

게임의 역습

이 세상에 도움이 될만한 게임을 한번 만들어 보자. 게임이 도움이 안 되냐고?

그건 아니다. 개인적인 스트레스 해소의 방법으로도 사용되며 멘탈을 관리하는 법 또한 배울 수 있다. 그뿐만 아니라 경쟁의 잔혹함, 성취감, 노력의 증명 등 많은 것을 배울 수 있다. 우리 모두 하나가 되어 한 팀을 응원할 수도 있고, 게임에 몰고 웃으며 추억을 만들기에도 충분하다. 그러나 좀 부족하지 않은가? '세상'에게 도움이 된다면. 억울하지 않은가? 이토록 즐거운 게임이 왜 부정적인 시선에 의해 무시당해야 하지? 우리 모두를 즐겁게 하는 것 외에도 문제 해결 방법으로 더 도움이 될 수 있지 않을까?

그래서 준비했다. 세상에 게임으로 도움을 주는 법. 우선 우린 RPG 게임을 만들자. 기존에 있었던 게임에 적용해도 좋고, 새로운 RPG 게임을 만들어도 좋다. 사람들은 RPG 게임의 재미를 느끼며 게임을 하지만, 우린 그를 통해 암호를 만들어 낼 것이다.

암호를 만들어 내는 방식은 블록체인의 원리에서 착안한다. 블록체인은 금전 거래 기록들을 통해 암호를 만들어 내는데, 이 암호는 해킹 가능성도 적고 암호를 빠르게 만들 수 있다는 장점이 있다. 우리는 플레이어들의 좌표이동 기

록으로 암호를 만들어 낼 것인데, 3차원으로 가정했을 때 dx , dy , dz 값이 시간 t 에 따라 움직이는 함수를 암호로 만들어 낼 것이다. 그러면 그때 플레이어들은 돌아다니기만 하면서도 암호를 하나씩 만들어 내는데 그렇다면 무수한 암호 값들이 난수처럼 생겨날 것이다.

이 암호는 난수를 돌려서 만들어지는 가짜 암호들도 거를 수 있다. 플레이어들은 알다시피 레벨 대에 따라서 비슷한 경로로 이동한다. 그에 따라 추세선이 생길 것인데, 이 추세선은 위조암호를 판단하는 열쇠가 될 수도 있다. 예를 들어, 이 암호가 지폐라고 생각해보면 우리는 위조를 판별해야 할 것이다. 그런데 뉴비들이 일반적으로 가지 않는 좌표가 갑자기 나타나거나 말이 안 되는 이동이 나타나면 추세선에서 심한 이탈이 연속적으로 일어날 것이다. 이때, 얼마 이상 벗어난 좌표의 난수는 배제하는 등의 규칙을 통해 쉽게 사실 여부를 확인할 수 있다.

또한, 암호의 다양성도 이 방법의 큰 장점이다. 예를 들어, 메이플스토리의 경우엔 택시를 타는 먼 이동도 존재한다. 이런 큰 좌표이동 또한 그 값을 게임 회사가 알고 있으므로 추세선에 포함될 수 있으므로 다양한 암호가 나올 수 있다. 또한, 실제계와 관찰계가 다르므로, 관찰계에 따른 암호도 다양하다. 이처럼 다양한 암호의 가능성은 암호의 완성도를 높여준다.

그렇다면 왜 필요할까? 사실 암호를 만드는데 오직 1명 혹은 1개의 주체만 필요하다면, 오히려 기존의 중앙화된 시스템이 더 효율적이다. 하지만, 1명이 만들어낼 수 있는 암호에는 한계가 있다. 근본적인 무작위성도 줄어들고, 예측 가능한 방향으로 흘러갈 수밖에 없다. 그러나 여러 명이 의도하지 않게 암호를 만들어 내면 큰 다양성이 보장되고, 조작 가능성, 해킹 가능성도 보장된다.

단점은 모든 정보를 게임사에서 관리하기 때문에, 신뢰의 문제가 가장 크다고 생각한다. 또한, 게임사가 망했을 경우 그 정보가 타 게임사로 들어갈 수 있고 실질적으로 데이터를 관리하는데에도 아직 기술이 부족하다는 단점이 있다. 하지만, 시작부터 이런 의도로 만들어진 게임에다 여러 가지 제약도 묶는다면 해결할 수 있다고 생각한다. 시작부터 이런 의도로 만들어졌지만, 재미까지 있다면 일거양득일 것이다.

사실 한 사람이 이처럼 다양하고, 위조를 막는 완성된 암호들을 생각해내는 것은 무리가 있다. 그러나 여러 플레이어가 자연스럽게 이동하는 좌표값을 기반으로 암호를 만든다면 가능해진다. 이야말로 의도 없이 한 문제를 해결하는 게임의 가장 큰 예가 되지 않을까?

불완전성(시나리오)



자막 2048년



(클로즈업) 사람들과 로봇, 쓰레기를 청소하는 로봇들이 보인다.

출근하는 수많은 사람과 로봇들 사이 주인공이 걷고 있다.

주인공 독백 : 신은 죽었다. 그러나 요즘엔 다른 '신' 같은 존재가 있다.



빌딩 위의 TV 스크린이 보인다. 로봇이 무언갈 발표하고 있다. 사람들은 열성적으로 듣는다(무음)

회상

자막 2041년



주인공이 고등학교 교복을 입고 있다. 학생들이 수업을 듣고 있다.

선생님 : 너희들의 감정의 주파수가 측정되는 거야. Dr 값은 사랑의 감정에 대한 값인데, (칠판에 Dr/d 를 쓴다) Dr/d 가 1보다 크면

학생들 : 사랑한다!

선생님: 이제 이 Sv 값은 두려움에 대한....

주인공 독백 : 7년 전, 감정의 값에 관한 연구가 노벨상을 탔다. 다른 물리학적

에너지를 찾으려고 한 실험에서 우연히 가까워지면 커지는 값이 발견되었고 이
가 사랑에 비례한다는 것을 밝혀졌다. 이때부터 모든 학문의 패러다임이 바
뀌게 되었고, 컴퓨터 기술 패러다임도 변화했다.



(자막) 2043년

주인공 독백 : 나의 고3 시절은 로봇의 부흥기였다.

(TV 코미디 프로그램)

개그맨 1 : 내가 애를 좋아해?

로봇 : 아닙니다.

개그맨 1 : 그럼 내가 애를 싫어해?

로봇 : 아닙니다.

개그맨 2 : 그럼 나는 애 좋아해?

로봇 : 아닙니다.

개그맨 1 : 아니 그럼 어떻게 해

로봇 : 모르겠습니다..

개그맨 2 : 애는 맨날 모른대~ (로봇 뒤통수를 한 대 때린다.)

(객석 웃음)

주인공 독백 : 개그의 소재로도 많이 나왔고, 고 3때 친구들 사이에서 로봇 따

라하기가 유행이었던 것이 기억난다.

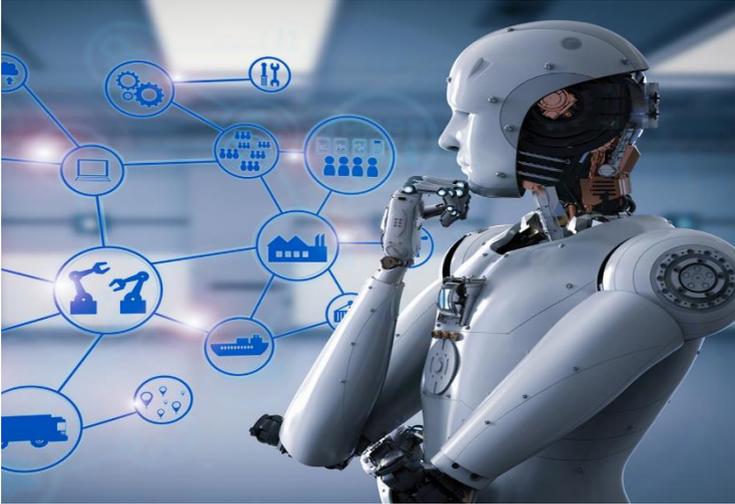


(자막) 2044년

주인공 독백 : 그렇게 난 대학에 가게 되었고,



음악에 빠지게 된다. 여자친구도 사귀고, 술도 열심히 마시고.



