



EVENT



핵심 정리

예제 Project

Project Name	EVENT
Project Type	Empty qmake Project
Module	QT += widgets
Source	main.cpp mouse_event.h keyboard_event.h window_event.h



핵심 정리

- 마우스 이벤트가 발생하면
 - ⇒ 약속된 가상함수가 호출.
 - ⇒ 마우스 이벤트를 처리하려면 해당하는 가상함수를 **override**
 - ⇒ 이벤트 종류에 따른 추가적인 정보(마우스 좌표등)는 인자로 전달.

```
void mouseDoubleClickEvent(QMouseEvent *event)
void mouseMoveEvent(QMouseEvent *event)
void mousePressEvent(QMouseEvent *event)
void mouseReleaseEvent(QMouseEvent *event)

void enterEvent(QEnterEvent *event)
void leaveEvent(QEvent *event)
void wheelEvent(QWheelEvent *event)
```



핵심 정리

● 키보드 관련 가상함수

```
void keyPressEvent(QKeyEvent *event)
void keyReleaseEvent(QKeyEvent *event)
```



● 키보드 번호

scan code

키보드가 컴퓨터로 보내는 번호
H/W적인 배치, Q(16), W(17), E(18) 등

virtual
key code

알파벳 순으로 키보드에 부여된 번호
대소 문자 구별 안함
A, a 모두 65, B 와 b 모두 66

ascii code

키보드가 아닌 문자에 부여된 번호
A=65, a=97



핵심 정리

- 윈도우를 “움직이거나 크기를 변경하는 등의 작업을 수행할 때 도 이벤트 발생”
⇒ 해당하는 가상함수가 호출됨.



Drawing



핵심 정리

예제 Project

Project Name	DRAW
Project Type	Empty qmake Project
Module	QT += widgets
Source	main.cpp draw1.h draw2.h draw3.h



핵심 정리

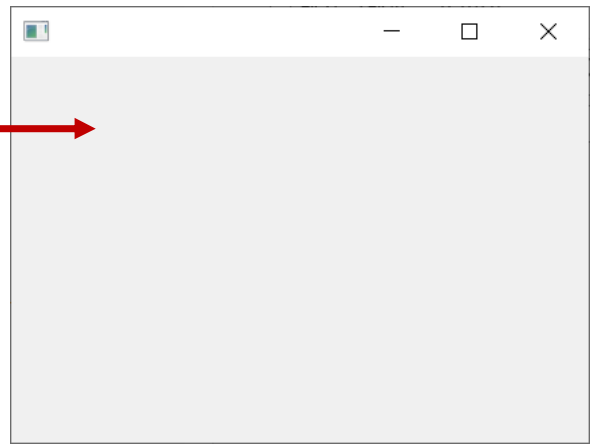
- 윈도우에 그림을 그리려면
 - ⇒ 규칙 1. `paintEvent` 가상함수에서만 그릴 수 있다.
 - ⇒ 규칙 2. `QPainter` 클래스를 사용해서 그린다.
- `QPainter` 클래스
 - ⇒ 그림을 그리기 위한 다양한 멤버 함수를 제공하는 클래스
- `mouseMove` 등에서 그리려면(서명 등)
 - ⇒ `image` 에다 그려야 한다.(별도의 강좌에서 설명)



핵심 정리

- `paintEvent` 함수는 언제 호출되나 ?
 - ⇒ 윈도우를 다시 그려야 할 때..
 - ⇒ 크기가 변하거나, 작아졌다 커지거나 ...

다시 그려져야 할 때마다
`paintEvent` 가상함수가
호출됨



- 코드로 윈도우가 다시 그려져야 한다고 명령하려면
 - ⇒ `QWidget::update()` 또는 `QWidget::repaint()`
사용



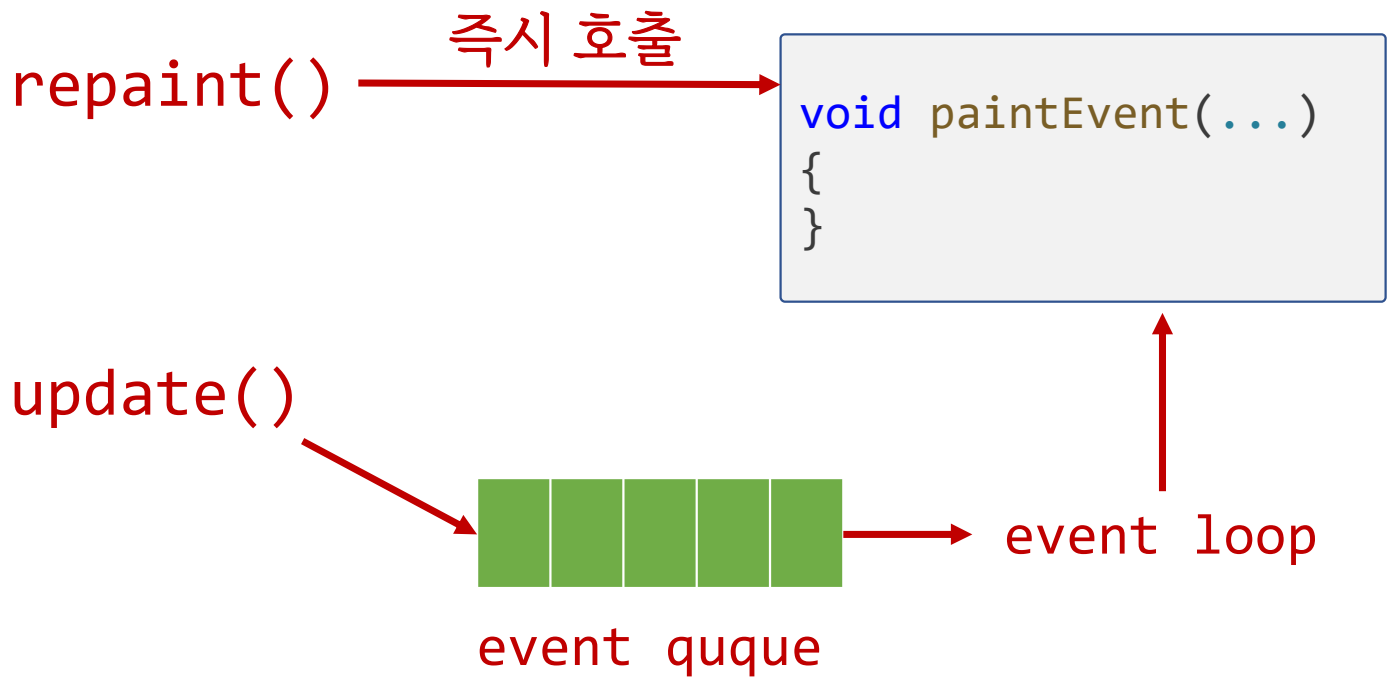
핵심 정리

- 다시 그려야 하는 영역의 크기 지정
 - ⇒ QPainter 로 그림을 그릴 때 해당 영역에만 출력할 수 있다.
 - ⇒ update() 할 때 다시 그려야 할 필요가 있는 부분의 크기를 전달하는 것이 좋다.



핵심 정리

- update() vs repaint()





Sliding Puzzle Game



핵심 정리

● 예제 Project

Project Name	PUZZLEGAME
Project Type	Empty qmake Project
Module	QT += widgets
Source	main.cpp puzzlegame.h

● **puzzle.jpg**

- ⇒ 게임에 사용할 그림
- ⇒ 다른 포맷도 가능
- ⇒ 적절한 크기로 준비





Step 1. 기본 코드

- 필요한 헤더 파일 포함



↑
EMPTY

- PuzzleWindow 클래스**

⇒ 블럭의 갯수를 관리할 상수

⇒ 2개의 가상함수 재정의



Step 2. 윈도우에 그림 출력하기

- **QPixmap 클래스**

- ⇒ 다양한 포맷의 이미지를 로드해서 출력할 수 있는 클래스.

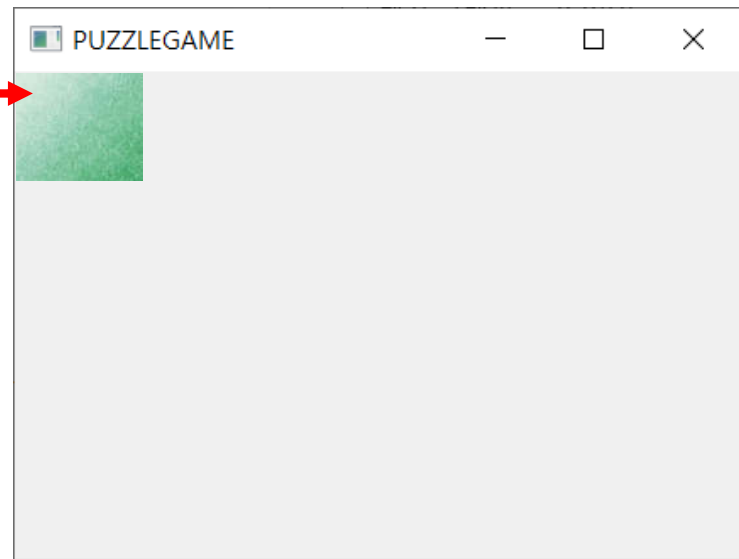
- ⇒ 전체 또는 부분 출력이 가능하다.

- **윈도우 크기 조절**

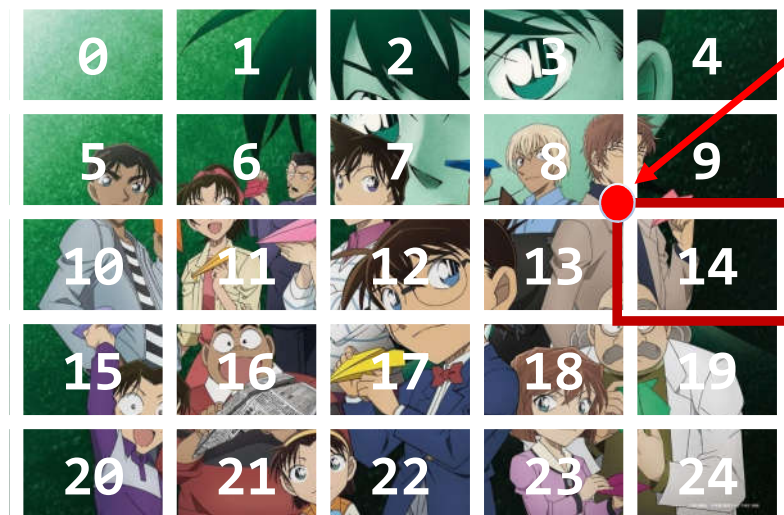
- ⇒ `setFixedSize()`



Step 3. 그림을 블록 단위로 출력하기



- 블록의 크기를 관리할 변수 필요
- 0번째 블록이 아닌 원하는 블록 출력



이곳의 x, y 좌표를
알아야 한다.

14번째 블록은

$14 \% \text{ COUNT} \Rightarrow 4$

$14 / \text{ COUNT} \Rightarrow 2$



Step 4. 게임판의 상태를 나타내는 2차원 배열



핵심 !!!!

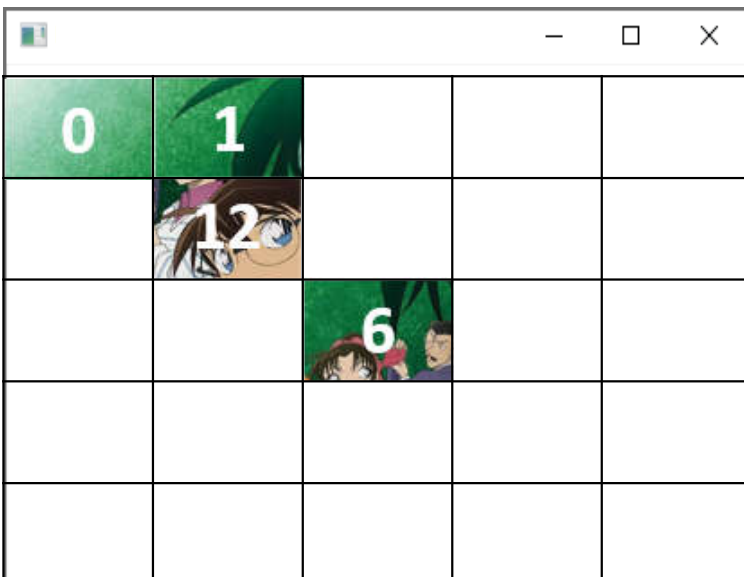


```
board[COUNT][COUNT]
```

게임판의 상태를
나타내는 2차원 배열

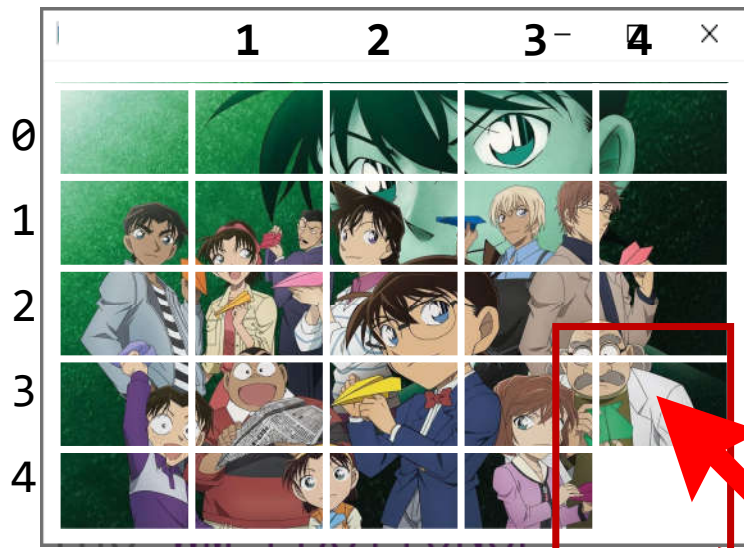
0	1	2	3	4
5	12	7	8	9
10	11	6	13	14
15	16	17	18	19
20	21	22	23	24

