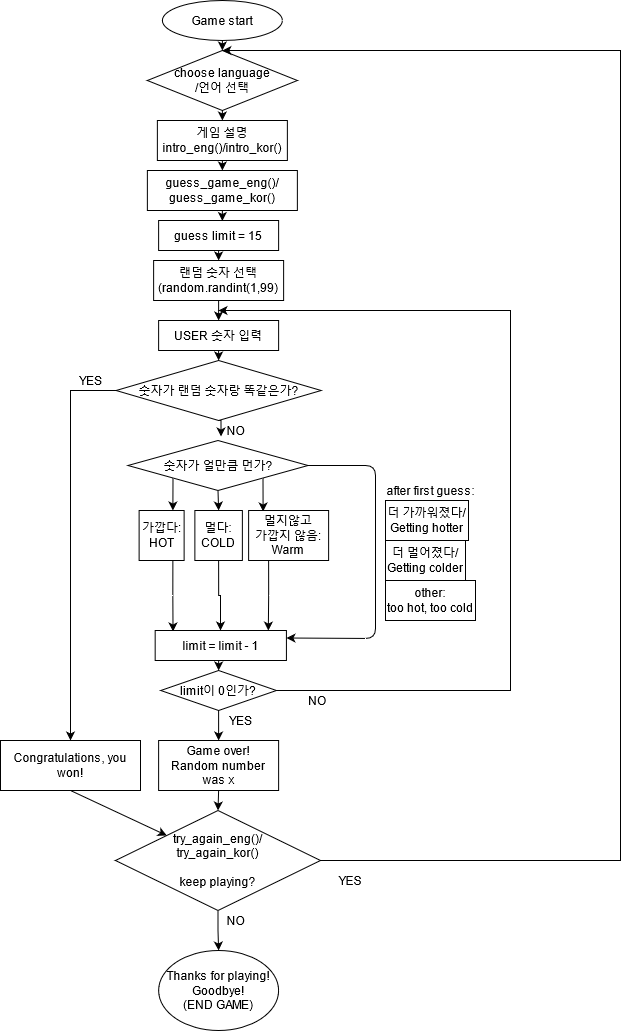
**제목**: Hot or Cold

**개발 목적 및 중요성**: 기존 수업에서 배웠던 함수/기능들(if-elif, while, random module, 비교연산자 등)을 응용해서 파이썬으로 숫자 맞히기 게임을 만들어본다.

**개발 내용 및 범위**:

* 방법(요구사항):
  + 개발환경 프로그램 PyCharm과 파이썬 언어를 이용한 숫자게임을 만들었습니다.
  + 놀이 중에 Hot & Cold라는 게임의 규칙도 조금 추가해봤습니다. (Hot & Cold를 하는 방법이 다양하지만 예를 들면 방안에 물건을 숨기고 물건을 찾는 게임을 할 때, 숨겨진 장소에 가까워질수록 ‘hot’, 멀어질수록 ‘cold’라고 힌트를 주는 게임입니다).
* 구조: (블록 다이어그램 + 설명)
  + 게임 시작/파일 실행하면 welcome()/언어 선택 화면이 실행된다. ‘ENG’, 또는 ‘KOR’를 입력하면 다음 함수로 넘어간다 (그 외 언어/입력은 ‘wrong input’이라고 출력하며 받지 않는다).
  + 선택한 언어로 게임 설명이 출력된다 (intro\_eng()/intro\_kor())
  + guess\_game\_eng(), guess\_game\_kor()이 실행되고 유저 숫자 입력을 받는다.
    - 맞히면 break하고 try\_again\_eng()/try\_again\_kor()이 실행된다.
    - 숫자가 랜덤 숫자와 다르면 얼만큼 가까운가에 따라서 출력되는 답/힌트가 달라진다.
      * 첫번째 기회 이후 출력되는 답/힌트들이 추가된다.
    - 입력할 때 마다 limit(guess할 수 있는 기회)가 1씩 줄어든다.
    - Limit이 0이 되면 try\_again\_eng()/try\_again\_kor()가 실행된다.
  + 마지막에는 try\_again\_eng()/try\_again\_kor()이 실행돼서, 계속 플레이할지 종료할지 선택할 수 있다.



* 기능(함수):
  + welcome(): 언어 선택 기능이 있다. ‘ENG’을 입력하면 intro\_eng(), try\_again\_eng()로 연결되고 ‘KOR’를 입력하면 intro\_kor(), try\_again\_kor()로 연결한다. ‘if language.upper() ==’로 모든 입력을 uppercase/대문자로 통일화 한다.
  + intro\_eng()/intro\_kor(): 게임 룰을 설명한다 (Hot/Cold/Warm 뜻 설명). 언어에 따라 guess\_game\_eng(), guess\_game\_kor()로 이어진다.
  + guess\_game\_eng(limit)/guess\_game\_kor(limit): 본 게임 (언어만 다를 뿐 같은 방법으로 실행된다). Random.randint로 1과 99사이의 숫자(num)를 하나 선택하고 숫자 입력(guess)을 받는다.

맞히지 못할 경우 limit(기회)가 1씩 줄어든다 (limit은 15로 설정되었다).

