과제 2

마감 : 5월 7일(일) 저녁 6시까지

내용 :

오락실의 하키 게임을 SW로 구현 (설명 참고)

Windows 프로그램 / C 언어 (C++등 다른 언어 안됨)

한쪽은 사람이 키보드로 진행하고 다른 한쪽은 컴퓨터가 자동으로 알고리즘으로 진행함 (업 다운 키)

스테이지가 올라가거나, 랠리가 한번씩 더해질수록 속도가 빨라지게 함.

컴퓨터쪽의 Bar 움직임 속도를 다르게 설정하여 난이도를 정하게 함. (난이도는 스테이지가 올라가도 유지됨)

사람쪽의 Bar 움직임은 적당한 속도로 일정하게 고정함. 최대한 단순하지만 Windows App 으로 진행함.

4,5월 수업이 도움이 될 수도 있고, 도움없이 진행할 수도 있음.

chatGPT 포함한 외부 자료 사용 절대 금함. 특성을 이미 잘 알고 있으므로,  사용흔적이 보이면 0점 처리

C언어 소스가 포함된 주피터노트북 파일(\*.ipynb 파일) 과 해당 파일을 PDF 로 저장한 파일을 보고서로 제출. 소스 설명과 주요 결과 화면, 소스 파일 자체가 포함되면 됨. (파이썬이 아니지만 ipynb 파일 형태로 제출. 작성이 용이함) Visual Studio 프로젝트 폴더를 통째로 압축해서 ZIP 파일로 같이 제출. 그래서 총 3개의 파일임

실행 결과는 절대 평가함. (제대로 되었는지 여부, 제출한 입력 포맷에 평가 데이터를 적용해서 진행)

전체 결과는 등급제로 상대 평가함. 열심히 코드 만들고, 열심히 최선의 결과를 잘 만든 사람에게 높은 등급

위 내용을 기준으로 추가로 필요한 부분은 스스로 판단해서 진행하면 됨. 의도에서 벗어나거나 축소되면 안됨.

이러한 게임은 탁구게임, 퐁게임으로 분류된다

c pong game

c ping pong game 검색하면 나옴

<https://github.com/samagragupta/ping-pong-C/blob/master/pingpong.c>

<https://www.youtube.com/watch?v=abHN47VsVEc>

<https://cboard.cprogramming.com/c-programming/151648-ping-pong-game.html>

<https://softuni.org/project-tutorials/how-to-make-a-pong-game-in-csharp-guide/>

비슷한 공튀기기 게임도 참고

<https://huiyu.tistory.com/entry/C%EC%96%B8%EC%96%B4%EB%A1%9C-%EA%B2%8C%EC%9E%84%EB%A7%8C%EB%93%A4%EA%B8%B0-5-%EC%B2%AB%EB%B2%88%EC%A7%B8-%EA%B2%8C%EC%9E%842Ball-%EB%A7%8C%EB%93%A4%EA%B8%B0>

공룡게임 참고(c언어에서 게임 만드는 법)

<https://blockdmask.tistory.com/344>

벽돌깨기 게임도 유사함

graphics.h 설치 (그래픽을 나타내려면 필요)

키보드 입력 받는 법

<https://m.blog.naver.com/sharonichoya/220875372940>

<https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=smilennv&logNo=220784833000>

<https://thisisvegetable.tistory.com/19>

공 알고리즘

<https://ansohxxn.github.io/c++%20games/chapter3-1/>

랜덤함수

<https://coding-factory.tistory.com/666>

일정한 각도로 공이 튀면서 게임이 진행되기 까지는 만듦

게임오버에서 system(“pause”) 후 재시작 안되는 문제 해결 (해결완료)

랠리 진행될수록 속도 증가 추가하기 (추가 완료)

난이도 선택 추가하기 (추가 완료)

공의 속도가 2배가 되었을 때 바를 통과하는 현상 제거하기 무조건 인식 x좌표에서 멈추도록 설정 해보기 (완료)

크기 줄이기 (완료)

공의 시작 방향을 랜덤으로 (완료)

현재 bar옆으로 한 칸 빠져도 그 다음 칸에서 bar의 위치에 볼이 존재해서 살릴 수 있음 허용할지 말지 선택 / 허용 안할거면 한 칸 빠졌을 때 바로 게임 종료 (x가 1이나 width일 때) (허용 안함으로 완료)

다른 각도로도 공이 튀도록 만들어보기 -> 이전 bar의 위치에 따라 튀는 방향이 바뀌게 만들어보기 (하지 않기로 결정 물리적으로 더 이상할 수도)