

Removed if-statements

Calc.getRandomIntBetween()

Removed a double If-statement that checked boundry values into one if-statement.

See: `git diff a0f05d2 2672a38`

VerticalPlatform.update() & HorizontalPlatform.update()

Beforehand, there even was an additional method `updateEnums()` with one if-statement which has moved into the `update` method and has been modified to remove all pre-existing if-statements in the `update()` method. This is because the platform is now self-aware of its speed, rendering the whole system present before useless and making the if-statements it used absolute.

See: `git diff 2672a38 4128f2c`

FileSystem.deleteFile()

There was a if in the `deleteFile()` method that checked if deleting the file was successfull and only logged something in that case. Now the filesystem will always log something, indicating whether or not deleting the file was succesfull.

See: `git diff b125ab2f 20c0949`

DoodleCamera.update()

The `update` method in the `DoodleCamera` previously had an if-statement checking if the camera position must be updated, now we use a little bit more complex maths to determine the new position without an if-statement.

See: `git diff d7c6f97 2ea775e`

AudoManager.Sound.play()

There was an if-statement which checked if there was actually a clip runnen and stopped that clip. This check is not necessary because the clip can always be stopped, whether it is playing or not.

See: `git diff 1e80cfc b14b80a`

Remaining if-statements

`Button.mouseClicked()`

This method contains an if-statement checking if the click was within certain boundaries, removing this would make the code unnecessarily complex.

`ButtonFactory.getButtonFactory()`

Contains one if-statement for a singleton implementation which we want to maintain.

`FileSystem.getResourceFile()`

Contains three if-statements, two of which ensure there are no null-values which we want to maintain as such because the method is public. And changing the other one, which prepends a character conditionally, would make the code unnecessarily complex.

`Logger.appendToTextFile()`

Contains one if-statement which ensures that the logger only logs when it should log (pending tasks).

`LoggerFactory.LoggerFactory()`

Contains one if-statement which makes sure the log file is being cleared only when a certain value is true. Removing this would make the code unnecessarily complex.

`LoggerFactory.createLogger()`

Contains one if-block that makes sure that classes that shouldn't log, don't log by returning a fake logger (an implementation of `ILogger` without any actual code). Removing this would only introduce more if-statements in each class that uses a `Logger`.

`Calc.getRandomIntBetween()`

Contains one if-statement which ensures the input values are correct. Removing this would make using the method unsafe. And replacing with an `assert` is not recommended because the method is public.

`Calc.getRandomDouble()`

Contains one if-statement which ensures the input value is correct. Removing this would make using the method unsafe. And replacing with an `assert` is not recommended because the method is public.

Renderer.drawSprite(...), Renderer.drawSpriteHUD(...)

All contain one if-statement which ensures the input sprite is not null. Removing this would make using the method unsafe. And replacing with an assert is not recommended because the method is public.

Renderer.setGraphicsBuffer(...)

Cotains one if-statement which ensures the given graphics buffer is not null. Removing this would make using the method unsafe. And replacing with an assert is not recommended because the method is public.