



TECH TALK

we start
avoiding ifs ?

2023

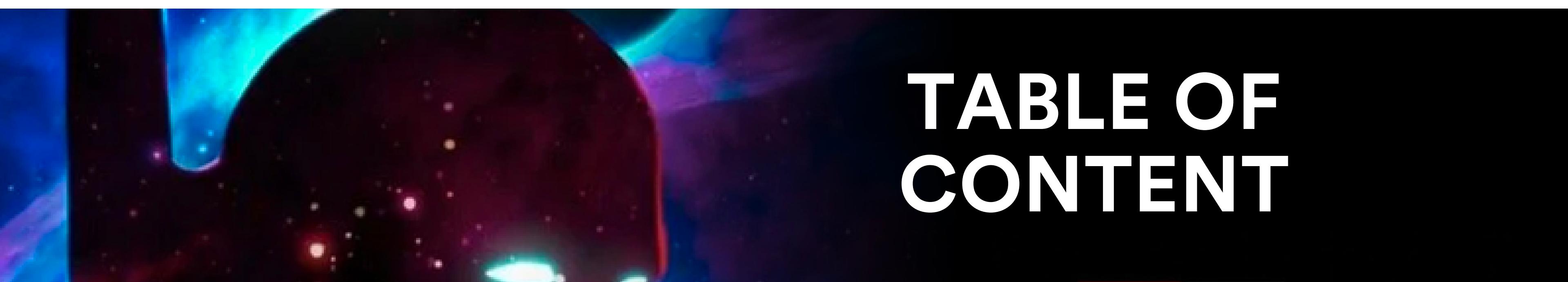


TABLE OF CONTENT

01 | INTRO

02 | SIMPLE
PROBLEM

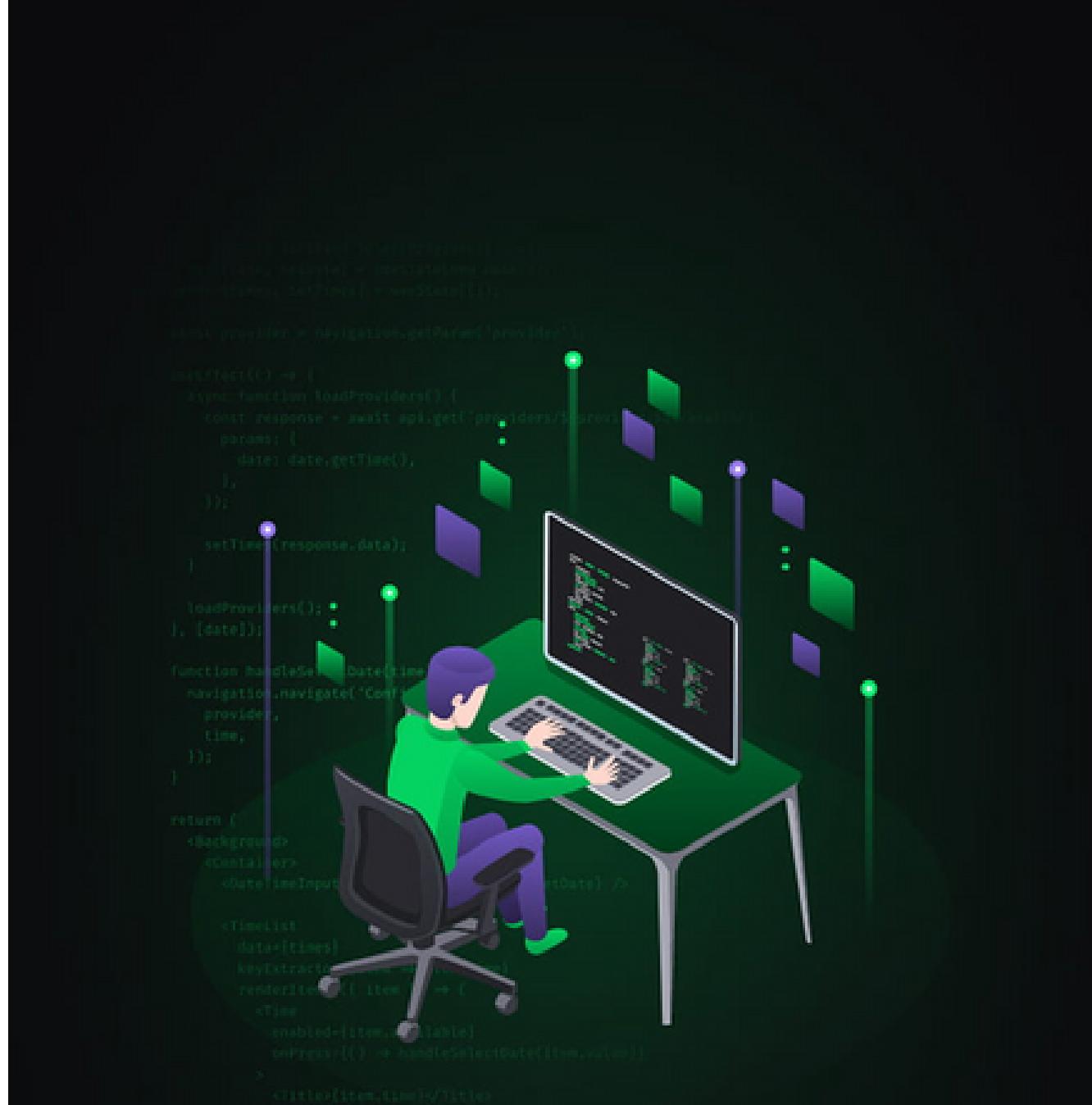
03 | COMPLEX
PROBLEM

04 | PROS & CONS

05 | HACKS

06 | Q&A

DAVID G KOTLIREVSKY



WHO AM I?

[HTTPS://DAVID-KOTLIREVSKY-RESUME.WEB.APP/](https://DAVID-KOTLIREVSKY-RESUME.WEB.APP/)
[HTTPS://AR.LINKEDIN.COM/IN/DKOTLIREVSKY](https://AR.LINKEDIN.COM/IN/DKOTLIREVSKY)

WHAT WILL WE DO ?



SOLVE ISSUES!



- We will solve two problems
- Solve it in a simple and quick way first
- Describe some issues
- Apply TDD (with some licenses)

