

2021년도 2학기
교과목 포트폴리오

20202827 김후목

목차

1. 강의 계획서
2. 깃허브
3. 자기주도 학습 내용

1. 강의 계획서

2021년도 2학기 강의계획서

교과목명	융합작품(종합설계)	과목 개설 전공	컴퓨터정보공학과
강의실 및 강의시간	화:5(3-218),6(3-218),7(3-218),8(3-218)		
교과분류	실습	학점	4 (시수 : 4)
담당교수	강환수	교수 소속 학부	컴퓨터공학부
교수연구실	2호관-706	연구실 전화번호	02-2610-1941
이메일	hskang@dongyang.ac.kr	면담가능시간	화 11:00~11:50

교육목표	
과목개요	○ 수업운영방안 : 비대면(코로나19 방역상황 호전시 대면으로 전환) ○ 온라인 수업 운영방법 : 녹화형/실시간 화상 수업으로 병행하여 진행 ○ 과목 개요 : 컴퓨터정보공학과에서 학습한 내용들을 종합적으로 활용하여 실제로 프로젝트를 구현해 보는 과목으로서 팀별로 하나의 프로젝트를 완성하는 것을 목표로한다. 이러한 프로젝트 개발 활동은 졸업 후 사회에 진출했을 때 실제적으로 할하게 될 일의 사전준비가 될 것이다. 또한 하나의 프로젝트를 처음부터 시작하여 끝까지 스스로 완성하여 봄으로서 학습내용들을 체계화하고 실용적인 실력을 키우는 데 그 목적이 있다. 팀별로 완성된 작품은 졸업작품 최종 발표와 학술대회 등을 통해 공개되며 평가된다.
학습목표 및 성취수준	대학 교육목표 및 학과 교육목표를 달성하기 위하여 본 교과목은 컴퓨터정보공학과 교육과정에서 배운 다양한 기술을 이용하여 팀별로 하나의 프로젝트를 수행해 프로젝트의 작품을 개발하는 것을 목표로 한다. 본 과목을 통해 다음을 성취할 수 있다. - 팀별로 자신의 역할을 감하고 팀 프로젝트 작품을 기획하고 설계할 수 있다. - 작성된 설계서를 기반으로 지금까지 배운 과목별 학습내용을 이용해서 팀 프로젝트 작품을 개발할 수 있다.
수업시 사용도구	각 팀별로 프로젝트 개발 시 필요한 재료 및 도구
평가방법	출석 20% + 중간고사(수행과정 평가) 30%(프로젝트보고서 1차발표/5, 2차발표/10, 중간제인/15) + 기말고사(과정 및 최종발표 평가) 30%(프로젝트 최종발표/15, 최종보고서/5, 개인 팀내기여도/10) + 교내외 학술대회 및 작품전 참가 및 수상 10% + 취업활동 및 기타 10%
수강안내	강의완료 후 학생들은 팀 프로젝트의 일정으로서 하나의 작품을 개발하기 위해 작품의 기획부터 개발, 그리고 발표까지 프로그램의 개발 프로세스 모든 과정을 경험할 수 있으며 자신의 역할을 충실히 수행할 수 있다. 선수과목으로 1학기의 시스템분석설계 과목을 학습한 학생들이 수강해야 한다

구분	해이름	저자	출판사	비고
주교재	유인물	-	-	

주	개요	내용	목표	교재	과제
1	[개강일(9/1), 수강신청(9/6~9/7)]	〈강의의 소개〉 첫 시간의 수강 학생을 확인과 이 과목에서 학습하게 되는 융합 작품에 대한 개요 및 강의 방법, 평가 방법 등에 대해 설명한다. 특히, 시스템분석설계 과목과 연계된 융합작품의 진행 방법과 완성된 융합작품을 평가하기 위한 졸업작품평가회와 전산학부학술대회의 순회에 대해서 소개한다. 더불어 강의 목표와 강의 시 유의사항에 대해서도 설명한다.	1. 학습대상자 출석 및 선수과목 이수 여부 확인 2. 융합작품 강의의 내용 및 목표 소개 3. 학습 방법 및 학습 평가 소개 4. 팀구성 및 확인	1. 팀별 작품 진행 상황 파악 2. 발표 중 팀별 활동 보고 준비	
2	[2주]	〈팀별 작품진척도 점검 및 지도〉 각 팀별로 현재 진행 중인 팀 프로젝트 작품에 대한 작품 진행 상황 점검과 발표 중 팀별 활동 내역 파악	1. 각 팀별 프로젝트 구성 인원을 확인한다. 2. 발표 중 팀별 활동 내역을 발표하여 각 팀별 프로젝트 진행상황을 파악한다. 3. 각 팀별 프로젝트 진행 상황에 따라 앞으로의 프로젝트 개발 계획을 수립한다. 4. 팀별 프로젝트 진행 상황에 따라 자신이 맡은 역할의 세부 계획을 수립한다.		융합작품 프로젝트 보고서 1차발표
3	[3주, 추석연휴(9/20~9/21) -> 보강(9/30,12/14)]	〈팀별 작품 변경 사항 확인〉 팀별 프로젝트 작품의 기획과 달리 변경 또는 수정된 기능들을 확인하고 변경된 사용자 인터페이스 등을 점검한다. 이를 통해 최종적으로 개발해야 하는 팀 프로젝트 사용자 인터페이스 및 기능 등을 확정한다.	1. 팀 프로젝트 시작 시 기획 및 설계했던 내용과 변경된 기능 및 구현 내역을 확인한다. 2. 변경 및 수정된 부분에 대한 사용자 인터페이스와 기능에 대해 재설계를 진행한다. 3. 이에 따라 팀별 또는 개인별 개발 계획을 변경하고 수정한다.		변경된 최종 설계서
4	[4주, 추석연휴(9/22) -> 보강(12/8)]	〈팀별 작품 개발을 위한 관련 기술 학습〉 수정된 팀별 작품을 위해 필요한 관련 작품 개발 자료 및 기술들을 수집하고 이를 학습한다.	1. 수정 및 추가된 팀 프로젝트 기능들을 위해 관련 기술 자료들을 수집한다. 2. 수집된 작품 관련 소스들을 분석하고 이를 작품에 반영할 수 있도록 학습한다. 3. 작품들을 작품개발서버에 설치하기 위한 관련 자료들을 학습하고 이해할 수 있다.		
5	[5주, 개천절(10/3) -> 보강(10/4)]	〈팀별 작품 개발 진행〉 최종 융합작품 발표회 전까지 기획한 팀 프로젝트 작품의 개발을 진행하며 이를 통해 팀 프로젝트의 각 역할을 학습한다.	1. 전체적인 팀별 작품 개발 진도 및 진행 상황을 파악할 수 있다. 2. 팀 프로젝트의 각 역할별 기능별 진행 상황을 파악할 수 있다. 3. 팀 프로젝트 구성원별로 개발 진행 상황을 보고하고 개발시 문제점을 토론하고 해결할 수 있다. 4. 다음주까지 개발 계획을 확인하고 필요한 도구사함을 요청할 수 있다.		융합작품 프로젝트 보고서 2차발표

6	[6주, 한글날(10/9) -> 보컬(12/13)]	<p><팀별 작품 개발 진행> 최종 졸업작품 발표회 전까지 기획한 팀 프로젝트 작품의 개발을 진행하며 이를 통해 팀 프로젝트의 각 역할을 학습한다.</p>	<p>1. 전체적인 팀별 작품 개발 진도 및 진행 상황을 파악할 수 있다. 2. 팀 프로젝트의 각 역할별 기능별 진행 상황을 파악할 수 있다. 3. 팀 프로젝트 구성원별로 개발 진행 상황을 보고하고 개발시 문제점을 토론하고 해결할 수 있다. 4. 다음주까지 개발 계획을 확인하고 필요한 요구사항을 요청할 수 있다.</p>		
7	[7주]	<p><팀별 작품 서버 설치> 최종 졸업작품 발표회 전까지 기획한 팀 프로젝트 작품의 개발을 진행하며 개발된 작품을 서버에 설치하여 동작여부를 확인한다.</p>	<p>1. 현재까지 개발된 작품을 서버에 설치하고 이를 동작시킬 수 있다. 2. 전체적인 팀별 작품 개발 진도 및 진행 상황을 파악할 수 있다. 3. 팀 프로젝트의 각 역할별 기능별 진행 상황을 파악할 수 있다. 4. 팀 프로젝트 구성원별로 개발 진행 상황을 보고하고 개발시 문제점을 토론하고 해결할 수 있다. 5. 다음주까지 개발 계획을 확인하고 필요한 요구사항을 요청할 수 있다.</p>		각 팀 프로젝트 구성원별 기능별 개발 진행 사항 전공동아리 킨츠스 작품전 참가
8	[8주]	<p><중간고사> 지금까지 개발한 팀 프로젝트 작품의 중간 평가를 위해 개발 내용을 발표 자료로 만들고 이를 프리젠테이션한다. 2. 지금까지 개발한 팀 프로젝트 작품의 중간 평가를 위해 작품의 동작 과정을 시연한다.</p>	<p>1. 지금까지 개발한 팀 프로젝트 작품의 중간 평가를 위해 개발 내용을 발표 자료로 만들고 이를 프리젠테이션한다. 2. 지금까지 개발한 팀 프로젝트 작품의 중간 평가를 위해 작품의 동작 과정을 시연한다.</p>		중간고사 또는 보고서 발표
9	[중간고사 기간 : 10/2(화)~11/2(화)]	<p><팀별 작품 개발 진행> 최종 졸업작품 발표회 전까지 기획한 팀 프로젝트 작품의 개발을 진행하며 이를 통해 팀 프로젝트의 각 역할을 학습한다.</p>	<p>1. 전체적인 팀별 작품 개발 진도 및 진행 상황을 파악할 수 있다. 2. 팀 프로젝트의 각 역할별 기능별 진행 상황을 파악할 수 있다. 3. 팀 프로젝트 구성원별로 개발 진행 상황을 보고하고 개발시 문제점을 토론하고 해결할 수 있다. 4. 다음주까지 개발 계획을 확인하고 필요한 요구사항을 요청할 수 있다.</p>		각 팀 프로젝트 구성원별 기능별 개발 진행 사항
10	[10주]	<p><학부 학술대회 및 경진대회 준비> 학부에서 개최하는 학술대회 및 경진대회에 각 팀별 프로젝트 작품을 출품하기 위한 작업을 진행한다.</p>	<p>1. 각 팀별 현재까지 개발 진행 상황을 파악하고 학부 학술대회 및 경진대회 출품여부를 지도교수와 협의한다. 2. 학부 학술대회 및 경진대회 브로셔를 위한 작품 소개 자료를 작성한다. 3. 학부 학술대회 및 경진대회 발표자료 및 대모 동영상 등을 준비한다.</p>		학부 학술대회/경진대회 브로셔 학부 학술대회/경진대회 발표자료 학부 학술대회 대모 동영상
11	[11주]	<p><작품 설치 및 최종 테스트> 개발된 작품을 서버에 설치하고 각 모듈별 동작 과정과 전체 동작 과정을 테스트한다.</p>	<p>1. 개발된 작품을 서버에 최종적으로 설치 완료한다. 2. 각 모듈별(입, 연, 서버 등) 동작 과정을 테스트한다. 3. 전체 작품의 동작 시나리오를 만들고 이에 따라 테스트를 완료한다.</p>		작품 동작 시나리오
12	[12주]	<p><최종졸업작품 발표> 학과 교수님들을 평가위원으로 한 최종졸업작품발표회를 개최하여 각 팀별로 개발된 작품을 발표하고 시연한다. 이를 통해 졸업작품 과목을 평가한다.</p>	<p>1. 각 팀별로 팀 프로젝트 작품 발표자료를 만들어 졸업작품평가전에서 발표하고 개발된 작품의 동작 여부를 시연한다. 2. 평가위원으로부터 지적받은 내용들을 체크리스트로 만들어 수정 보완 사항을 파악한다.</p>		
13	[13주]	<p><개발된 작품 보완 작업> 졸업작품발표회에서 평가위원들로부터 지적받은 수정사항 및 추가사항들의 개발을 보완한다. 이에 대한 체크리스트를 만들어 보완여부를 확인한다.</p>	<p>1. 평가위원들로부터 지적받거나 부족한 기능들에 대한 체크리스트를 만든 후 만들어진 체크리스트에 따라 보완사항들을 개발한다. 2. 보완된 내용들을 테스트하여 작품 개발을 완료한다.</p>		
14	[14주]	<p><개발된 작품 보완 작업 확인> 지도교수를 통해 졸업작품발표회에서 지적된 사항들의 수정 및 보완이 완료되었는지 확인한다.</p>	<p>1. 작품평가회에서 지적사항 보완 여부를 확인한다. 2. 아직 부족한 부분에 대해서는 재작업이 필요한지 여부를 판단하고 이에 따라 최종 작품 개발을 완료한다</p>		
15	[기말고사 기간 : 12/15(수)~21(화)]	<p><최종 완성된 작품 제출> 졸업 작품 과목을 통해 개발된 모든 내용을 제출하고 개발된 작품을 할시 확인할 수 있도록 서버에 유지관리 작업을 수행한다.</p>	<p>1. 최종적으로 완성된 작품을 디지털 형식(CD, USB 등) 으로 제작하여 제출한다. 2. 개발된 작품을 서버에 설치 완료하여 할시 개발된 작품이 동작할 수 있도록 유지한다. 3. 작품 개발을 위해 작성한 관련 자료(기획서,설계서,발표자료 등)를 정리하여 제출한다.</p>		최종작품 관련 일체 자료

