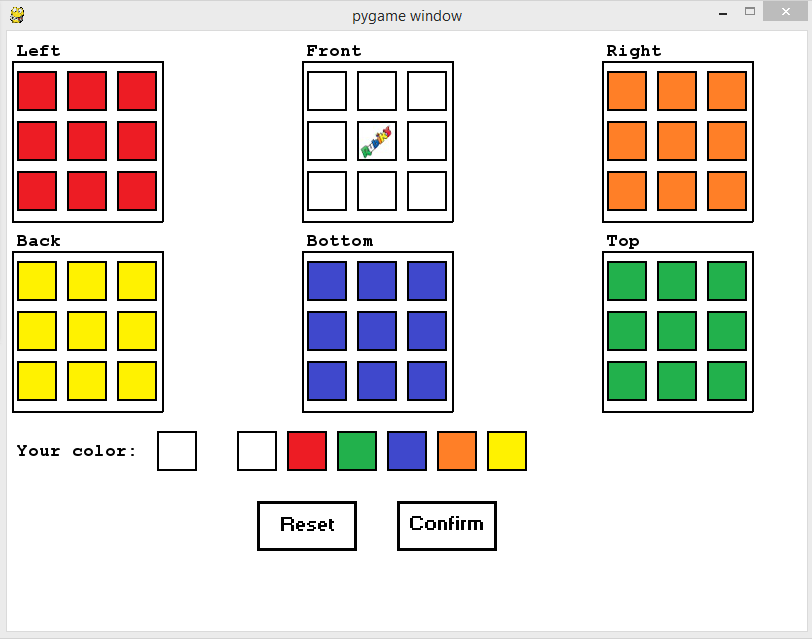
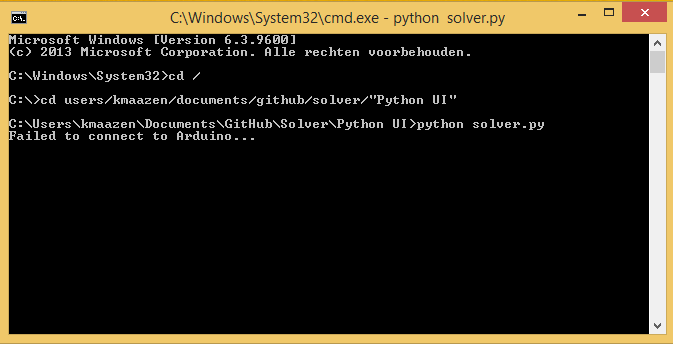
**Stappenplan Project Rubik’s Kubus overdracht 23-7-2018***Quickstart voor het aan de praat krijgen van de Solver*

Stap 1:  
(Als je niet weet hoe je met Git moet werken volg dan onderstaande uitleg, zo wel gebruik dan gewoon Git zoals gewoon met de URL van Stap 3).  
Ga naar <https://desktop.github.com> en download + installeer Github Desktop(Maak een nieuw Github account aan als er om wordt gevraagd).  
  
Stap 2:  
Als Github is opgestart zie je 3 opties:   
-Create a new repository  
-Add a local repository  
-Clone a repository  
Kies Clone a repository.  
  
Stap 3:  
Kies hierna het tabblad URL van de bovenste 3 tabbladen.  
Gebruik de volgende link: <https://github.com/frankhuurman/Solver.git>  
Kies een local path waar je je Git bestanden wilt opslaan.  
Druk op clone en hierna worden de bestanden die horen bij het Rubik’s kubus project van github afgehaald en in je local path neergezet.  
  
Stap 4:  
Open je command line in Windows(zoek naar cmd.exe).  
Check of Python aanwezig is: type python –V en druk op enter om dit te checken.  
Als je iets in de vorm van “Python 3.6.2” terug krijgt dan staat Python er op, zo niet installeer Python dan vanaf hier: <https://www.python.org/getit>

Stap 5:  
Om het programma te draaien zijn 2 belangrijke dependencies nodig namelijk “Pygame” en “Pyserial”.  
Om deze te installeren herstart je je cmd.exe nadat je Python installatie klaar is, en type je pip install pygame. Als het installeren hiervan klaar is type je pip install pyserial.  
(Krjig je errors dat pip niet gevonden kan worden of iets dergelijks? Dan zit het er waarschijnlijk in dat je cmd pip niet kan vinden in je omgevingsvariabelen.  
Druk in dit geval op windows knop + pause/break en vervolgens op geavanceerde systeeminstellingen(waarschijnlijk heb je hier Martin of Hans even bij nodig gezien de rechten die je nodig hebt). Als je online zoekt naar iets met environment variables en pip python install dan krijg je voldoende hulp om dit rond te krijgen).  
  
