**시스템 프로그래밍 팀 프로젝트 보고서**

**프로젝트명: System\_BB**

2013020011 김윤흥

2014105052 신주용

2015111255 육태훈

2014105107 황재성

**1. 벽돌깨기 제작 이유**

* 고전게임의 대표 게임 중 하나

(갤러그도 생각했으나 오브젝트의 공격기능 관리문제 및 디자인의 폭이 좁음)

* 수업 시간에 7장(ncurses), 14장(thread)에서 배운 내용을 심층적으로 구현 가능
* 간단하면서 다양한 시스템을 추가하기 용이하고 확장성이 좋아보였음
* 역할 분담이 테트리스, nsnake, pong 등의 게임보다 수월
* 그래픽이 비교적 간단하고 cursor의 부하가 적어서 선택

**2. 개발 과정**

* 0주차 : 각자 프로젝트 주제 조사
* 1주차 : 게임의 방향성에 대해 토의 & 프로젝트 선정
* 2주차 : 각자 생각하는 방향으로 프로토타입 개발 시작
* 3주차 : Github repository 생성 및 각자 프로토타입 업로드
* 4주차 : 코드 비교 & 통합
* 5주차 : 파트 나눠서 개발 시작
* 6주차 : 버그 수정 & 개선
* 7주차 : 최종 버그 수정 & 발표 준비

**3. 역할 분담**

* 김윤흥 : 시스템 UI 개발
* 신주용 : master branch 관리
* 육태훈 : 게임 엔진 구현(브릭 핸들링 함수)
* 황재성 : 게임 엔진 구현(공 핸들링 함수)

**4. 개발 중 문제점 및 해결 방법**

* 화살표키 읽어오기 문제와 패들의 움직임 문제(쓰레드 활용)
* 기존에는 기본 출력함수로 단순하게 구현 계획 (V0.1)

⇒ ncurses라이브러리를 이용하여 인터페이스 개선

⇒ 윈도우 분할 및 색 출력 기능 사용

⇒ display, UI 등은 nsnake 게임의 형태를 차용.

* 개발 도중 공의 움직임에 대한 변수가 생각보다 다양해서 곤란

⇒ 로그 파일을 이용해 버그 해결(폭발 블럭, 공의 위치 파악)

⇒ 로그 파일은 공의 위치(움직임), 공의 블럭 타격, 폭발 블럭의 작동을 기록

* 맵은 처음에 코딩으로만 구현하다 맵 파일을 읽어오는 방식으로 전환

⇒ 다양한 모양의 맵 구현 가능.

* 화면 깨짐

⇒ 블록, 공, 패드 정보를 수정하는 쓰레드와 게임 화면을 출력하는 함수가 동시에 같은 정보에 접근하면서 화면이 깨지는 현상이 종종 발생

⇒mutex\_lock을 이용해 race condition 방지

**5. 핵심 라이브러리 및 코드 설명**

**(1) Header File**

* ncurses.h

⇒ 좌표를 이용한 출력, WINDOW를 이용한 화면 공간 분배, Color pair 사용

* pthread.h

⇒ 공의 움직임 및 벽돌 판정, 타이머를 멀티쓰레드로 구현

* BB.h

⇒ 다른 헤더파일, 매크로를 통합 관리

**(2) 함수 정의**

* initialize()

⇒ 벽돌마다 color pair 설정, 맵 설정 및 점수 박스, 공 위치 좌표 박스 초기화, 맵 배열 초기화 함수, 초기화면, 맵을 불러오지 못했을 경우 처리.

* refreshMap()

⇒ 벽돌, 공, 보드의 변경사항이 있을때마다 화면을 새로 출력해주는 함수

* setBallPos()

⇒ 현재 공의 위치를 파악, setBallDel()함수와 연계

* setBallDel()

⇒ 공이 보드나 벽 등의 충돌 판정에 대해 델타값을 변경

1. 벽과 충돌 시 delta값을 역으로 변경 + 예외 처리
2. 보드와 충돌 시 보드의 부위에 따라 공의 델타값을 다르게 변경
3. 벽돌과 충돌 시 deleteBrick() 호출

* \*ballThreadFunc()

⇒ 보드의 움직임과 공의 움직임에 대해 서로의 간섭을 피하기 위해 공의 움직임을

쓰레드로 분리하여 구현

* deleteBrick()

⇒ 공에 맞은 블럭의 처리 (1단계, 2단계, 3단계, 투명, 폭발 블럭) 및 스코어 계산

1. 2, 3단계 블럭의 경우에는 이전 단계로 이동
2. 1단계 블럭의 삭제
3. 투명 블럭의 경우 1단계 블럭으로 변경
4. 폭발 블럭을 때렸을 경우는 recursive call과 flag 변수를 통해서 처리
5. 폭발 블럭은 기본적으로 상하를 제거, 좌측을 맞으면 좌측, 우측을 맞으면 우측을 제거

**6. 프로젝트를 진행하면서 느낀 점 & 아쉬운 점**

1. **느낀 점**

* Github를 통해 협동 개발을 해볼 수 있는 좋은 기회였음
* 프로젝트의 규모가 커질수록 함수명과 변수명 관리의 필요성 체감
* 협업에 있어 코딩 스타일과 변수명의 통일의 중요성을 깨달음
* 시간 배분의 중요성에 대해 체감
* 수업시간에 배운 것을 응용하고 복습할 수 있는 좋은 기회였음

**(2) 아쉬운 점**

* 프로젝트가 복잡해질 것을 예상하지 못해 하나의 C code 파일로 관리해야해서 힘들었음 (github에서 Auto merge X).
* Github 명령어를 능숙하게 사용하지 못해 차질이 있었음
* 아이템, 맵 추가/편집 등 콘텐츠 개발 필요
* C언어 및 터미널 기반이라 게임 구현에 한계가 있었음.
* 충돌 판정 세분화 필요