https://github.com/davidgk/avoid_ifs_kopius_talk

1º ROUND

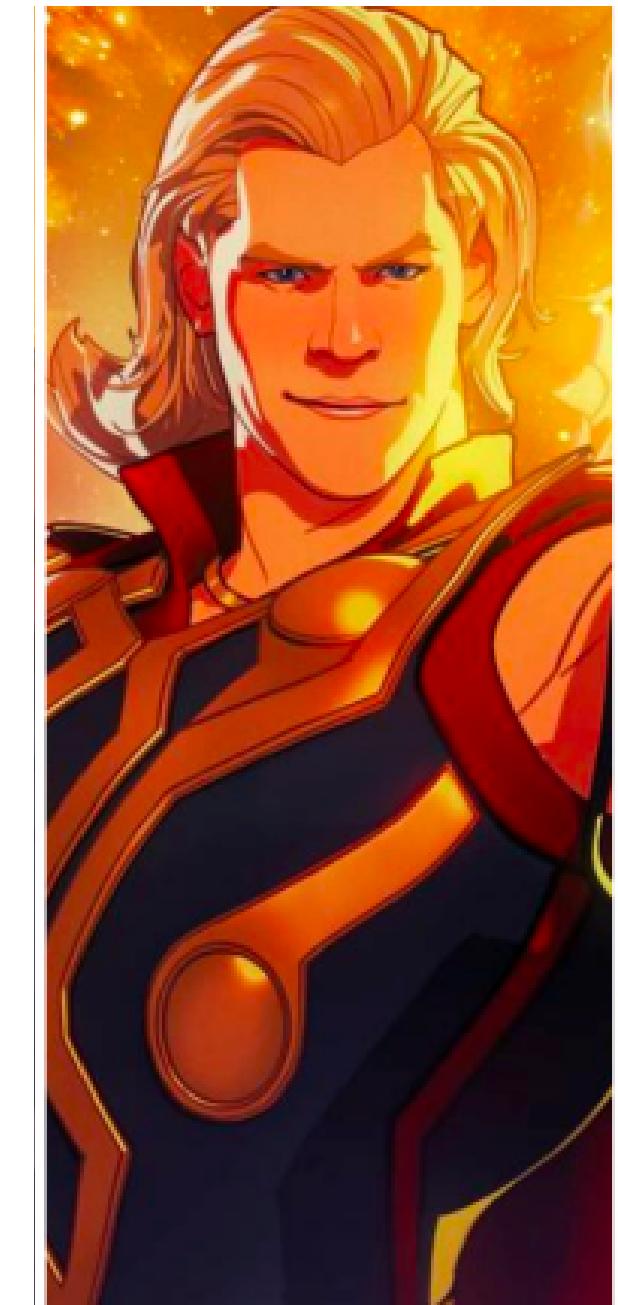


DRIVER CHECK GEARS

Let's try to solve first a simple problem. It's about modeling some interactions based on a Driver who wants to change gears to a Car , so it will tell us the maximum velocity that it can go.

We'll use TDD; implementing an initial solution, but it might not be the best one

