**개별 프로젝트 최종 보고서**

**작성자: 박창주**

1. **개발 목적**

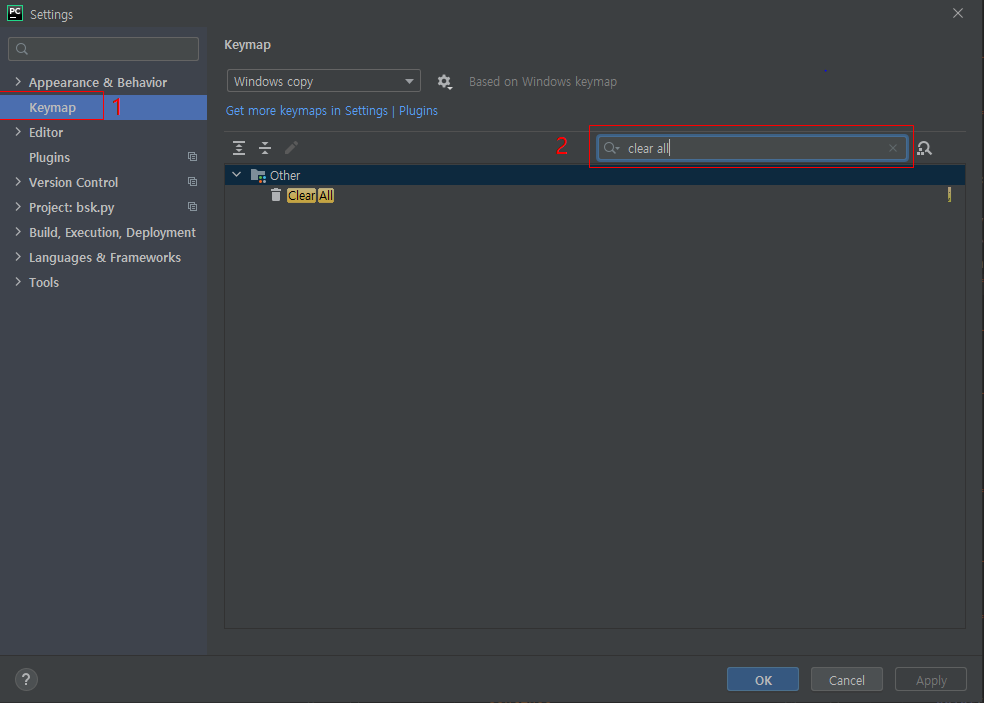
간단한 게임 진행으로 컴퓨터와 대결을 하여 간단한 두뇌의 활성화를 목적

1. **개발 내용 및 범위**
2. **방법**

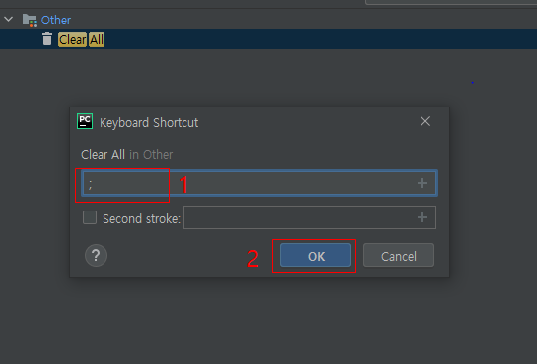
* **진행 간 요구하는 입력 값을 입력하며 단계적으로 진행**
  + - 하단 프로그램 동장 방법 참조(page 8~9)
* **게임 실행 전 기능 설정**
  + - 키 설정 (화면 초기화 키)
      * Pycharm 프로그램 실행 후 ctrl + alt + s 를 누른다.

