JAVA 기반 프레임워크 응용 SW 개발자 양성 과정 1차 프로젝트

Project [DICERS]

**작성자**

김근호

**관련 능력 단위**

요구사항 분석

화면 설계

애플리케이션 설계

**작업계획서(WBS)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **구분 코드** | **작업명** | **작업자** | **선행작업** | **공통 모듈** | **시작일시** | **완료일시** | **완성도** |
| Main | DICERS\_main 화면, 기능 구축 | 김근호 | - | pygame | 20.08.18 | 20.08.28 | 100 |
| Rank\_1 | DICERS\_rank 화면 구축 | 김근호 | Main | pygame | 20.08.19 | 20.08.26 | 100 |
| Game\_1 | DICERS\_game 게임 함수 구축 | 김근호 | Main | pygame | 20.08.20 | 20.08.28 | 100 |
| Game\_2 | DICERS\_game 기록 저장 함수 구축 | 김근호 | Game\_1 | PyQt5 | 20.08.21 | 20.08.25 | 100 |
| Rank\_c | DICERS\_rankcal 기능 구축 | 김근호 | Game\_2 | operator | 20.08.22 | 20.08.24 | 100 |
| Game\_3 | DICERS\_game 화면 구축 | 김근호 | Game\_1  Game\_2 | Pygame | 20.08.23 | 20.08.27 | 100 |
| Rule | DICERS\_rule 화면, 기능 구축 | 김근호 | Main | pygame | 20.08.28 | 20.08.28 | 100 |

**목차**

1. 프로젝트 정보
2. 요구사항 정의서
3. 업무분담
4. 설계사양서
5. 개발 및 서비스 환경
6. 어플리케이션
7. 보완사항
8. **프로젝트 정보**

프로젝트명: DICERS

팀: 김근호(개인)

개발기간: 2020.08.18 – 2020.08.28

* **개발 목적**
* 누구나 가볍게 즐길 수 있는 베팅 게임
* 파이썬 GUI 패키지(pygame, PyQt5)를 이용한 윈도우 어플리케이션 개발
* **개발 과정**

|  |  |  |
| --- | --- | --- |
| **단계** | **과정명** | **산출물(완료보고서)** |
| 1 | 요구분석 | 요구사항 정의서 |
| 2 | 설계 | 설계사양서  개발 서비스 환경  어플리케이션 |
| 3 | 계획 | 작업계획서(WBS)  업무분담 |
| 4 | 중요/공통 모듈 개발 |  |
| 5-1 | 우선순위 개발 및 테스트 |  |
| 6 | 배포 | 보완사항 |
| 5-2 | 낮은 순위 개발 및 테스트 |  |

1. **요구사항 정의서**

* 현행분석
* 요구사항 정의서
* 요구사항 상세
* **현행분석**

학원 동기생들끼리 간단하게 즐길 수 있는 로컬 게임 필요

* **요구사항 정의서**

|  |  |  |
| --- | --- | --- |
| **구분코드** | **요구사항** | **사용자** |
| 1 | 무작위 주사위 눈을 이용해야 한다. |  |
| 2 | 보유 포인트와 베팅 포인트가 실시간으로 반영되어야 한다. | 포인트 입력 |
| 3 | 랭킹 정보를 파일의 형태로 저장하여 언제든 불러올 수 있어야 한다. |  |
| 4 | 랭킹의 정보에 중복된 이름이 있을 경우 가장 높은 값으로 바뀌어야 한다. |  |
| 5 | 랭킹 화면은 실시간 순위를 반영해야 한다. |  |
| 6 | 프로그램은 윈도우 10에서 작동되어야 한다. |  |
| 7 | 모든 화면은 GUI 환경으로 제공되어야 한다. |  |

* **요구사항 상세**

|  |  |  |
| --- | --- | --- |
| **구분코드** | **요구사항** | **상세** |
| 1 | 초기 랭킹 값이 있어야 한다. | long, clever, rich 랭킹 텍스트에 기본값 지정 |
| 2 | 무작위 주사위 눈을 이용해야 한다 | random 내장함수 이용 |
| 3 | 랭킹 정보를 파일의 형태로 저장한다. | long, clever, rich 랭킹 텍스트 파일에 결과값을 저장하는 함수 지정 |
| 4 | 랭킹 정보를 실시간 정렬하고 최신화해야 한다. 랭킹 정보에 중복된 이름의 경우도 계산해야 한다. | long, clever, rich 텍스트 파일에 저장된 결과값을 순차적으로 재정렬하고 필요없는 데이터를 삭제하는 rankcal 함수 지정 |
| 5 | 모든 프로그램의 과정은 시각화되어야 한다. | Pygame, PyQt5 이용하여 전체 시각화한다. |

* **사용환경**

Windows 10 32bit 이상

Python 3.5이상

1. **설계사양서**

* 프로세스
* 테이블(데이터)구조
* 네이밍 규칙
* 패키지 구조
* ERD
* **프로세스**

**개체, 시계이(가) 표시된 사진

자동 생성된 설명**

**- 테이블(데이터)구조**

데이터베이스를 사용하지 않고 텍스트 파일을 사용함

|  |  |
| --- | --- |
| **입력** | **출력** |
| 1위 플레이어 이름, 점수, 라운드, 2위 플레이어 이름, 점수, 라운드, 3위 플레이어 이름, 점수, 라운드 | GUI 상으로 출력 |

* **네이밍 규칙**

|  |  |  |
| --- | --- | --- |
| **공통사항** | 가급적 풀네임 사용 |  |
| **디렉토리** | 공통 모듈은 같은 위치에 있다. | DICERS\_main  DICERS\_game  DICERS\_rank  DICERS\_rankcal  DICERS\_rule |
| **Html/txt** | 소문자로 사용한다. | long.txt  clever.txt  rich.txt |

* **디렉토리 구조**

|  |  |  |  |
| --- | --- | --- | --- |
| **구분** | **폴더명** | **파일명** | **설명** |
| 게임 본체 | DICERS | DICERS\_main.py | 첫 화면 GUI |
|  |  | DICERS\_game.py | 게임 화면 GUI, 게임 메커니즘, 랭킹 기록 함수 |
|  |  | DICERS\_rank.py | 랭킹 화면 GUI |
|  |  | DICERS\_rankcal.py | 랭킹 텍스트 파일 정리 모듈 |
|  |  | DICERS\_rule.py | 도움말 GUI |
| 화면 데이터 | Material | NEXONFootballGothic.ttf | 게임 화면 구성 폰트 |
|  |  | rule1.png | 도움말 1페이지 |
|  |  | rule2.png | 도움말 2페이지 |
|  |  | rule3.png | 도움말 3페이지 |
|  |  | rule4.png | 도움말 4페이지 |
| 랭킹 데이터 | ranks | clever.txt | clever 랭킹 데이터 |
|  |  | long.txt | long 랭킹 데이터 |
|  |  | rich.txt | rich 랭킹 데이터 |

* **패키지 구조**

|  |  |  |
| --- | --- | --- |
| **패키지명** | **클래스명** | **설명** |
| pygame | display.set\_mode() | GUI surface 객체 설정 |
|  | display.update() | surface 변경사항 갱신 |
|  | draw.rect() | surface 위에 사각형을 그림 |
|  | surface.fill() | surface 객체에 색 표시 |
|  | surface.blit() | surface 객체에 객체 표시 |
|  | font.Font() | 텍스트 객체 설정 |
|  | image.load() | 이미지 파일 객체 설정 |
|  | object.render() | 텍스트 객체 구현 |
|  | object.get\_rect() | 객체의 화면 좌표 설정 |
|  | object.collidepoint() | 해당 객체 좌표의 충돌여부 확인 |
|  | event.get() | pygame 내에서 event라고 정의된 행동들 수집 |
|  | event.type | event의 type을 받는 변수 |
|  | event.pos | event의 위치를 받는 변수 |
|  | MOUSEMOTION | surface 내에서 마우스 좌표 event |
|  | MOUSEBUTTONDOWN | 마우스 클릭 event |
|  | time.delay() | 시간 지연 설정 |
| random | randint() | 정수 난수 추출 |
| operator | itemgetter() | 랭킹 딕셔너리 정렬 |
| PyQt5.QtWidgets | QInputDialog | 플레이어 이름을 받는 Dialog 호출 |
|  | QWidget | Qt5 창 호출 |
|  | QApplication | 상호작용 |

* **ERD**

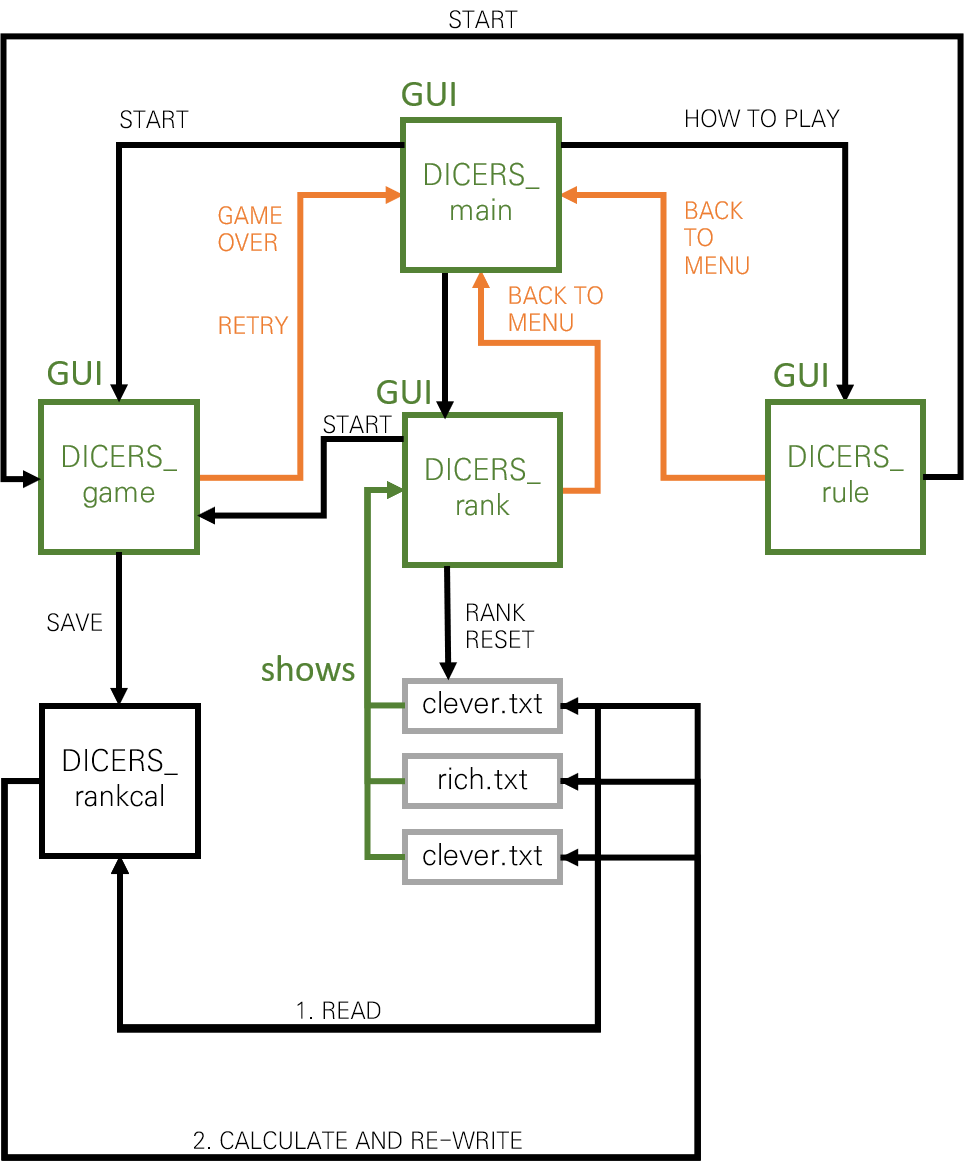
해당사항 없음

1. **개발 및 서비스 환경**

|  |  |  |  |
| --- | --- | --- | --- |
| **구분1** | **구분2** | **개발 환경** | **서비스 환경** |
| SW | OS | Windows 10 64bit | Windows 10 32bit 이상 |
|  | 개발 언어 | python 3.8 | python 3.5 이상 |
|  | 프레임워크 라이브러리 | pygame  (python 3.5 이상) |  |
|  |  | PyQt5  (python 3.5 이상) |  |

1. **어플리케이션**

* 프로시져
* 서버 및 기타 설정
* **프로시저**

****

**스크린샷이(가) 표시된 사진

자동 생성된 설명(메인 화면 GUI)**

**스크린샷이(가) 표시된 사진

자동 생성된 설명(게임 화면 GUI)**

**스크린샷이(가) 표시된 사진

자동 생성된 설명(랭킹 화면 GUI [초기값])**

**스크린샷이(가) 표시된 사진

자동 생성된 설명(게임 설명 GUI)**

* **서버 및 기타 설정**

윈도우 어플리케이션 프로젝트로 해당사항 없음.

1. **보완사항**

|  |  |  |
| --- | --- | --- |
| **이름** | **보완사항** | **보완내용** |
| 김근호 |  |  |
|  |  |  |
|  |  |  |