|  |  |  |  |
| --- | --- | --- | --- |
| **제출일** | 2012-07-15 | **프로젝트명** | SailingText |
| **참여자** | 김이석, 김진협, 지상현 | | |

**프로젝트 수행계획서**

**-안드로이드 플랫폼 기반의 G센서를 이용한 사용자 참여 게임 개발-**

**차 례**

1. 프로젝트 제목 --------------------------------------------------------2
2. 프로젝트의 동기 및 목적 -------------------------------------------2
3. 프로젝트의 내용 구성 -----------------------------------------------2
4. 프로젝트에 필요한 기술 및 이론 ----------------------------------3
5. 프로젝트 수행 방법 --------------------------------------------------3
6. 업무분담 및 구체적 역할 --------------------------------------------
7. 프로젝트 일정 계획 --------------------------------------------------
8. 예상되는 장애요인 및 해결방안 (risk) -----------------------------
9. 기대효과 및 활용방안 ------------------------------------------------

**1. 프로젝트명**

**Sailing Text**

**2. 프로젝트 동기 및 목적**

**2.1. 프로젝트 동기**

* 한이음 멘토링 프로젝트와 학과 내 프로젝트 수업을 통하여 안드로이드 프로그래밍 기술을 습득할 수 있도록 한다.
* 프로그래밍 기술을 습득함에 있어 단순히 개인적 기술이 아닌 참여자(팀원) 간 협력함으로써, 실제 업무에서 단체 개발에 필요한 기술을 습득할 수 있도록 한다.
* 단순히 쉽고 간단한 것이 아닌, 이번 기회를 통해 보다 많은 기술을 학습 하는 것을 목표로 한다.

**2.2. 프로젝트 목적**

* 안드로이드 어플리케이션 개발 프로젝트를 진행함으로써 참여자들이 안드로이드 플랫폼의 기술을 접하고 응용 할 수 있도록 한다. 또한 보다 많은 기술을 접할 수 있고 만들고자 하는 의욕이 높은 게임 어플리케이션을 개발하도록 한다.
* 1to50, Angry Birds같은 단순하면서도 반복적으로 할 수 있는 게임을 개발한다.
* Handset의 방향센서를 이용하여 참여자가 좀더 능동적으로 게임에 몰입할 수 있도록 한다.
* 게임을 보다 현실성 있도록 물리 엔진을 만든다.
* 단체 개발에 있어 소스 코드 제어와 프로그램 버전 관리를 위해 Git를 이용하며, 이를 통해 프로젝트의 개발 투명성을 유지하도록 한다.

**3. 프로젝트 내용 구성**

**3.1. 구성**

* 처음에 로고가 뜨는 Splash 화면이 지나가고 메인 메뉴가 나타난다.
* 메뉴는 ‘Start Game’(게임 실행 메뉴), ‘Highscores’(기록확인메뉴), ‘Options’(설정 메뉴)로 나뉜다.
* ‘Start Game’을 누르면 게임을 할 스테이지 목록이 뜨고, 선택하면 게임이 시작된다.
* ‘Highscores’를 누르면 게임을 했던 기록과 점수가 뜨는 화면으로 이동한다.
* ‘Options’를 누르면 게임의 옵션 (배경음 On/Off 여부 등)을 조절할 수 있는 화면으로 이동한다.

**3.2. 기능**

* 방향 센서를 이용하여 Handset의 방향을 알아내 아래 방향을 찾아, 공이 아래쪽으로 이동하도록 한다.
* 아래로 떨어지는 공을 골에 도달시키는 것을 목표로 한다.
* 공을 움직이는 동안 다양한 장애물이 사용자의 이동을 방해하여 게임의 재미를 높인다.
* 스테이지 마다 장애물의 배치 등에 따라 난이도가 달라진다.
* 사용자가 게임을 한 기록은 DB에 기록되며, 메인 메뉴에서 Highscores 메뉴를 통해 확인 가능하다.

**3.3. 환경**

* + **JDK 6.0**
  + **Android 2.3 Gingerbread (Target)**
  + **IDE : Eclipse Indigo**
  + **Test Device : Galaxy S2, ATRIX, Optimus 3D 및 AVD**

**4. 프로젝트 수행에 필요한 기술 및 이론**

**4.1. 물리 엔진 구현 사항**

**4.**1.1. 운동 및 충돌의 물리 처리

이 게임에서는 현실과 같은 물체의 운동을 표현하기 위해 동역학의 이론을

적용한다. 그러나 처리를 단순하게 하도록 다음과 같은 가정에 기반한다.

* 스테이지 안에 존재하는 모든 물체는 강체(rigid body)입니다.
* 움직이는 물체는 평행 이동만 고려합니다. 즉, 토크(torque)를 고려하지 않습니다.
* 충돌 처리에 관여하는 물체들은 모두 균일한 이상적인 상태라고 가정하며, 질점(particle) 취급합니다.
* 충돌 처리에 가담하는 물체는 공(ball)과 장애물(obstacle)뿐입니다.
* 움직이는 물체는 공뿐이고, 장애물은 고정된 물체(fixture)입니다.
* 물체의 모양(shape)은 정해져 있습니다. 공은 항상 원(circle)이고, 장애물은 항상 직사각형(rectangle)입니다.

**4.1.2. 운동의 표현 방법**

* 스테이지 안에서 움직이는 의미 있는 물체는 오로지 공(ball)이며, 단순히 크기를 질점(particle) 취급하여 움직임을 표현한다.
* 게임 안에서 단위 시간(1/30초로 설정)에 맞추어 프레임을 처리하기 때문에 Handset의 처리 속도가 달라도 똑같이 처리할 수 있다.
* 공에는 중력과 항력이 작용하며, 중력 가속도의 방향은 센서로 구하고 항력은 속도에 비례하는 근사식으로 구한다.

**4.1.3. 충돌 처리**

공이 장애물에 부딪혔을 때 뚫고 지나가지 못하도록 하며 튕겨 나오도록 처리한다. 그러나 프레임이 이산적으로 처리되므로, 공이 장애물을 뚫고 들어가는 상황을 고려해야 한다. 따라서 충돌 시점에서 관통된 변위를 구해 올바른 방향으로 튕겨나간 것으로 교정해야한다.

요약해서, 현재 위치와 새 위치를 잇는 이동 경로와 겹치는 선분(변)이 있다면 충돌된 것이며 뚫고 들어간 부분만큼 위치와 속도를 교정한다.

**4.1.4. 게임 내 물체 관리**

충돌 처리를 할 때 충돌 검사를 할 장애물만 빠르게 골라낼 수 있도록 공간을 Quad Tree로 분할하여 관리한다.

Space partitioning by Quad Tree

4.2. **HTML5 - Canvas**

게임 내 Map 제작에 사용되는 맵 에디터는 실행환경에 제약 없이 빠르고 간편한 사용을 위해 HTML5 기반의 Canvas 기술을 사용하며, 이에 따른 기술을 알고 있어야 한다.

**5. 프로젝트 수행 방법**

* 매주 3회 이상 오프라인 모임을 갖고, 작업 진행 확인 및 방향 설정을 하도록 한다.
* 해결해야 할 과업과 우선순위를 정한 뒤, 우선순위에 따라 적절한 인원에게 과업을 할당한다.
* 참여자 간 작업 내용 공유는 git를 이용하고 저장소는 github 를 이용한다.  
  (저장소: https://github.com/ssidang/dimension-drive-downside.git)

**6. 업무분담 및 구체적 역할**

1. **김이석**

* Project Manager
* Game UI 를 비롯한 Graphic 효과 Design
* Level Design 및 게임 진행 설계

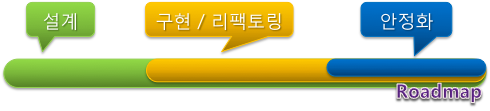
1. **김진협**

* Database 작업(SQLite)
* Activity Organization 및 Source Merge작업
* Sound 작업 및 맵에디터 작업

1. **지상현**

* World Management
* 충돌 효과 구현 작업
* 게임 제어 작업

**7. 프로젝트 일정 계획**



|  |  |  |
| --- | --- | --- |
| 일정 | **작업계획** | **참여자 별 역할** |
| 06.25 ~ 06.28 | **프로젝트 설계** | **김이석**: 프로젝트 설계 |
| **김진협**: DB및 기타 작업  설계 |
| **지상현**: 충돌효과 계획 |
| 06.29 ~ 07.08 | **프로젝트 구현** | **김이석**: Graphic 작업 |
| **김진협**: DB구축 및 |
| **지상현**: 물리 엔진 구현 |
| 07.09 ~ 07,20 | **발견된 수정사항 및 오류사항에 대한 작업 및 안정화 작업** | **김이석**: 수정 및 오류사항  파악 |
| **김진협**: 소스 통합 작업 |
| **지상현**: 물리엔진 최적화  및 안정화 |

**8. 예상되는 장애요인 및 해결 방안 (Risk)**

1. **충돌처리의 어려움**

* 물리엔진의 활용 및 이를 이용하기 위한 수식 처리를 통한 미세한 물리 작용 처리가 관건으로 판단된다.
* 이를 해결 하기 위해 프로젝트 막바지까지 지속적인 테스트와 수식 수정을 계획하고 있다.

1. **참여자간 작업한 소스의 통합 작업**

* 참여자간 분담하여 작업한 소스는 참여자의 개인적 특성이 반영되므로 이를 통합하는 것이 쉽지 않을 것으로 예상된다.
* 가급적 참여자들에게 소스의 캡슐화를 요구하여 통합 내부 알고리즘에 대한 고민 없이 직접적인 클래스 선언 및 메서드 호출의 배치 만으로 작업이 될 수 있도록 설계를 요청한다.
* 프로젝트 마감 3~4일 전에는 팀원이 모두 모여 소스 코드 통합 시 생기는 오류에 즉각적으로 대처 할 수 있도록 한다.

1. **열악한 테스트 환경**

* 테스트는 IDE에 포함되어 있는 AVD 와 참여자가 가지고 있는 Device (Galaxy S2, Atrix, Optimus 3D)로 테스트 환경이 제약된다.
* 가급적 프로젝트 마감 전 더 많은 테스트 장비들을 획득하여 테스트 할 수 있도록 한다.
* 프로젝트 마감 전 하지 못한 테스트 환경은, 마켓 등록 후 사용자들의 피드백을 통하여 에러의 습득 및 즉각적인 수정을 할 수 있도록 한다.

**9. 기대효과 및 활용방안**

* 참여자들이 프로젝트 작업을 통하여 안드로이드 프로그래밍을 이론이 아닌 실제 응용이 가능하게끔 익힌다. 이를 바탕으로 추가적인 기술 습득 및 향후 졸업 후 실무에서 활용하도록 한다.
* 단체 제작 프로젝트를 통하여 개인이 아닌 다수가 참여하는 프로젝트에서의 작업 방법을 익힌다. 세부적으로 소프트웨어 공학 및 프로젝트 관리 과목 등에서 배우는 설계적 이론을 습득 하고, 실무에서 활용 할 수 있도록 한다.
* 실제로 구글 마켓에 등록하여 성취감을 느낄 수 있도록 하며, 추후 마켓에 등록된 어플리케이션의 현황을 파악하고 이를 위한 마켓 분석을 통하여 실제 시장에서의 어플리케이션 동향을 파악하고, 제작으로 끝나는 것이 아닌 유지보수에 대한 개념도 확립 할 수 있도록 한다.