2

* + - * Settings 창 내 Keymap 클릭 후 검색 창에 ‘clear all’ 입력 후 검색



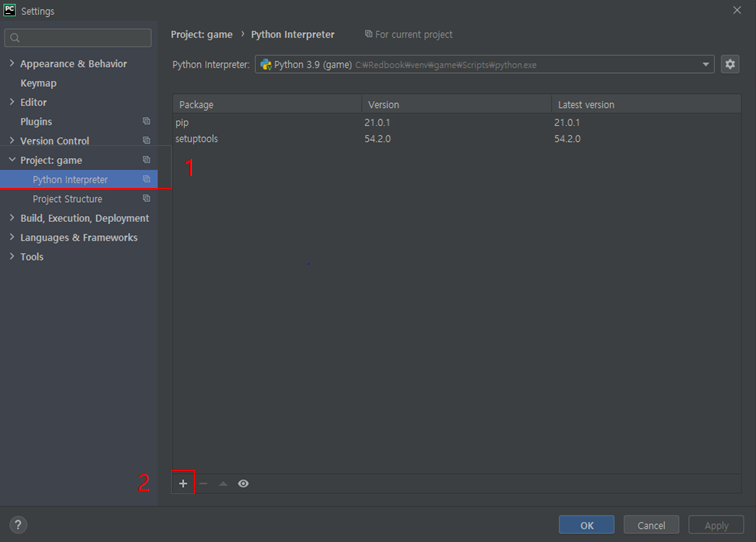
* + - * Clear All 항목에 마우스 우 클릭 후 Keyboard Shortcut 클릭
      * Keyboard Shortcut 창에 입력 바에 ‘;’ 을 입력 후 OK 버튼 클릭



* + - 모듈 설치
      * Pycharm 프로그램 실행 후 ctrl + alt + s 를 누른다.

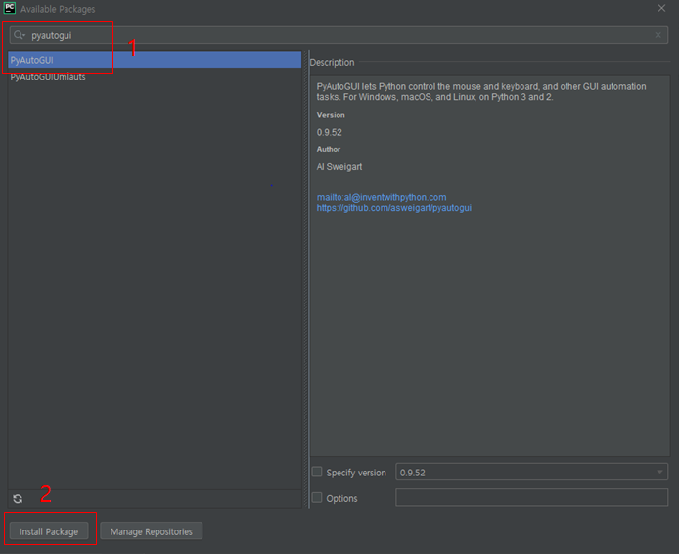
2

* + - * Settings 창 내 게임 파일이 저장 된 프로젝트 폴더에서 ‘python interpreter’ 클릭 후 화면 내 ‘+’ 버튼 클릭

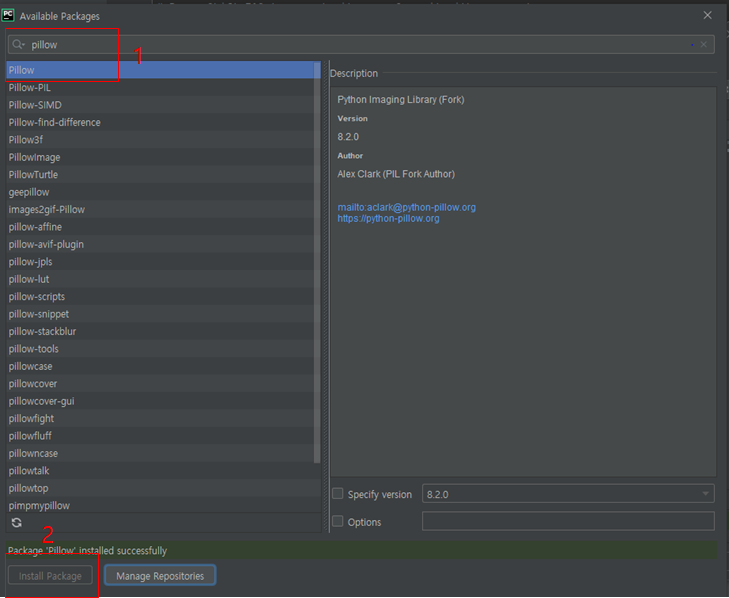


* + - * Available Packages 실행창 내 ‘pyautogui’과 ‘pillow’ 를 검색 후 하단 ‘Install Package’ 클릭

