Margrete, Teis, Jonas, Nanna
Number of characters: 19115
AU, Aesthetic Programming 8/5-2018

Exam project - CAPTCHA

Our program revolves around aspects of human intelligence, code and coding practices through a CAPTCHA program. In our synopsis we are using human natural language as tool for analyzing the differences and similarities between language and code in its expressive form. In this assignment we highlight certain aspects of our code to draw the line between compressing complex human intelligence into algorithms and how the programmer also shapes a program to prejudice.

Our program and its concept

Our program judges the humanness of the user. Inspired by the Turing test, our program sets up a numbers of tests for the user to complete in order to determine if the user is a human or not. The program itself is made up of 6 different levels. These levels challenge the user in different kinds of human intelligence, starting with very easy challenges, and gradually increasing in complexity. The following are the levels and what kind of intelligence the puzzles challenges:

1. **Pattern recognition**
   In this puzzle the user is asked to find the warped letters and write them in the inputbox. This challenges the user's ability to recognize letters.
2. **Leet speak**
   In this puzzle the user is asked to recognize the sentence written in leetspeak. This challenges the user's ability to decipher words from an otherwise incomprehensible sentence.
3. **Object recognition**
   In this puzzle the program shows the user three pictures of three different objects, in which the user is asked to find the picture of a fish. This challenges the user's ability to distinguish between the objects and recognize the right one.
4. **Sense making in a sentence**
   In this puzzle the program proposes an incomplete sentence where the user is asked to complete it by choosing the right word. This challenges the user's linguistic understanding.
5. **Emotions**
   In this puzzle the program shows an image of a domestic abused woman and asks

how the user feels towards the picture. This checks whether or not the user gets emotionally affected by strong pictures.

6. **Facial recognition**
   The last puzzle of our program authorizes the human by checking if the user has a face or not. This program turns on the user's webcam, asks the user to look into it and detects for a face by applying a face model onto the video capture. This authorizes the user by his or her bodily properties.

The structure of the program is meant to show how authorization programs like CAPTCHA is evolving in parallel with the advancement of the intelligence of technology.
Looking at the program superficially one could argue that there is a difference in the difficulty of the levels. The first three levels are based on the traditional CAPTCHAs. They can seem easy to solve for both the human and AI, since they rely on our natural ability to recognize and differentiate. This ability is something easy for the AI to learn since it's rule-based and easy to simulate, and the consequences of this is that the traditional CAPTCHAs no longer provide the security that they once used to. The last three levels tries to envision how future CAPTCHAs might look like because of the advancement of AIs that can beat them. These levels are what we think as unbeatable by current AI, since they demand semantics, intuition and emotional aspects from the user. These demands can seem fundamentally more complex, for a human this might not seem the case as it's a cultural part of being a human. With that said, it doesn't mean that these demands are less complex, it's just more natural for us to meet these demands and it's therefore easier for us to complete the last three levels.

It's this kind of human complexity that we want to explore further. Why do we see these levels as more complex? How can we define our intelligence and compare it to the machine's intelligence? We will explore this field by looking at language, since these complexities can be expressed through language. When we talk about language, we do not only mean articulation performance of it, since *" Speech is so much more than the articulation of words".* (Geoff Cox, Vocable Code)

If we look closer to natural language, it's made up of rules, grammatics and there are rules in how you pick your words in situations. Grammatics is something that expresses human intelligence and is something leanable to all. And since code is based on rules and

algorithms, grammar is something fairly easy to teach AI. It's possible for the computer, to distinguish between true and false and therefore make decisions based on statements we feed the program. The syntax within the code is pre-programmed and is set in stone. Natural language contains infinite possibilities of expressions, but it's set within boundaries, consisting of grammar-rules. These rule-based systems enables our communication, since it defines a common understanding of how to express our language. But as Geoff Cox states in Speaking Code, language can be expressed in so many various ways. He uses irony as an example of expressing yourself in a manner that doesn't come to show verbally, but is "hidden" in speech and writing. Compared to grammar, irony can twist the underlying meaning and intention of the words, whereas grammar can be seen as a more as a constant. Grammar is not capable of conforming or shuffle rules in it's individual context, the same way as syntax can't be changed. In this aspect, there is a similarity between syntax and grammar, since they both are based on rules and thus enables a common foundation for expressing ourselves socially, but there is still a big difference in syntax and grammar.