Based tests we'll implement another approach



https://github.com/davidgk/avoid_ifs_kopius_talk/tree/develop/developer-gears

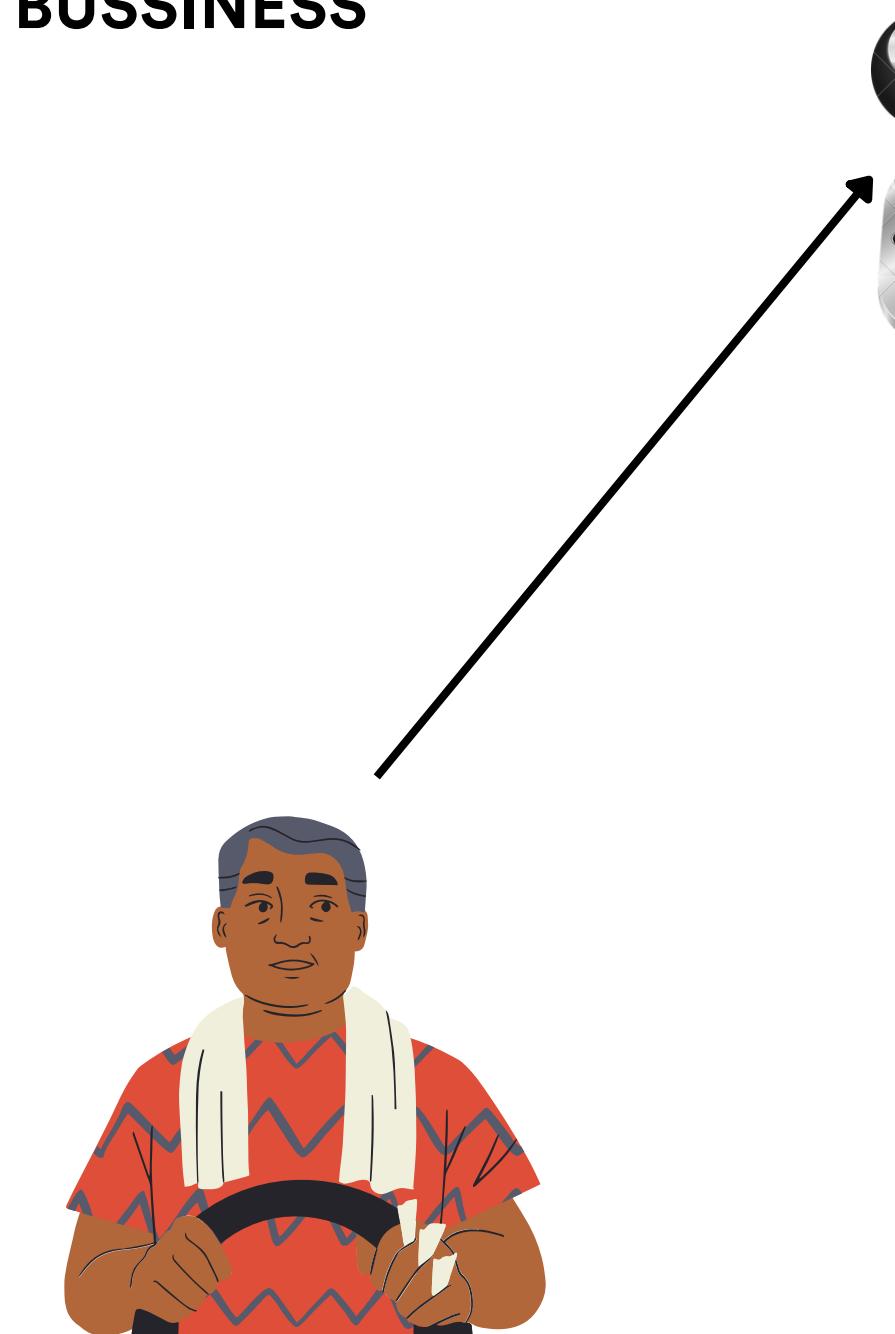
1° ROUND



BUSSINESS

DRIVER CHECK GEARS

EXPECTED



https://github.com/davidgk/avoid_ifs_kopius_talk/tree/develop/driver-gears

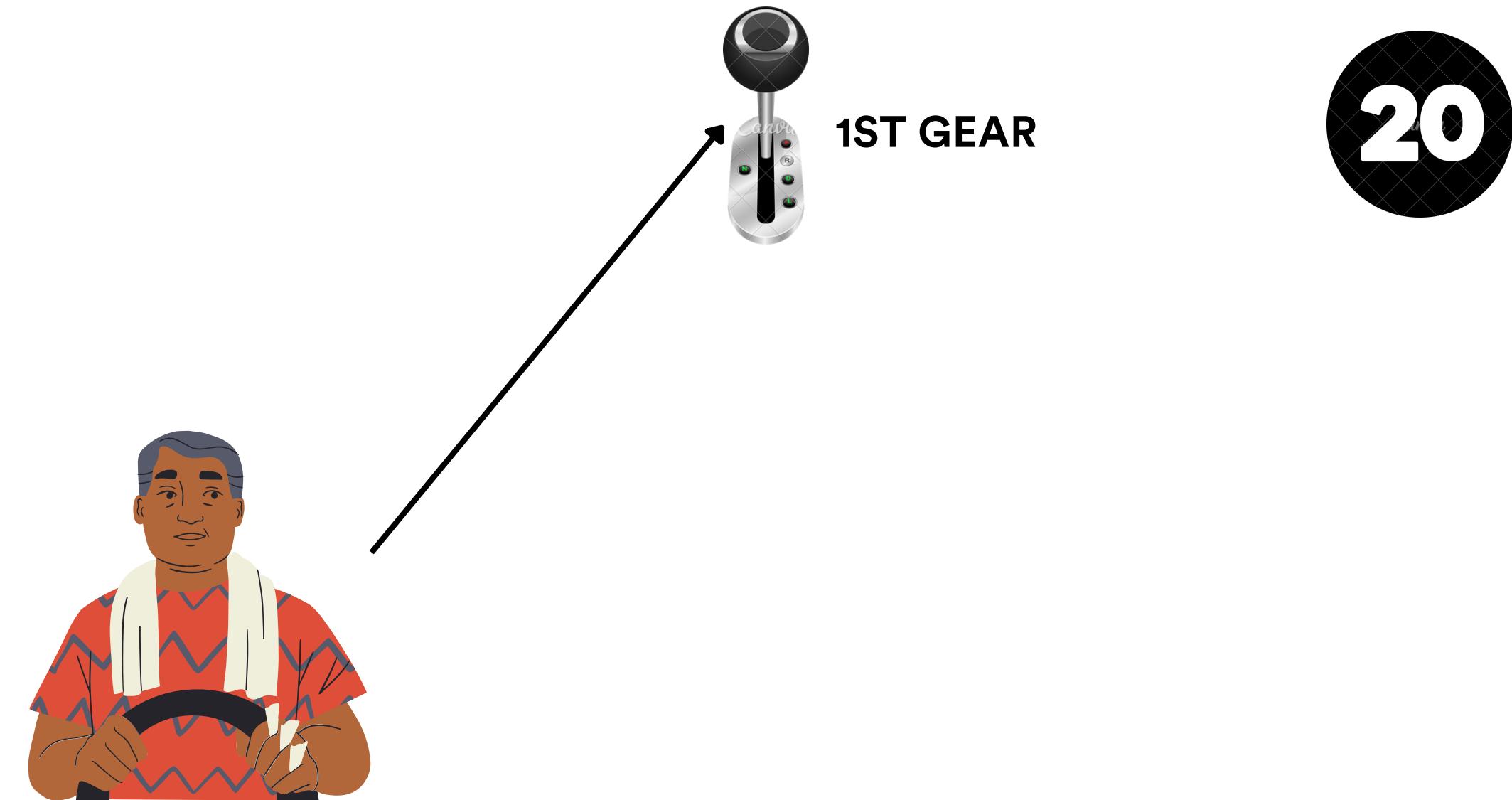
1° ROUND



BUSSINESS

DRIVER CHECK GEARS

EXPECTED



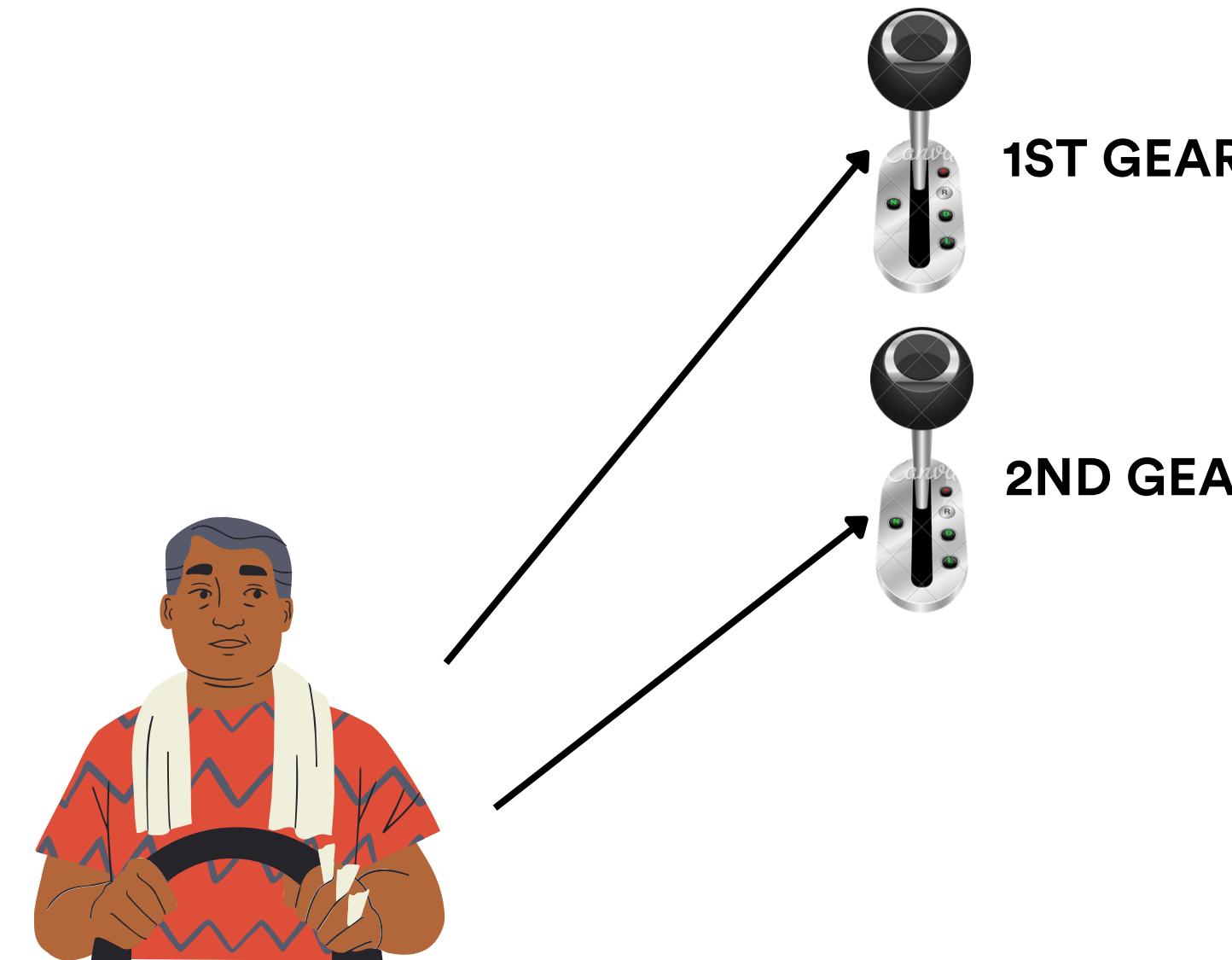
https://github.com/davidgk/avoid_ifs_kopius_talk/tree/develop/driver-gears