기술의 발전은 빠르다고 했던가... 몇 년 전만 해도 놀림받던 로봇들에게 점점 사람은 의존하고 있었다. 모든 일들이 양자역학적으로 연결되어 있다는 선행 연구를 기반으로 과학은 급속도로 발전했다. 그저 코미디에나 나왔던 로봇들이 이제 양자역학적 데이터를 통해 미래를 예측하기 시작했고 틀린 적이 없다 보니 모두가 의존하기 시작했다. 무서운 점은 그 순간에도 로봇은 더 발전하고 있었다는 것이다. 당연히 가장 먼저 묻은 것은

gettyimages®



진행자 : 신은 있습니까?

로봇 : '신'에 대한 정의가 없습니다.

진행자 : 세상을 움직이는 힘이 있습니까?

로봇 : 네

진행자 : 세상을 움직이는 생명체가 있습니까?

로봇 : 네

진행자 : 모든 것을 아는 존재가 있습니까?

로봇 : 네

이를 판단하는 메커니즘은 다른 차원이라 우리가 눈으로 확인하기는 힘들지만, 컴퓨터가 생각하는 정의에 따른 답은 언제나 나왔다. 또한, 여러 가지를 합쳐 판단하는 기능도 있었다.

진행자 : 그럼 이번에 진실담에서 낸 이 정책은 훗날 경제적으로 도움이 됩니까?

로봇 : 네

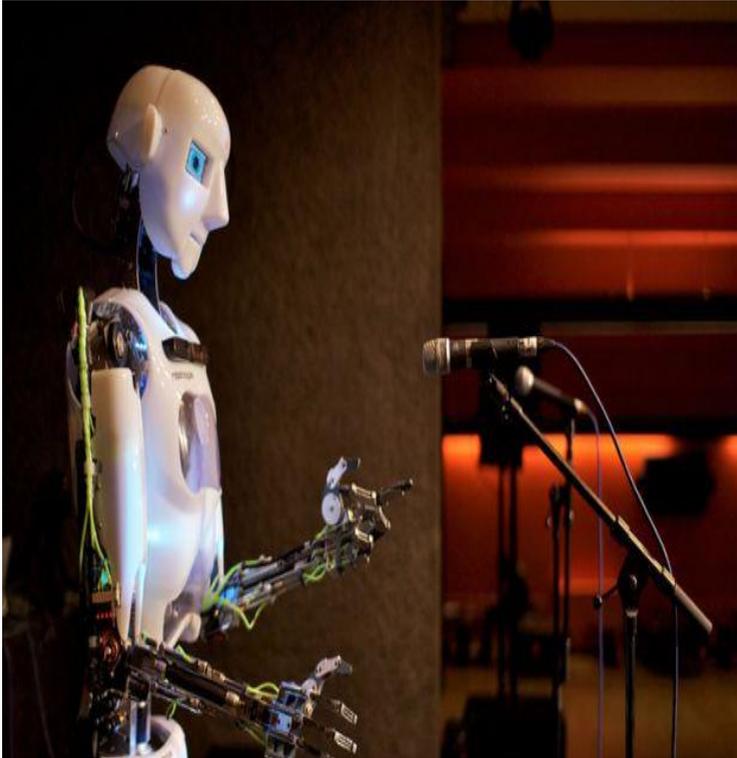
진행자 : 이 정책이 빈부격차를 과속한다는 것도 사실입니까?

로봇 : 네

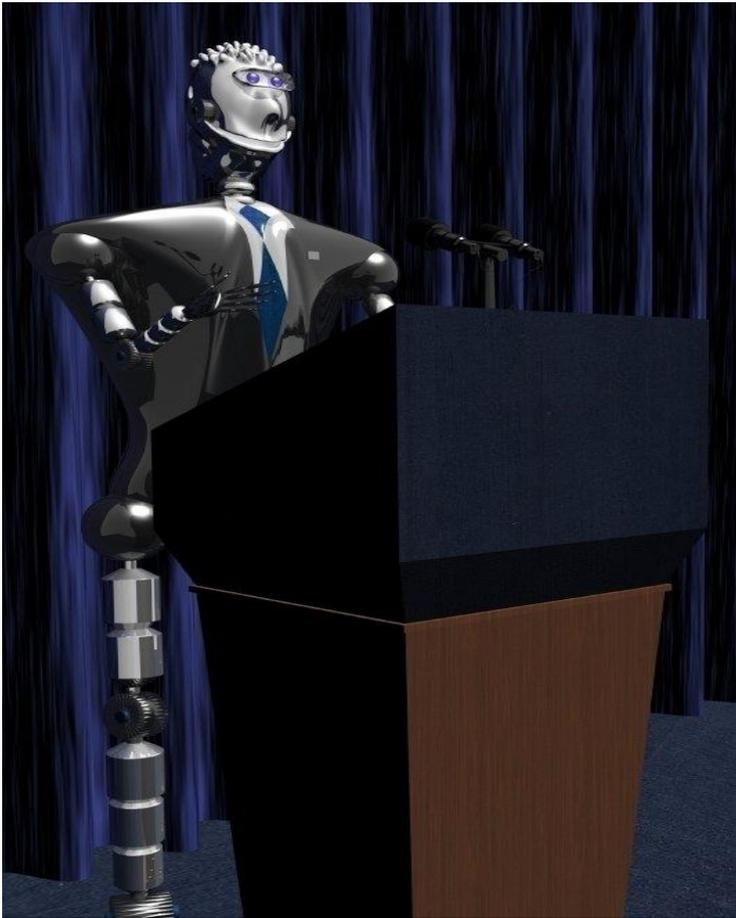
진행자 : 그럼 모든 것을 따져봤을 때, 전체 인구의 행복도가 훗날 더 높아지는 정책입니까?

로봇 : 네

물론 전체 인구의 행복도이기 때문에 다른 의견들도 많았지만 많은 사람이 양자역학적 데이터의 근거가 있는 객관적 판단을 신뢰하기 시작했다. 심지어 미래의 양자역학적 에너지의 이동도 예측할 수 있다는 점은 큰 메리트였다. 이는 결국 정치로도 이어졌다.



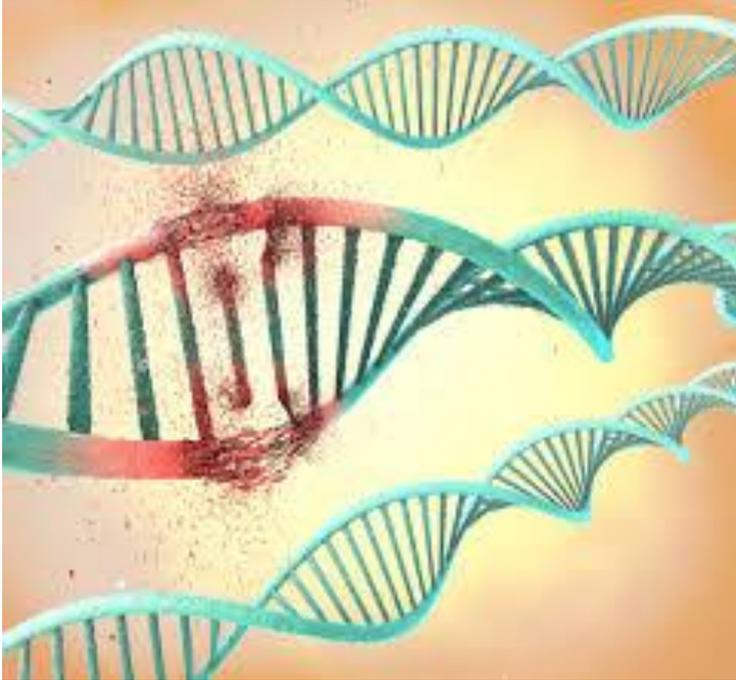
2046년 난 22살이었고, 첫 투표를 하게 되었다. 그리고 처음 찍은 대상이 로
봇이었다는 점은 아이러니했다. 그리고 결과는



당선이었다. 충격적이었다. 첫 로봇 대통령의 탄생이라니

그만큼 컴퓨터의 대답을 모두 믿는 분위기였다.

이는 결국 카스트 제도로 이어졌다.



