Designing the Games

305900
2008년 가을학기
9/25/2008
박경신

Game Design

- Why do we play?
  - 설계자의 문제가 아님
- What is the nature of games?
  - 설계자의 문제가 아님
- How is a game formed of parts?
  - 설계자가 고민해야 할 문제임

Game Design is not Software Design

- 소프트웨어 디자인은 게임 디자인을 지원하기 위해 필요함
- 게임은 소프트웨어 보다 훨씬 오래 전부터 존재해 왔음
- 대부분의 전자 게임(electronic game)은 전자게임이 아닌 형태로도 플레이 될 수 있음

What is Game Design?

- 게임의 규칙 (rules)
- 플레어의 역할 (role)
- 플레이어가 겪게 될 도전 (challenges)
- 받게 될 보상(rewards)/처벌(punishments)
- 어떻게 게임이 제어되는지 (interactivity)
- 서사적인 흐름 (narrative flow)
- 결과의 예측 불허함을 향상 시키는 밸런스 (balance)
  - 매번 지거나 언제나 이기게 하는 전략은 피할 것!

What is “Fun” in Games?

- Size of state space in game, learning of patterns (Raph Koster)
- Sensation of flow during play (Salen/Zimmerman & Chris Crawford)
- A vague term to be avoided in critical discussion (Chris Crawford)
- Emotionally interesting choices (David Freeman)

What makes a Game “Fun” or Not?

- 내 게임 설계를 위해서 먼저 현존하는 다른 게임 디자인을 평가해 볼 것
- 무엇이 게임을 재미있게 또는 재미없게 만드는가?
- 매우 매우 많은 게임을 플레이해 볼 것
- 매우 많은 게임을 얘기해 볼 것

Build Fast Playable Prototypes

- 가능한 빠르게 게임 프로토타입 (rapid prototype)을 만들어서 플레이를 해볼 수 있도록 함
- “Find the Fun” (Shigeru Miyamoto)
- 좋은 게임 아이디어는 의외로 매우 형편없는 것에서부터 시작함
- 과정(process)은 지속적인 정제 (refinement) 와 개정 (revision)의 하나임

Take Care of Your Players

- 지속적인 자극을 주자 (continuous stimulus)
- 보상을 자주하라 (reward frequently)
- 도전을 밸런스하라 (balance the challenge)
- 모험 (adventure), 탐색 (exploration) 하게 하라
- 혼동을 줄이자 (minimize confusion)
- 반복을 없애자 (avoid repetition)
- 플레이를 재미있게 배우도록 하게하라 (fun to learn to play)

Lessons Learned from Industry

- (학교에서의) 비디오 게임 소프트웨어 개발은 도구나 방법 측면에서 업계에 뒤지고 있음
- 좋은 설계(good design)는 엄청난 노력을 필요함
- 게임 디자인은 소프트웨어를 개발하지 않고도 연습할 수 있음
  - Pen-pencil prototyping
  - Model mockups
  - Documentation

Miyamoto's Game Design Principles

- Start with a simple concept
- Design around computer limitations
- Minimize player confusion
- Importance of play testing
- Incorporate a smooth learning curve
- Accommodate all skill levels

Gender-Inclusive Game Design

- 소비자의 절반 이상이 여성임
- 그러나 게임은 주로 사춘기 남성에게 초점이 맞춰져 있음
- 하지만 전자게임 외엔 여성이 남성과 거의 동등하게 플레이하고 있음
- 다양성만이 창의적인 면에서 효과적일 수 있음

You can have the best technical programming skills in the world, but if your game design is bad, your game will be bad.
-from Gamedev.net

Basic Elements of a Modern Video Game

1. 개발자 & 제작자 로고 (Logo) 스크린
2. 게임 문맥을 제공하는 오프닝 애니메이션 또는 컷신
3. 게임 구성 스크린
   - 콘트롤 레이아웃 선택, 튜토리얼, 게임 타입 (일인용 또는 멀티유저), 속임수, 엑스트라, 크레딧 (Credits)
4. 게임 레벨 또는 준비 스크린 또는 컷신
   - 게임 캐릭터의 속성을 선택함 – 사람 또는 우주선을 선택하는 등등
5. 게임 레벨 (저장 기능)
6. 레벨에서 컷신의 끝
7. 게임 저장
8. 4로 반복
9. 게임 컷신의 끝
10. 앤딩 크레딧 (Credits)