2. 깃과 깃허브

깃허브(GitHub, 원래 이름: Logical Awesome LLC)는 분산 버전 관리 툴인 깃을 사용하는 프로젝트를 지원하는 웹호스팅 서비스이다. 부리 온 레일로 작성되었다.

GitHub는 영리적인 서비스와 오픈소스를 위한 무상 서비스를 모두 제공한다.

2009년의 Git 사용자 조사에 따르면 GitHub는 가장 인기있는 Git 호스팅 사이트이다.

또한 2011년의 조사에서는 가장 인기있는 오픈소스 코드 저장소로 뽑혔다.

깃이 텍스트 명령어 입력 방식인데 반해, 깃허브는 화려한 그래픽 유저 인터페이스(GUI)를 제공한다.

깃허브는 페이스트빈(pastebin)과 유사한 서비스인 Gist와 위키를 각 저장소마다 운영하고 있으며, 깃 저장소를 통해 고칠 수 있다.

2.1 깃

깃(Git /git)은 컴퓨터 파일의 변경사항을 추적하고 여러 명의 사용자들 간에 해당 파일들의 작업을 조율하기 위한 분산 버전 관리 시스템이다.

소프트웨어 개발에서 소스 코드 관리에 주로 사용되지만 어떠한 집합의 파일의 변경사항을 지속적으로 추적하기 위해 사용될 수 있다.

기하학적 불변 이론을 바탕으로 설계됐고, 분산

버전 관리 시스템으로서 빠른 수행 속도에 중점을 두고 있는 것이 특징이며 데이터 무결성, 분산, 비선형 워크플로를 지원한다.

깃은 2005년에 리눅스 커널 개발을 위해 초기 개발에 기여한 다른 커널 개발자들과 함께 2005년에 리눅스 토르발스가 처음 개발한 것이다.

2005년부터 지금까지 주니오 하마노(Junio Hamano)가 소프트웨어의 유지보수를 맡고 있다.

다른 대부분의 분산 버전 관리 시스템처럼, 또 대부분의 클라이언트-서버 시스템과 달리, 모든 노드의 모든 깃 디렉터리는 네트워크 접속이나 중앙 서버와는 독립적으로 동작하는 완전한 이력 및 완전한 버전 추적 기능을 갖춘 성숙한 저장소이다.

2.2 깃허브

깃허브의 프로젝트는 표준 Git 명령줄 인터페이스를 사용하여 접근하고 조작할 수 있으며 모든 표준 Git 명령어들이 이 인터페이스와 함께 동작한다.

깃허브는 또한 등록 사용자와 비등록 사용자 모두에게 사이트의 공개 저장소를 찾아볼 수 있는 권한을 제공한다.

여러 데스크톱 클라이언트와 Git 플러그인 또한 깃허브, 그리고 플랫폼을 연동하는 다른 서드파티에 의해 개발되고 있다.

사이트는 피드, 팔로어, 위키(골룸이라는 이름의 위키 소프트웨어 사용), 소셜 네트워크 그래프 등

소셜 네트워크와 같은 기능을 제공하여 개발자들이 어떻게 자신의 저장소의 버전(포크)을 작업하는지, 어느 포크(및 해당 포크의 브랜치)가 최신인지를 보여준다.

사용자는 사이트에 내용을 기여하기 위해 계정을 만들어야 하지만 공개 저장소는 누구든지 탐색 및 다운로드가 가능하다. 등록된 사용자 계정이 있으면 사용자는 토론, 저장소 관리, 다른 저장소로의 기여 제출, 코드의 변경 사항 검토를 할 수 있다.

깃허브를 구동하는 소프트웨어는 루비 온 레일즈와 얼랭을 사용하여 깃허브 개발자들인 완스트래스(Wanstrath), 하이엣(Hyett), 프레스턴-워너(Preston-Werner)에 의해 개발되었다.

2.3 깃허브의 범위

깃허브는 대부분 코드를 위해 사용된다.

소스 코드뿐만 아니라 깃허브는 다음의 포맷과 기능을 지원한다

- 다양한 마크다운식의 파일 포맷의 자동으로 렌더링되는 README 파일을 포함한 문서화
- 레이블, 마일스톤, 할당받느이(assignee), 검색 엔진을 갖춘 이슈 추적(기능 요청 포함)
- 위키
- 코드 검토 및 댓글을 지원하는 풀 리퀘스트(Pull request)
- 커밋 역사

- 그래프: 펄스(pulse), 기여자, 커밋, 코드 주기, 편지 카드, 네트워크 멤버
- 연동 디렉터리(Integrations Directory)
- diff 통합 및 분리
- 이메일 알림
- 다른 사용자를 구독하는 옵션 (다른 사용자의 언급 알림).
- 에모지
- 깃허브 페이지: 작은 웹사이트들은 깃허브의 공개 저장소의 호스팅을 받을 수 있다. URL 포맷은 <https://username.github.io>.
- 파일 내의 중첩 작업 목록
- 지형공간 분석 데이터 시각화
- 새로운 통합 STL 파일 뷰어를 사용하여 미리 보기가 가능한 3D 렌더 파일 ("3D 캔버스" 상의 파일 표시). 뷰어는 WebGL, Three.js의 지원을 받는다.
- 포토샵의 네이티브 PSD 포맷은 미리 볼 수 있으며 동일 파일의 다른 버전과 비교가 가능하다.
- PDF 문서 뷰어
- 각기 다른 패키지에서 공통 취약점 및 노출로 알려진 보안 경보

3. 자기주도 학습내용

다른 앱으로 간단한 데이터를 보내는 Android Sharesheet 와 Android 인텐트 리졸버가 있습니다.

데이터를 보낼 때 인텐트를 만들고 작업을 Intent.ACTION_SEND로 설정합니다. Android Sharesheet를 표시하려면 Intent.createChooser()를 호출하여 Intent 객체에 전달해야 합니다. 그러면 Android Sharesheet를 항상 표시하는 인텐트의 버전을 반환합니다.

우선 제일 간단한 텍스트를 보내는 예제입니다.

```
Intent sendIntent = new Intent();
sendIntent.setAction(Intent.ACTION_SEND);
sendIntent.putExtra(Intent.EXTRA_TEXT, "This is my text
to send.");
sendIntent.setType("text/plain");
```

```
Intent shareIntent = Intent.createChooser(sendIntent,
null);
startActivity(shareIntent);
```

ACTION_SEND 작업을 사용하여 바이너리 데이터를 공유할 수 있습니다. MIME 유형을 설정하고 extra(EXTRA_STREAM)의 데이터에 URI를 배치할 수 있습니다. 일반적으로 이미지를 공유하는 데 사용되지만, 다음 예시와 같이 바이너리 콘텐츠 유형을 공유하는 데 사용될 수도 있습니다.

```
Intent shareIntent = new Intent();
shareIntent.setAction(Intent.ACTION_SEND);
shareIntent.putExtra(Intent.EXTRA_STREAM,
    uriToImage);
shareIntent.setType("image/jpeg");
startActivity(Intent.createChooser(shareIntent,
    getResources().getText(R.string.send_to)));
```

보내는 데이터와 관련하여 구체적인 MIME 유형을 보내야 합니다. 예를 들어 일반 텍스트를 공유할 때는 text/plain을 사용해야 합니다. 다음 예시는 Android에서 간단한 데이터를 보낼 때의 몇 가지 MIME 유형입니다.

수신자가 text/*에 등록해야 하는 text/plain, text/rtf, text/html, text/json 이 있고, 수신자가 image/*에 등록해야 하는 image/jpg, image/png, image/gif 등이 있습니다.

여러 콘텐츠를 공유하려면 콘텐츠를 가리키는 URI 목록과 함께

ACTION_SEND_MULTIPLE 작업을 사용할 수 있습니다. MIME 유형은 공유하는 콘텐츠의 혼합에 따라 달라지는데, 예를 들어 세 개의 JPEG 이미지를 공유한다면 유형은 "image/jpg"입니다. 이미지 유형이 혼합된 경우 모든 유형의 이미지를 처리하는 활동과 일치하는 "image/*"여야 합니다. 혼합된 유형을 공유할 수 있지만 수신자가 전송될 내용을 명확히 알 수 없어서 자주 사용되지 않습니다. 여러 유형을 보내야 한다면 "*/*"를 사용해야 합니다. 예시는 다음과 같습니다.

```
ArrayList<Uri> imageUris = new ArrayList<Uri>();  
imageUris.add(imageUri1); // Add your image URIs here  
imageUris.add(imageUri2);
```

```
Intent shareIntent = new Intent();  
shareIntent.setAction(Intent.ACTION_SEND_MULTIPLE);  
shareIntent.putParcelableArrayListExtra(Intent.EXTRA_STREAM,  
imageUris);
```

```
shareIntent.setType("image/*");  
startActivity(Intent.createChooser(shareIntent, "Share images  
to.."));
```

Android 인텐트 리졸버는 데이터를 다른 앱에 전송할 때 가장 적절하게 사용할 수 있습니다. Android 인텐트 리졸버를 사용하려면 Android Sharesheet를 호출할 때와 같이 인텐트를 만들고 extra를 추가하면 됩니다.

ACTION_SEND 및 MIME 유형과 일치하는 필터를 사용하는 여러 애플리케이션이 설치되어 있다면 시스템에서 사용자가 공유할

타겟을 선택할 수 있는 인텐트 리졸버라는 대화상자를 표시합니다.
하나의 애플리케이션이 일치하면 이 애플리케이션이 실행됩니다.

다음은 Android 인텐트 리졸버를 사용하여 텍스트를 보내는
방법의 예입니다.

```
Intent sendIntent = new Intent();  
sendIntent.setAction(Intent.ACTION_SEND);  
sendIntent.putExtra(Intent.EXTRA_TEXT, "This is my text to  
send.");  
sendIntent.setType("text/plain");  
startActivity(sendIntent);
```

출처

안드로이드 디벨로퍼스

<https://developer.android.com/?hl=ko>