

AD Project

소프트웨어 상세 설계서

(DDS)

20182209 김수빈

Class MainWindow		
Attribute (properties)	dbfilename	scoredb 파일 이름을 저장한다.
	scoredb	scoredb 파일 내용을 리스트로 저장한다.
	N	mapsize를 저장한다.
	mapsizeText	[3x3, 4x4, 5x5, 6x6] 리스트를 저장한다.
	mapsizeLabel	mapsizeText를 레이블로 저장한다.
	mapPhotoBtn	Item 클래스에서 img, size를 받아와서 Button 클래스를 이용해 mapPhoto 버튼을 설정한다. 비활성화 시킨다.
	arrowBtn	Item 클래스에서 img, size를 받아와서 Button 클래스를 이용해 왼쪽 오른쪽 버튼을 설정한다. sizeFunc 매서드와 연결해준다.
	startBtn	Item 클래스에서 img, size를 받아와서 Button 클래스를 이용해 게임 시작 버튼을 설정한다. showGamewindow 매서드와 연결해준다
	rankBtn	Item 클래스에서 img, size를 받아와서 Button 클래스를 이용해 순위표 버튼을 설정한다. showRankwindow 매서드와 연결해준다
Method	initUi	메뉴의 GUI를 설정해준다. 윈도우 배경 색과 크기를 설정한다. self.N은 맵 사이즈로, 3으로 초기화한다. 첫번째 행에 맵 사이즈 그림 버튼을 설정한다. 두번째 행에 왼쪽 버튼과, 맵 사이즈 레이블과, 오른쪽 버튼을 설정한다. 맵 사이즈 레이블은 3x3, 4x4, 5x5, 6x6으로 이루어져 있다. 왼쪽 버튼은 leftfunc 매서드에 연결해준다. 오른쪽 버튼은 rightfunc 매서드에 연결해준다. 세번째 행에 게임 시작 버튼을 설정한다. 게임 시작 버튼은 showGamewindow 매서드에 연결해준다. 모든 버튼들은 Item 클래스에서 self.itemGroup에 dictionary 형식으로 저장되어 있는 size와 img를 받아와서, Button 클래스에서 설정해준다. 3개의 행으로 전체 레이아웃을 구성한다
	readScoreDB	게임 점수가 저장되어 있는 scoreDB.dat을 불러와서 self.scoredb에 저장한다.
	sizeFunc	sender의 objectName이 'Left'이면 self.N을 1만큼 감소시킨다. 'right'이면 1만큼 증가시킨다.

		<p>맵 사이즈 그림의 img 리스트에서 self.N-3 인덱스 위치의 그림으로 바꿔준다.</p> <p>self.mapsizeText 리스트에서 self.N-3 인덱스 위치의 text 로 맵 사이즈 레이블을 바꿔준다.</p>
	showGamewindow	<p>메뉴 윈도우 창을 숨기고, 게임 윈도우 창을 불러온다.</p> <p>이때, self.N 과 self.scoredb 를 인자로 보내준다.</p> <p>게임 윈도우가 close 될 때까지 대기하고, close 가 되면 다시 메뉴 윈도우 창을 보여준다.</p>
	showRankwindow	<p>메뉴 윈도우 창을 숨기고, 순위표 윈도우 창을 불러온다.</p> <p>이때, self.scoredb 를 인자로 보내준다.</p> <p>순위표 윈도우가 close 될 때까지 대기하고, close 가 되면 다시 메뉴 윈도우 창을 보여준다.</p>

