2022-01-14 (금)

1조 : 천지창조

내일은 내가 바로 숫자야구의 신

빅데이터 전문가 양성 과정

김동훈, 김민정, 임낙현, 주현우, 채지혜

목차

01

추진 배경

프로젝트 개요

추진 배경

어릴 적 즐겨하던 숫자 야구게임을 구현해봤습니다!

프로그램 특징

숫자 야구 게임에 랭킹 시스템을 더하여 플레이어가 자신의 스코어를 확인 할 수 있습니다.

```
1. 게임 시작하기
2. 랭킹 확인하기
3. 끝내기
번호를 입력해주세요 :
```

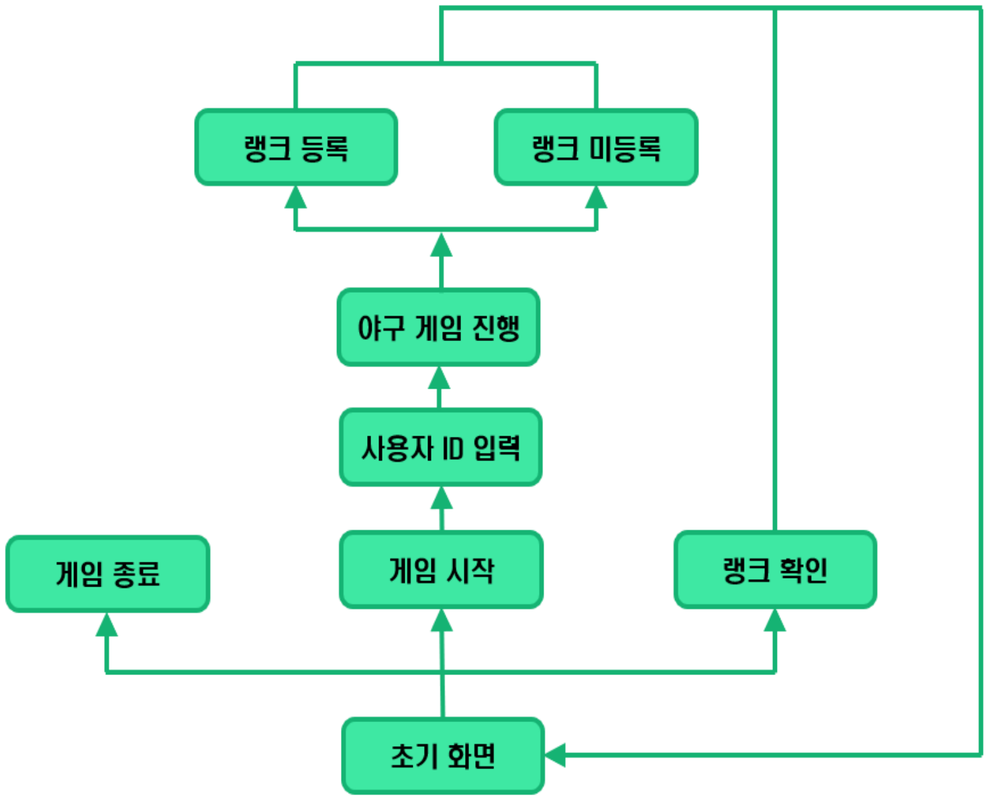
02

시나리오

프로그램 진행 순서

프로그램 진행 순서

시나리오 다이어그램



게임 진행 순서

1. 플레이가 프로그램을 켭니다.
2. 게임 종료, 게임 시작, 랭크 확인 중 선택합니다.
3. 게임을 시작할 때 사용자 ID를 입력합니다.
4. 재미있는 숫자 야구 게임을 즐깁니다.
5. 게임이 끝이 난다면 본인의 랭크를 등록 하거나 미등록 할 수 있습니다.

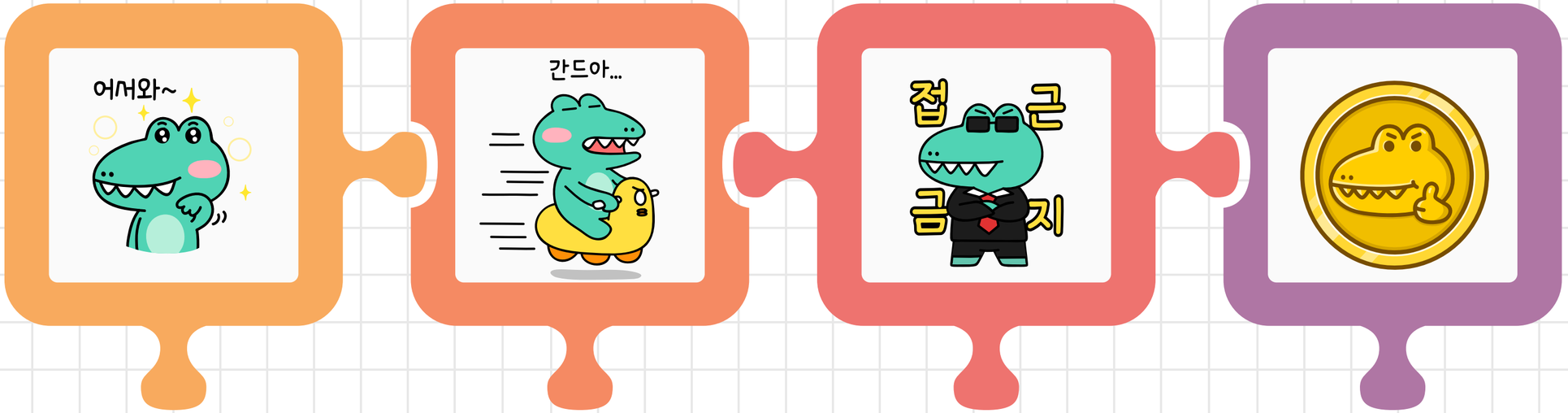
03

구현 방법

프로그램 코드 구성

프로그램 코드 구성

코드 소개



게임 메뉴화면

screen

게임 진행

- game_algorithm
- rank_save_ask

게임 DB

DB

게임 Class

- player_class
- game_classs

게임 메뉴화면

〈 screen 〉

code

```python
from controller.db_controller import select_top10_minimum_attemps
from game.game_algorithm import create_numbers
from lib.lib import clear_terminal_by_os
from db.db import DB
from classes.player import Player
from classes.game import Game
from uuid import UUID, uuid4
import sys


@clear_terminal_by_os
def init_player():
    print('\n이름을 입력해주세요! : ',end='')
    name = sys.stdin.readline().replace('\n','').strip()
    this_player:Player = Player(uuid4(),0,name)
    return this_player

def init_game(player_id:UUID):
    ans = create_numbers()
    this_game = Game(player_id,ans)
    return this_game

# @clear_terminal_by_os
def game_start(p:Player,g:Game):
    print(1)
```

①

```python
@clear_terminal_by_os
def rank_output(db:DB):
    vals,columns = select_top10_minimum_attemps(db)
    print(columns)
    for i in vals:
        print(i)

@clear_terminal_by_os
def in_game(db:DB):
    while True:
        print("1. 게임시작!")
        print("2. 랭킹 보기")
        print("3. 게임 종료\n")
        e_c = input("1~3 사이의 번호를 입력해주세요. : ")
        if e_c == '1':
            this_player = init_player()
            this_game = init_game(this_player.get_id())
            game_start(this_player,this_game)
        elif e_c == '2':
            rank_output(db)
        elif e_c == '3':
            print("게임을 종료합니다.")
            break
        else:
            print("1~3 사이의 번호를 입력해주세요.")
        print(" ")
```

②

게임 진행

code

〈 game_algorithm 〉

① ②

```python
from random import shuffle
from sys import stdin
from typing import Any, Union

from lib.lib import check_type_int


def create_numbers() -> list[int]:
    arr = [_ for _ in range(0, 10)]
    shuffle(arr)
    return arr[:4]


def get_input()->Any:
    print("\n숫자(0~9) 4개를 입력해주세요! EX. 1 2 3 4")
    print("\n** 포기하시려면 n을 입력해주세요! **\n")
    inp = []
    while True:
        inp:Union[list[str],list[int]] = stdin.readline().split()
        inp_set = set(inp)
        if 'n' in inp:
            return []
        if inp_set.__len__() != 4 or inp.__len__() != 4 or not check_type_int(inp):
            print("띄어쓰기 구분 고유한 4개의 숫자(0~9)를 입력해주세요!")
            continue
        else:
            inp = list(map(int,inp))
        if list(filter(lambda x:x >= 10 or x < 0, inp)).__len__() > 0:
            print("0부터 9까지의 숫자를 입력해주세요!")
            continue
        break

    return inp


def Check_number(ans: list[int], my_ans: list[int])->list[int]:
    Strike = 0
    Ball = 0
    for i in range(0, 4):
        if my_ans[i] in ans:
            if my_ans[i] == ans[i]:
                Strike = Strike + 1
            else:
                Ball = Ball + 1
    return [Strike, Ball]
```

게임 DB

code

〈 DB 〉

①

```python
from typing import Any, Optional
import psycopg2 as pg2
from typing import Optional
from lib.lib import error_logger


class DB:
    def __init__(self) -> None:
        self.__connection: Optional[pg2.connection] = None
        self.__cursor: Optional[pg2.cursor] = None  # typing 해결 안됨..

    def connect(self, options: str) -> None:
        self.__connection = pg2.connect(options)
        self.__connection.autocommit = True
        # self.set_isolation_level()
        self.__cursor = self.__connection.cursor()

    @error_logger
    def execute_query_has_return(self, query: str) -> [list[tuple], list[str]]:
        self.__cursor.execute(query)
        column_names = [r[0] for r in self.__cursor.description]

        return [self.__cursor.fetchall(), column_names]
```

②

```python

    @error_logger
    def execute_query_no_return(self, query: str)->None:
        self.__cursor.execute(query)

    def close(self) -> None:
        try:
            self.__connection.close()
        except pg2.DatabaseError as e:
            exit(0)

    @error_logger
    def get_connection(self) -> Optional[Any]:
        return self.__connection

    @error_logger
    def get_cursor(self) -> Optional[Any]:
        return self.__cursor
```

게임 Classes

code

〈 player_class 〉

```python
from uuid import UUID


class Player:
    def __init__(self, ids:UUID, attempt:int, name:str):
        self.__id = ids
        self.__attempt:int = attempt
        self.__name:str = name

    def set_id(self, ids:int)->None:
        self.__id = ids

    def get_id(self)->UUID:
        return self.__id

    def set_name(self, name:str)->None:
        self.__name = name

    def get_name(self)->str:
        return self.__name

    def set_attemp(self, attempt:int)->None:
        self.__attempt = attempt

    def get_attempt(self)->int:
        return self.__attempt
```

〈 game_class 〉

①

```python
from uuid import UUID
from lib.consts import GAME_STATE


class Game:
    def __init__(self, id_:UUID, answer:list[int]=[0,0,0,0], ball:int=0, strike:int=0, situation:str=GAME_STATE['게임시작 전']):
        self.id = id_
        self.ball = ball
        self.strike = strike
        self.situation = situation
        self.answer = answer

    def get_id(self) -> UUID:
        return self.id
    def set_id(self, id):
        self.id = id

    def get_answer(self)->list[int]:
        return self.answer
    def set_answer(self, l:list[int])->None:
        self.answer = l

    def get_ball(self):
       return self.ball
    def set_ball(self, ball):
        self.ball = ball
```

②

```python
    def get_ball(self):
       return self.ball
    def set_ball(self, ball):
        self.ball = ball


    def get_strike(self):
        return self.strike
    def set_strike(self, strike):
        self.strike = strike


    def get_situation(self):
        return self.situation
    def set_stituation(self, situation):
        self.stituation = situation
```


QnA

발표를 들어주셔서
감사합니다