

구조 설계서

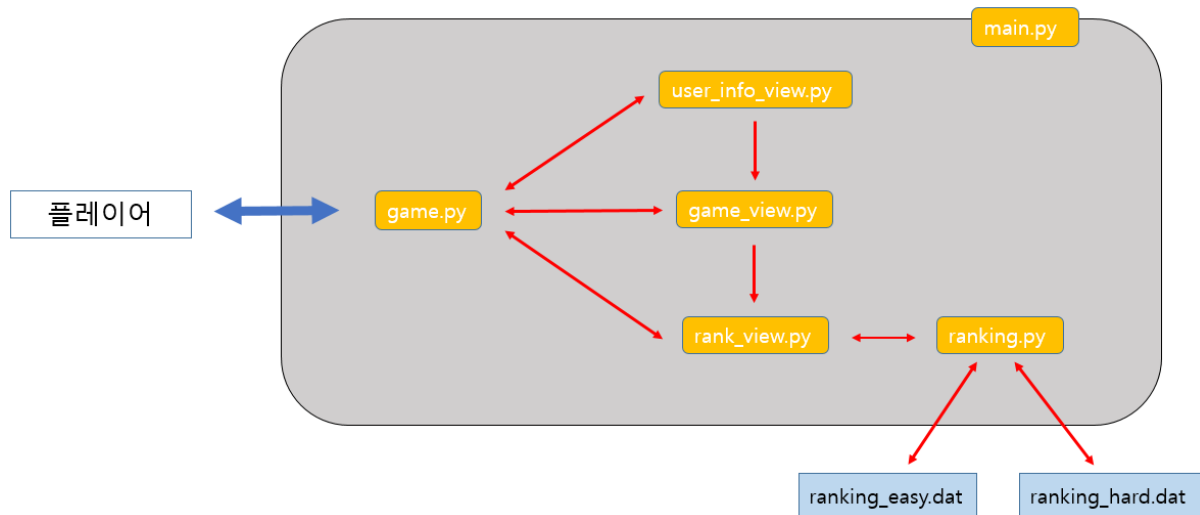
20213102 황수민

20213360 김혜은

(1) 파일의 구조와 역할 소개

모듈(파일)	클래스	역할
main.py	-	main.py를 실행하면 게임이 실행된다.
user_info_view.py	UserInfoView	유저의 정보를 입력 받는다.
game.py	Game	게임이 실행되면 user_info_view.py에서 받은 정보를 가지고 '정답', '버튼을 누르는 횟수'를 미리 정하고 '정답 확인', '시간 측정' 및 '사운드 재생'을 한다.
game_view.py	GameView	다이얼을 돌려 게임을 플레이한다.
rank_view.py	RankView	게임이 끝난 후 보여질 랭킹 창이다.
ranking.py	Ranking	랭킹 파일을 새로 생성하거나 기록을 저장한다.
ranking_hard.dat	-	Hard 난이도의 기록을 저장한다.
ranking_easy.dat	-	Easy 난이도의 기록을 저장한다.

(2) 전체적인 소프트웨어 구조



상세 설계서

20213102 황수민

20213360 김혜은

Class Main

구분	이름	기능
method	main	Game()와 UserInfoView() 객체를 생성하고 게임을 실행시킨다.

Class UserInfoView

구분	이름	기능
attribute (properties)	game	Game 객체, 지금 실행되고 있는 game_data를 불러온다.
	game_view	GameView 객체
method	initUI	사용자의 정보를 입력받는 창이다.
	button_clicked	<ul style="list-style-type: none"> - 이름을 입력하지 않은 경우 QMessageBox를 이용해 에러 창을 띄운다. - 입력한 경우, 이름과 선택한 난이도를 game객체를 통해 new_game함수의 인자로 전달하고, game_view창을 띄운다.
	__init__	game 객체, game_view 객체, initUI를 불러온다.

Class Game

구분	이름	기능
attribute (properties)	answer_list	Game()와 UserInfoView() 객체를 생성하고 게임을 실행시킨다.
	user_name	사용자의 이름을 담는다.
	difficulty	선택한 난이도를 담는다.
	current_pw	현재까지 입력된 password를 담는다.

	max_page	버튼을 누르는 횟수를 담는다.
	current_page	현재 버튼을 몇 번 눌렀는지 저장한다.
	sound_effect	QSoundEffect 객체, 다이얼을 돌릴 때 사용할 beep음을 저장한다.
	start_time	게임의 시작 시간을 저장한다.
	time_record	게임을 하는데 걸린 시간을 저장한다.
	max_number	다이얼의 최대 눈금을 지정한다.
method	new_game	user_name과 difficulty 인자를 받아서 정답, 페이지등 게임 시작에 필요한 것들을 정한다.
	create_password	난수로 지정된 max_page에 따라 랜덤한 password를 만든다.
	get_password	answer_list 속의 정답을 string으로 변환한다.
	check_password	최종 정답을 확인한다.
	record_time	플레이 시간을 측정한다.
	goto_next	버튼을 누른 경우 current_password를 갱신하고 current_page에 1을 더한다.
	play_beep_sound	다이얼을 돌릴 때 beep음을 출력한다.
	__init__	필요한 attribute들을 초기화한다.

Class GameView

구분	이름	기능
attribute (properties)	game	Game 객체, 지금 실행되고 있는 game_data를 불러온다.
	rank_view	QWidget 객체

method	initUI	메인 게임 창을 구성한다.
	button_clicked	<ul style="list-style-type: none"> - 현재 다이얼의 숫자를 받아서 game객체에게 보내고 current_pw를 갱신한다. - LCD 판에 current_pw를 출력한다. - 마지막 페이지인 경우 버튼 텍스트를 'OK'로 변경한다. - 페이지가 끝까지 가지 않았는데 이미 6자리가 된 경우 실패로 처리한다.
	value_changed	다이얼이 움직이면 beep음을 출력하는 함수를 부른다..
	update_game	난이도가 easy일 때 버튼을 눌러야 하는 횟수(힌트)를 화면에 띄워준다.
	__init__	game 객체 불러오고, rank_view 객체 생성, initUI 불러온다.

Class RankView

구분	이름	기능
attribute (properties)	game	Game 객체, 지금 실행되고 있는 game_data를 불러온다
	user_name	사용자의 이름을 불러온다.
	difficulty	난이도를 불러온다
	ranking	Ranking 객체
	rank_data	ranking객체에서 rank_data를 가져온다.
	success	게임의 성공 여부를 담는다.
method	initUI	기존에 저장 되어있던 랭킹과 사용자의 기록을 보여주는 창을 구성한다.
	button_clicked	<ul style="list-style-type: none"> - 종료 버튼을 누르면, 성공 시, 기록을 저장할 지 선택하고 종료한다. - 실패 시, 프로그램이 종료된다.
	__init__	game 객체를 불러오고, 필요한 attribute를 불러온다.

Class Ranking

구분	이름	기능
attribute (properties)	file_name	난이도에 따른 파일 이름이다.
	rank_data	현재 사용자의 기록을 리스트 형태로 담는다.
method	read_from_file	file_name 명의 파일을 연다.
	write_to_file	file_name 명의 파일에 rank_data를 dump한다.
	rank	rank_data에 사용자의 기록을 저장한다.
	update_game	난이도가 easy일 때 버튼을 눌러야 하는 횟수(힌트)를 화면에 띄워준다.
	__init__	file_name을 정하고, rankdata를 초기화한다.

Game 클래스의 인터페이스

메소드	입력	출력
new_game	user_name, difficulty	-
get_password	-	result_text
check_password	-	self.current_pw == self.get_password()
goto_next	value	self.current_pw

Ranking 클래스의 인터페이스

메소드	입력	출력
rank	user_name, time	-