Designing and Developing the Game

- 게임 디자인과 개발의 기본적인 단계
  1. 게임 아이디어에 대한 브레인스토밍 – 이 게임의 FOCUS가 무엇인지 어떤 Factor가 작용할 지
  2. High-level 디자인 문서화 – 스토리보딩
  3. 기술적 디자인 문서화 – 자료구조
  4. 점진적인 프로토타입 개발
     1. 게임 루프
     2. 원하는 그래픽 효과를 지원하는 아트작업 또는 프로그램 코딩
  5. 게임 플레이 및 테스팅
     1. 정확성을 위해
     2. 재미를 위해
  6. 프로젝트 만기일까지 4와 3단계를 반복함 - 게임 튜닝

Starting a Game Idea

- By Gameplay
  - 예, 이 게임은 DOOM 같은 일인 슈팅 게임…
- By Story
  - RPGs 게임이나 게임 기반의 영화에서 가장 흔히 찾을 수 있음
  - 종종 게임 플레이가 결정된 후에 스토리를 생성함 – 예, Jedi Knight
  - 종종 게임프로그래머가 자신을 소설가로 착각하는데, 이런 실수를 피하도록 해야 함
- By Technology
  - 예, Quake, UnReal 같은 게임엔진은 전략적인 게임보다는 일인칭 또는 삼인칭 액션 또는 슈팅에 적합함
- 게임은 실시간 시뮬레이션이므로 하드웨어의 제안사항, 기간 내에 마칠 수 있는 프로그래머의 개발 능력 등을 감안하여 브레인스토밍을 해야 함
  - 예, 동시에 1000적을 한 화면에 보여줄 수 있는지
  - 예, 컴퓨터 인공지능으로 작동되는 적군을 어떻게 만들어 낼 지

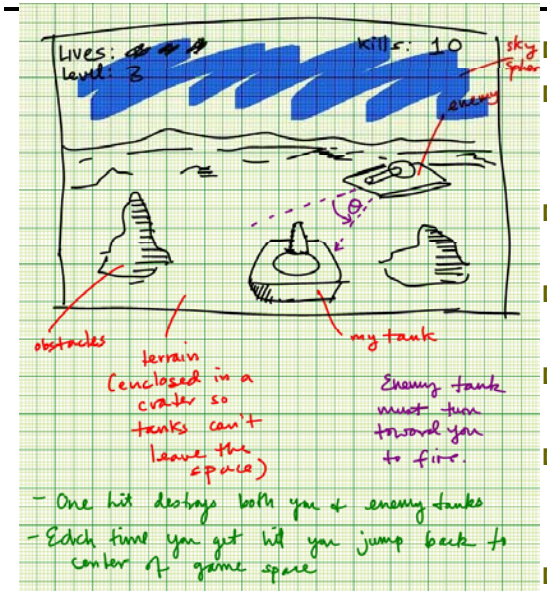
Commonly Encountered Documents

- 게임 컨셉 문서 (Concept / Sales Pitch Document)
  - 게임플레이 설명서, 스크린 mockups, 스토리보드, (캐릭터, 배경) 아트 작업, 팀원 bio, 구현하고자 하는 플랫폼, 시장성, 예산, 개발에 소요되는 예상 기간, 등등.
- 게임 설계 문서 (Design Document)
  - 게임 설계에 필요한 상세서 - 구체적인 스토리 보드, 배경 스토리 등등. 게임 설계 내용은 개발과정 중에 종종 바뀌게 됨.
- 게임 기술적 설계 문서 (Technical Design Document)
  - 자료 구조, 인공지능 알고리즘, 그래픽 알고리즘