Furthermore, we see a similarity in the expression of humans and computers. John Langshaw Austin describes how when we say something we also do something (fodnote s. 35). He claims that our individual utterances have a certain effect and outcome on our surroundings. It's described as a speech act and by wording it that way, it's emphasized that our verbal expression is not empty, as such that it has consequences and is therefore more to be seen as an action. This comes very close to the execution of a program, since it's based on syntaxes which affects the outcome. On the other hand, there is no gap between a command and the execution. It happens at the same time. This is also where the differences can be seen human and machine. Everything that are being produced in code can be reduced to a computational analysis. Speech express the variety of articulations, tempers, physical capabilities and cultures. They are so resistant to compress into an algorithm and furthermore analyze the lines of codes, since it's not scalable.

The way we express ourselves and how we adjust our language behaviours is something very unique from person to person, and thus it becomes hard to generalize it. It depends on factors like previous experience, social environments, personality, feelings and so on. Like Geoff Cox says:

"The meanings of words are not derived from an inherent logical structure alone that manipulates symbols into particular sequences (like a program), but also from their social usage."(Geoff Cox, Vocable Code)

It is therefore difficult for technology to ever perform human language perfectly since they all operate algorithmically and pragmatically. Computers are not idiosyncratic in themselves, they depend on human input which states it's values and can thereby simulate different expressions. But this mere simulation is not convincing enough. By looking at advanced chatbot technologies we see that the sentences makes sense, but at the same time we can feel the lack of emotions and social context in these conversations. Our ability to feel this kind of disparity is why the requirements of passing the Turing test is for the examinator to be convinced that the examinee is human through conversation. It's fairly easy to teach the computer to simulate human language and converse, but to duplicate the way we humans think and feel seems almost impossible since it's so complex for even us to understand. As Geoff Cox says about John Searles' observations and conclusions:

"His observation is that the syntactical, abstract or formal content of a computer program is not the same as semantic or mental content associated with the human mind. The cognitive processes of the human mind can be simulated but not duplicated as such." (Cox, p. 31)

When it comes to human intelligence some aspects are more easily described and understood, while others still seem intricate to even ourselves. While we have a full understanding on how to use intelligence, we have a hard time explaining why and how. This also become challenges during the programming or learning of an AI. Mathematics and object recognition is some of the easy abilities to recreate as it includes a lot of similar learning data and defined answers. Language on the other hand could be seen as more complex. Not only does it rely on complicated rules, it also is forever evolving in parallel with the society or culture who uses it. Understanding a language also includes an awareness of both its linguistic context and the context in which it is expressed. Lastly there is intelligences that seem rather impossible to make concrete. These are abilities like understanding feeling, both each own and those of others, but also that of creating art.

Because of the advancements of AI technology that can beat the CAPTCHAs it forces the CAPTCHA to challenge more complex intelligences lige the last spoken of. And by doing this the programmer of the newer CAPTCHA quickly become more biased than the old ones, since the newer contains questions that evolve feelings and subjectivity. Our program wants the users to reflect upon and emphasize how the computer is judging/testing our humanness based on some parameters and algorithms that someone has determined for the program. As mentioned before, human thought and feelings are something very hard, if not

impossible, to recreate or duplicate. And therefore our program can seem rather absurd and even extreme in its way of testing the user. But the point of doing this is to emphasize on the humanness and subjectivity behind the program.