못 맞힐 경우 guess가 num에서 얼만큼 가까운지에 따라서 출력되는 답이 다르다:

* + - HOT: guess가 num과 가까운 범위에 있다; (num < guess < num + 10) or (num > guess > num - 10)
    - COLD: guess가 num보다 많이 멀다; guess > (num + 20) or guess < (num - 20)
    - Warm: guess가 HOT, COLD사이의 범위에 있다; (num + 10 <= guess <= num + 20) or (num - 5 >= guess >= num - 20)
    - 첫번째 guess 이후 상황에 따라 다른 답들도 출력된다:
      * Getting Hotter!/더 가까워졌어요!: 이전에 입력한 값을 임의로 pre\_value에 저장해서 현재 입력 값을 비교한다. 만약에 guess가 pre\_value보다 num에 더 가까워졌으면 이 답이 출력된다; ((pre\_value < guess < num) or (num < guess < pre\_value))
      * Getting Colder…/더 멀어졌어요…: 만약에 guess가 pre\_value와 num보다 멀어졌으면 이 답이 출력된다; ((guess < pre\_value < num) or (num < pre\_value < guess))
      * Getting VERY HOT!/많이 가까워졌어요!!: num과 아주 가까운 범위에 있다; ((num < guess <= num + 5) or (num > guess >= num - 5))
      * That’s TOO COLD!/너무 멀어졌어요!!: COLD보다 더 멀어졌다; (guess > (num + 25) or guess < (num - 25))
    - 숫자를 맞히면 게임을 이겼다는 문자가 출력이 되며 break한다. try\_again\_eng()/try\_again\_kor()로 이동한다.
    - limit이 0이되면 기회가 더 이상 없다고 출력하며, num/정답을 출력한다. try\_again\_eng()/try\_again\_kor()로 이동한다.
  + try\_again\_eng()/try\_again\_kor(): 게임을 다시 하겠냐고 물어본다. 모든 입력은 ‘if again.lower() ==’로 소문자로 통일화 한다. 입력은 ‘y’, ‘n’만 받고, 그 외에는 입력을 잘못했다고 출력한다. ‘y’면 welcome()이 다시 실행되고, ‘n’이면 ‘Thanks for playing! Goodbye!’라고 출력하고 프로그램이 종료한다.
* 구현(인터페이스): PyCharm의 run창에 실행된다.

**결과:**

* 성공 여부: 요구사항과 제안서에 적었던 기능들에 일치/실행할 수 있는 게임이 됐습니다.
  + 장점:
    - 선택에 따라서 한국어, 영어로 게임이 가능하다.
    - 힌트에 따라서 추리하는 재미(?)가 있을 수 있다.
  + 단점:
    - 엄연히 숫자 맞히기 게임이라서 힌트가 달라지는 것 외에는 구조가 단순해서 오래 플레이할 가능성이 낮다.
    - 간혹 guess가 num, pre\_value를 지나칠 때 ‘getting colder’대신 ‘Hot’, ‘Warm’이 출력되는 오류가 있다.
* 프로젝트 기대 성과: 의도했던 기능들을 구현 가능 해졌다 (특히 임시로 전 값을 보관해 현재 값과 비교하는 방법, Getting hotter/colder 출력하는 방법). 하지만 가끔 위 단점처럼 출력이 다르게 되는 것을 완전히 해결 못한 것이 아쉬운 점이다. (게임을 망치지는 않지만 100% 의도한 것이 아니 라서…) 그래도 나름 만족할 수 있는 게임을 만들었다.
* 실행 내용:

|  |  |
| --- | --- |
|  | welcome() |
|  | intro\_kor() |
|  | guess\_game\_kor()  첫번째 출력 (HOT/ COLD) |
|  | guess\_game\_kor()  Getting Colder…/ Getting Hotter!  출력 |
|  | guess\_game\_kor()  랜덤 숫자/정답에 많이 가까워진 경우 출력되는 힌트 |
|  | guess\_game\_kor()  랜덤 숫자/정답에서 너무 멀어진 경우 출력되는 힌트 |
|  | guess\_game\_kor()  게임에서 진 경우/기회가 더 이상 없는 경우. 랜덤숫자도 알려준다. |
|  | guess\_game\_kor()  게임에서 이긴 경우 |
|  | try\_again\_kor()  ‘n’을 입력한 경우 ‘Thanks for playing!’이 출력되며 종료 |
|  | intro\_eng()  welcome()에서 영어를 선택한 경우 |
|  | guess\_game\_eng()  첫번째 출력 (HOT) |
|  | guess\_game\_eng()  첫번째 힌트 이후 추가되는 힌트 |
|  | guess\_game\_eng()  게임에서 이긴 경우 |
|  | guess\_game\_eng()  게임에서 진 경우 |
|  | try\_again\_eng()  ‘y’를 입력한 경우 welcome()이 다시 실행된다. |
|  | 전체적인 화면 (한국어 버전 |

**기대성과 및 활용방안:**

* 프로그램 동작 방법:
  + PyCharm에서 파일을 열어서 실행한다.
  + 선택할 언어를 입력한다 (‘eng’, ‘kor’ 중 입력한다).
  + 설명을 읽은 후 숫자 입력을 한다 (1부터 99포함해서).
  + 게임을 완료하면 ‘y’나 ‘n’을 입력해서 다시 플레이 해도 되고, 종료해도 된다.