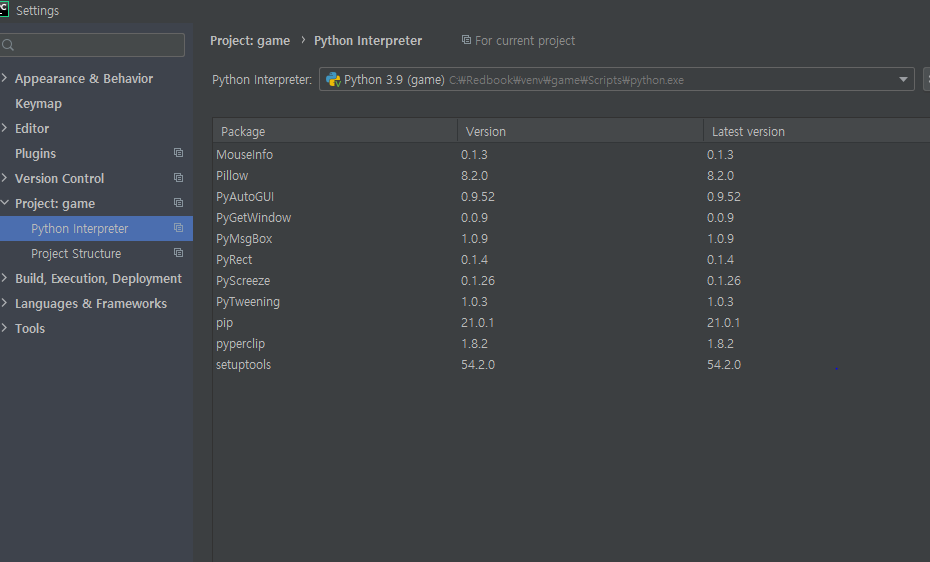
# pyautogui



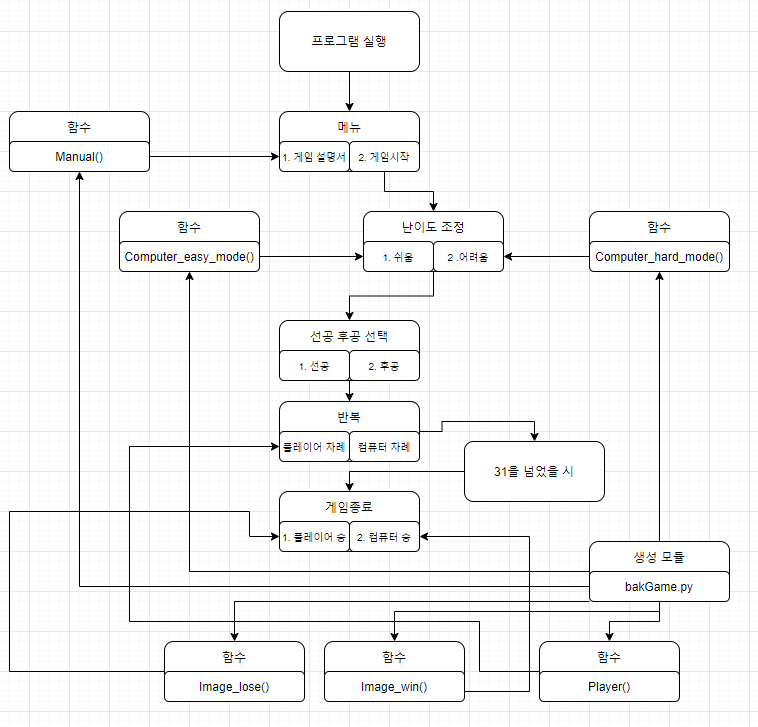
# pillow



# 설치 완료 후 setting 실행창 화면



* **구조(블록 다이어그램)**



* **시나리오**

1) 시스템은 플레이어에게 게임설명서 또는 게임진행 여부를 묻는다

2) 플레이어는 시스템에 요구에 따라 선택한다

3) 시스템은 플레이어에게 게임 난이도를 선택을 요구한다

4) 플레이어는 게임 난이도를 선택한다

5) 시스템은 난이도에 따라 등록된 클래스 함수를 불러온다

6) 시스템은 플레이어의 선/후공 여부를 묻는다

7) 플레이어는 선/후공를 선택한다

8) 시스템은 입력 받은 플레이어의 선택을 게임에 적용한다

9) 시스템은 플레이어에게 부를 숫자를 묻는다

10) 플레이어는 매 턴 1~3 까지 부를 숫자의 개수 을 입력한다

11) 시스템은 플레이어가 1~3 이내 범위 숫자를 입력했는지 확인 후 아닐 시 ‘다시 입력하세요' 를 출력 후 이전 단계로 돌아가게 한다

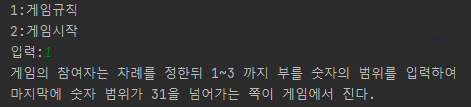
12) 시스템은 플레이어가 부른 숫자를 제외한 나머지 숫자를 최대 3개 까지 출력 한다

13) 이후 시스템과 플레이어는 입력 수가 31을 초과 할 때 까지 반복 한다

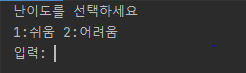
14) 시스템은 부른 수가 31을 초과했는지 확인 후 반복 과정을 종료한다

15) 시스템은 마지막에 31을 부른 쪽에 패배를 출력한다.

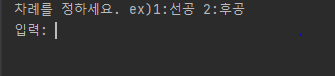
* **기능(함수)**
  1. While (continue, break)
     + 조건에 만족 시까지 동작
  2. 생성한 모듈 함수
     + Manual: 게임내 설명서
     + Player: 플레이어
     + Computer\_easy\_mode: 난이도 쉬움 # random 모듈 활용
     + Computer\_hard\_mode: 난이도 어려움
     + Image\_win: 결과 출력(승리)
     + Image\_lose: 결과 출력(패배)
  3. If / elif / else
     + 게임 진행 간 조건 설정
  4. Input
     + 사용자 인터페이스 구현
  5. Print
     + 원활한 인터페이스 구현 목적
  6. Pyautogui 모듈 ‘press’
     + 실행창 내 화면 초기화 활용
  7. Random 모듈 ‘randiant’
     + 컴퓨터 난이도 설정에 활용
  8. Pollow 모듈 ‘Image’
     + 게임 종료 시 승리, 패배 이미지 불러오기
* **구현**
  + - 초기 화면



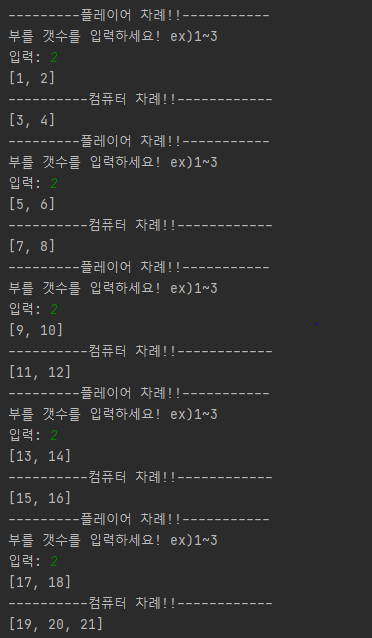
* + - 난이도 선택 화면



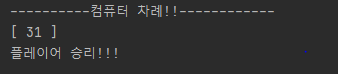
* + - 공수 선택 화면



* + - 플레이 화면



* + - 결과 화면





* **장단점**

1. **장점**
   * + **단시간 내에 가볍게 플레이가 가능하다.**
     + **게임이 대중적이라 누구나 쉽게 접근 할 수 있다.**
     + **난이도 조정이 가능하여 수준별 플레이가 가능하다**
     + **선공 후공을 설정 할 수 있어 다양한 플레이가 가능하다.**
   1. **단점**
      * **어려움 난이도가 너무 어렵다.**
2. 프로그램 동작 방법
3. 초기 실행 시 활성화된 창 게임 설명 및 게임 진행 입력
   * + 1번 입력 시 게임 설명 2번 입력 시 게임 시작
4. 게임 시작(2번) 입력 후 활성화된 창에 난이도 설정 입력
   * + 1번 입력 시 쉬움(난이도) 2번 입력 시 어려움(난이도)
5. 난이도 설정 후 공수 설정 입력
   * + 1번 시 선공 2번 시 후공
6. 공수 설정 후 게임 진행
   * + 1~3까지 자신이 부를 숫자 입력

Ex) 마지막 부른 수 + 1~3까지 숫자 = 마지막 부른 수