Class RankWindow		
Attribute (properties)	scoredb	menuWindow클래스에서 파일 내용을 리스트로 저장되어 있는 scoredb를 받아온다.
	menuBtn	Item 클래스에서 img, size를 받아와서 Button클래스를 이용해 메뉴 버튼을 설정한다. showMenuwindow 매서드와 연결해준다.
	scoreText	scoredb에서 score 리스트를 받아와서 '1위 1234점'과 같이 text로 변환시켜 저장한다.
	rankLabel	scoreText를 레이블 텍스트로 저장한다.
	sizeText	'3x3'과 같이 text를 저장한다.
	sizeLabel	sizeText를 레이블 텍스트로 저장한다.
Method	initUi	순위표의 GUI를 설정해준다. 윈도우 배경 색과 크기를 설정한다. 첫번째 행에 메뉴 버튼을 설정한다. 메뉴 버튼은 Item클래스에서 menuBtn의 size와 img를 받아와서, Button 클래스에서 설정해준다. 두번째 행에 맵 사이즈에 따라 scoredb에 저장되어 있는 게임 점수 순위를 rankLabel과 sizeLabel을 설정해 rankLayout으로 만들어 보여준다. 2개의 행으로 전체 레이아웃을 구성한다.
	showMenuwindow	순위표 윈도우 창을 닫아준다.

Class GameWindow		
Attribute (properties)	N	mapsize를 저장한다.
	dbfilename	scoredb 파일 이름을 저장한다.
	scoredb	scoredb 파일 내용을 리스트로 저장한다.
	sizeScoredb	Scoredb 리스트에서 size가 N인 score 리스트를 저장한다.
	Title	'2048' 레이블을 저장한다.
	scoreText	'SCOREWn0'과 같이 저장한다. 게임을 시작하면 'SCOREWn+score'로 업데이트 한다.
	bestScore	'BESTWn+sizeScoredb[0]'을 저장한다. 게임을 시작하고 현재 score가 sizeScoredb[0] 보다 커지면 'BESTWn+score'로 업데이트 한다.
	restartBtn	Item 클래스에서 img, size를 받아와서 Button클래스를 이용해 게임 초기화 버튼을 설정한다. CMap 클래스에 displayGameinit 매서드와 연결해준다
	menuBtn	Item 클래스에서 img, size를 받아와서 Button클래스를 이용해 메뉴 버튼을 설정한다. showMenuwindow 매서드와 연결해준다
	backgroundRect	background의 위치와 size를 QRect로 저장한다.
	button	Item 클래스에서 img, size를 받아와서 Button클래스를 이용해 블록 버튼을 설정해서 self.N * self.N 사이즈만큼 리스트로 저장한다. 비활성화 시킨다.
Method	CMap	CMap 클래스를 불러와 저장한다.
	__init__	메뉴 윈도우로부터 self.N 과 self.scoredb 를 받아온다. scoredb 는 dictionary 형식으로 [{'size':3, 'score':[]}, {'size':4, 'score':[]}, { 'size':5, 'score':[]}, { 'size':6, 'score':[]}] 로 설정했다. 여기서 size 가 self.N 인 score 리스트만 self.sizeScoredb 에 따로 저장해준다. initUi를 보여준다.
Method	initUi	게임 윈도우의 GUI 를 설정해준다. 윈도우 배경 색과 크기를 설정한다. 첫번째 행에 게임 타이틀 레이블과 게임 점수, 게임 최고 점수를 설정한다. 게임 최고 점수는 self.sizeScoredb[0]이다. 두번째 행에 메뉴 버튼, 게임 초기화 버튼을 설정한다. 세번째 행에 mainWindow 에서 받아온 맵 사이즈만큼 맵 블록 버튼을 설정하고, 레이아웃을 구성해준다. 모든 버튼들은 Item 클래스에서 self.itemGroup 에

		<p>dictionary 형식으로 저장되어 있는 size 와 img 를 받아와서, Button 클래스에서 설정해준다.</p> <p>3 개의 행으로 전체 레이아웃을 구성한다.</p> <p>CMap 클래스를 불러온다.</p> <p>메뉴 버튼은 showMenuwindow 매서드와 연결한다.</p> <p>게임 초기화 버튼은 CMap 클래스에 있는 displayGameinit 매서드와 연결해준다.</p> <p>키보드 방향키 함수를 설정한다.</p>
	writeScoreDB	<p>self.scoredb 를 저장해주는 함수이다.</p> <p>게임 창을 닫거나, 게임 초기화할 때 그 때의 score 를 저장해준다.</p> <p>score 가 scoredb 에 이미 있다면 저장하지 않는다.</p> <p>scoredb를 저장할 때, 내림차순으로 정렬한 후 상위 5개로 잘라서 저장한다.</p>
	showMenuwindow	<p>WriteScoreDB 매서드를 불러와 score 를 저장한다.</p> <p>게임 윈도우 창을 닫는다.</p>
	keyPress	<p>사용자가 키보드 방향키를 누르면 발생한다.</p> <p>위젯과 독립적으로 키보드 이벤트를 캡처할 수 있는 QShortcut 을 사용해 방향키 입력을 받아온다.</p> <p>방향키를 누르면 CMap 클래스 안에 changeMap 매서드와 연결해준데,</p> <p>이때 왼쪽 방향키를 누르면 'L'을 인자로 주고,</p> <p>오른쪽 방향키를 누르면 'R'을 인자로 주고,</p> <p>위쪽 방향키를 누르면 'U'를 인자로 주고,</p> <p>아래쪽 방향키를 누르면 'D'를 인자로 준다.</p>
	paintEvent	<p>게임 맵의 background 를 그려준다.</p> <p>게임 맵은 QGridLayout 으로 이루어져 있는데 레이아웃은 배경을 설정할 수 없어서 paintEvent 를 이용해서 배경을 색칠한다.</p>
	blockUpdate	<p>블록을 업데이트 해주는 함수이다.</p> <p>Map 이 0 인 곳에는 빈 공간 그림으로 설정해주고,</p> <p>Map 이 2 의 거듭제곱인 곳에는 그에 해당하는 그림으로 설정해준다.</p> <p>각 그림은 item 클래스에 block 의 img 리스트를 받아오고, Button 클래스에 setImg 를 통해서 설정해준다.</p>

Class CMap		
Attribute (properties)	N	Mapsize를 gameWindow 클래스에서 받아와 int로 저장한다.
	score	게임 진행 중의 게임 점수를 int로 저장한다.
	MoveBlock	MoveBlock클래스를 불러와 저장한다.
	map	게임 맵을 나타낸다. self.N * self.N 크기만큼 2차원 리스트 배열으로 0으로 초기화 한다. 게임 진행 중에 블록을 생성하거나 움직일 때마다 map은 업데이트 된다.
	zeroIndex	map이 0인 위치(row, col)를 2차원 리스트 배열로 저장한다.
	nextMap	키보드 방향키를 누를 때, MoveBlock클래스에 nextMove 매서드를 통해 map을 이동시킨 후의 map을 nextmap에 2차원 리스트 배열로 저장한다.
	nextScore	키보드 방향키를 누를 때, map을 이동시킨 후의 score를 nextScore에 int로 저장한다.
Method	gaimInit	게임 맵을 초기화 해주는 함수이다. self.map 을 self.N x self.N 크기만큼 모두 0 으로 초기화 해준다. self.score 를 0 으로 초기화 해주고, GameWindow 클래스에 scoreText 에 보여준다. self.map에 랜덤한 위치에 2블록을 두 개 생성한다.
	displayGameinit	게임을 초기화 했을 경우, GameWindow 클래스에 writeScoreDB 매서드를 이용해 현재 점수를 scoredb 에 저장한다. gameinit 을 통해 게임 맵을 업데이트 시키고 그것에 따라 GameWindow 클래스에 blockUpdate 매서드로 블록 img 를 업데이트 해준다.
	generateBlock	게임 맵에 새로운 2 또는 4 블록을 빈 공간인 랜덤한 위치 한 곳에 생성한다. 2는 90% 확률로, 4는 10% 확률로 선택된다.
	changeMap	키보드 방향키에 따라 게임 맵을 변경해주는 함수이다. moveBlock 클래스 안에 nextMove 매서드를 통해 skey 에 따라서 self.map 이 변경되어야 하는 nextmap 과 nextScore 를 설정한다. 현재 점수가 최고 점수보다 커지면 최고 점수도 같이 업데이트 해준다. 키보드 방향키를 누르기 전과 후의 게임 맵이 다르면

		<p>generateBlock 메시지를 통해 새로운 블록을 생성하고, GameWindow 클래스에 blockUpdate 메시지를 통해 블록 그림을 업데이트 해준다.</p> <p>success 메시지와 gameover 메시지를 통해 게임 성공과 게임 실패 여부를 확인하고, True 이면 message 메시지를 통해 MessageBox 를 생성해서, 다시 시작할 것인지, 게임 종료를 할 것인지 정한다.</p>
	message	<p>QMessageBox 를 통해 yes or no 를 설정해준다.</p> <p>Yes를 누르면 displayGameinit 메시지를 통해 게임을 초기화 해주고,</p> <p>No를 누르면 ScoreDB를 저장한 후, 게임 윈도우 창을 닫아준다.</p>
	gameover	<p>self.map 에 빈 공간(0 블록)이 있으면 False 를 리턴한다.</p> <p>어느 방향으로도 더 움직일 수 없으면 True 를 리턴한다.</p>
	success	<p>self.map에 2048블록이 있으면 True를 리턴한다.</p>

Class MoveBlock		
Attribute (properties)	nextMap	키보드 방향키를 누를 때, map을 이동시킨 후의 map을 nextmap에 2차원 리스트 배열로 저장한다.
	NextScore	키보드 방향키를 누를 때, map을 이동시킨 후의 score를 nextScore에 int로 저장한다.
	sKey	키보드 방향키에 따라서 'L','R','U','D'중 하나를 저장한다.
	N	mapsize를 저장한다.
Method	nextMove	arrT 매서드를 통해 sKey에 따라 배열을 transpose 하거나 하지 않고, moveZero 매서드를 통해 sKey에 따라 0을 모두 왼쪽으로 보내거나 모두 오른쪽으로 보낸다. sKey가 'L' 또는 'U'이면 범위를 0부터 self.N-1까지로 잡고, 'R' 또는 'D'이면 범위를 self.N-1부터 0까지로 잡는다. 다음 column을 구하기 위해 num을 설정한다. 다음 column이 0과 self.N 사이에 있고, 다음과 현재 column에 해당하는 블록이 같은 숫자라면 다음 column 블록을 두 배 해주고, 현재 column 블록은 0으로 설정해준다. 그 두 배 해준 값을 nextScore에 더해주고, 다시 sKey에 따라 0을 모두 왼쪽으로 보내거나 모두 오른쪽으로 보낸다. 이것을 반복한다. 그 후 다시 sKey에 따라 배열을 transpose 하거나 하지 않는다. 이렇게 구한 nextMap과 nextScore를 리턴해준다.
	arrT	sKey에 따라 배열을 transpose 하거나 하지 않는 함수이다. sKey가 'L' 또는 'R'이라면 transpose를 하지 않고 바로 리턴해준다. sKey가 'U' 또는 'D'라면 2차원 리스트 배열을 위 또는 아래로 검사할 수 없기 때문에 numpy 모듈을 사용해 배열을 transpose 해준다.
	moveZero	sKey에 따라 0을 모두 왼쪽으로 보내거나 모두 오른쪽으로 보내는 함수이다. sKey가 'L' 또는 'U'라면 모든 블록을 왼쪽으로 보내야 하기 때문에 모든 0을 제거한 뒤, 그 개수만큼 0을 왼쪽에 붙여 넣는다. sKey가 'R' 또는 'D'라면 모든 블록을 오른쪽으로 보내야 하기 때문에 모든 0을 제거한 뒤, 그 개수만큼 0을 오른쪽에 붙여 넣는다.

Class Button		
Attribute (properties)	img	Item클래스에서 img를 받아와서 QPixmap으로 저장한다. paintEvent를 통해 버튼에 그려진다.
	size	Item클래스에서 size를 받아와서 저장한다.
Method	setImg	Img를 다시 설정하고, 업데이트 해서 버튼에 그려준다.
	sizeHint	size를 받아와 버튼의 가로 세로를 같은 크기로 맞춘다.
	paintEvent	Pixmap형식의 img를 그려주고 업데이트한다.

Class Item		
Attribute (properties)	itemGroup	background, block, menuBtn, restartBtn, arrow, mapPhoto, startBtn, rankBtn 각각의 size와 QPixmap 형태의 img를 dictionary 형태로 저장한다. img가 여러 개면 리스트로 만들어 저장한다.