Designing the Games

305900
Fall 2009
9/21/2009
Kyoung Shin Park

Game Design

- Why do we play?
  - Not a designer's problem
- What is the nature of games?
  - Not a designer's problem
- How is a game formed of parts?
  - A designer's problem

Game Design is not Software Design

- Software design is done to support a game design
- Games have existed much longer than software
- Most electronic games can be played in a non-electronic form

Games are expression

- Games are expression, not simulation
- Simulation emulates the real world, games exist outside of that context
- Simulation is typically more expensive than expression

What is Game Design?

- Rules of the game
- Role of the game player
- Challenges the player must face
- Rewards/Punishments received
- How the game is controlled (Interactivity)
- Narrative flow
- Balance to improve uncertainty of outcome
  - Avoidance of no-win and always-win strategies

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?

- Evaluate existing game designs
- What makes a game fun? Or not?
- Play lots and lots of games
- Talk about lots of games

Build Fast Playable Prototypes

- Build to play as soon in the design as possible (Rapid prototype)
- "Find the Fun" (Shigeru Miyamoto)
- Even good game ideas start out bad
- Process is one of continual refinement and revision

Take Care of Your Players

- Continuous stimulus
- Reward frequently
- Balance the challenge
- Adventure, Exploration
- Minimize confusion
- Avoid repetition
- Fun to learn to play

Lessons Learned from Industry

- Video game software development lags behind the industry in tools and methodologies
- Good design beats massive effort
- Game design can be practiced without developing software
  - Pen-pencil prototyping
  - Model mockups
  - Documentation

Avoid Programmer Think

- Don't hide the game within the technology (Sheri Graner Ray)
- Programmers tend to excel at sequential reasoning
- Most non-programmers don't enjoy this kind of thinking
- Try to minimize your sequential reasoning challenges

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

- More than half of the consumer world is female
- Yet games continue to focus on adolescent males
- Outside of electronic games, women play games equally to men
- Diversity can only benefit a creative field

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

Designing and Developing the Game

- Basic steps for designing and developing the game
  1. Brainstorm game idea- define the FOCUS or X Factor of the game
  2. High level design document – storyboards
  3. Technical design document – data structures
  4. Develop iterative prototype
     1. Game loop
     2. Art and code to support desired graphical effects
  5. Play testing
     1. For correctness
     2. For fun-ness
  6. Repeat at step 4 or 3 until reach publishing deadline. Tune game.

Starting a Game Idea

- By Gameplay
  - E.g. "It's a first-person shooter, like DOOM"
- By Story
  - Tends to occur most often in RPGs or games based on films – e.g. Infocom Adventures, Knights of the Old Republic, Spiderman.
  - Often stories are created after the gameplay is decided. – E.g. Jedi Knight
  - Every game programmer thinks he/she's a novelist- avoid this mistake.
- By Technology
  - E.g. Using a game engine like Quake, UnReal would typically suggest a first or third person action/shooter rather than a strategy game.
- Because games are real time simulations that play on hardware that has limitations, the brainstorms often need to be constrained by technological capabilities and the abilities of a programming team to develop them.
  - E.g. Can you put 1000 enemies on the screen at the same time.
  - E.g. Can you simulate the AI needed to create a compelling computer opponent.

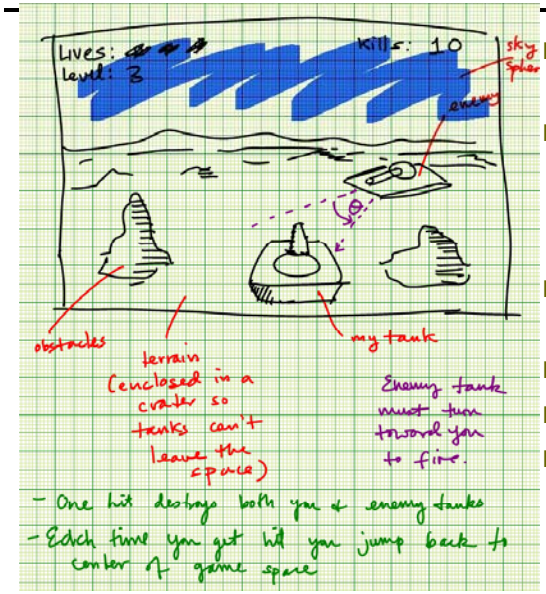
Commonly Encountered Documents

- Concept / Sales Pitch Document
  - Gameplay explanation, screen mockups, storyboards, description of art style, bios of team members, chosen platform, marketplace, budgets, estimated timelines
- Design Document
  - Different from traditional functional spec of say, a Word Processor. There is no way to spec a "winning" game. The game is often tweaked during the course of development. Spec consists of storyboards, backstory, etc.
- Technical Design Document
  - Data structures, AI algorithms, algorithms for implementing graphical elements – like explosions.