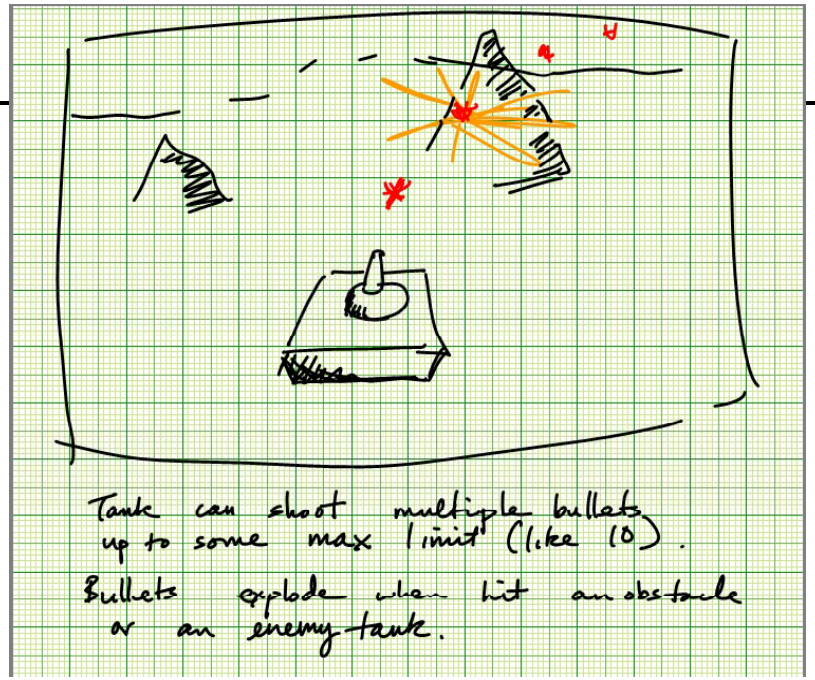
Game Design Document

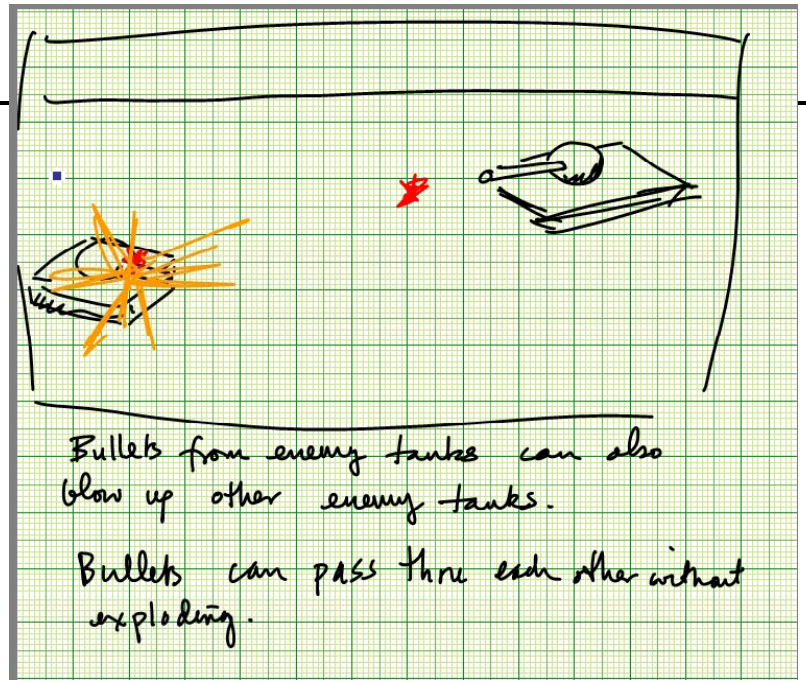
- **Executive summary** - 1 page 전체적인 요약 – 게임의 FOCUS 를 설명.
- **Game mechanics** - 어떻게 사용자가 게임세상과 인터랙션하는지?
- **Artificial Intelligence** - 게임세상이 사용자에 어떻게 반응할 지? 적수가 사용자에게 어떻게 대항할 지?
- **Game Progression** - 게이머가 경험하게 될 이벤트, 게임 레벨, 게임 종료 시 얻는 rewards, 게임 단계별 종료 시 얻는 rewards, 다음 단계와 차별점?
- **Game Elements** - 캐릭터, 아이템, 객체/메카니즘
- **Story** - 스토리가 있다면, 스토리는 게임플레이 요소가 왜/어떻게 영향을 미치는지 설명해줘야 함.
- **Script** - 스토리의 마지막 실행, 컷신 장면.
- **Sound effects** – 스토리에서 필요한 사운드 효과, 배경 음악.
- **Menus and interface design** – 사용자가 어떻게 환경을 제어할 지, game dashboard, 시작메뉴, 스크린 저장/열기, 옵션 또는 속임수.
- **Art Bible** – 객체, 캐릭터, 아이템의 스케치와 mockups.
- **Storyboards** – 게임플레이 설명에 중요함- 예를 들어, 카메라 앵글, 게임 스크린의 배치. 컷신에는 스토리보드가 매우 필요함.