  
  
  
  
Stap 6:  
Alles gelukt met het installeren van pygame en pyserial? Nice! Let’s launch it!  
Solver.py is het hoofdprogramma die te vinden is in:  
<Door jou ingestelde local path> \GitHub\Solver\Python UI  
Doe dit weer in je cmd.exe met behulp van onderstaande stappen.  
Stel je hebt je documenten folder ingesteld als local path dan kom je er met de volgende commando’s in cmd:  
cd / (Zorgt ervoor dat je naar het begin gaat van je C: schijf)  
cd users/jouwusernaam/documents/github/solver/”Python UI”  
  
In mijn\* geval zou dit commando worden:  
cd users/fhuurman/documents/github/solver/”Python UI” <- zorg dat je de aanhalingstekens toevoegt anders snapt Windows het niet!  
  
\*omdat dit deel van het document op een andere computer wordt getypt staat er in onderstaande afbeelding niet fhuurman maar kmaazen.  
  
Als je in deze directory het volgende commando uitvoert zou het programma moeten starten: python solver.py  
  
  
Stap 7:  
Probeer het altijd nog een keer op te starten als het de eerste keer niet lukt, het kan soms zijn dat de dependencies de eerste keer niet goed worden gezien.  
Als alles nu draait en je ziet de prachtige Python GUI dan zie je nog de melding “Failed to connect to Arduino…” in je cmd.exe staan.  
Dit betekent dat je de Arduino die op de Rubik’s Kubus solver zit nog aan moeten sluiten via een USB poort.  
Op regel 137 van Solver.py(indien het niet ondertussen is veranderd) vind je de regel: serial.Serial("COM4", 38400, timeout=0.1).  
Als je de Arduino hebt aangesloten en na het opnieuw starten van solver.py nog steeds Failed to connect to Arduino krijgt dan zal het aanpassen van “COM4” waarschijnlijk de oplossing zijn.(Vaak zit hij op COM3 of COM4 overigens).  
Oftewel open solver.py in een goede tekst editor zoals notepad++, sublime tekst of atom en pas COM4 aan naar een andere port om te zien of het dan werkt(zo niet dan moet er in apparaatbeheer gezocht worden naar de goede com poort die wordt gebruikt voor de Arduino).  
  
De Arduino code is al aanwezig op de Arduino zelf, wil je dit aanpassen of inzien dan moet je in de volgende map zijn:  
<Jouw local path>\GitHub\Solver\Arduino\final  
Dit final.ino bestand kan worden geopend in een aparte IDE voor Arduino en via daar worden geüpload naar de Arduino(let erop dat solver.py niet openstaat tijdens het uploaden, anders is de serial port nog in gebruik en kan de Arduino IDE niet communiceren met de Arduino!)  
  
De map <Jouw local path>\GitHub\Solver\Documentatie is puur voor documentatie zoals dit bestand en wordt verder niet gebruikt in het programma.  
  
Alle gebruikte Python code en overige onderdelen kun je vinden in de “Python UI” map.  
LUT.txt wordt gebruikt als lookup table om delen van het algoritme te scheiden van de uit te voeren Python code.  
  
Cube.py(credits: Joost Saarloos) is het kubus object in code die bijhoudt wat de stand is van het kubus object na het uitvoeren van een draai.  
  
Algorithm.py(credits: Timo Strijker) is waar de magie van het algoritme zich bevindt. Deze heeft een functie die de tekst leest van LUT.txt.  
  
Solver.py(credits: Frank Huurman) is voornamelijk de Python GUI en bevat functies om te verbinden met de Arduino om vervolgens een lijst aan moves te versturen zodat de motoren kunnen gaan draaien.  
  
  
Tips:  
-Leer wat meer over Git of Github om als team jullie eigen Github project te maken en vervolgens alle bestanden die jullie net er vanaf hebben gehaald te pushen/committen naar jullie eigen master branch zodat ieder teamlid er onafhankelijk aan kan werken en het werk bewaard blijft voor in de toekomst. Het zou tof zijn om het grondwerk van ons verder te zien groeien in een nieuw Github project die ons team eventueel zelf van Github af kan plukken om eens te bekijken!  
-Het algoritme heeft een lange tijd 9 van de 10 keer gewerkt tijdens het oplossen. We lopen er nu voornamelijk op vast dat hardware matig niet alle motoren goed worden aangestuurd waardoor het debuggen van die laatste 1/10e keer niet vooruit komt. Waarschijnlijk is er iets mis met de bekabeling. Ook met het zicht op stages en afstuderen van Frank & Joost is er te weinig tijd om het project af te ronden dus we hopen dat het volgende team dit wel lukt!  
-Laat ons graag even weten(REA Docenten hebben onze contactgegevens nog wel) wanneer het project is afgerond, hier willen we graag bij zijn natuurlijk ☺  
-Alles alle motoren goed werken kan de Mario theme worden afgespeeld, de code hiervoor staat op de Arduino. Be careful, with great power comes great responsibility!  
  
  
Thanks to:  
  
Frank Huurman – REA Student  
Joost Saarloos – REA Student  
Timo Strijker – REA Student  
Martin Steenman – REA Docent