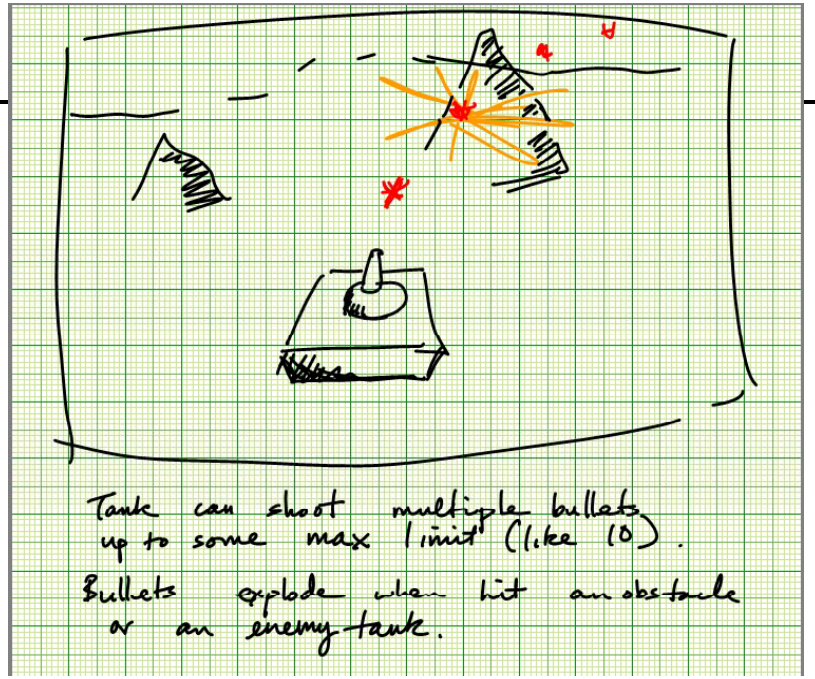
Game Design Document

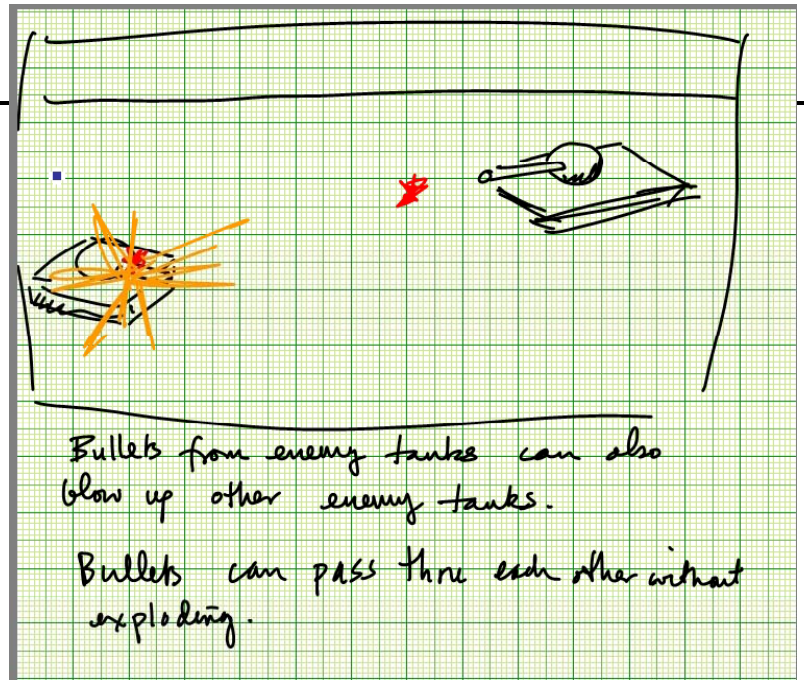
- **Executive summary** – the 1 page "big picture" summary – specifies the FOCUS of the game.
- **Game mechanics** - how the user reacts to the world. What can the user do in the world?
- **Artificial Intelligence** – how the world reacts to the user. How do opponents respond to the users?
- **Game Progression** – the events the player experiences. Game levels. What happens when the player finishes a task? What happens when you finish the level? What is different in the next level?
- **Game Elements** - Characters, items, objects/mechanisms – elements that are encountered, accessible, at each level of the game.
- **Story** – if there is one, the story helps define why certain gameplay elements make sense- e.g. why can the player fly?
- **Script** – final execution of the story and cutscenes will require a script.
- **Sound effects** – elements in the story that generate sound effects and the sound effects they generate. Background sounds.
- **Menus and interface design** – how users control the environment, game dashboard, opening menus, save/load screens, options & cheat screens.
- **Art Bible** – sketches and mockups of objects, characters, items.
- **Storyboards** – useful in illustrating aspects of the gameplay- e.g. camera angles or layout of game screens; storyboards are absolutely necessary for cutscenes.

Brainstorm the Gameplay (Using Storyboards)



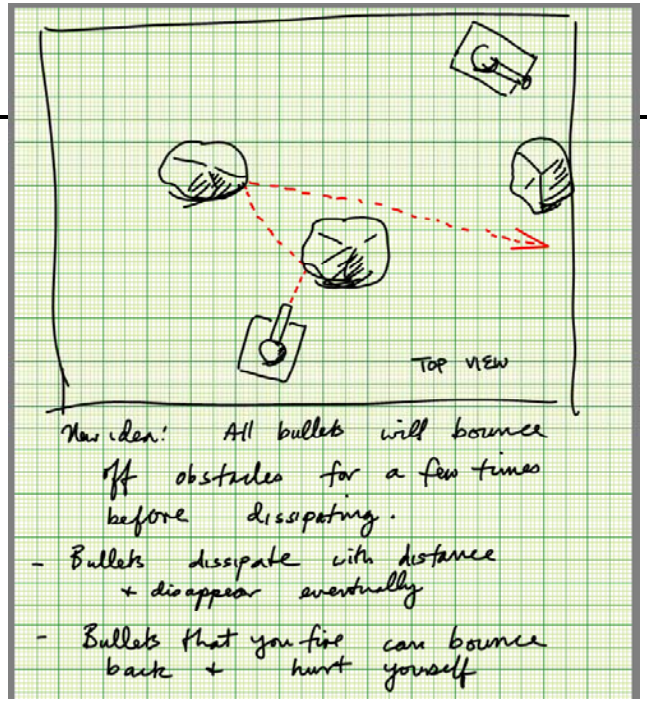
- Brainstorm the GAMEPLAY!
- DO NOT start by brainstorming the opening cutscene of the game.
- Draw them by HAND. NOT BY COMPUTER.
- Do them FAST.
- Let the IDEAS flow.
- DO NOT JUDGE the Ideas





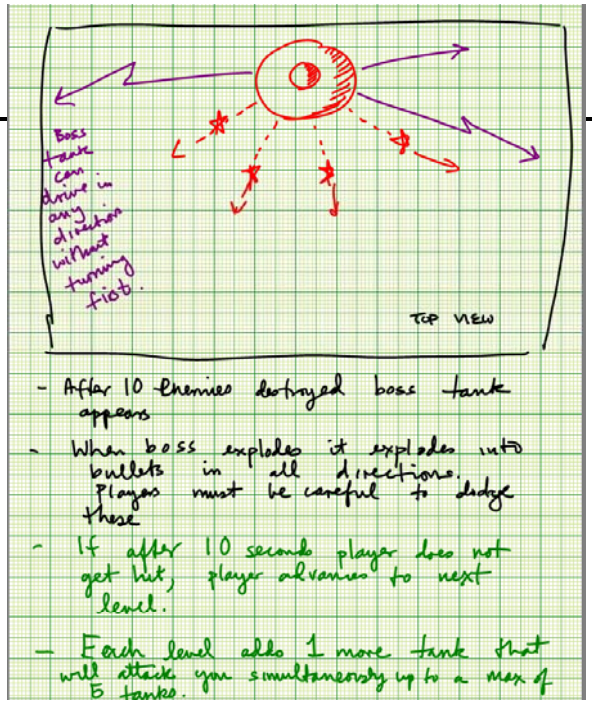
Bullets from enemy tanks can also blow up other enemy tanks.

Bullets can pass thru each other without exploding.



Hard Mode: All bullets will bounce off obstacles for a few times before dissipating.

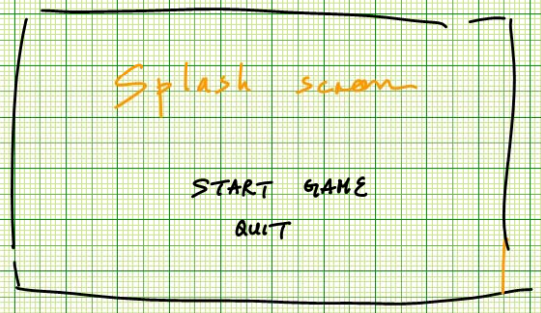
- Bullets dissipate with distance + disappear eventually
- Bullets that you fire can bounce back + hurt yourself

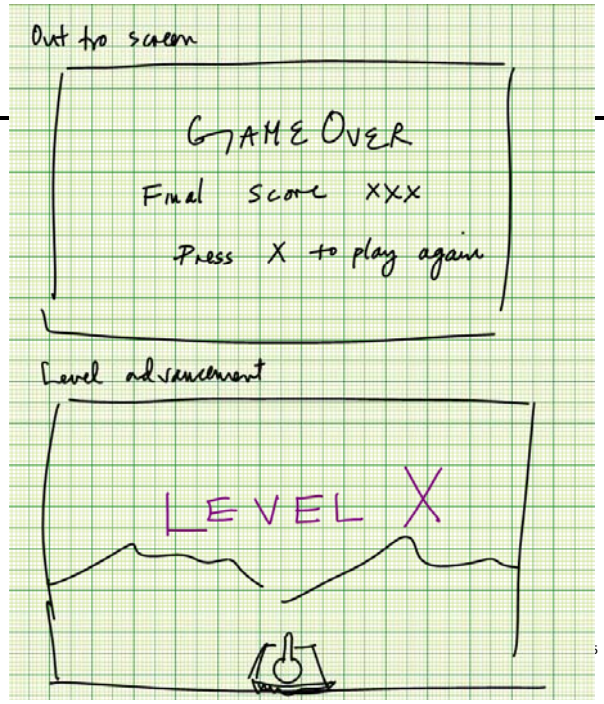


- After 10 enemies destroyed boss tank appears
- When boss explodes it explodes into bullets in all directions. Player must be careful to dodge these
- If after 10 seconds player does not get hit, player advances to next level.
- Each level adds 1 more tank that will attack you simultaneously up to a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





Final Report

- What I expect from the demo:
  - Game introduction
  - Gameplay instructions and user interface description (include diagrams and illustrations)
  - Gallery of all sketches, prototype images of designs, various versions of the game in progress, pictures of all aspects of the game during play. In total expect to see about 30 pictures
  - Technical documentation:
    - Finite State Machine diagrams – main loop, NPCs (Non player Character)
    - Data structures used
    - Descriptions and illustrations on how graphical effects are implemented – e.g. glowing fireball
    - Descriptions of basic AL algorithm
    - Descriptions and diagrams of any networking message handshakes used
    - Descriptions on what sound effects are in the game and how they were created – perhaps by merging several sound effects together
  - Full download of the entire game with source code and executes

Project Management

- You are not here to destroy your colleague. You are here to all work 110% because frankly that should be your philosophy in everything you do.
- There is too much mediocrity in the world why contribute to it?
- In life the only time positive things happen to you is if you do something well. Even then there are ways to do things better- so start with the best you can and always try to do better.
- The project manager has the most unenviable role in a company.
  - The project manager is the one who lies awake at night thinking about how to do things better & what the backup plans should be if something is late or falls through.
  - The manager has to be the one to speak frankly to the team member that is not delivering on time or to the right degree of quality.

Project Management

- When discussing issues:
  - Give everyone a chance to explain their decisions.
  - Provide criticism on the work, NOT the person.
  - Provide constructive but mindful ideas of how to improve something. E.g. "have you thought about trying X? The reason is, if you try X…."
  - Never take criticism personally.
  - No Whining- just get it done
- Everyone is busy - Always respect everyone's time. So:
  - Be on time at meetings. If you are late for a meeting you inconvenience others. It shows lack of respect for other people's time. My schedule does not revolve around yours.
  - If you have to be late due to something you cannot control, call someone at the meeting.
  - Set an agenda for the meeting well before the meeting.
  - After a meeting write down a list of resolutions / tasks / directives for each person. If someone comes away from a meeting with nothing to do- something is very wrong.

Project Management

- Put together a timeline by starting backwards from the deadline.
- Update timeline or check off DONE items at each weekly meeting and update it on the Web.
  - The project manager should be in charge of updating the timeline.
- Make a table with the following entries:
  - Due Date
  - Milestone (ie major goals)
  - Milestone Status / Comments
  - Tasks for Team Member A (ie tasks to achieve the goals)
  - Status of Tasks for Team Member A
  - Tasks for Team Member B
  - Status of Tasks for Team Member B
  - Etc..

Project Management

- Put down any special dates like family travel / religious holidays etc on the timeline so your colleagues know well in advance that you will be tied up.
- Exchange cell phone and email addresses.
- Create backups of your work on multiple computers / media. Backup includes your web site.
- Use a version control system like Subversion, if possible.
- Integrate parts of your game as often as possible- preferably weekly. DO NOT WAIT TILL THE END TO INTEGRATE.
- Keeping a constantly working version provides visible progress & therefore encouragement.
- Test this game on your friends for feedback. Write down the feedback and save it for your web site.

Reference

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