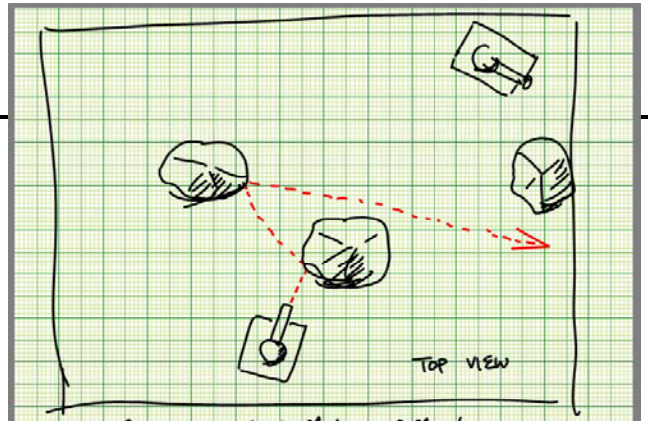
Brainstorm the Gameplay (Using Storyboards)



- 게임플레이 브레인스토밍
- 게임의 오프닝 컷신부터 브레인 스토밍을 시작하지 말 것
- 손으로 직접 그릴 것 – i.e., 컴퓨터로 그리지 말 것.
- 빨리 마칠 것 (시간은 1시간 정도로)
- 자유분방하게 아이디어가 흐르는 데로 맡길 것
- 아이디어가 좋은지 나쁜지 판단하지 말 것 (상호비판 엄금)
- 질보다는 양

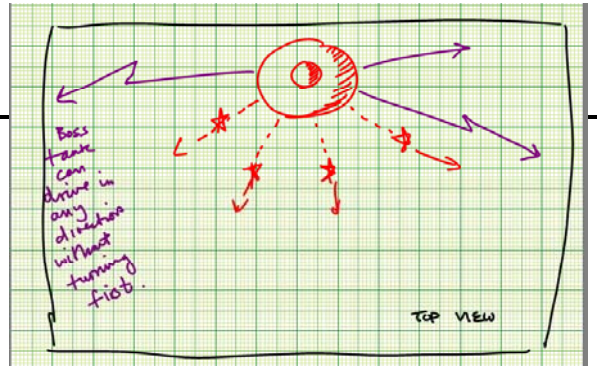






New idea! All bullets will bounce off obstacles for a few times before dissipating.

- Bullets dissipate with distance + disappear eventually
- Bullets that you fire can bounce back + hurt yourself

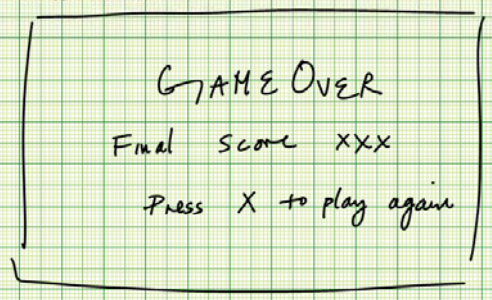


- After 10 enemies destroyed boss tank appears
- When boss explodes it explodes into bullets in all directions. Players must be careful to dodge these
- If after 10 seconds player does not get hit, player advances to next level.
- Each level adds 1 more tank that will attack you simultaneously up to a max of 5 tanks.

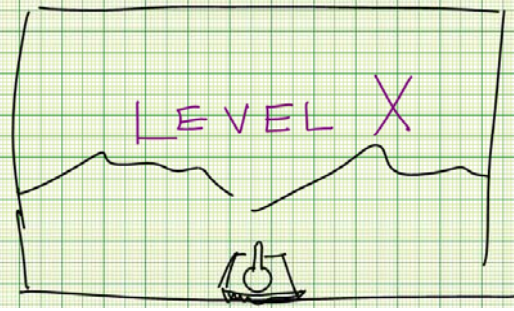
- Each new level will grant you an additional life.
- Beyond level 5 tanks get smarter + faster.
- At each level, destroying each boss causes more bullets to spew out.

Intro screen

Out tro screen



Level advancement



Final Report

- 최종보고서는 게임 디자인 문서의 상세한 내용이 담긴 파일
- 최종보고서에 들어갈 내용:
  - 게임 소개
  - 플레이 방법, 사용자 인터페이스 설명 (다이어그램과 그림을 포함시킬 것)
  - 스케치, 디자인에 사용된 이미지, 게임 개발을 진행하면서 만들어진 다양한 버전, 플레이하는 게임 장면 사진을 포함하여 약 30개 정도의 그림
  - 기술적 문서:
    - 메인루프나 NPCs (Non Player Characters) 등을 포함하는 Finite State Machine (FSM)
    - 자료구조
    - 구현된 그래픽 효과에 대한 설명과 그림
    - 기본적인 인공지능 알고리즘 설명
    - 네트워크 게임경우, 사용된 네트워크 메시지 다이어그램 및 설명
    - 게임의 사운드 효과에 대한 설명과 어떻게 만들어졌는지 – 예를 들어, 여러 개의 사운드를 샘플링해서 새로 만들었다던지
  - 게임의 전체 소스와 실행파일

Project Management

- 공동 프로젝트를 수행하는 팀원으로써 110%로 일하길 기대한다.
- 프로젝트를 하면서 다른 팀원의 성적까지 망치지 말자!
- 우리는 무엇인가 잘 수행해 냈을 때 좋은 기억으로 남으니까, 먼저 본인이 최선을 다하고, 언제나 더 잘할 수 있도록 노력하자.
- 프로젝트 매니저는 막중한 책임을 가진 위치에 있다.
  - 프로젝트 매니저는 밤 늦게까지 깨어 있으면서 어떻게 하면 더 잘할 수 있을까를 고민하고 그리고 만약 안될 경우를 대비해 어떤 백업 계획을 세워야 하나를 생각해야 함
  - 프로젝트 매니저는 계획된 일정 안에 프로젝트를 마칠 수 있도록 팀원들에게 아주 솔직하게 얘기할 수 있어야 함

Project Management

- 토론 시:
  - 모든 사람에게 자신의 의견을 피력할 수 있는 기회를 줄 것
  - 작업에 대한 비평을 할 것이며, 사람에 대한 비평을 하지 말 것
  - 어떻게 무엇인가를 향상시킬지에 대한 건설적, 사료있는 아이디어를 낼 것
  - 비평을 개인적인 감정으로 받지 말 것
  - 우는 소리하지 말고 일을 끝내도록 할 것
- 모두가 다 바쁘다 - 다른 사람의 시간을 존중하도록 하자:
  - 미팅 시간을 정확히 지킬 것
  - 연락도 없이 늦는다는 것은 다른 사람을 불편하게 하는 것이며, 다른 사람의 시간을 존중하지 않는 것
  - 만약 어쩔 수 없이 늦게 된다면, 미팅에 있는 사람에게 미리 연락할 것
  - 미팅 전에 미팅을 위한 논의사항(Agenda)을 미리 준비할 것
  - 미팅이 끝나고 나면 반드시 어떻게 문제를 해결할 것인지, 어떤 새로운 작업이 생겼는지, 또 각자 해야 할 일 등에 대해 미팅노트를 남기고 모든 사람에게 알릴 것

Project Management

- 프로젝트의 일정은 deadline으로부터 거꾸로 내려오면서 각 단계별로 마쳐야 하는 "일정"을 적을 것
- 일정은 계속해서 업데이트할 것
- 주간 미팅에서 끝낸 아이템이 있으면 체크해 놓을 것
- 지속적으로 블로그/웹에 프로젝트 내용을 갱신할 것
- 프로젝트 매니저는 이러한 프로젝트 일정의 갱신을 담당할 것
- 아래의 내용이 담긴 TODO 리스트 테이블을 만들 것:
  - Due Date
  - 주요 목표가 담긴 Milestone
  - Milestone의 진행상태 / 코멘트
  - 각 팀원별로 담당한 작업내용
  - 각 팀원별로 담당한 작업의 진행 상태
  - 등등

Project Management

- 다른 팀원도 알 수 있도록 가족 여행이라던가 종교적 휴일 등과 같은 특별한 일정을 적어둘 것
- 핸드폰번호나 이메일 주소를 서로 교환하여 공유할 것
- 여러 개의 컴퓨터와 미디어에 프로젝트 작업을 모두 백업 할 것 - 이 때 백업은 웹사이트 정보도 포함
- 가능하면, "version control" 시스템을 사용할 것
- 일주일 단위로 자주 게임 부분들을 통합할 것 – 즉, 마지막까지 통합을 미루지 말 것
- 일정하게 "working version"을 잘 둘 것 – 그래야 어떻게 게임개발이 진행되고 있는지를 알 수 있으며 격려를 받을 수 있음
- 친구들에게 개발하고 있는 게임을 플레이해 보도록 자주 테스트할 것 - 그리고 플레이한 후의 반응을 웹사이트 같은 곳에 잘 적어 두고 참고할 것

Reference

- http://www.evl.uic.edu/spiff/class/cs426/Notes/Design2005.ppt
- http://nihlen.us/speak/CNMGDI2007.pdf
- http://www.vancouver.wsu.edu/fac/peabody/game-book/Coverpage.html