1° ROUND



BUSSINESS

DRIVER CHECK GEARS



EXPECTED

20

50

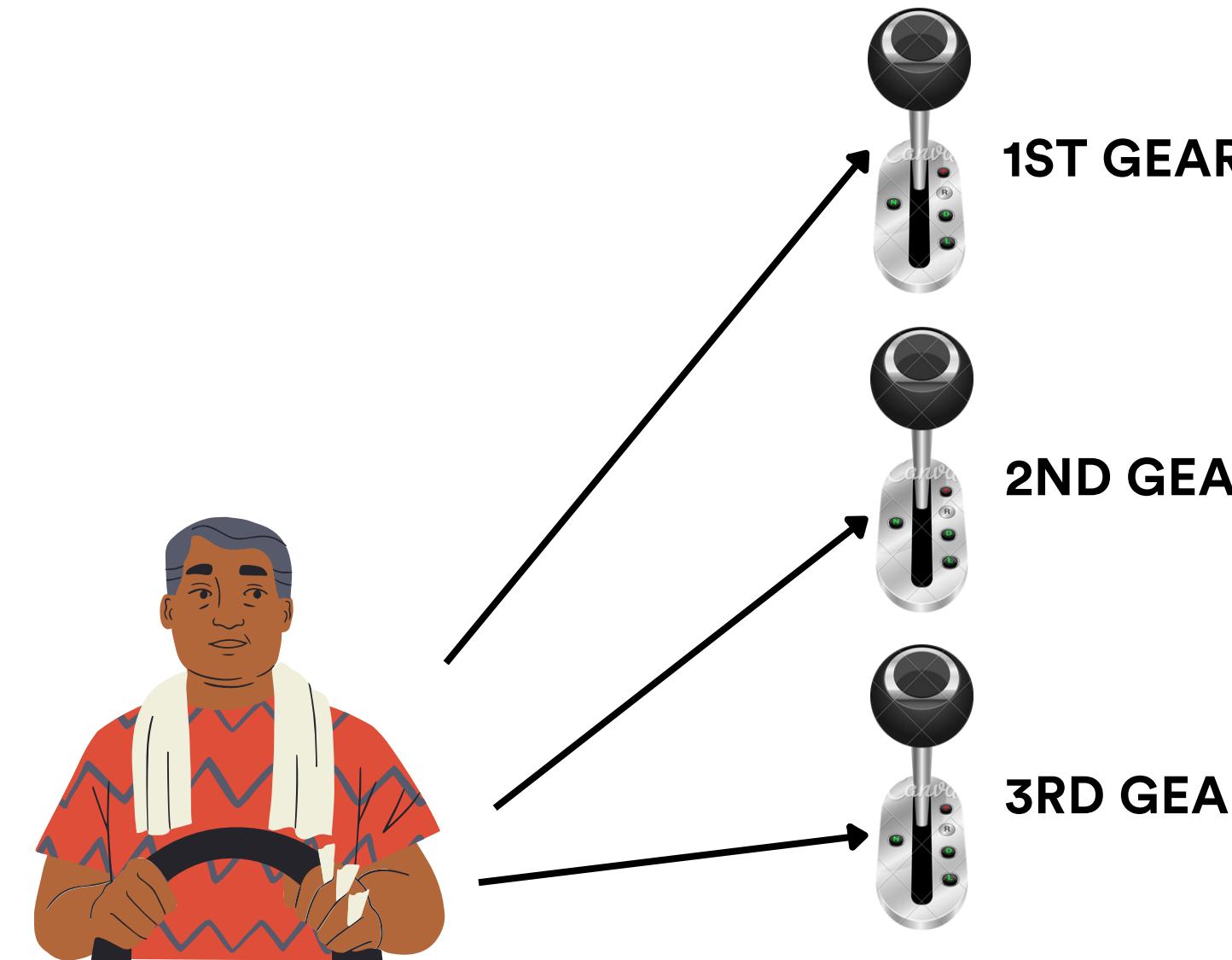
https://github.com/davidgk/avoid_ifs_kopius_talk/tree/develop/driver-gears

