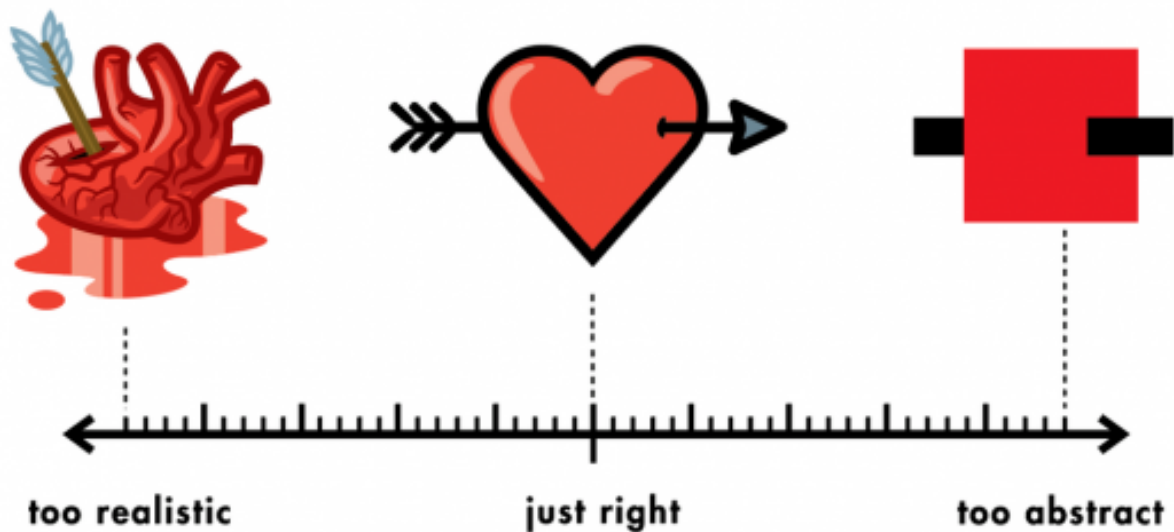


# ABSTRACTION

HIDING COMPLEXITY

## THE ABSTRACT-O-METER



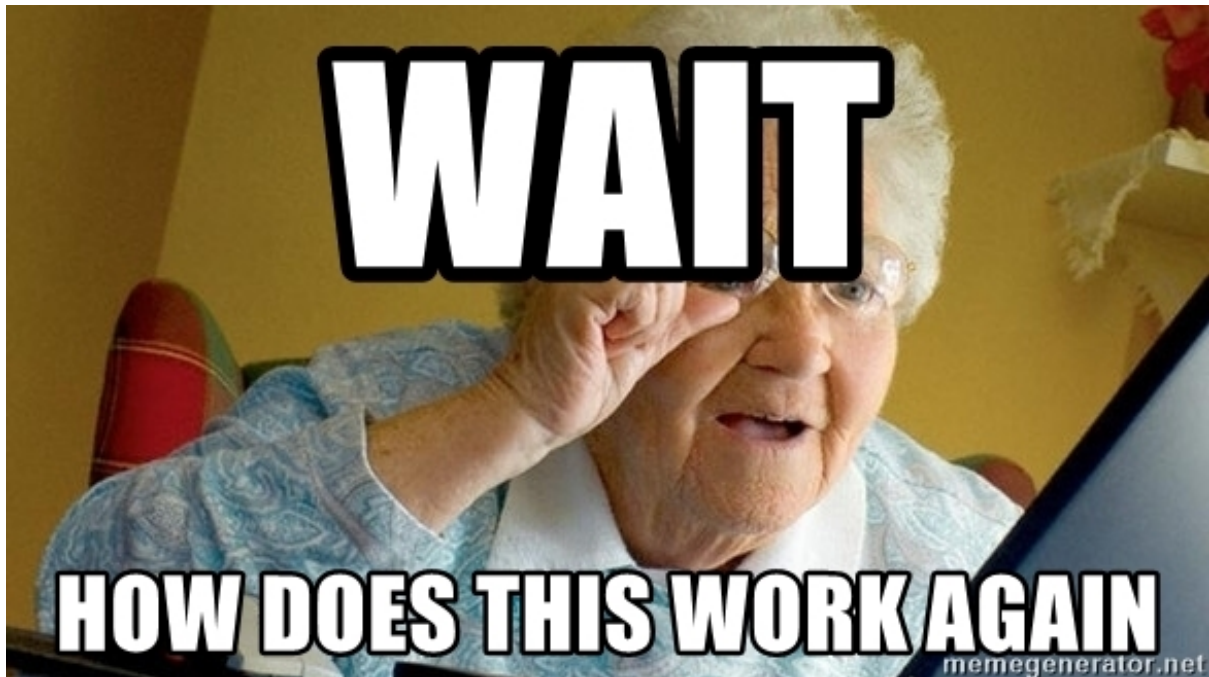
As a kid you would often ask questions that sound something like “How does this work?” In IT we often don’t ask that question because we don’t care how it works, only that it does. But sometimes it helps us to do our job better if we understand computers on user level, as well as on the bits and bytes level. In this chapter, we explain why it is important to understand abstraction and how abstract are computers really.