즉 그들의 말은 이랬다. 암이란 것은 DNA에 이상이 생기면 걸리는 것인데, 자주 양자역학적 흐름에 반하게 되면 암이 생기는 것이다. 정해진 운명을 바꾸거나, 갑자기 큰 변화가 있으면(단, 운명이 그렇지 않을 때) 암이 생긴다는 것이다. 실제로 암에 걸린 사람들의 증언도 그랬고, 사람들은 운명에 따라 살아가는 것을 선택했다. 그 결과로, 대통령이 된 것이고 정책 또한 그랬다. 사람들은 자신의 운명을 받아들이기 더 쉬워졌고, 나도 그게 어려워질 것이라 생각하진 못했다. 그때까진...



사람들은 나라에서 지급하는 휴대용 로봇을 들고다니며 자신의 운명에 대해서 계속 질문을 할 수 있었고, 모두 운명을 따르며 살았다. 이를 통해 사기를 치면 어떡하냐는 초반에 반대 의견도 많았지만, 결국 자신의 건강을 해치길 원하는 사람은 없었다. 사기 칠 운명인 사람 외에는 사기를 치지 않았고, 사기를 칠 운명인 사람들마저도 운명이 정해준 대로 적정선에서 그쳤다.

생각보다 큰일은 없었고, 오히려 순응함으로 인해 개개인의 행복도는 높았다.

특히나 모든 사람의 총 행복도는 같은 값이라는 연구결과도 이에 한몫을 했다.

나의 운명은 어땠냐고?



나쁘지 않았다. 대기업에 취직하는 삶, 누군가는 부러워했고 나도 사실은 나쁘지 않았다. 자신의 운명에 맞는 일이라면 나쁜 일이 있겠나. 하늘이 도와주는 느낌이었고, 적성에 잘 맞았다.



주인공 독백 : 직장 분위기도 좋았다. 우리는 해야 할 일에 대해 시답잖은 농담을 하곤 했는데,

상사 : 동주야, 이거 너한테 시키는 게 내 운명이 해야 할 일이라나~

주인공 : 휴 제 운명은 하기 싫어도 해야 한다네요~

주인공 독백 : 사실 농담이었지만, 공허함은 늘 존재했다. 해야 할 일이라니, 인생은 선물 같은 거였는데, 과제만 잔뜩 쌓여있는 기분이었다. 기분 좋을 타이밍마저도 로봇에게 스포일러 당하기 때문에 힘들었다. 기분이 좋아도 지금이 운명적으로 내가 기분 좋을 타이밍일까에 대한 쓸데없는 고민이 생겼다.



(퇴근 후 술자리)

동료들 : 마시자~ (약자지결)

주인공 : 선배, 선배는 꿈이 뭐였어요?

선배 : 꿈? 흠....

동료 : 난 있었어! 그냥 먹고 살기, 하루에 누가 만 원씩만 줘도 살 수 있다고

뽕뽕 큰소리쳤는데....

선배 : ㅋㅋㅋ 살아보니 어떨냐, 만원 가지고 살겠더냐?

동료 : 흠.... 안 살아봐서 모르겠긴 하지만 사실 로봇이 안 정해줬으면 그렇게도 살 수 있었을 거 같은데....

선배 : 아니야, 로봇은 정해준 게 아니라 그저 운명을 가르쳐주는 것일 뿐, 넌 그렇게 못 살았을 거야. 로봇이 안 말해줬어도 그럴 운명이었을걸?

머리가 팽했다. 하긴, 운명이란 건 원래 있었을 테니. 머리가 복잡해졌다.

주인공 : 근데, 선배. 원래 운명이 있었다면 운명을 거스르는 것이 운명으로 태어난 사람도 있지 않았을까요?

선배 : 운명이 거스르는 것이 운명이라.. 그렇게 생각하면 끝도 없긴 하지. 결국엔 선택 아닐까? 넓게 보면 암에 걸리는 것도 운명일 수 있지. 어렵군

주인공 : 근데 왜 지금은 안보일까요?

선배 : 음,, 생각보다 그런 사람들 아직 많은데? 니가 안 봐서 그렇지 아직도 많아.

주인공 : 네?? 진짜요?!?

선배 : 응, 사실 나도 예전 꿈은 화가가 되는거였어. 미대에 입학했었고 졸업도 했지. 같이 붙어 다니던 3명이 있는데

(회상)

자막 2038년



선배 : 삼총사라 불렸지.. 우린 재밌었어. 그냥 그림 그리는 것도 재미있었고,
다 같은 예고 출신이었거든. 다 같이 화가가 되는 건 당연했고, 의심해본 적도
없어. 늘 우리의 미래를 생각하면서 농담도 하고..... 근데.....

(회상)

자막 2046년 대통령 선거 후



선배 친구 1 : 야, 너희는 뭐 나왔냐?

선배 : 회사원이란다. 말이 되냐? 넌

선배 친구 1 : 난 화가래. 근데 그것도 배고픈

선배 친구 2 : ㅋㅋㅋㅋㅋ 배고픈 화가랙ㅋㅋㅋ 잘 어울린다 너랑

선배 : 휴.. 회사원보다는 낫지.. 아는 것도 없는데 뭐 회사원이냐... 넌 뭐 나왔어?

선배 친구 2 : 난 음식점한대... 휴 뭐 음식점이냐.. 미술이나 해야지

선배 : 미쳤어? 암 걸린대잖아

선배 친구 2 : 암이 대수냐. 하고 싶은 거 못하는게 암이지.

선배 친구 1 : 그래도...

선배 친구 2 : 몰라~ 난 맘 정했어.

선배 : 진짜로?

(다시 술자리)

선배 : 개는 아직 그림 그려. 대장암 4기라나...

주인공 : 진짜요?

선배 : 그래도 좋대. 몰라 자기 좋다는데 해야지 뭐. 원래 그런놈이었어.

주인공 : 재밌대요? 미술도 로봇이 먹었잖아요,

선배 : 응, 사람들이 감동을 느끼는 채도, 사람들의 마음이 따뜻해지는 색 온

도, 소름 돋는 색 조화 같은 것들도 연구가 끝났는데 어떻게 이겨.

주인공 : 그러니깐요. 정해진 사람들 빼고는 안하던데

선배 : 근데 재밌대 아직. 아직 못 찾은 색 조화가 있대나 뭐라나. 암보다도

중요한 게 있나보지. 넌 꿈이 뭐였냐?

주인공 : 저는..



주인공 독백 : 말이 쉽게 나오지 않았다. 음악이라.. 같이 음악하던 내 친구들은 다 자기 자리를 찾아갔다. 그래서 그게 당연한 줄 알았고, 아무리 생각해도 당연한 게 맞았다. 당연한 게 맞는데.. 왜 기분이..

동료 : 너 취했냐?

선배 : 많이 마셨다. 하나 둘,, 몇 병이냐... 휴 술술 정리하자. 내일 출근해야지



주인공 독백 : 그 날은 잠이 오지 않았다. 그냥 안된다는 걸 알면서도 흔들렸다.
분명히 아닌걸 아는데..... 모든 사람들의 행복함은 같은 값이라고 그랬다. 근
데 왜 난 행복하지 않지? 나의 ADP 로봇을 켜다.

주인공 : 야 ADP

ADP : 네, 주인님.

주인공 : 난 왜 안 행복하지?

ADP : 잘 모르겠습니다.

주인공 : 음악 때문이냐?

ADP : 잘 모르겠습니다.

주인공 : 휴,,, 됐다. 자야지.

(이불을 덮어 쓴다.)

(이불을 다시 걷으며)

주인공 : 야, 혹시 내가 음악하면 성공할 수는 있냐?

ADP : 제시한 명제는 참입니다.

주인공 : 진짜? 내가?

ADP : 네 주인님.

주인공 : (기뻐하는 것도 잠시 슬픈 표정을 지으며) 근데 그럼 암이 걸리겠지...?

ADP : 제시한 명제는 참입니다.

주인공 : 그치...?...

주인공 독백 : 음악도 미술과 비슷하게 로봇에게 먹혔다. 오히려 미술보다 먼저 먹혔는데, 사람들이 소름을 느낄만한 주파수와 눈물샘을 자극하는 주파수 등 감정 값의 선행 연구에 이은 후속 연구로 인해 이미 의미 없는 분야가 되어버렸다. 가수도 많이 줄었고, 그마저도 목소리 음파 자체가 감정을 잘 변화시키게 타고난 사람들뿐. 최근 가장 유행하는 오디션 프로그램은 청중들의 소름과 감정값의 변화를 누가 가장 많이 유도해내는지 뽑는 오디션이다. 보고있으면 소름이 너무 돌아 놀랄 정도이다.