1° ROUND



BUSSINESS

DRIVER CHECK GEARS



EXPECTED

- 20
- 50
- 80

https://github.com/davidgk/avoid_ifs_kopius_talk/tree/develop/driver-gears

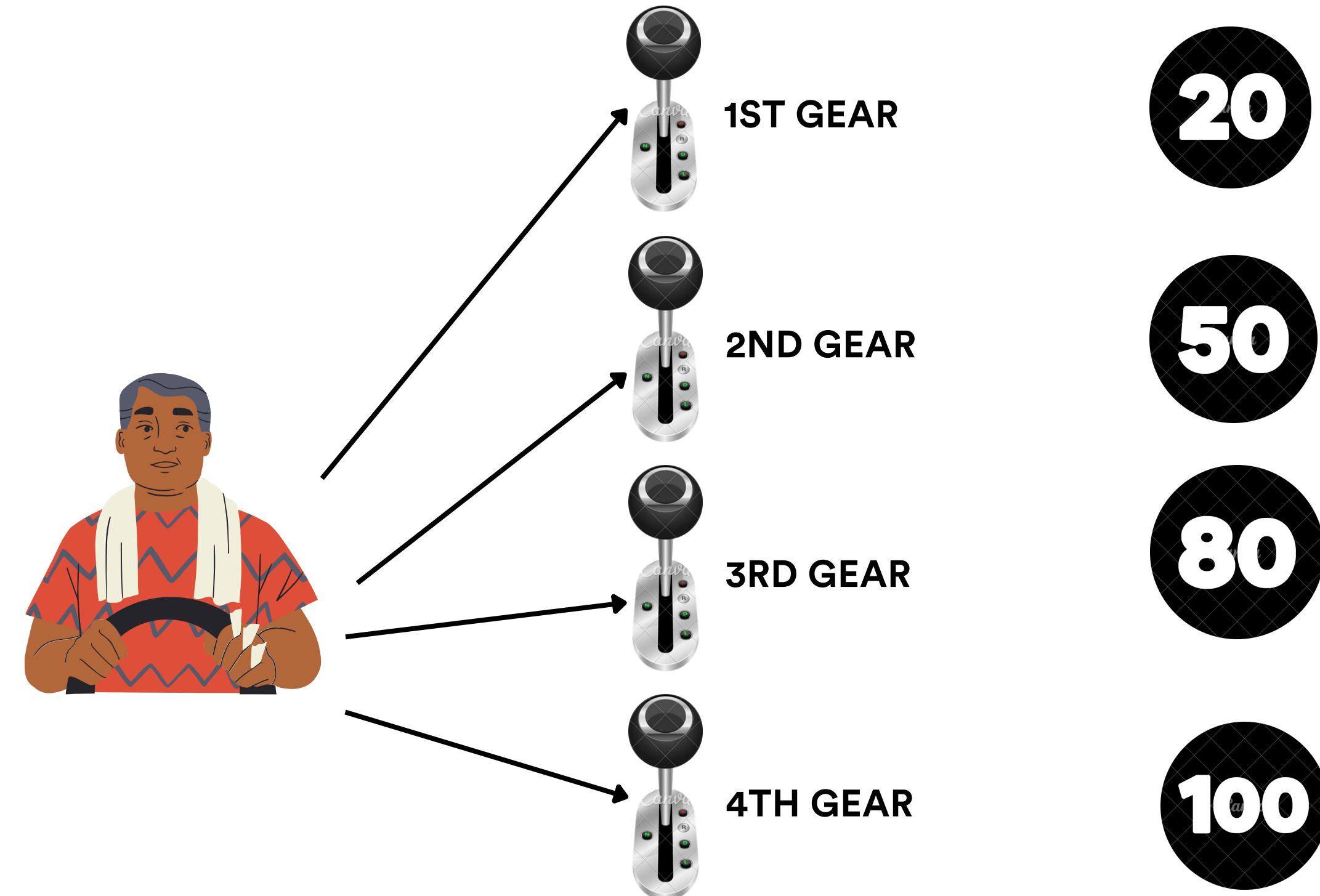
1° ROUND



BUSSINESS

DRIVER CHECK GEARS

EXPECTED

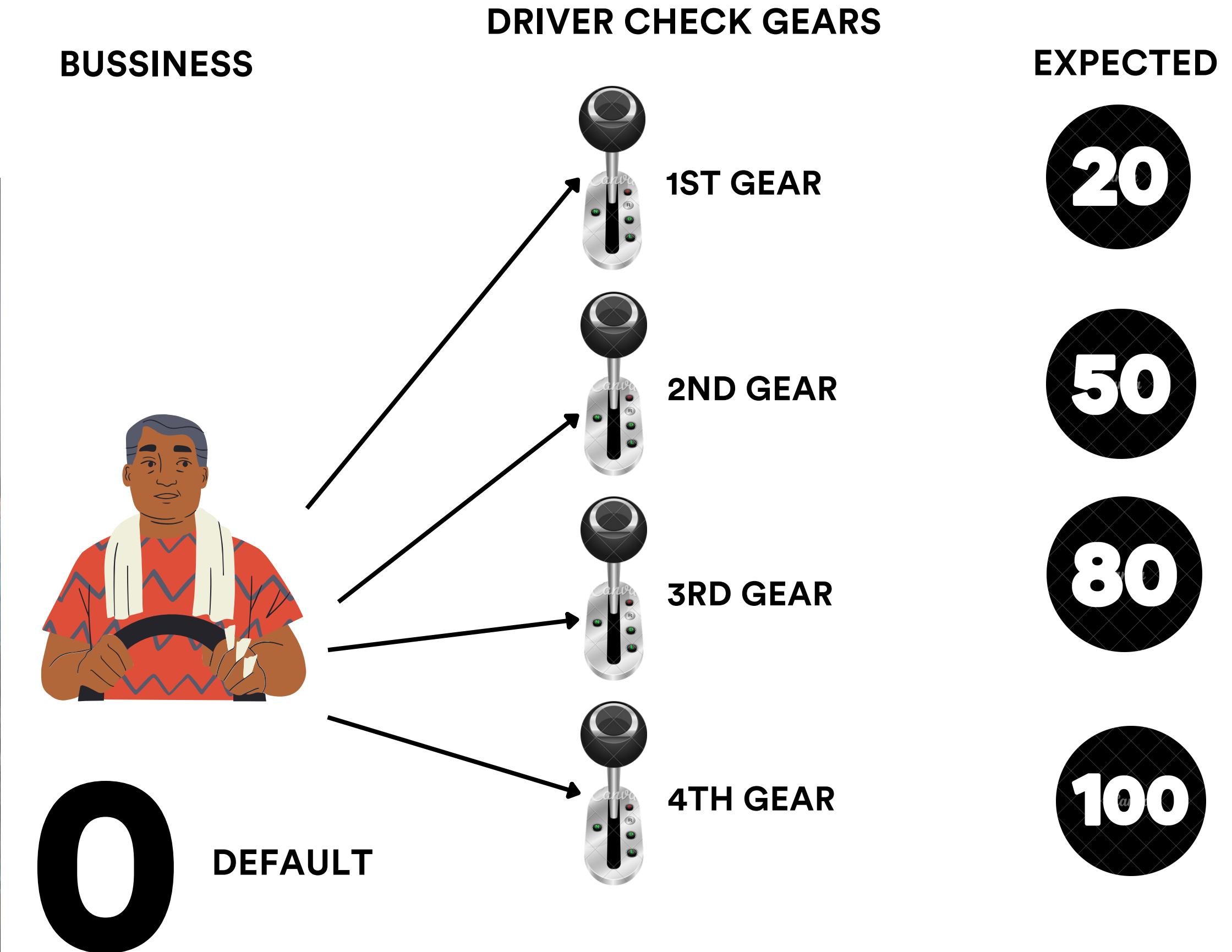


https://github.com/davidgk/avoid_ifs_kopius_talk/tree/develop/driver-gears

1° ROUND



BUSSINESS



https://github.com/davidgk/avoid_ifs_kopius_talk/tree/develop/driver-gears

1° ROUND



**LETS
CODE
IT!**

[https://github.com/davidgk/avoid_ifs_kopius_talk/tree/develop driver-gears](https://github.com/davidgk/avoid_ifs_kopius_talk/tree/develop	driver-gears)

2° ROUND



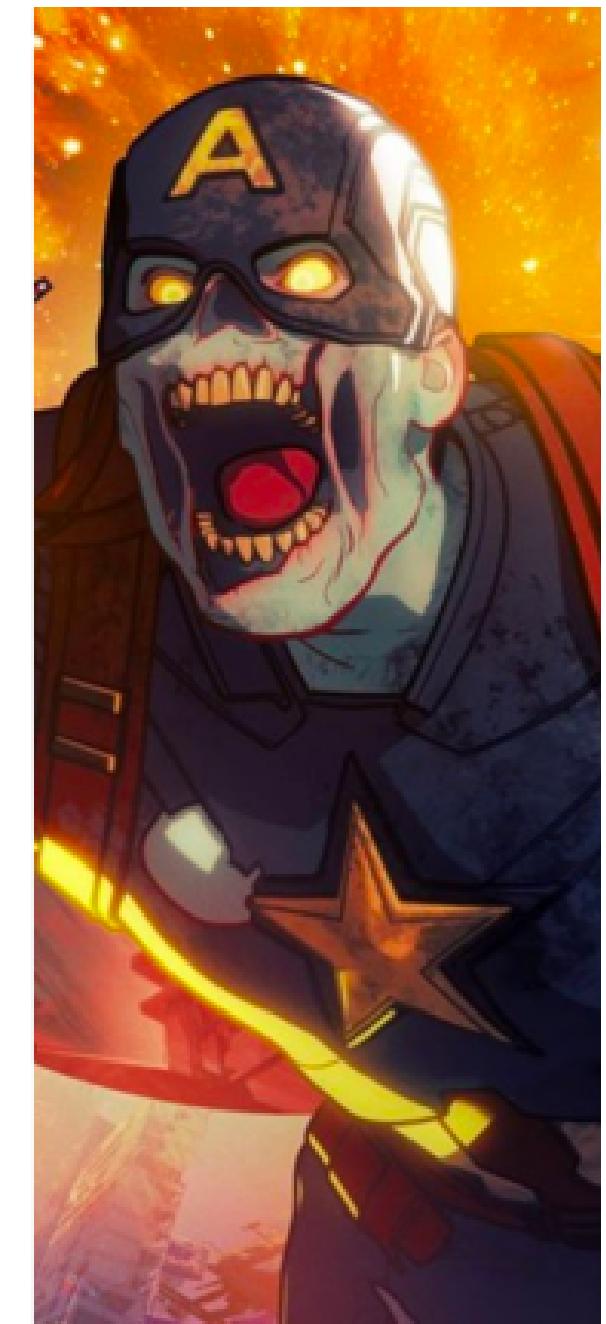
© 2021 MARVEL



DRIVER AND MOTORS

Let's try now to solve a little bit more complex problem. We'll check some actions based on a Driver who wants to Test a motor performance based on the fuel we use

We'll use TDD, and we'll go from implementing an easy solution, but not the best one. Then, based on the test we'll try another approach



[https://github.com/davidgk/avoid_ifs_kopius_talk/tree/develop driver-motors](https://github.com/davidgk/avoid_ifs_kopius_talk/tree/develop	driver-motors)