Looking deeper into a syntax

In the following piece of code we want to show the algorithm we've made to accept the human face. This is done by using the clmtrackr. And by using the function ctracker.getScore(), we can get an output of how well the face fits the predefined face model. The output of this score is what we've called "face". We chose to make an if-statement in which the tracker needs to find 80% of the facial points before being executed. But why have we chosen only 80% instead of 100%?

```
if (face > 0.8){
      fill(0, 255, 0);
      text('Human confirmed',width/2,height/1.1)
}
```

To figure out what the value of the condition should be, we have made a console log as so:

```
console.log(face);
```

Our average score when looking into the console.log were 85%. So to make the if-statement executed at all times when looking into the webcam we gave a small buffer of 5% so the text (Human confirmed) won't flicker on the screen. This argument is made with a boolean expression, so that the face needs to be more than 80% tracked before the if-statement is executed. We will now dissect this line of code into two subjects: the boolean expression and the pModel:

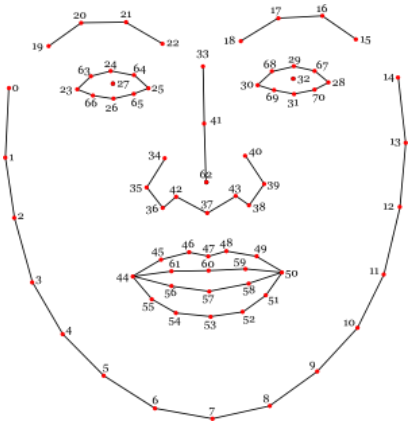
Boolean expression

A boolean expression is in its nature a way of constantly evaluating whether a statement is true or false, and thereby judging between right and wrong. The conditions put around the boolean expression plays a very essential role of our program. We've used these conditions quite often in our program to be able to create the leveling system. If a statement is fulfilled the program grants access to the next line of code which can execute a new process.

In the snippet of code that we're focussing on, we see this process clearly in the triggering of human confirmation when face is clearly detected. This syntax is therefore biased since the programmers are the ones deciding how much the facial model has to fit the face before confirming the user. By using a boolean expression in general gives no room for deviation.

pModel

One of the problems by having a predefined facial model to determine the "humanness" of the user is that it seeks out specific looks of a person. It won't be able to look past facial properties which doesn't fit the clmtrackr's face model(pModel) which the tracking points is made up from.



```
ctracker = new clm.tracker();
ctracker.init(pModel);
ctracker.start(capture.elt);
```

What this line of code is actually doing is judging the users face. One of the specs the tracker is looking at is a person's eyes. The eyes is tracked by 18 spots alone, which takes a lot of percentage of the whole tracking. So if the user wears glasses or lacks an eye, there will suddenly be more room for error. This roughly means that a person with a minor handicap might not be judged as being an actual human by our program.

Another interesting thing about this pModel is how it simulates the way humans recognize faces. It seeks to find out how much a person looks like the reference model. A bouncer might look for specific features in a person to see if they are allowed in. If the person doesn't meet the bouncer's demands the bouncer might get suspicious about the presence of the person. The same way goes for this program, that if the user doesn't meet the demands made from our algorithm, he or she won't be granted access to the next level. The only way that this pModel differentiate from us humans is that it is more mathematically precise in handling the captured data. This collected data is measurable and we can see how much the user is similar to the original model. But the pModel is still biased since it can't imagine a face that differs from the modeled face.

As always when talking about code it is important to notice the bias it contains. This can be a problem since it excludes individual opinions, thoughts and in some situations specific

minorities. In our code it is no different from the rest. No matter how hard you try to avoid these things, you would still be able to find biased language in the code. By language we mean the choice of variables and syntaxes used in the program. Firstly the CAPTCHA as we know it contains a correct input. And when we talk about the leet speak part of the program it is clear that there might just be one correct answer. In object recognition there is a "rightbutton" and a "wrongbutton" to determine the outcome.

The biased programmer becomes even more visible when looking into our emotion level in level 5. First of all we are talking about feelings and what the right way to feel is. In line 189-203 our code has a really strong indicator of a biased programmer. Firstly, we have only chosen 3 different kind of emojis to show 3 different kind of feelings. But as we want to show with our program, the human is a really complex being and you can't really compress all feelings into 3 different variables. And to make it even more clear with the biased code, we have chosen to call the variables: "*wrongEmotion*" and "*rightEmotion*":

```
sadSmiley4 = createImg('Images/sad.png')
  sadSmiley4.position(windowWidth/3-50,height/1.3-sadSmiley4.height/2);
  sadSmiley4.size(100,100);
  sadSmiley4.mousePressed(rightEmotion);
```

Again this choice in code can really exclude other opinions and feelings. But again this is something that we want to show with this program. It is somewhat impossible for the computer to mimic a human through algorithms. As we've mentioned above and as Geoff Cox writes in the paper Vocable Code:

"The cognitive processes of the human mind can be simulated but not duplicated as such." (Cox, p. 31)

if the question of level 5 "How does this make you feel" should be more righteous and justifiable, then we could have made all answers the correct option. But in relation to our original idea of the concept, we wanted to show that the computer doesn't have these feelings but merely operates by algorithms. AI can not yet understand abstract human attributes such as feelings and how one should feel in relation to specific topics and contexts. But how far away is AI in relation to human intelligence? Is it so far as we think?