배열에 담긴 값으로
화면에 24개의 블록을
그린다. EMPTY가 담긴
곳은 흰색 사각형 으로.





Step 5. 블록 이동 하기



0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	24
20	21	22	23	19

`board[COUNT][COUNT]`

2개의 블록을 다시 그리게 한다.

- ① 마우스 좌표로부터 x축, y축으로 몇 번째 블록인지 조사 ($x = 4, y = 3$)
- ② board 배열을 조사해서 상/하/좌/우에 EMPTY가 있는지 조사
⇒ $x < 4$ 의 경우만 오른쪽을 조사하면 된다.
- ③ EMPTY가 있는 경우 2개의 배열 값을 교환 하고 2개 블록의 영역을 다시 그리게 한다.(update)



도전 과제

- ① update() 할 때 영역의 크기를 계산해서 꼭 필요한 부분만 update
- ② 다 맞추었는지 확인 하는 기능
- ③ 섞는 기능(메뉴 등을 추가)
- ④ COUNT 를 변경할 수 있는 기능(5 => 7)
- ⑤ 그림을 변경할 수 있는 기능
- ⑥ 맞추는 데 걸린 시간, 횟수 등을 관리하는 기능(score)



Sketch Example



핵심 정리

예제 Project

Project Name	SKETCH
Project Type	Empty qmake Project
Module	QT += widgets
Source	main.cpp sketch.h

SketchWindow 에 4개의 가상함수 override

```
void paintEvent(QPaintEvent *event)

void mouseMoveEvent(QMouseEvent *event)
void mousePressEvent(QMouseEvent *event)
void mouseReleaseEvent(QMouseEvent *event)
```



핵심 정리

paintEvent가 아닌
함수에서 그리기



QImage 객체

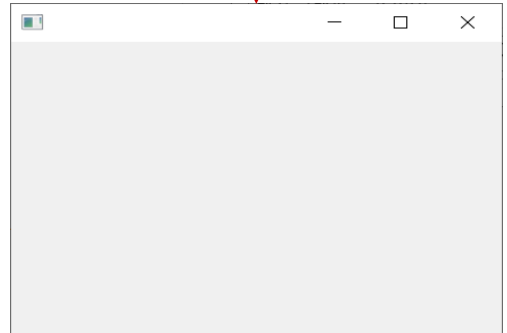
paintEvent()



윈도우에 직접 그리려면



paintEvent 에서 그려야 한다

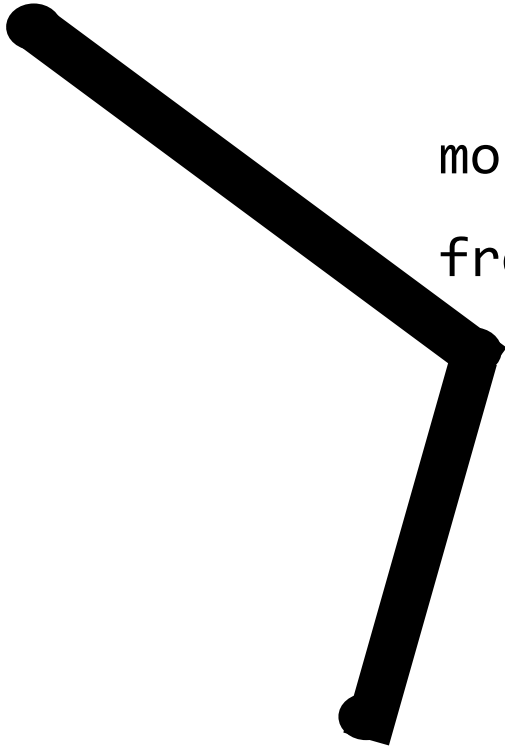




핵심 정리

마우스 버튼을 누르면

현재 좌표를 보관하고(from 변수)



mousemove 에서

from~현재좌표까지 선을 그리고

from = 현재좌표

mousemove 에서

from~현재좌표까지 선을 그리고