2° ROUND



BUSSINESS

DRIVER CHECK FUEL

EXPECTED
PERFORMANCE



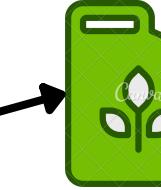
EXPERT DRIVER



ECO FUEL



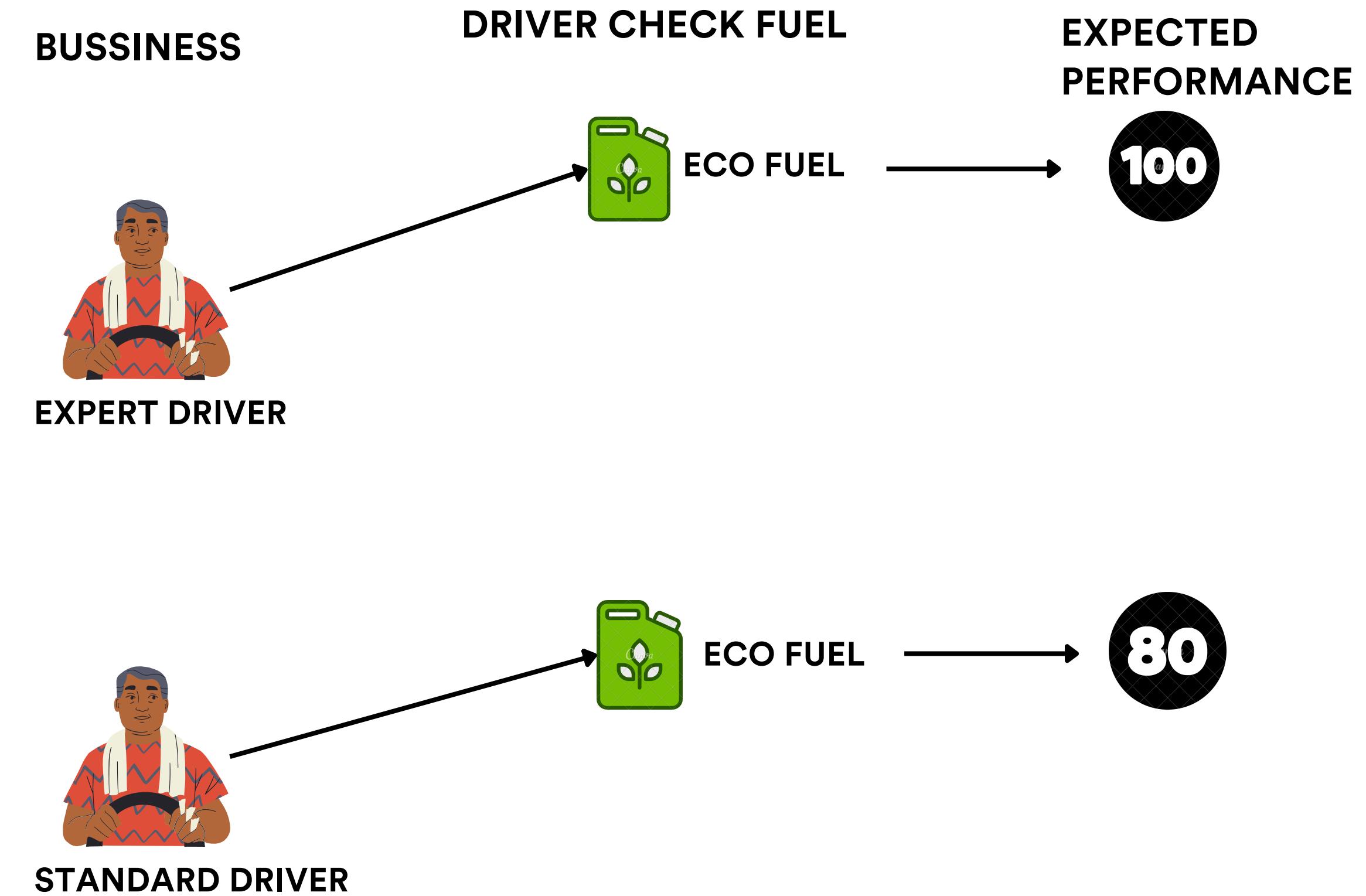
STANDARD DRIVER



ECO FUEL

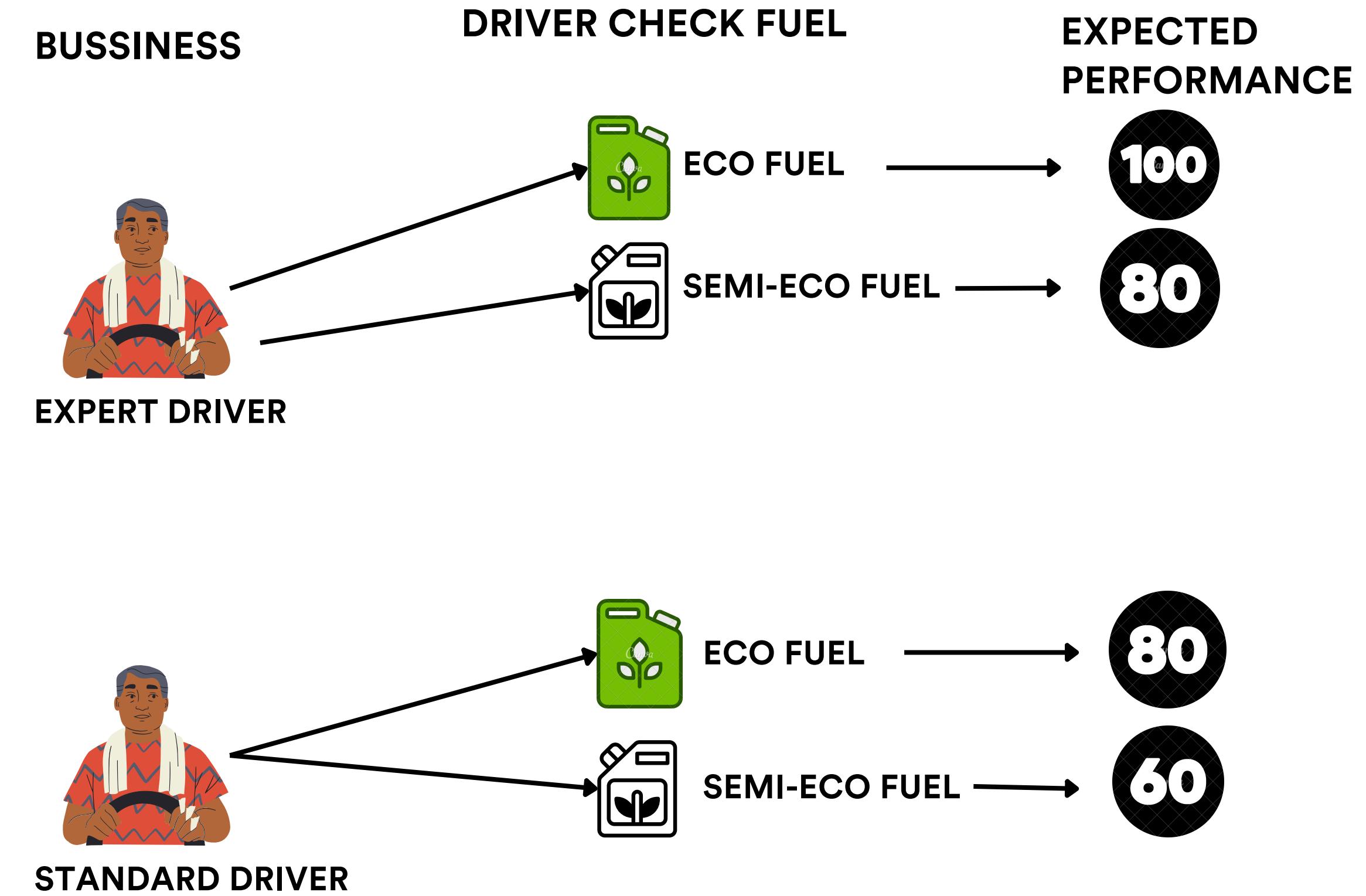
https://github.com/davidgk/avoid_ifs_kopius_talk/tree/develop/driver-motors

2° ROUND



https://github.com/davidgk/avoid_ifs_kopius_talk/tree/develop/driver-motors

2° ROUND



https://github.com/davidgk/avoid_ifs_kopius_talk/tree/develop/developer-motors

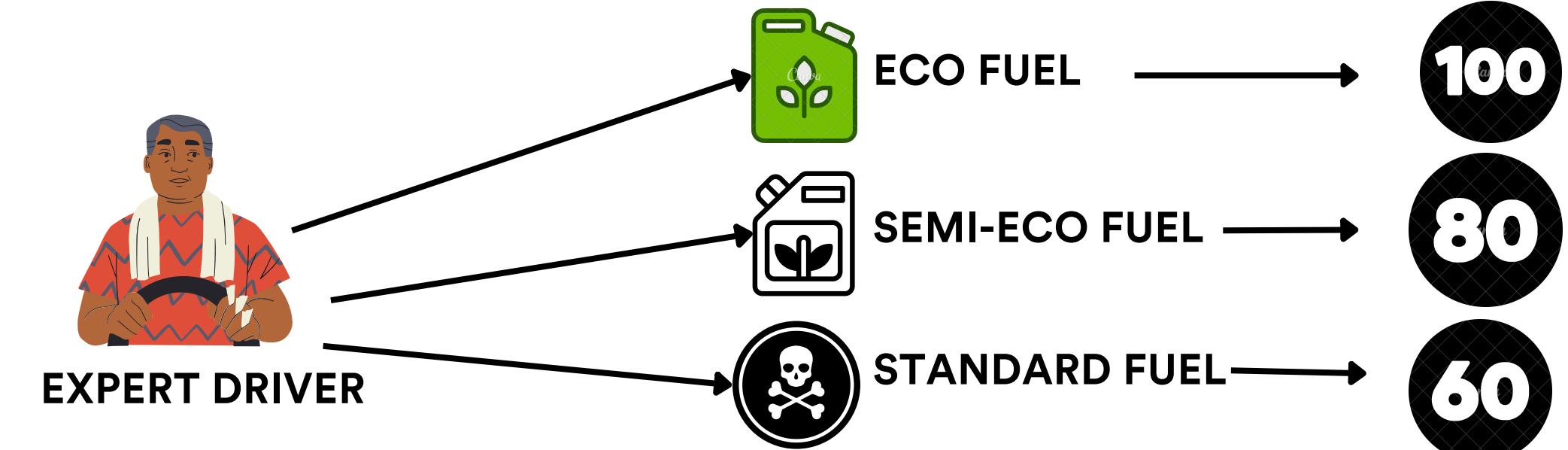
2° ROUND



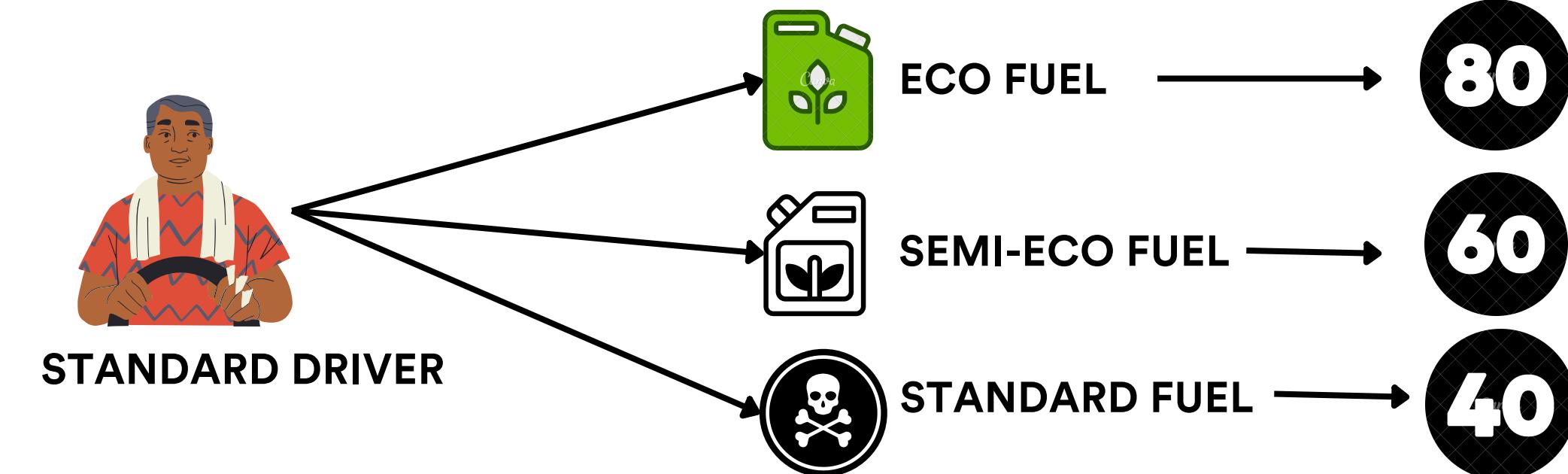
BUSSINESS

DRIVER CHECK FUEL

EXPECTED
PERFORMANCE

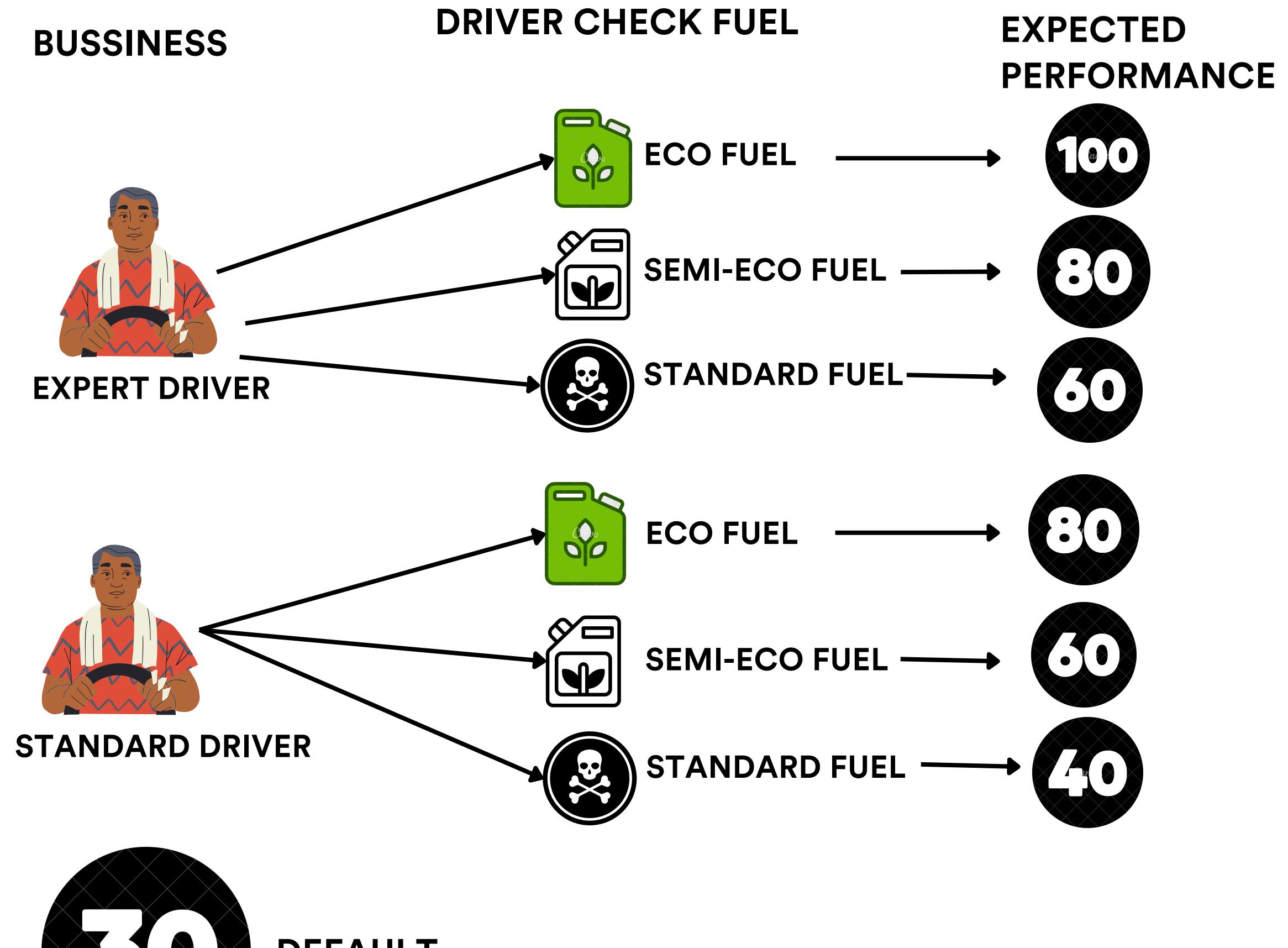


STANDARD DRIVER



https://github.com/davidgk/avoid_ifs_kopius_talk/tree/develop/developer-motors

2° ROUND



2° ROUND



LETS
CODE
IT!