This opens up some deeper discussions about intelligence and what it is. It's interesting to talk about whether human intelligence is as complex as we think and speculate whether or not we possess a "higher dimension" of intelligence where we can feel, judge and have consciousness. Is this kind of intelligence the "real" intelligence, and is simulated intelligence just as "real" since it's functional and effective? One could argue that the simulated intelligence is just as intelligent as us, but just in another way than us. We just have a hard time understanding since we're exposed to the idea, that we're unique and are supreme beings with awareness of our existence. What if our brains are actually very similar to computers, logical and algorithmical, but just so abstract and incomprehensible that we can't put it into words?

Conclusion

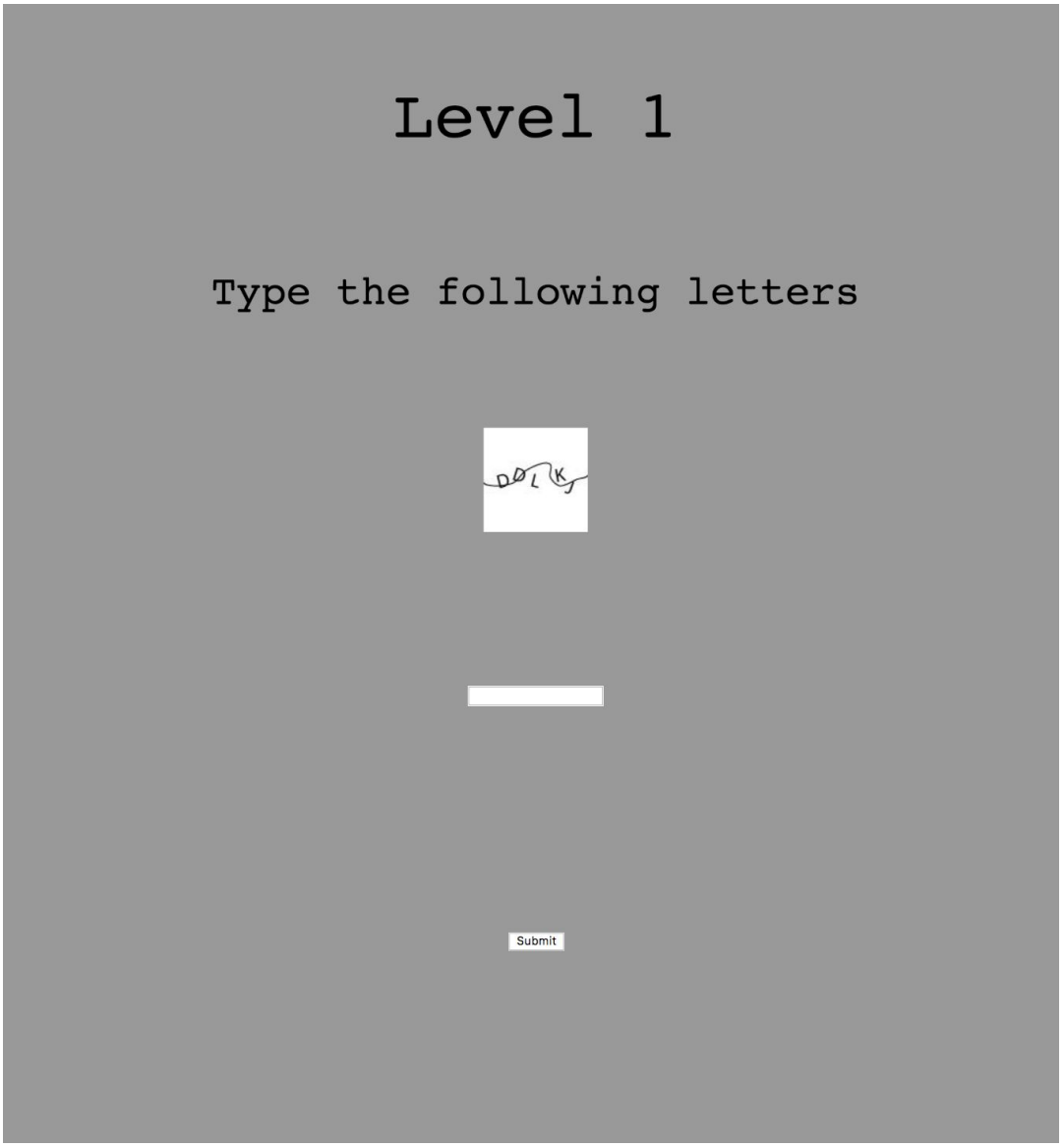
By making a CAPTCHA program we've made a critical work on how one might differentiate human from computers. In doing this, one needs to figure out the differences between human intelligence and computer intelligence. In this paper we have explored how this might become more and more difficult as the development in human-like AI progresses. But we have also reflected upon how some intelligences and abilities might be hard to compute and therefore must in some way be simulated. This way of simulating might eventually lead to a lack of understanding in the original ability as it is simplified into a simulation.

