Sonic collects a ring, it makes a specific sound and the ring counter increases matchingly.

Yes

No

Music

The music matches the current level. If it's not the original one (or a remix), it must at least come from a Sonic game.

Yes

No

Spinning goal plate and score

When Sonic runs by the goal plates, it starts spinning and the level and music is triggered. The score appears 6 seconds after the music starts.

Yes

No

Ex02, the end of the joyride

Traps and tricks are added. You must have a second playable level to grade this exercise.

Holes

There is one or several holes in the level. If Sonic falls into it, he loses a life. The matching sound is played and the life count is updated in the playerprefs.

Yes

No

Spikes - dead

You must make sure the expected behavior is triggered when Sonic touches the spikes. Check the code to make sure all of the subject's requirements are observed and test it to make sure nothing bugs. If Sonic doesn't carry any ring, the dead method is called.

Yes

No

Spikes - bump

If Sonic has carries rings, he doesn't die but is bumped back with velocity set to zero and then applying a ForceMode2D.impulse.

Yes

No

Spikes-bools

The Sonic.cs script's animator's bools are defined as required in the subject. isHit passes to true and the stopHit method is invoke 2 seconds later.

Yes

No

Spikes - Invincible coroutine

A coroutine making Sonic invincible for 5 seconds is launched.

Yes

No

Spikes - rings

Sonic loses all his rings. Half of them are scattered around him as required in the subject.

Yes

No

Ex03, fast as lightning

Add bumpers and tv's. You must have a third playable level to grade this exercise.

Bumper physics

There are several bumpers in the level. They're triggered when Sonic hits the center part of the sprite.

Yes

No

Animation and bumper sound

When a bumper is triggered, its sprite is animated as required in the subject. You can hear a matching sound.

Yes

No

TV physics

TV's feature a collider and can be destroyed when Sonic hits them rolling. If so, the Sonic.cs Destroy() method is called.

Yes

No

TV animation

TV's are animated as required in the subject. When they're broken, the matching sprite is displayed and the collider is off.

Yes

No

Rings tv

The TV gives 10 rings to Sonic.

Yes

No

Power sneakers TV

When the TV is destroyed, Sonic's max speed and the music pitch are increased. They both return to normal after 15 seconds.

Yes

No

Shield tv

The destroyed tv creates a shield around Sonic. It behaves as required by the subject.

Yes

No

Ex04, enemies!

Enemies are added in the game. You must have a fourth playable level to grade this exercise.

Shooter

A still enemy that shots projectiles (straight, airborne, homing, whatever).

Yes

No

Spiker

A slow moving enemy that stops and turns into a spike ball.

Yes

No

Runner

An fast moving enemy that has no other specific skill.

Yes

No

Gonna kill 'em all

To destroy an enemy, Sonic must be in a ball and roll or jump. The Destroy() method is called. If he touches the enemy in another state, the getHit() method is called.

Yes

No

Bonus exercise, Dr Robotnik

The final boss. He appears at the end of the 4th level or in a 5th one.

Final fight

The final fight must be cool, or whichever word you'll use. Anyway, the boss must have different attack patterns, several life points and he must blink when he's hit. A matching sound must play when he's hit. Don't panic. The grading is saved. This exercise doesn't count in today's scale. It's just a fun little pastime.

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

Ratings

Don't forget to check the flag corresponding to the defense

Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation