



소프트웨어공학개론

Electronic Door Lock System

Team #3: 5bps
김태현 201211341
송병우 201011338
은다함 201013388
장호성 200611515



Unit test 설계

EDLS.UTC_019_000	2.4.2.3 Disable user input	Trigger 되어 사용자 입력 S/B flag를 끈다.	EDLS.UTC_019_000	Trigger in	(SBFlags>>NB&1==0)&& (SBFlags>>KS&1==0)
EDLS.UTC_020_000	2.4.2.4 Initialize1	Trigger 되어 사용자 입력 S/B flag를 켜다.	EDLS.UTC_020_000	Trigger in	(SBFlags>>NB&1==1)&& (SBFlags>>KS&1==1)
EDLS.UTC_020_001	2.4.2.4 Initialize1	Trigger 되어 PQ에 타이머 프로세스를 삭제한다.	EDLS.UTC_020_001	fn==Timer	PQManager.find(fn)<0
EDLS.UTC_021_000	2.4.2.5 Self terminate	PQ에서 Authorize 프로세스를 삭제한다.	EDLS.UTC_021_000	fn==Auth	PQManger.find(fn)<0

```
assert ("EDLS.UTC_019_000", !BitGet(SBFlags, KS+MASK) && !BitGet(SBFlags, DB+MASK));
```

```
assert ("EDLS.UTC_020_000", BitGet(SBFlags, KS + MASK) && BitGet(SBFlags, DB + MASK));
```

```
assert ("EDLS.UTC_020_001", !PQ->find(timer, NULL, NULL));
```

```
assert ("EDLS.UTC_021_000", !PQ->find(authorize, NULL, NULL));
```

Unit test 방법

- Assert문 설계 : 런타임 UT이므로 **옳은것은 표현 안한다.**

```
int assert(char *msg, int val){  
    if (!val){  
        gotoxy(0, 22);  
        printf(msg);  
    }  
    return !!val;  
}
```

- **사용**

- 예) PQ포인터가 비어있지 않은지 확인

```
assert("EDLS.UTC_003_000", PQ);
```

- 비어있다-> PQ==0 -> UTC번호 출력 -> 0반환
- 비어있지 않다 -> PQ!=0 -> 1반환

- **응용 : 반환 값을 이용**

- If(assert("...",...)) 로 옳은 경우만 실행되는 코드 작성 가능
- If(!assert("...",...)) 로 틀린 경우만 실행되는 코드 작성 가능



Unit test 적용 결과

- 버그 발견

- 단순 UT 조건문 오류
 - EDLS.UTC_009_000, EDLS.UTC_010_000, EDLS.UTC_011_000, EDLS.UTC_011_001
- 타이머 관련 버그 : EDLS이 열린 상태에서 문을 다시 열려고 하는 경우
 - EDLS.UTC_005_000, EDLS.UTC_013_000

- 후속 조치

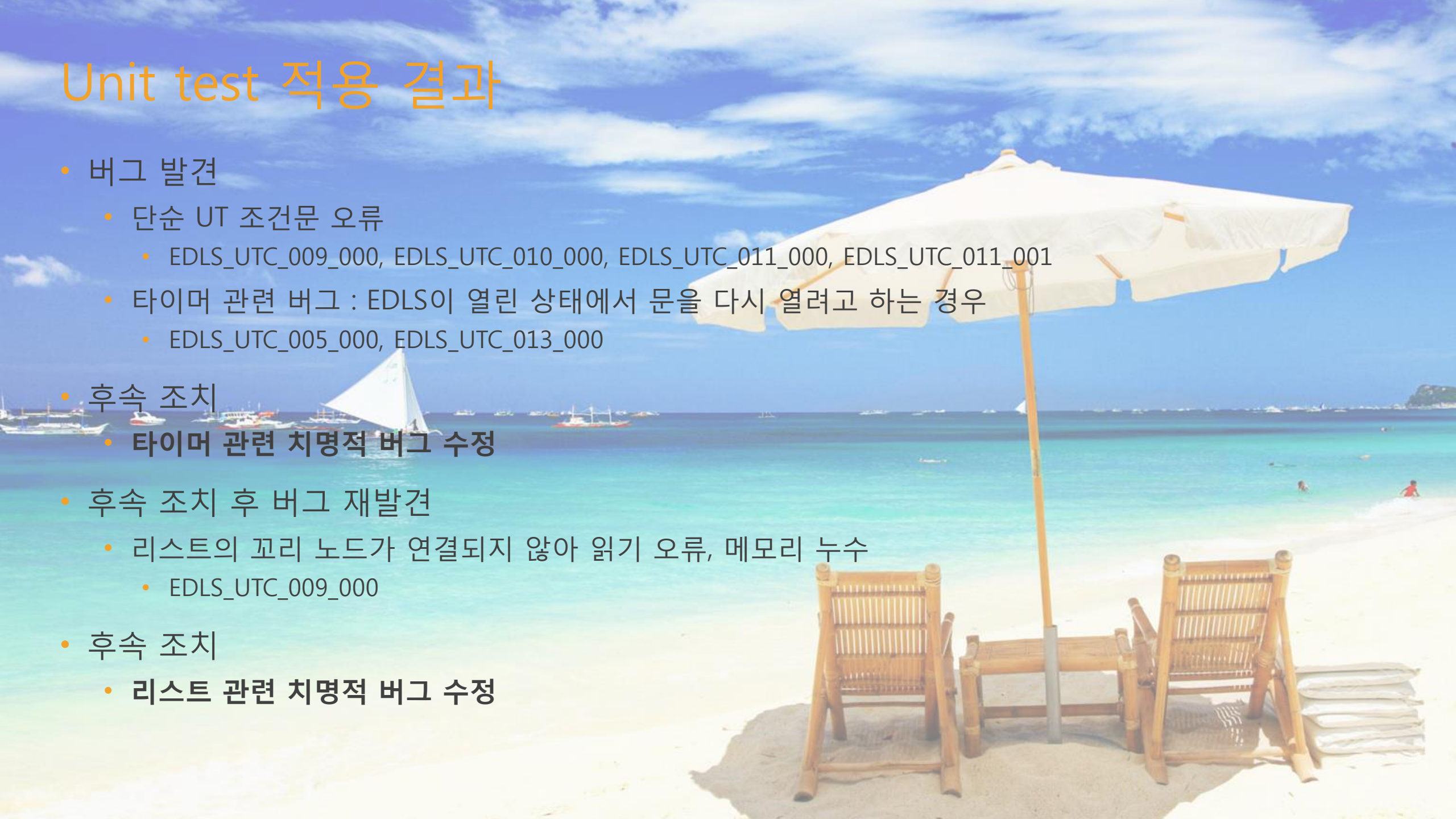
- 타이머 관련 치명적 버그 수정

- 후속 조치 후 버그 재발견

- 리스트의 꼬리 노드가 연결되지 않아 읽기 오류, 메모리 누수
 - EDLS.UTC_009_000

- 후속 조치

- 리스트 관련 치명적 버그 수정



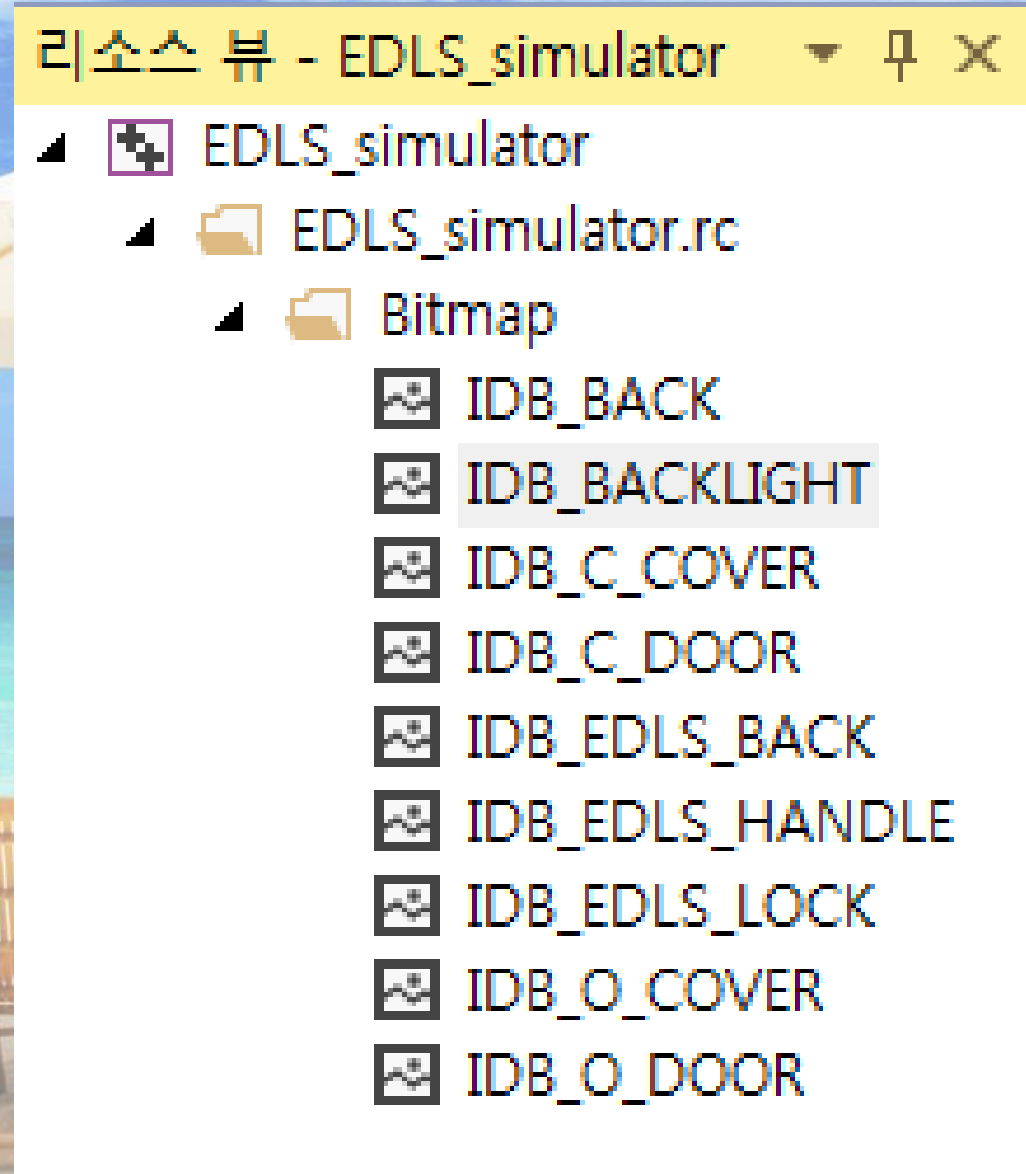
Interval fixation

- 무한루프 과정에서 Sleep(TICK) 을 사용하면 실제 루프는 루프 런타임+TICK의 런타임을 잡아먹는다. 이를 보정하기 위해서 Sleep 시간에서 루프 런타임을 빼준다
 - 매 루프의 실행시간을 TICK ms로 보장.

```
while (!GetAsyncKeyState(VK_ESCAPE)){
    time(&ts); //측정시작
    ... //루프 런타임 소비
    time(&te); //측정 끝
    timelapse = te - ts; //비교
    if (timelapse < TICK && timelapse >= 0)
        Sleep(TICK - timelapse); //보정
    else
        printf("Wn#####runtime calculation error#####Wn");
}
```

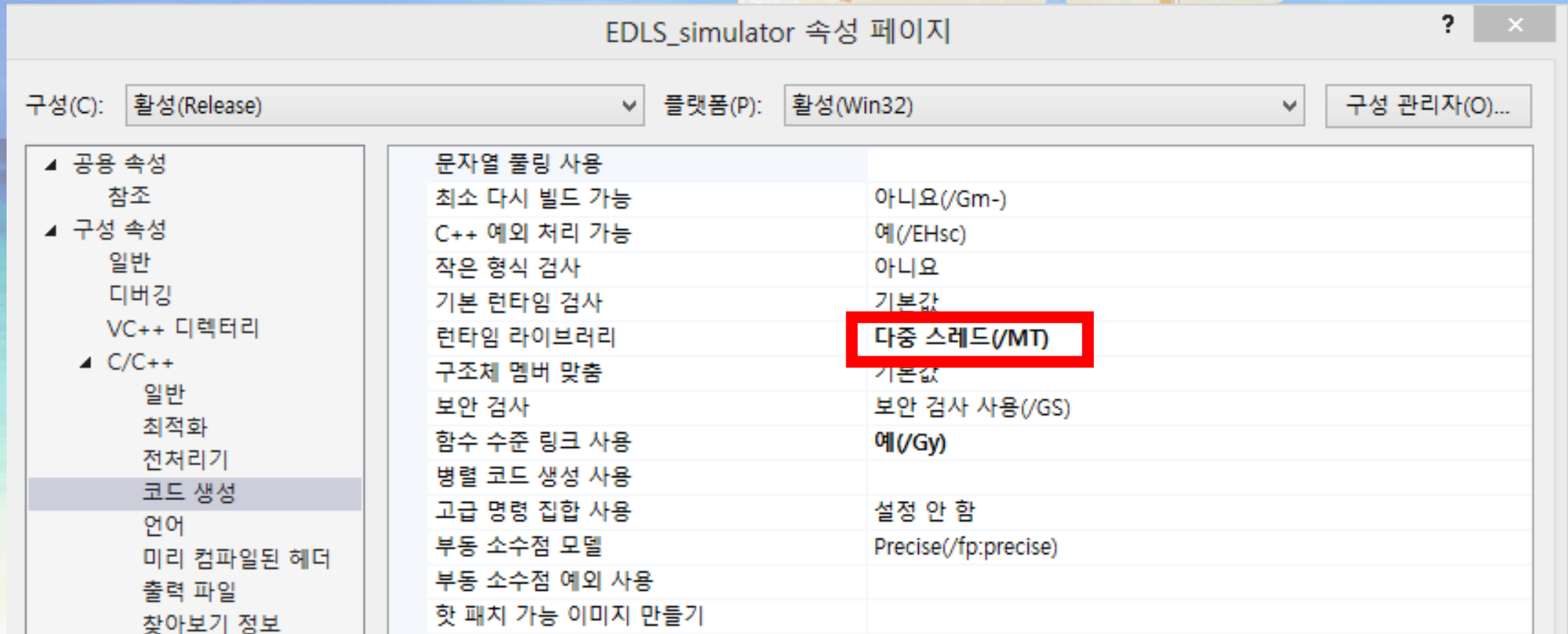
GUI 적용

- 자작 비트맵 그리기 함수 적용
 - 다중 HDC레이어 합성
 - 1함수 다중 출력
 - 투명색 지정기능
 - 회전 그리기 기능
 - 모터 손잡이에 적용
- 그림판 노가다로 리소스 확보
- 적용 비트맵의 리소스화
 - EXE에 포함된다.
 - 2KB->1.2MB EXE 크기 증가
- 게임처럼 리소스가 많아진다면?
 - 리소스 설계방법?



시연

- 컴파일 시 독립실행 불가능 -> 다른 환경에서 실행 불가
- 독립실행 가능하도록 컴파일 설정 변경
 - 프로젝트 속성에서 **다중 스레드 /MT** 옵션을 설정



시연

※본 프리젠테이션의 배경 화면은 GUI의 클리셰임.

실행

QNA