Through our program we want to show the development of CAPTCHAs and reflect upon how one might differentiate human from computer as the advancement of AIs continue. The future CAPTCHAs might be more complex in the sense that they could challenge the user's semantics, intuition and feelings. While doing this we have reflected critically upon how programs are simulating human intelligence and therefore biased by the programmer. We have also reflected upon the computer's difficulties in simulating complex human attributes by looking at the relation between natural human language and code as an illustration of this. Natural language is not just limited to speech, but also body language, articulation, social environment, personality and so on, it gives us the ability to express our thoughts, ideas and feelings and therefore a part of our intelligence. By looking deeper into a piece of our code we've explored this further in providing our perspective on how this differentiation in intelligence unfolds in our program in forms of biased code and judging algorithms. Further on we've engaged in a discussion about human intelligence and artificial intelligence. The discussion questions whether computer intelligence is as intelligent as we are, whether the advancement of this intelligence should be something to fear and lastly, whether human intelligence is as complex as we think.

References:

Cox, G. (2013). *Speaking code*. Cambridge (Mass.): The MIT Press.

Screenshot of our program:



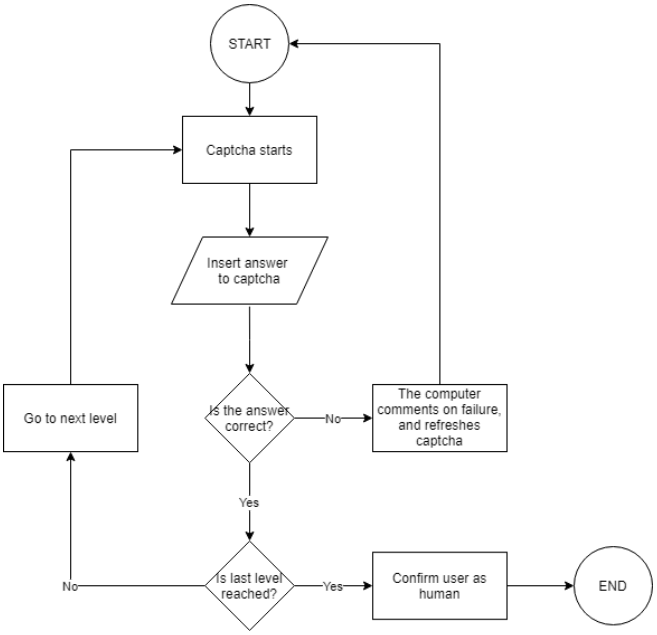
Link to our program:

https://rawgit.com/AUAP/AP2018_Submission/master/Final_Project/Group2/Final_project/Captcha/empty-example/index.html

Link to video:

https://youtu.be/C3vfXswa1yA

Flowchart:



More technical Flowchart: