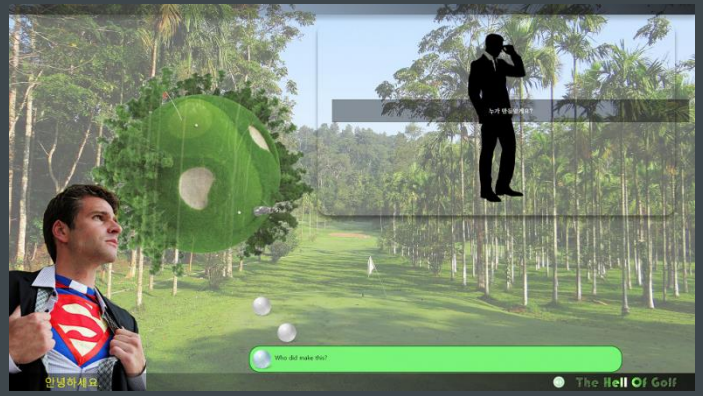
MINIGOLF



개요

웹 기반의 골프 게임

HTML5 표준과 CSS3 및 canvas를 적극 활용한 경쟁형 게임. javascript 버전의 물리엔진과 애니메이션 기술 라이브러리를 활용, 플레이어간의 랭킹을 비교하며 경쟁하는 2D 물리 골프 게임.

특징

- 프로젝트 과제 제출
- 2d 물리 엔진을 적용한 골프 게임 제공
- 사고력 향상과 경쟁심 유도

Products

MiniGolf

Owlet

Ichatt

Noriter

BetterBrain

개발환경

형상관리	SVN
개발 언어	Javascript, HTML
개발 도구	Eclipse, Aptana, notepad
Database	Mssql

Products

MiniGolf

Owlet

Ichatt

Noriter

BetterBrain

역할 및 이용기술

총 참여인원 3명
본인은 클라이언트 전체 개발
서버 통신, 엔진 적용, 이미지 렌더링, 사용자 이벤트 처리, 사운드 재상 등 작업

Preload.js, box2d.js 및 canvas API 이용.
SVN으로 소스 관리. 서버는 MS의 Asure를 이용.

기술상세

Preload.js
- 게임이다 보니 자원(resource)의 용량이 상당하여 사용자의 부담을 덜기 위해 사용

box2d.js
- 여러 물리 엔진 중 가장 배우기 쉽고 간단히 사용가능하여 선택

canvas API
- 학습목표라 canvas API를 거의 그대로 이용해야 하여 사용. 일부 공통 처리는 render 라이브러리로 canvas 사용용이성을 매우 높여주는 easeljs 선택하여 사용.

SVN
- 소스 관리를 위해 당시 NAVER에서 제공하던 SVN 을 이용

MS의 Asure
- Node Server 개발 서버 배포

Products

MiniGolf

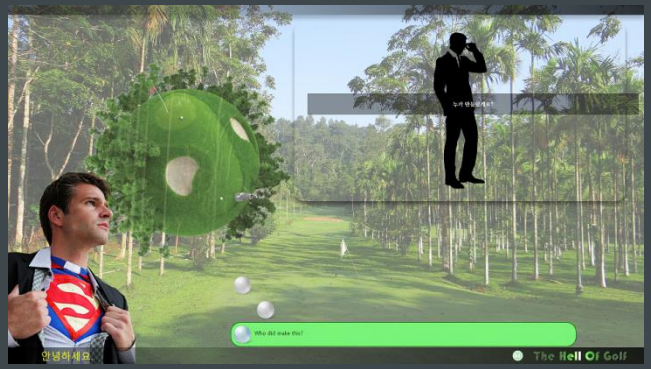
Owlet

Ichatt

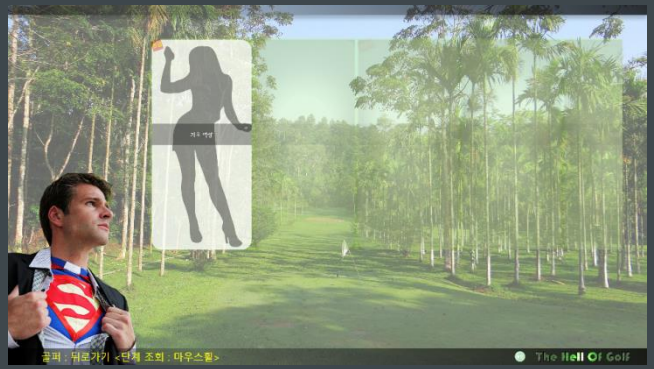
Noriter

BetterBrain

스크린샷



주 화면



단계 선택 화면

플레이 장면 1

플레이 장면 2

Products

MiniGolf

Owlet

Ichatt

Noriter

BetterBrain

Products

MiniGolf

Owlet

Ichatt

Noriter

BetterBrain

고찰

물리 엔진을 적용하는데 Physics.js, p2.js, verlet.js, Matter.js 등을 살펴보는데 의외로 당시 코드의 관습에 대한 이해가 부족하여 적용에 매번 실패함. 일정을 맞출 수 있을까 걱정했으나 Box2d의 js 포팅 버전을 찾게 되어 가까스로 해결.

Canvas가 아닌 영역은 모두 DOM 구조를 이용하여 애니메이션 렌더링을 해야 했는데 초기 Javascript로 조절하다가 CSS3을 3일 빠르게 훑고 js로 처리하는 부분을 css로 모두 교체.

UX에 대해 이해폭이 상승했고 여러 라이브러리들을 살펴보고 뜯어보면서 공통된 엔진들의 관습에 대해 견문을 넓힐 수 있었음.

OOP 방식으로 철저하게 짜려고 했으나 prototype의 동작에 브라우저별 일부 차이가 있어 크로스 브라우징을 위해 일부 상속구조를 지키지 못함.

시간 한계로 원했던 멋진 효과를 모두 넣지 못함.

통신은 모두 비동기로 json 이용.