**mRla모바일 어플리케이션 설명서**

|  |  |  |  |
| --- | --- | --- | --- |
| **모바일 어플리케이션 이름**  **(1점)** | 구구단을 외자! | | |
| **모바일 어플리케이션 설명**  **(1점)** | **기능** | 누가 구구단 계산을 더 잘하는지 겨룰 수 있는 게임이다. 구구단 문제가 주어지면, 최대한 빨리 답을 하는 것이 중요하다. 구구단 10문제가 나오며, 시간 초를 재서 가장 빠르게 맞춘 사람이 똑똑한 사람으로 인정받을 수 있다. 잘못된 답을 클릭하면 패널티로 1초가 부과됩니다. ( 비교적 어려운 5-9단 문제만 나오도록 했습니다. ) | |
| **모바일 어플리케이션**  **스크린샷**  **(3점)** |  | | C:\Users\derba\AppData\Local\Microsoft\Windows\INetCache\Content.Word\image (2).png |
| 구구단을 외자! 게임의 시작 화면이다.  START 버튼을 누르면 게임을 시작한다. | | 10문제를 푸는 데 걸린 시간을 표시하는 결과 화면. 버튼을 누르면 이동한다. |
| C:\Users\derba\AppData\Local\Microsoft\Windows\INetCache\Content.Word\image (1).png | | |
| 구구단 문제를 풀 수 있는 화면이다. 질문에 맞는 답을 누르면 다음 문제로 넘어가며, 틀린 답을 누르면 1초씩 패널티가 추가된다. | | |
| **구현 기술**  **(조건에 맞춰 서술)**  **(2점)** | 1. 새로운 클래스 만들어 보기  - MainActivity.java에서 구구단 문제와 보기를 자동으로 생성하는 stage 클래스를 만들어 보았습니다.  2. View 사용 및 속성 변경  - TextView, Button을 사용하였으며, TextView와 Button의 text값을 MainActivity.java의 stage 클래스 등에서 변경하였습니다.  3. 이벤트 구현  - 각 버튼에 onClick 함수를 연결했습니다.  4. Activity 만들기  - activity\_main.xml / activity\_start.xml / activity\_end.xml 로 총 3개의 액티비티로 구성되어 있습니다.  5. Thread나 Service 구현  - thread를 사용하여, 구구단 10문제를 푸는 데 걸린 시간을 측정했습니다. | | |