[https://github.com/davidgk/avoid_ifs_kopius_talk/tree/develop driver-motors](https://github.com/davidgk/avoid_ifs_kopius_talk/tree/develop	driver-motors)

CONCLUSIONS



PROS & CONS IF'S WAY



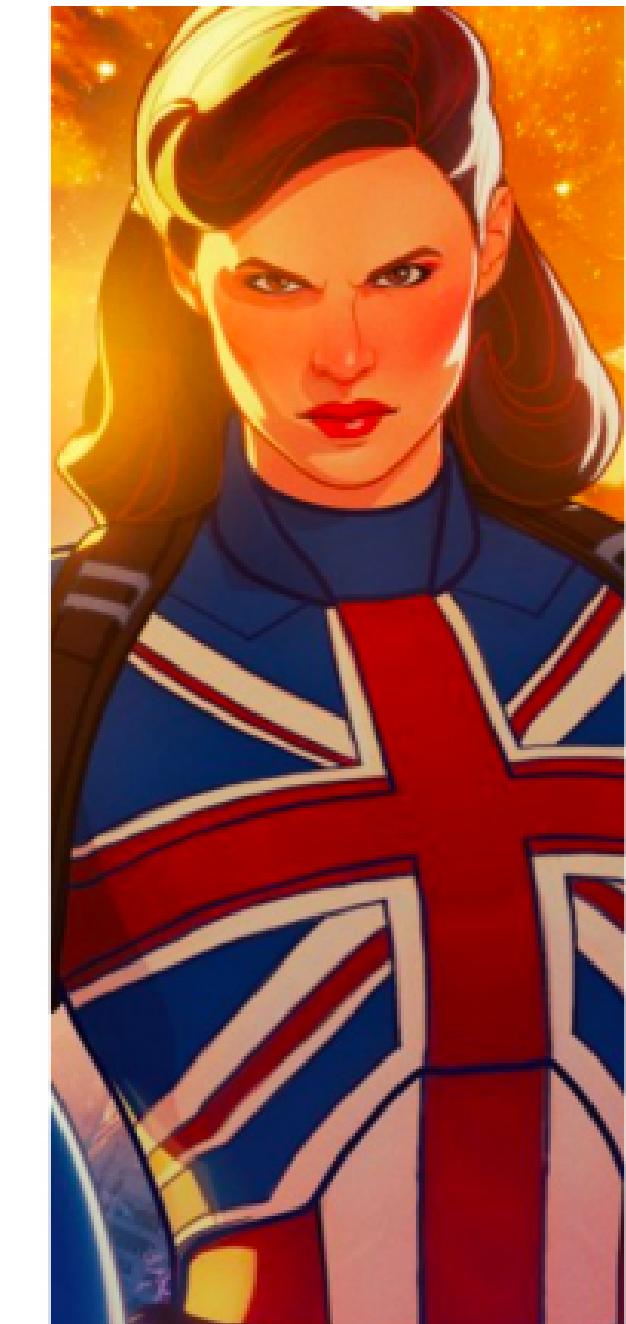
CONS

- High Coupling
- Low Cohesion
- Anemic Class. Incomplete Objects
- Breaks Encapsulate
- Spaghetti Code
- A new condition, means the code is getting more complex
- More complexity, means the maintainability complexity increase



PROS

- Simplicity
- Quicker implementation



PROS & CONS OOP'S WAY



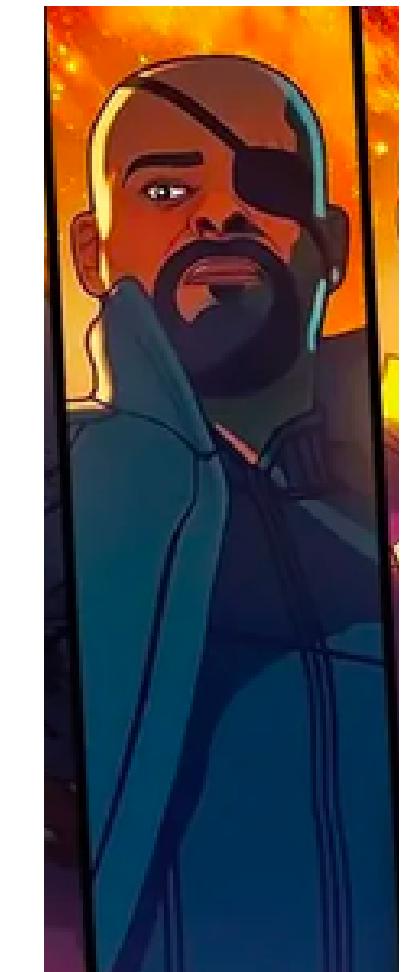
PROS

- Low Coupling
- High Cohesion
- Complete Classes
- Don't Breaks Encapsulate
- Simpler Code
- A new condition, goes to the specific owner
- With more complexity, it goes to the specific owner, not adding complexity in any place



CONS

- At beginning lower velocity then it's payed by lower complexity on new stuffs
- Over design if the problem is simple



HACK - 1



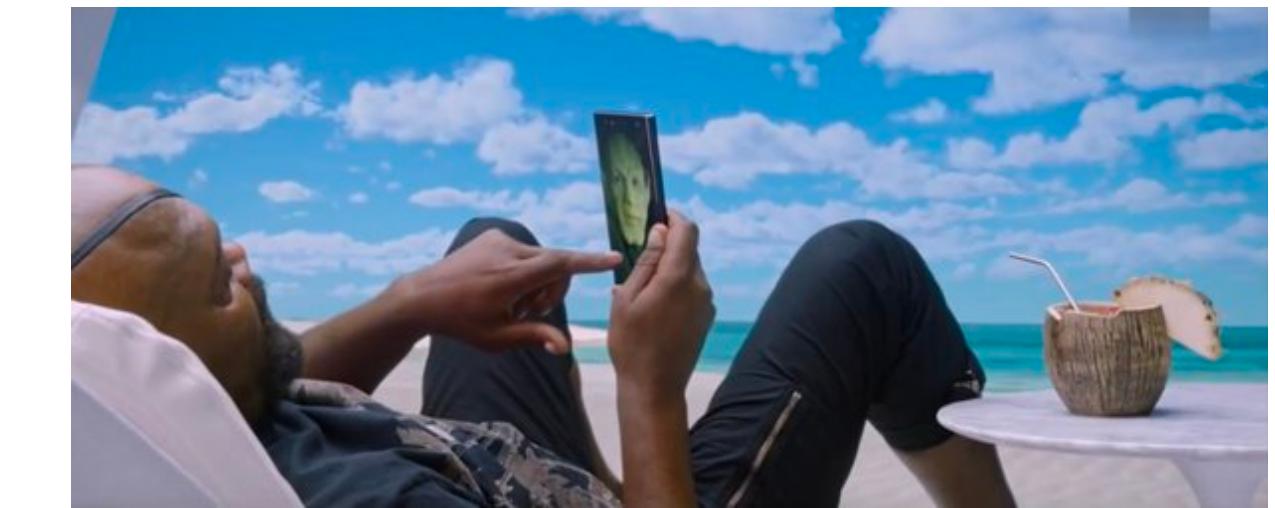
**WHEN IF CONTAINS A CONTEXT,
CAN BE REPLACED BY A CLASS**



HACK - 2



**CONDITIONS NESTED, MEANS THAT
MIGHT BE THINGS THAT HAPPENS
INTO A CLASS**



HACK - 3



**TESTS
MATTERS!!!**

MAKE IT RUN

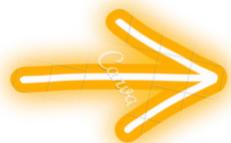


HACK - 3



**TESTS
MATTERS!!!**

MAKE IT RUN



MAKE IT RIGHT



HACK - 3



**TESTS
MATTERS!!!**

MAKE IT RUN

→ MAKE IT RIGHT →

MAKE IT FAST





THANK YOU

FOR WATCHING

KOPIUS ACADEMY | 2023