## WHY IS ABSTRACTION IMPORTANT

Abstraction is the process of simplifying concepts by removing details and hiding complexity to focus on what is important.

If a child asks you how a car works, you wouldn’t start explaining how fuel is burned and laws of thermodynamics. You would probably say that a car is a device that gets you from point A to point B, using some source of energy. Later you would explain how to drive a car, and after

that you might explain what is under the hood, but you definitely wouldn't start with laws of thermodynamics.



<https://memegenerator.net/instance/80849458/grandma-finds-the-internet-wait-how-does-this-work-again>

Example of how the same thing can have different numbers of abstraction layers would be the network models, ISO-OSI and TCP/IP. Even though they describe the same thing we use different models with different numbers of layers depending on our needs and work we do.

## **ABSTRACTION IN REAL LIFE**

Now let's take a look at how complex things get if we look beyond the abstraction, and if we start wondering how things work? We can use the car analogy to see how far abstraction can go, and we will compare it to the abstraction of a computer. It's important to understand that the number of layers in abstraction is not defined. In this example we will compare playing computer games and using a car. Examples in the table below describe abstraction from the highest level, where we only use the product, to the lowest level where we get an in depth explanation of how something works.

COMPUTER GAME

CAR

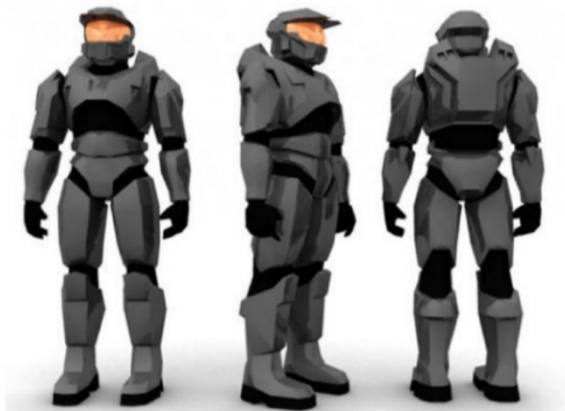
HIGHEST LEVEL



GAME - ALL YOU NEED TO KNOW IS HOW TO KILL THE ENEMIES



PASSENGER - ALL YOU NEED TO KNOW, IS THAT THE CAR WILL GET YOU PLACES



GAME ENGINE - ALLOWS YOU TO CONTROL WHAT WILL BE IN THE GAME



DRIVER - ALLOWS YOU TO CONTROL, WHERE YOU WANT TO GO

```

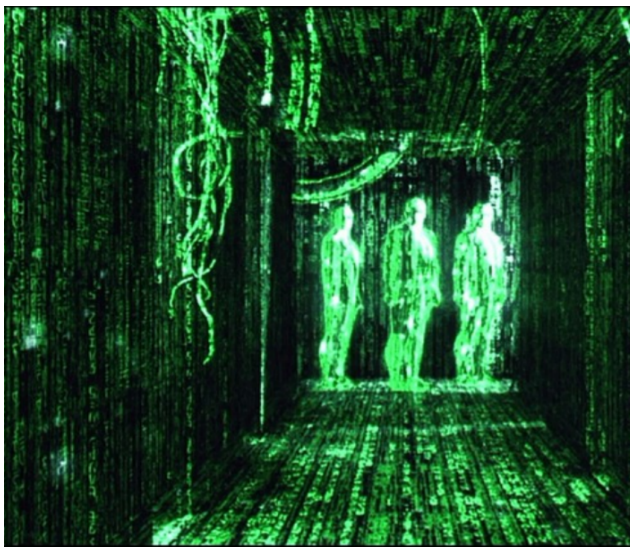
181 public void actionPerformed(ActionEvent e)
182 {
183     if (e.getSource() == forwardJButton)
184     {
185         Transform3D temp = new Transform3D();
186         viewObjectFromGroup.getTransform(temp);
187         Transform3D tempDelta = new Transform3D();
188         tempDelta.setTranslation(new Vector3f(0.0f, 0.0f, -1.0f));
189         temp.mul(tempDelta);
190         System.out.println(temp);
191         float matrix[] = new float[16]; //declare array of 16 floats for a
192         temp.get(matrix);
193         if (matrix[11] <= 1.0)//object front face z = 1
194         {
195             System.out.println("Don't multiply Transform3D at: "+matrix[11]
196         }
197         else //setTransform
198         {
199             viewObjectFromGroup.setTransform(temp);
200         }
201     }

```

HIGH LEVEL SOURCE CODE - THE MEAN OF CREATING THE GAME



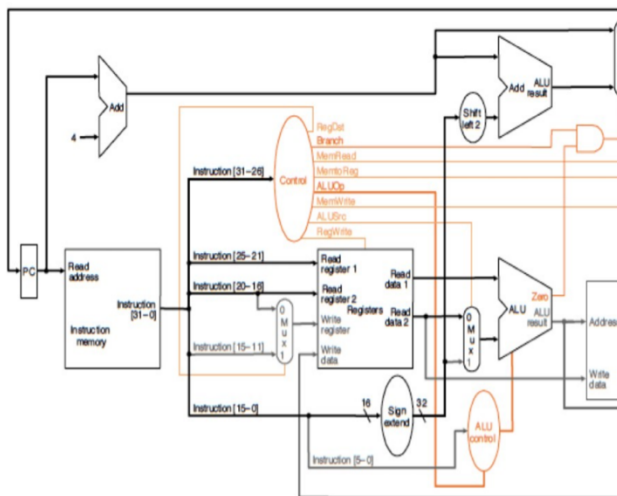
CONTROL PANEL AND GEAR - MEAN OF CONTROLLING THE CAR



MACHINE CODE - HOW DOES THE COMPUTER "UNDERSTAND" WHAT YOU MEANT

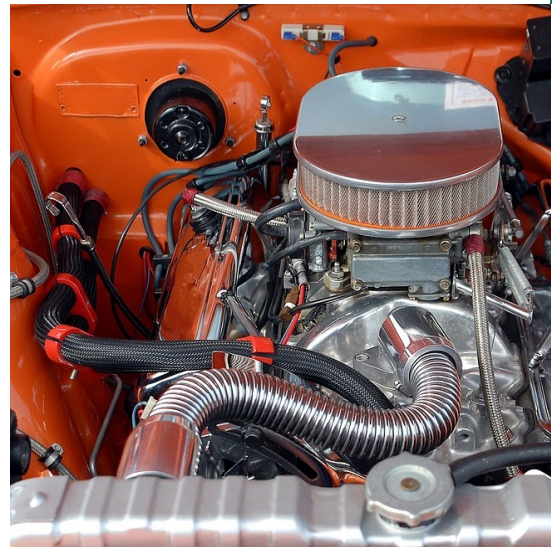
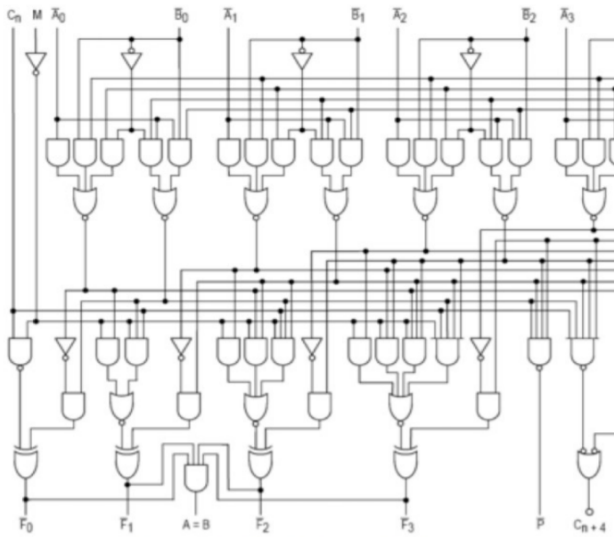


SHAFTS, GEARS, PIPES - TRANSLATING YOUR ACTIONS INTO CAR'S LANGUAGE



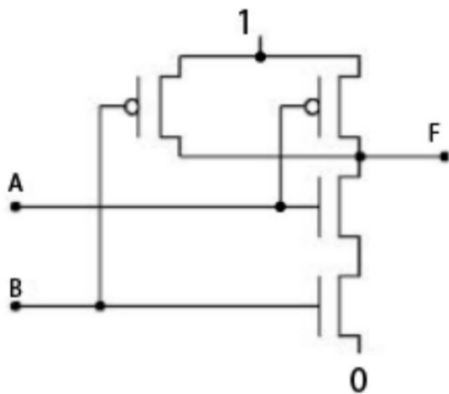
CPU - THE BODY OF THE COMPUTER

UNDER THE HOOD - THE BODY OF THE CAR



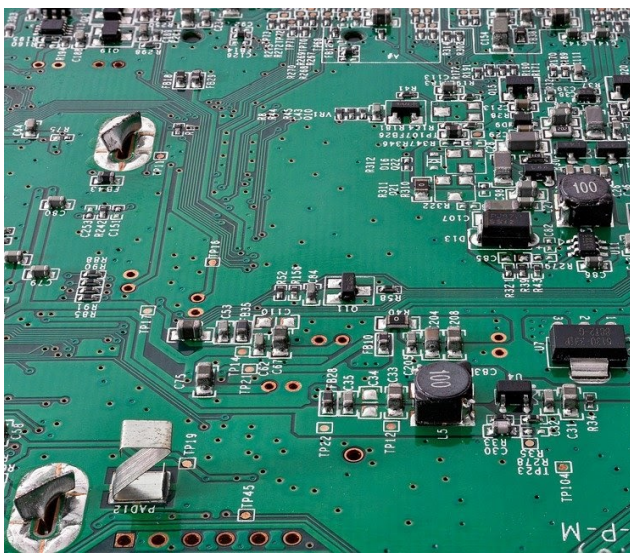
ALU - THE BRAIN OF THE COMPUTER

ENGINE - THE HEART OF THE CAR



LOGICAL GATE - ALLOWS COMPUTER TO PERFORM OPERATION ON BITS

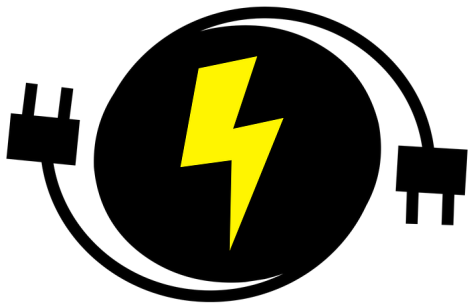
PISTON - ALLOWS ENGINE TO GENERATE MOVEMENT/ENERGY



TRANSISTOR - PREVENTS OR ALLOWS CURRENT TO FLOW THROUGH



FUEL - RESOURCE THAT ALLOWS PISTONS TO MOVE



ELECTRICAL CURRENT - FLOW OF ELECTRONS WITHIN CONDUCTOR



HYDROCARBON AND OXYGEN REACTION TO IGNITION - SET OF CONTROLLED EXPLOSION

LOWEST LEVEL

When you start to understand the concept of abstraction levels and you just can't stop getting more into it

