

[506489] System Programming (F'18)

Term Project (Week 3, Final) Report

팀 정보

No.	이름	학번
1	정재민	20145170
2	이종학	20145153
3	김형훈	20145114
4		
팀 이름: 물음표		

<목차>

팀 정보.....	1
결과물 제출	2
보고서 작성 가이드	2
Part I: 최종 보고서	3
Part II: 데모/시연 시나리오 및 결과물	6
Part III: 프로젝트 진행중 발생한 문제점 및 해결방안.....	7
Part IV: 개선방향	8

* 보고서 작성이 완료되면 <목차>를 업데이트 해 주세요. (방법: “목차” 클릭>”목차 업데이트”>”목차 전체 업데이트”)

결과물 제출

- SmartCampus 에 업로드 해야 하는 제출물 목록은 다음과 같습니다.
 - Report(프로젝트 보고서), **PDF 형식**
 - 프로젝트 발표자료 (ppt, pptx, pdf 등)
 - GitHub 프로젝트를 다운로드 한 zip 파일
- 제출 기한을 넘기면 자동으로 0 점 처리됩니다.
- 팀 과제이며, 팀별로 한 명만 제출하면 됩니다.

* 보고서 및 발표자료도 GitHub 에서 관리해야 합니다!!!

보고서 작성 가이드

- Week 3 시작: 2018. 12. 12. (Wed) 1:00pm
- Week 3 마감: 2018. 12. 17. (Mon) 12:59pm
- **매 주차별로 반드시 1 회 이상 GitHub 에 commit 한 내역이 있어야 합니다.** (팀별로 1 회 이상, 개인별 1 회 아님). GitHub 에 commit 한 내역이 없을 경우, **큰 감점**을 받게 됩니다.

본 보고서는 총 3 개의 파트로 구성되어 있습니다.

- Part I: 최종 보고서
 - 1 주차 보고서 내용 전체를 포함하여 작성하고, 그 동안 추가/수정된 내용이 있다면 그에 맞게 보고서를 수정해야 합니다.
- Part II: 데모/시연 결과물
 - 개발한 프로그램이 정상적으로 구동된다는 것을 보여주세요.
 - 데모/시연을 위해 사용한 시나리오를 자세하게 설명하고, 해당 시나리오에 대한 단계별 데모 결과물(예: 스크린샷)을 첨부하거나 또는 데모 영상을 촬영해서 첨부파일로 제출해도 됩니다.
- Part III: 프로젝트 진행중 발생한 문제점 및 해결방안
 - 2 주차 보고서 내용중 “프로젝트 진행 중 발생하는 문제점 및 해결방안”에 대한 내용을 포함하는 내용으로 작성하세요.
 - 그 외에 추가로 발생한 문제점이 있었다면 해당 문제점 및 해결방안을 기술하세요.
- Part IV: 개선방향
 - 개발 결과물을 앞으로 어떻게 개선할 수 있을지에 대한 의견을 서술하세요.

Part I: 최종 보고서

[프로젝트 제목]

NCURSES를 이용한 벽돌 깨기 게임

[프로젝트 개요]

- 선정 동기

게임을 좋아하는 팀원들끼리 프로젝트를 구상하다가 리눅스 NCURSES를 이용해서 게임을 만들어 보자는 생각으로 게임을 만들게 되었습니다.

- 프로젝트 내용

리눅스 소켓 프로그래밍과 리눅스 NCURSES를 이용한 벽돌깨기 게임 입니다

[프로젝트 개발목표 및 개발/구현 내용]

- 프로젝트 개발목표

단순히 벽돌게임을 하는 것이 아니라 게임을 이용하는 시간대마다 게임의 성적을 측정해서 어느 시간대에 성적이 좋고 나쁜지에 대한 데이터를 저장하고 이러한 데이터를 나중에 다른 게임을 만들 때 고려 할 수 있다면 좋겠다고 생각했습니다.

- 프로젝트 개발/구현 내용

C를 이용해서 로그인, 회원가입 버튼을 구현한다.

로그인을 하면 게임을 시작한다.

Single 플레이 모드 클릭 시 바로 게임을 시작한다.

-----게임-----

게임 KEY : 왼쪽 : J, 오른쪽 : L, 공 발사 : K

발사 시(K키 입력) 공이 동작하는 것을 구현

벽을 맞을 때 공이 반대 반사 각으로 튕기는 것을 구현

BAR의 오른쪽을 맞을 때 오른쪽으로 튕기는 것을 구현

BAR의 왼쪽을 맞을 때 왼쪽으로 튕기는 것을 구현

BAR의 가운데를 맞을 때 기존에 공이 오는 각의 반사각으로 튕기는 것을 구현

BLOCK을 맞을 때 블록의 왼쪽 오른쪽 위쪽 아래 쪽 맞을 때 반사각을 따로 구현

BLOCK의 오른쪽이나 왼쪽을 맞을 때 왼쪽 오른쪽 위로 튕기는 것을 구현

BLOCK의 아래쪽 맞을 때 왼쪽, 오른쪽 아래로 튕기는 것을 구현

BLOCK의 위쪽 맞을 때 왼쪽 오른쪽 위로 튕기는 것을 구현

BALL이 BAR보다 아래쪽으로 갔을 때 게임종료

게임종료시 점수 저장하는 것 구현

-----소켓-----

서버를 열면 클라이언트가 접속

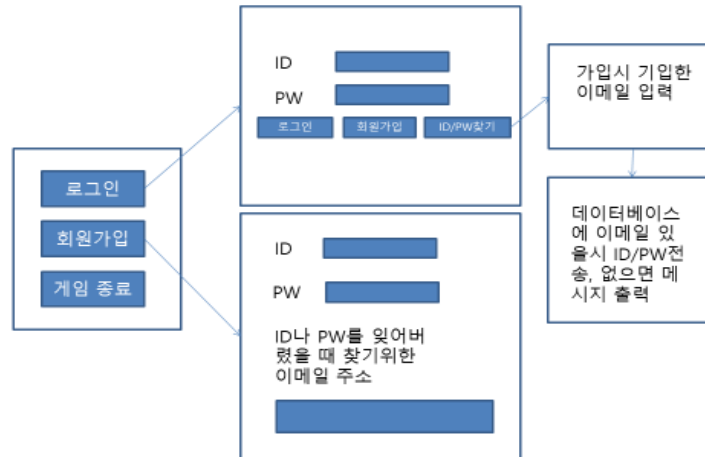
서버에서 어떤 클라이언트가 접속하였는지 확인

클라이언트가 게임을 플레이 한 후에 점수를 저장한 텍스트 파일을 이용하여 순위를 지정

클라이언트가 종료하면 서버에서 어떤 클라이언트가 종료하였는지 확인

-----GUI-----

[프로젝트 구성도]



회원가입을 한 뒤에 로그인을 할 수 있도록 구현한다.



로그인을 하고 게임 종류를 선택한다.(MULTI 추후 개선)

|

게임을 시작한다.

[기대효과 및 활용방안]

시간대마다 성적이 좋고 나쁜지에 대한 데이터를 이용해서 다른 게임을 만들 때 성적이 나쁜 시간대에는 집중이 잘 안되고 피곤한 것으로 판단하여서 게임 내부에서 접속중인 유저들에게 이벤트를 발생시키는 시간대를 결정하고 성적이 좋은 시간대에는 유저들이 게임을 즐기고 집중이 잘될 때는 대회를 여는 시간대를 결정할 때 사용합니다. 이 프로젝트가 성공할 시 다른 게임들을 추가할 것이며 게임회사의 한 획을 그을 것입니다.

Part II: 데모/시연 시나리오 및 결과물

영상 첨부

Part III: 프로젝트 진행중 발생한 문제점 및 해결방안

GUI 부분에서는 Gtk 버튼 클릭에 대한 액션 처리하고 싶었지만 계획한 알고리즘대로 잘 되지 않았다. 그래서 Gtk 를 사용하지 않고 콘솔창에서 메뉴선택에 따른 처리를 하도록 하였다. Gtk 부분에서 버튼을 클릭하면 두 개 이상의 데이터(ID 와 Password)를 ActionHandler 로 전송하여 처리하고 싶었지만 이 부분이 너무 어려워서 Gtk 를 사용하지 않을 수밖에 없었습니다.

게임 부분에서는 막대기를 움직일 때 계속 누르고 있으면 입력이 중첩되어서 바의 움직임을 제어할 수가 없게 된다. 블록과 막대기 사이에 거리가 너무 좁아서 공이 튕기는 각을 줄이는 것이나 막대기와 블록 사이의 간격을 넓히는 것을 적용하였다. 이제 막대기의 어느 부분에 맞으면 왼쪽 오른쪽을 구별하여 왼쪽을 맞으면 왼쪽으로 튕기고 오른쪽을 맞으면 오른쪽으로 튕기고 가운데를 맞으면 기존에 있던 반대각으로 튕기게 하였다.

소켓 부분에서는 원래의 계획은 다수의 클라이언트가 서버에 접속하여서 서로 경쟁할 수 있는 Multi 벽돌깨기 게임을 구성하고 싶었지만 예상외로 구현하기가 어려워 Multi 게임은 구현하지 못하였고 각자 실행할 수 있는 Single 벽돌깨기 게임을 구현하였다. Single 게임만 가능하지만 다른 클라이언트들이 실행하였던 점수를 파일에 저장하여 서버에서 클라이언트의 랭킹을 확인할 수 있게 하였다.

Part IV: 개선방향

GUI 부분에서 클라이언트가 게임을 마친 후 첫 페이지로 돌아가면 다시 로그인 해야 하는 번거로운 부분을 수정하고 로그인 되어있으면 로그아웃 할 수 있게 추가할 것이다. 콘솔창을 이용하지 않고 ncurses 나 gtk 를 사용하여 더 깔끔한 디자인을 구현할 것이다.

게임 부분에서는 막대기를 계속 누르고 있으면 막대기를 제어할 수가 없다. 예를 들어, 막대기를 왼쪽 방향으로 꼭 누르고 있다가 오른쪽 방향으로 바뀌서 누르면 입력을 받지 못하고 계속 왼쪽 방향으로 향한다. 키보드의 입력을 확실하게 받을 수 있게 수정할 것이며 막대기의 움직임 뿐만 아니라 공의 움직임도 더 부드럽게 할 것이며 게임의 난이도를 추가할 것이다.

소켓 부분에서는 서버에서 클라이언트들의 점수를 확인할 수 있지만 클라이언트들은 자신을 포함한 다른 클라이언트들의 점수를 확인하지 못하여 클라이언트들의 랭킹을 구현하지 못하였다. GUI 부분에 랭킹 확인 창을 추가하여 클라이언트들이 랭킹 확인 창에 들어가면 랭킹을 확인할 수 있게 수정할 것이며 클라이언트가 ID 나 Password 를 까먹었을 때 이메일 전송으로 정보를 찾을 수 있게 추가할 것이다.

[끝]