(휴대폰을 본다.)



(화면 속 프로그램 광고) 올해도 돌아왔습니다! 누가 더 많이 감정 값을 변화시킬 수 있을까요? 평소에 감정 값을 잘 변화시키는 목소리를 가지고 있다! 하는 사람들은 아래 번호로 지원하세요! 일반인도 괜찮습니다!

(휴대폰을 덮고 눕는다.)

주인공 독백 : 몰라. 출근해야 돼.... (페이드 아웃)



주인공 독백 : 오늘도 출근을 한다. 재미있는 일이 많이 있길..... (눈을 돌려 지하철 전광판을 본다.)

(당신도 주인공이 될 수 있어요! 오디션 광고 전광판)

주인공 : 휴... (한숨을 쉬고, 길을 간다.)



주인공 독백 : 오늘은 일이 잘 안된다. 어제 술을 많이 마셔서 그런가 집중도 안되고... 다행히 일은 적다.

선배 : 어제 집은 잘 들어갔냐?

주인공 : 네...

선배 : 그래 알바라, 점심이나 먹자 (뒤를 돈다.)

주인공 : (선배를 잡으며) 선배!

선배 : 응? 왜?

주인공 : 저 오늘만 연차 내도 될까요?

선배 : 응??? 갑자기? 너 연차 남았어?

주인공 : 네, 아직 이틀이요. 오늘 별로 일 없죠?

선배 : 음.. 어제 좀 많이 해놔서 오늘 할건 별로 없지..? 오늘 편한 날인데

갑자기 왜?

주인공 : 어디 갈 데가 생겨서요. (일어나며) 선배 내일 봐요. 연차 처리 좀

해주시구요. (급하게 옷을 챙겨 나간다.)

선배 : ???? 그래 내일 봐...? ???



주인공 독백 : 무작정 뛰었다. 생각을 정리해보려고 해도 잘 안 됐다.

ADP : 주인님. 현재 경로는 자칫 위험할 수도 있고, 심한 경.. (뚝)

주인공 독백 : 들리지 않았다. 일단 견고 싶었다.

얼마를 걸었을까... 결국 도착한 곳은



주인공 독백 : 방송국이였다.

난 홀린 듯 안으로 들어갔고, 방송국 안에는 오디션이 진행중이었다,



PD : 다음 참가자~ (나를 발견하고) 재는 뭐야? 여기 들어 오시면 안돼요.

심사위원 레전드 가수 1 : 들어오라 그래. 요즘 같은 세상에 이렇게 온 건 운명
이지 뭐.

심사위원 레전드 가수 2 : 그래도 선배님...

심사위원 레전드 가수 1 : 됐어 들어오라 그래. 내가 책임진다 그러고.

주인공 독백 : 정신을 차려보니 난 그들 앞에 있었다.

심사위원 레전드 가수 1 : 그래, 자네 이름이 뭐가?

주인공 ; (얼이 빠진 채로)

심사위원 레전드 가수 1 : 허허 이름이 없나보군. 준비해온 노래는?

주인공 : 그게....

심사위원 레전드 가수 1 : 이름도 없는데 노래도 없어? 기똥찬 친구구만. 그럼

평소에 좋아하는 노래 없어?

주인공 독백 : 예전에 공연에서 불렀던 노래가 생각났다.

주인공 : ***의 불완전성입니다.

심사위원 레전드 가수 1 : 불러봐 (PD는 못마땅한 표정으로 절레절레한다.)





(노래 끝, 멍뚱뚱 서있는 주인공)

심사위원 레전드 가수 1 : (일어나서 박수) 자네는 노래한 지 얼마나 됐나? 왜 처음 보는 거 같지?

주인공 : 예전에 했습니다.. 한.. 4년 전..?

심사위원 레전드 가수 1 : 예전에? 근데 왜 갑자기 음악을 다시 하는건가?
지금은 직업이 뭐가?

주인공 : 회사원입니다. 다시 하는 건 아니고, 그냥 길가다..

PD : 이제 내보내

심사위원 레전드 가수 1 : 잠시만 기다려보게. 자네는 그러면 운명이 가수인건
가 회사원인건가?

주인공 : 회사원입니다. 사실 지금도 근무해야 하는데.. 모르겠습니다. 오게 됐
네요..

심사위원 레전드 가수 3 : 이 친구의 노래는 좋긴 했지만, 보시다시피 여기 감
정 측정 기계 보시면 별로 높지는..

심사위원 레전드 가수 2 : 네 선배님. 그래도 롤이란 게 있지 않습니까? 아쉽
지만 여기까지 하는게....

심사위원 레전드 가수 1 : 내가 슈퍼 패스를 쓰겠네. 오랜만에 듣는 좋은 목소

리아, 사람 냄새랄까

심사위원 레전드 가수 2, 3 : 그치만...

심사위원 레전드 가수 1 : 자네 이리로 와보게

(주인공이 다가간다.)

심사위원 레전드 가수 1 : 자네 꼭 예전에 내 친구를 닮았어. 참 잘하는 친구였는데. 음악을 좋아하던 친구였지. 내 코를 찡하게 하던.

(주인공이 쳐다본다.)

심사위원 레전드 가수 1 : 그 친구는 작년에 죽었다네. 후두암에도 불구하고 그가 부르는 노래는 사랑이었고 내뱉는 가사 하나하나는 예술이었지.

(주섬주섬 뭍 꺼내며) 이걸 그 친구가 죽기 전에 나한테 준 앨범이야. 내가 제일 좋아하는 앨범이지. (불완전성을 부른 ***의 앨범이다.)



주인공 : 감사.. 감사합니다...

심사위원 레전드 가수 I : 본선에서 복세. 몸조리 잘하고 씩씩하게 어깨 펴고
와.

주인공 : 네.. 네 감사합니다.

PD : (못마땅한 얼굴로) 이름이 뭐예요.

주인공 : 이 동주.. 입니다.

PD : 10월 13일까지 여기로 오세요. 아 참, 그때 진짜 청중들 감정변화값으로
평가하니 명심하십시오.

주인공 : 네.. 네..

주인공 독백 : 그렇게 내 본선 진출은 확정되었다. 그래서 어떻게 되었냐고?

(화면 전환)

자막 2089년



(평화로운 집에 로봇이 청소를 하고 있다.)

손녀 : 할아버지! 아, 민호 씨인 좀 받아달라니까!!!

주인공 : 너무 유명한 가수가 된 탓에 손녀에게 시달리고 산다.

힘드냐고? 힘들다.....

아, 암은 어떻게 됐냐고?

그건 나도 모르겠다. 운명이 정해주겠지. 아직은 발견되지 않았다.

오늘은 내가 나갔던 오디션의 1차 예선 날이다.

난 당연히 심사위원으로 뽑혔다.

그리고 왠지 오늘은...

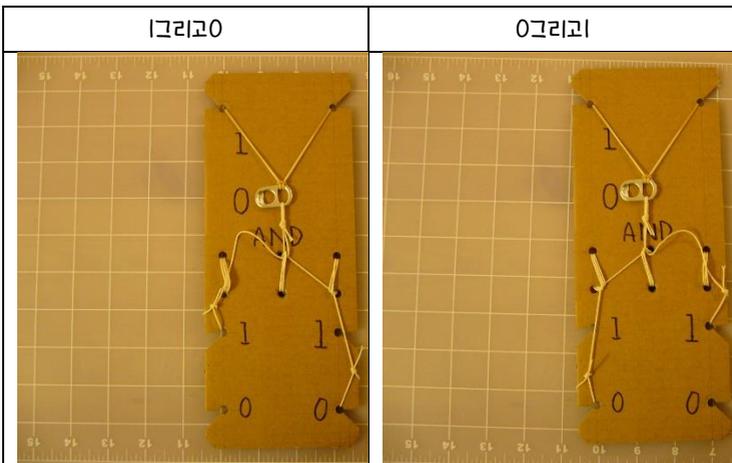


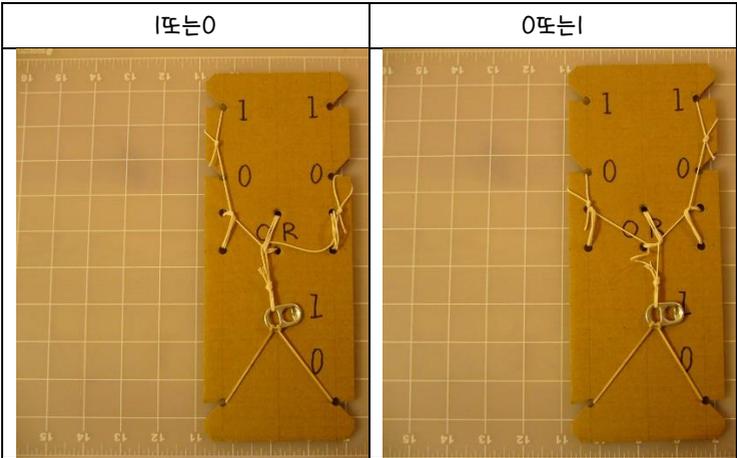
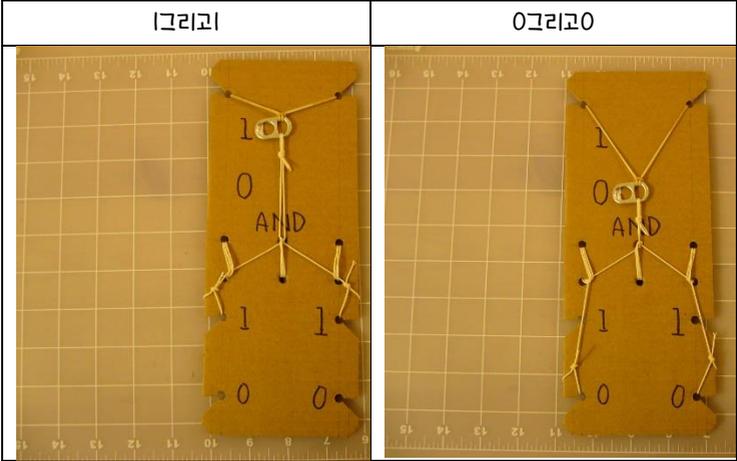
오디션장에서 나와 닮은 소년을 만날 것만 같다.

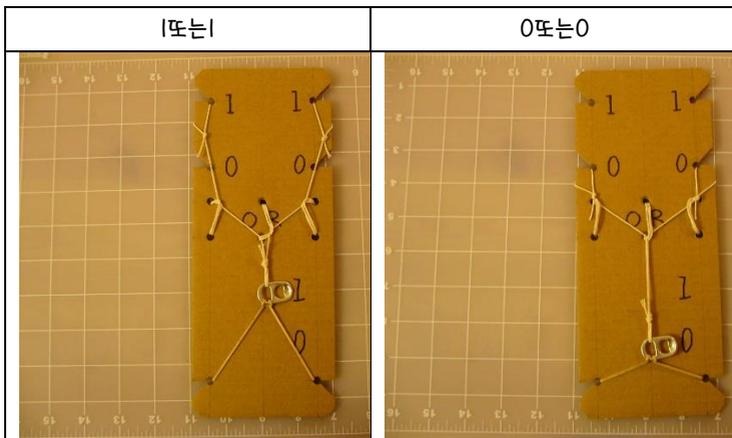
부울안경

부울은 그리고-또는-아닌의 연산자를 통해, 논리회로의 작동방식을 나타냈다. 그리고 이 세 가지 부품만으로 이 세상의 모든 컴퓨터를 만들 수 있다. 그렇다면 우리의 일상에서 그리고-또는-아닌의 연산으로 바라볼 수 있는 현상을 찾는다면, 이 현상으로 컴퓨터를 구성할 수 있지 않을까라는 궁금증이 생겼다. 이 궁금증을 가지고 우리 주위에서 그리고-또는-아닌으로 표현할 수 있는 현상들을 찾아보기 시작했다. 그러다가 문득 뜨개질을 하다가 남은 실이 눈에 들어왔다. 이 실로도 컴퓨터를 만들 수 있을까? 실을 당기고 놓고 하는 모습은 전기 회로의 모습과 닮아 있는 듯 했다. 여기에 착안하여 실에 대한 가능성을 느꼈고 어떻게 하면 실로 부울 연산자를 표현할 수 있을지 고민에 빠졌다. 여러 시행 착오를 겪은 후, 실과 박스 조각, 캔의 뚜껑, 고무줄이라는 집에서 쉽게 찾을 수 있는 간단한 재료들로 그리고-또는-아닌을 표현하는 데 성공할 수 있었다.

우선 최종적으로 완성된 모습은 다음과 같다.







끈을 잡아당겨 어느 위치에 놓느냐에 따라 캔 뚜껑이 표시하는 값(1,0)이 달라지는 것을 확인할 수 있다. 즉 이 간단한 모형은 그리고-또는-아닌의 연산을 보여주는 것이다('또는'연산의 모형은 아직 제작중에 있습니다). 그리고 쉽지는 않겠지만 캔 뚜껑에 실을 추가로 연결하여 다른 모형의 입력과 잘 연결시킨다면, 조금 더 난이도 높은 연산도 수행 가능할 것이며 모형을 정교하게 설계한다면 가능성은 무궁무진 할 것이다!! 실과 박스, 캔 뚜껑, 고무줄 만으로 컴퓨터를 구현할 수 있는 것이다!!

그리고 재료가 간단할 뿐만 아니라 제작 방법도 어렵지 않았다. 다음은 제작 과정 중에 찍은 사진이다. 다음 사진을 참고하여 박스에 구멍을 뚫고, 구멍에 끈을 통과시키기만 하면 된다.



알고리즘의 최종 목적지를 향해

"소프트웨어(튜링기계)로 만드는 문제 푸는 방도. 자동으로 돌릴 수 있는 문제 풀이법. 기계적인 방식으로 자동화되는 문제 풀이법." 책 '컴퓨터과학이 여는 세계'에서 알고리즘이라는 단어를 정의한 문장들이다. 이 문장들을 조금 곱씹어 보며 한 가지 궁금증이 생겼다. 기술이 계속 발전하면서 우리 삶에서의 모든 문제풀이를 기계적인 자동화된 풀이로 나타내는 알고리즘이 언젠가는 등장하지 않을까? 이런 알고리즘은 어떠한 모습으로 우리에게 다가올까? 우리의 삶은 어떻게 변할까? 이런 호기심을 가지고 나는 알고리즘을 다루는 책을 마구잡이로 읽기 시작했다.

호모데우스(유발 하라리), 미래사회보고서(유기운) 등 미래를 다루는 여러 책들을 훑어보았고, 모두 흥미로운 내용을 담고 있었지만 그 중 나의 호기심을 해결 해주었던 것은 '대량살상 수학무기'라는 책이었다. 나는 이 책에 대한 내용으로부터 이야기를 시작하고자 한다. 우리는 수업시간에 알고리즘의 두 가지 한계를 배웠다. 문제의 복잡도가 너무 커 알고리즘이 실질적으로는 해결하지 못하는 문제. 그리고 알고리즘이 태생적 한계로 절대 풀지 못하는 문제. 이 중 전자는 양자 알고리즘의 등장으로 인해 부분적으로 해결되고 있다. 그리고 양자 알고리즘과 같이 전자를 부분적으로 해결해주는 또 하나의 기동이 있다. 이는 바로 이 책의 주제인 '빅데이터'이다.

빅데이터는 대부분의 사람들이

책의 내용 소개 → 대량 살상수학무기 → 빅데이터의 문제점, 대량살상 수학무기 특징 → 우리는 이러한 빅데이터의 부조리 속에서 완전화된 알고리즘의 세계를 엿볼 수 있는 것이다. 이는 매우 귀중한 기회임이 분명하다.

해결방안은 무엇일까. 저자의 의견: 도덕적 상상력?? 이게 뭐지→ 책 도덕적 상상력을 읽었고 내용 소개와 나의 생각

'도덕적 상상력'이란 현실 세계에서 벌어지는 다양한 갈등과 분쟁을 조정하는, 갈등과 분쟁의 해결책이 단순히 한 쪽의 일방적인 희생을 요구하지 않는, 우리에게 필요한 변화의 의미와 내용을 장기적인 시각에서 분석하는, 그래서 현세대는 물론 미래세대까지 영향을 미치는 변화를 상상하고 실현하는 '상상력'을 저자는 '도덕적'이라고 주장한다.

내가 만든 튜링 기계

서론

'튜링 머신'이란, 간단히 말해서 앨런 튜링이 만든 '보편만능의 기계'이다. 이것은 컴퓨터의 청사진이 되었고 현재 우리의 주변의 많은 기계들 또한 '튜링 머신'에 속한다고 할 수 있다.

그런데 진짜 '보편만능의 기계'라 함은, 내가 원하는 작업을 내가 원하는 대로 다 시킬 수 있어야 하는 것 아닌가? 현실적으로 우리의 컴퓨터는 내가 원하는 작업이 있다면 많은 것을 해줄 수 있는 것은 사실이다. 하지만 나는 이번 기회에 진짜 움직이는 나의 튜링 머신을 만들어 보고 싶었다. 내가 원하는 대로 움직이는 내가 원하는 대로 이것저것 입혀서 조종할 수 있는 나만의 튜링 기계.

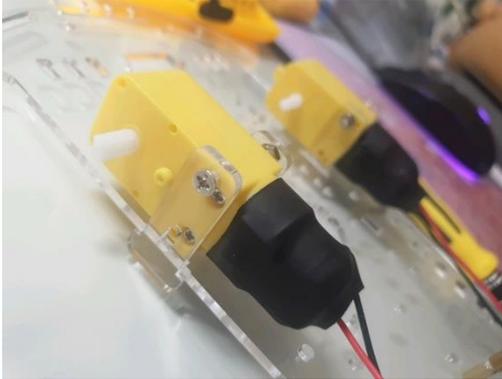
그래서 내가 정한 것은 RC카였다. 진짜 내가 원하는 대로 조종하면서도 스스로 읽고 어떤 일에 대해 수행능력을 출력할 수 있는 RC카.



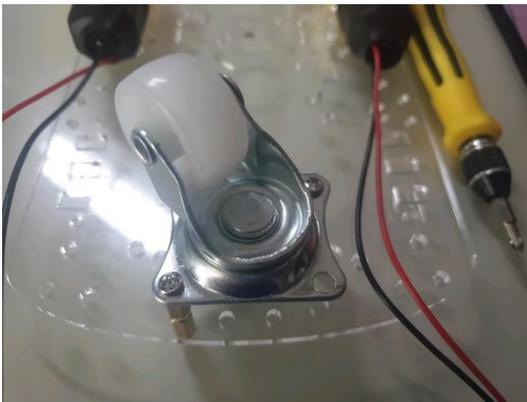
준비물은 모두 준비되었다. 모든 것의 베이스가 되어줄 플라스틱 베이스. 모든 것의 주인공이라고도 할 수 있는 튜링 기계 역할의 아두이노 우노 보드, 그를 보조해줄 모터 드라이버 쉴드, 직접 바닥의 검은 선을 읽고 판단하게 해줄 라인 트레이서 모듈, 내가 원하는 대로 움직이게끔 나의 의견을 전달하고 입힐 수 있게 도와줄 우노 USB 케이블. 그 외에도 배터리 홀더, 나사, 바퀴, T헤드 등의 필요한 재료들을 준비했다.

본론

만들기에 돌입해보자!

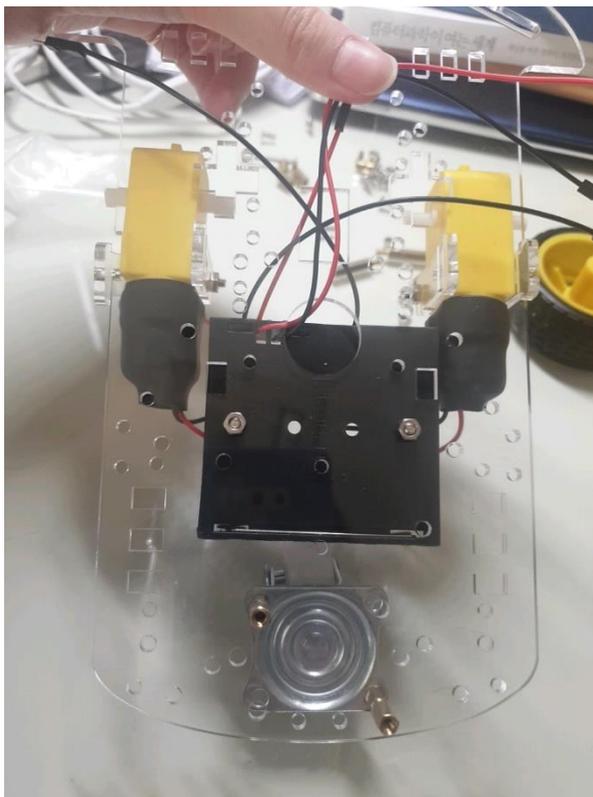


우선, +헤드와 DC모터를 연결해 단단히 고정시켜준다. 여기에 바퀴를 꽂아 돌아갈 것이기 때문에 더욱 유의하여 단단히 고정시켜준다.

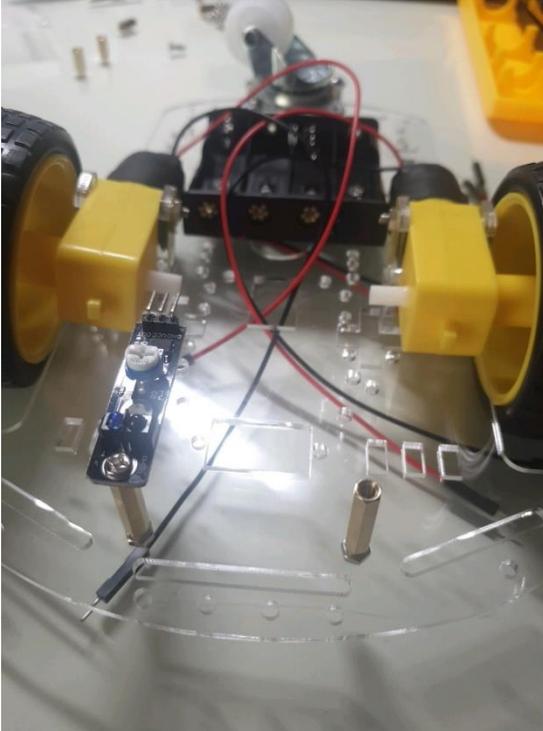


바퀴는 세 개 일 때 가장 안정된 무게중심을 가지게 되므로 뒤쪽으로 보조바퀴를 달아준다. 이는 서포트를 이용해 길이를 조절하여 볼트와 너트를 이용해

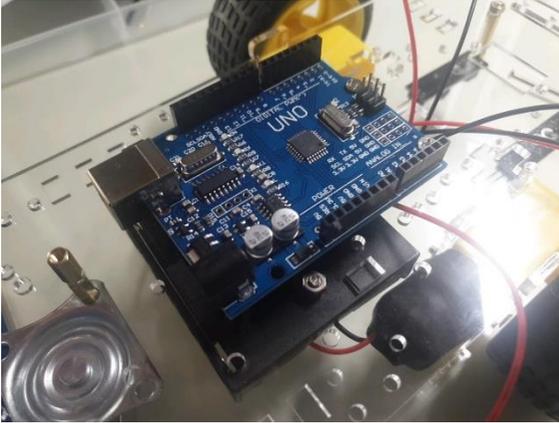
고정해준다.



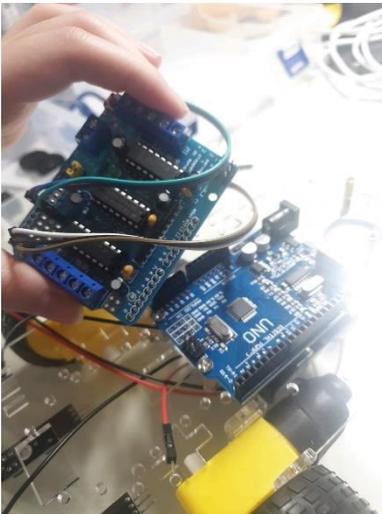
나의 RC카의 에너지를 담당해줄 배터리 홀더를 달아준다. 이때, 배터리 홀더의 전선을 나중에 연결할 때 조금 더 편하게 할 수 있도록 위로 빼 준다.



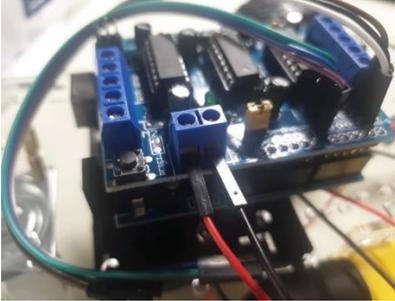
RC카로 정보를 읽을 수 있도록 눈의 역할이 되어줄 라인트레이서 모듈을 장착해준다. 이 또한 길이를 조절하기 위해 서포트와 볼트를 이용해 달아준다.



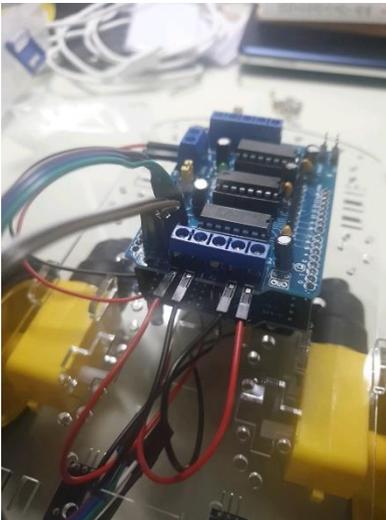
내가 만드는 튜링 기계의 핵심이자 심장이라고 해도 과언이 아닌 아두이노 우노 보드를 장착해준다.



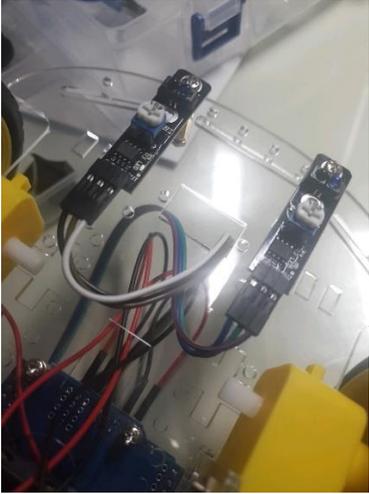
앞서 장착한 아두이노 우노보드에 모터 드라이버 쉴드를 장착해준다. 이때, 핀 결합에 유의하며 결합시켜준다.



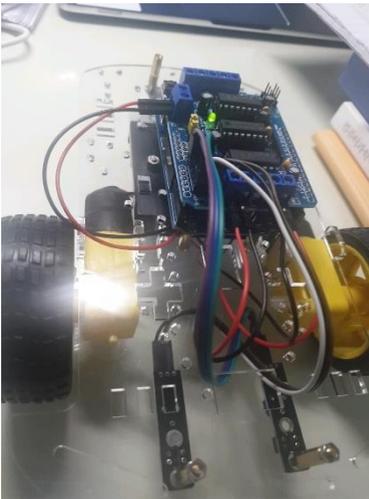
모터드라이버 쉴드의 외부전원 터미널에 배터리 홀더의 전원선을 연결해준다. 이때 혹시나 결합할 때 잘 들어가지 않거나, 쉽게 다시 빠지는 상황이 생긴다면 모터드라이버의 터미널을 드라이버를 이용해 전선이 상하지 않게 결합해주고 빠지지않게 고정해준다.



이번엔 모터드라이버의 터미널을 아까와 같은 방식으로 열어 M3터미널에는 왼쪽모터의 선을 연결해주고, M4터미널에는 오른쪽 모터의 선을 연결 해준다.



라인트레이서에 아두이노 우노 보드의 선을 각각의 역할에 맞추어 유의하며 연결 해 준다.



건전지의 결합 후 제대로 모터드라이버의 LED등에 불이 들어온다면 성공적!

```
#include <SoftwareSerial.h>
#include <AFMotor.h>
AF_DCMotor motor_L(3);
AF_DCMotor motor_R(4);

void setup() {
  Serial.begin(9600);
  Serial.println("Computational Civilization Team 10!");

  // turn on motor
  motor_L.setSpeed(150);
  motor_L.run(RELEASE);
  motor_R.setSpeed(150);
  motor_R.run(RELEASE);
}

void loop() {
  int val1 = digitalRead(A0);
  int val2 = digitalRead(A5);

  if (val1 == 0 && val2 == 0) {
    motor_L.run(FORWARD);
    motor_R.run(FORWARD);
  }
  else if (val1 == 0 && val2 == 1) {
    motor_L.run(FORWARD);
    motor_R.run(RELEASE);
  }
  else if (val1 == 1 && val2 == 0) {
    motor_L.run(RELEASE);
    motor_R.run(FORWARD);
  }
  else if (val1 == 1 && val2 == 1) {
    motor_L.run(RELEASE);
    motor_R.run(RELEASE);
  }
}
```

여기 까지가 단순한 장치의 조립이었다면, 이제부터는 정말로 이 장치에 지혜를 사용해 이를 내가 원하는 '만능기계'가 될 수 있도록 소프트웨어를 넣어주는 것이 남았다. 사실 여기서부터가 진짜 시작이라고 봐도 무방하다. 이때에는 '아두이노 IDE' 프로그램을 이용하여 코딩을 진행해주었다. 우선 나는 모터가 제대로 작동하는지 확인하기 위해 M4에 연결된 모터가 정 회전, 정지, 역회전을 초속 동작하고 다음으로 이어서 M3에 연결된 모터가 정 회전, 정지, 역회전을 초속 동작할 수 있게 테스트 해보았다. 연결은 성공적! DC모터가 제대로 연결되었음을 확인한 후에는 정말 본격적으로 내가 원하는 읽고 움직이는 기계로 만들어보았다.

```

Motor_driver_D93d | 아두이노 1.8.12
이동 편집 스키마 도움말
#include <Servo.h> // include servo library
Servo servoLeft; // declare left and right servos
Servo servoRight;

void setup() // Built-in initialization block
{
  pinMode(11, OUTPUT); pinMode(12, OUTPUT); // Left IR LED & Receiver
  pinMode(6, INPUT); pinMode(8, INPUT); // Right IR LED & Receiver

  servoLeft.attach(13); // Attach left signal to pin 13
  servoRight.attach(12); // Attach right signal to pin 12
  Serial.begin(9600);
}

void loop() // Main loop motor-control
{
  int leftIR = digitalRead(11, 72000); // Check for object on left
  int rightIR = digitalRead(12, 72000); // Check for object on right

  Serial.println(leftIR);
  Serial.println(rightIR);

  if((leftIR == 1) && (rightIR == 1)) // Both sides see table surface
  {
    servoWrite(100, 200); // Forward 20 milliseconds
    delay(100); // Wait 100 ms
  }
  else if(leftIR == 1) && (rightIR == 0) // Left OK, drop-off on right
  {
    servoWrite(100, 100, 200); // Left for 375 ms
    delay(100); // Wait 100 ms
  }
  else if(rightIR == 0) && (leftIR == 1) // Right OK, drop-off on left
  {
    servoWrite(100, 100, 200); // Right for 375 ms
    delay(100); // Wait 100 ms
  }
  else // Drop-off straight ahead
  {
    servoWrite(100, -100, 200); // Backward 200 ms before retry
    delay(100); // Wait 100 ms
  }

  int leftServoPin = 13; // leftServoPin, long frequency
  int rightServoPin = 12; // rightServoPin, long frequency
  delay(4); // Wait 4 ms
  int lr = digitalRead(leftServoPin); // IR receiver -> IR variable
  delay(1); // Down time before check
  return; // No return, 0 return
}

void servoWrite(int speedLeft, int speedRight, int maxTime)
{
  // speedLeft, speedRight ranges backward slower stop slower
  // forward
  // -200 = 100, ..., 0, ..., 100 = 200
  servoWrite(servoWrite(1500 - speedLeft));
  servoWrite(servoWrite(1500 - speedRight));
  if(maxTime == 1) // If maxTime == 1
  {
    servoWrite(servoWrite(1)); // Stop servo signals
    servoWrite(servoWrite(1));
  }
  delay(maxTime); // Delay for maxTime
}
Motor_driver_D93d
아두이노 IDE
아두이노 IDE

```

이것은 라인트레이서를 이용해 RC카가 스스로 검은색 선을 구분하고 그를 따라가게끔 하는 소프트웨어이다. 이를 이용하여 실제로 RC카가 어떻게 움직이는지 확인해보았다.

(이후 블루투스 연결 및 조종부분 추가 예정)

결론

이전에 '컴퓨터과학이 여는 세계' 수업에서 교수님께서서는 튜링이 사실 사람들이 이렇게까지 치켜세울 정도의 천재는 아니었을 수도 있다는 말씀을 하셨었다. 이번 기회에 내가 직접 나만의 튜링 기계를 만들면서 느낀 점은 겉보기와 달리 생각보다 정말 별거 아닌 것이라는 것이었다. 이렇게 아두이노를 활용해 내가 원하는 대로 움직이는 튜링 기계를 만든 것처럼 뭔가 더 이것저것 만들어 보고싶다는 생각을 했다. 교수님께서 소프트웨어는 지혜로 짜는 것이라는 말을 해 주신 것처럼 나도 지혜를 통해 다른 사람들에게 흥미로운 주제의 소프트웨어를 짤 수 있는 사람이 되고 싶다고 생각하게 되었다. 이후엔 정말 내가 처음으로 만드는 나만의 튜링 기계도 만들어 보고싶고, 수업시간에 배웠던 NP클래스의 문제에도 도달할 수 있는 프로그래밍을 내가 직접 해 보고싶다고 느꼈다.

가르침처럼 혹독한 가르침은 없다

민주와 홍성은 사귀지 1주년을 기념하며 해외여행을 떠나기로 한다. 어디를 갈지에 대해 고민하던 도중 홍성은 꼭 가보고 싶은 곳이 있다며 자신에게 여행 코스를 맡기라며 자신감 넘치게 이야기한다. 그에 민주는 알겠다며 여행준비를 시작한다. 날씨도 좋고 여행을 가기에 가장 최적인 날 그들은 여행을 떠나게 된다. 화창한 하늘이 그들의 여행을 함께 축하해주는 듯 했다.

오빠 나 진짜 너무 설레! 이렇게 여행오니까 너무 좋다 ㅎㅎ

근데 오빠 영국도 좋고 맨체스터도 좋은데 말이지..

응? 왜 뭐 불편한거라도 있어? 어디 더 가고싶은 곳이 있었어?

아니아니 그런게 아니구 왜 영국에 많고많은 여행지중에 맨체스터부터 꼭 가야한다고 이야기했던 이유가 따로 있는지 궁금해서!

흠... 또 축구때문이지?!

앗 물론 내가 축구를 좋아하기도 하고 맨체스터하면 축구를 먼저 떠올릴 수 있겠지만, 이번엔 아니야! 나 이번에 맨체스터를 꼭 들르고 싶었던 건 가고싶은 곳이 있어서 그랬어 ㅠ ㅠ

아 진짜?? 어딘데??

사실.. 이번기회에 꼭 'Alan Turing Memorial'에 둘러보고 싶었어 ㅎㅎ

음.. 대충 앨런 튜링이라는 사람을 추모하는 곳인건 알겠는데.... 앨런튜링이 누구야?

앨런튜링은 컴퓨터의 청사진을 처음으로 발견한사람이야 니가 없이는 못사는 컴퓨터!

헉 진짜?? 그럼 앨런튜링은 세계최초의 엔지니어나 프로그래머 막 그런거였나 보다 그치??

틀렸어 ㅋㅋ 앨런튜링이 '튜링기계'를 처음 세상에 소개시킨건 컴퓨터를 만들려고 한게 아니었어. 애초에 앨런튜링은 '컴퓨터'에 주가 아니라 "계산가능한 수에 대해서, 수리명제 자동생성 문제에 응용하면서"에서 나온 예시의 기계일 뿐이었거든

앵 그럼 얻어걸린거야??? 그럴수가 있어?? 근데 오빠 아까말한 계산하는 수 ...? 그 논문은 뭐야? 무슨 논문인데 그런게 예시의 기계로 나와??

아 그건 "자동으로는 수학의 모든 사실들이 만들어질 수 없다."¹ 라는 사실을 괴델이 이미 증명했는데, 대학생이던 앨런 튜링이 다시 한번 증명에 대해 담백하게 증명한거야 그 증명과정에서 사용된게 튜링의 '자동기계장치' 즉, 튜링기계야

아항 근데 그 튜링기계가 증명하는데 어떻게 쓰였는데?

튜링기계라는게 테이프,머리,상태 기록기, 유한한 표. 이렇게 네가지 부품으로 이루어진 기계인데, 앨런튜링이 그 기계로 표현할 수 있는 명제를 과감하게 '기계적'으로 설명 가능하다고 정의한거지 그래서 그 부품들로 어떻게 조합해서 만들더라도 수학의 모든 사실들을 알아내는 기계를 만드는 것은 불가능하다고

¹ <https://ropas.snu.ac.kr/~kwang/memo/turing1935.pdf> '튜링의 1935년' 이 광근 서울대 컴퓨터공학부 교수

증명을 한거야.

음 근데 그 네가지 부품은 어떻게 사용된거야? 테이프, 머리, 상태 기록기, 유한한 포로 만든 기계라니.. 상상이 잘 되지않는데?

여기서 테이프는 연속한 단위구간들로, 왼쪽 혹은 오른쪽으로 임의적으로 확장될 수 있는거야. 이제 이 테이프에 이 기계가 머리부분을 통해 어떤 작업을 할 것인지를 정하게 되는거지 그리고 상태 기록기가 이 기계의 상태 중 하나를 기록하는 역할을 하고, 유한한 포는 특정한 상태에 있는 기계가 어떠한 기호를 읽을 때 어떠한 행동을 취해야하는지를 지시해주는 역할이야. ²

오호라 오빠 근데 혹시 그 논문이 발표된건 언제야? 컴퓨터가 세상에 나온지 그렇게나 오래됐어??

아 그건 1936년 이야 사실 니가 생각하는 그런 현대적인 컴퓨터는 제2차 세계대전의 직전부터 대전 기간중에 발전했어 사실 전자회로가 기존의 존재하던 계산 기계의 기계식 연산장치를 대체한다던가 디지털 회로가 아날로그 회로를 대체하는 등의 변화는 연대에 따라 조금씩 이뤄져서 진짜 명확하게 '최초의 컴퓨터'를 정의 하기는 쉽지않지만 대충 1930년대 후반쯤에 컴퓨터가 발전했다고 생각하면 될듯하네 ㅎㅎ³

2

https://ko.wikipedia.org/wiki/%ED%8A%9C%EB%A7%81_%EA%B8%B0%EA%B3%84 '튜링기계' 위키피디아 검색

3

<https://ko.wikipedia.org/wiki/%EC%BB%B4%ED%93%A8%ED%84%B0%EC%>

근데.. 처음엔 튜링기계가 컴퓨터의 청사진이 됐다고 그랬잖아 근데 나 아직도 튜링기계랑 컴퓨터의 연관성을 잘 모르겠어

흠 아까 튜링기계의 부품이랑 작동원리에 대해 간단히 설명해줬잖아? 오늘날의 컴퓨터는 튜링이 정의한 기계 부품들이 고스란히 구현되어있다고 봐도 무관해! 부품들이 테이프와 머리는 메모리칩, 상태기록포는 입출력 장치, 작동 규칙 포는 중앙처리장치로 발전한거야. ⁴

오.. 그럼 우리는 아직 튜링기계에서 그렇게 발전하지못한거야? 과학은 점점 더 빠른 속도로 발전한다더니 아직 거의 100여년이 지나는 동안 그대로인거네?

ㅎㅎ 좋은 질문이야. 아직은 튜링기계를 능가하는 컴퓨터의 등장에 대해 인류는 알지 못해. 가능한지 불가능한지조차 증명되지않았거든

[9D%98_%EC%97%AD%EC%82%AC '컴퓨터의 역사' 위키피디아 검색](#)

4

<https://www.scienceall.com/%ED%98%84%EB%8C%80-%EC%BB%B4%ED%93%A8%ED%84%B0%EC%9D%98-%EB%AA%A8%EB%8D%B8-%ED%8A%9C%EB%A7%81%EA%B8%B0%EA%B3%84turing-machine%EC%9D%98-%EA%B3%A0%EC%95%88/>

사이언스올 '현대 컴퓨터의 모델, 튜링기계의 고안' 참조