Hitbox
Contact: Alan Richardson
ajrich@ucsc.edu
What is Hitbox

- The 'how to play' fighter
- Player training
  - Game is chiefly a series of tutorials
  - Cover fundamental mechanics common to genre
  - Use and teach vocabulary developed by competitive community
- Drawn objects correspond 1:1 with simulator objects
  - Draw all hitboxes
  - No graphical 'clutter'
  - Color code to show state & kind
Why Hitbox?

• There is a huge gap in AAA fighting game design.

• No bridge between intuitive play and competitive play.
  – Learning competitive play involves “doing homework.”
  – “Practice modes” undirected, not particularly fun.

• To maximum extent possible, players should learn to play by playing.
How will Hitbox teach?

- Vocabulary lessons
  - Frames, hitboxes, priority, safe/unsafe, etc.
- General structure:
  demonstrate, define, test
- Custom movesets to
  accentuate the term at
  hand, scripted adjustments
  to framerate.

- Scripted challenges
  - Execute a given combo
  - Escape an AI performed
    combo
  - Matches against custom
    AI designed to test
    mastery of specific
    skills.
Project: Nightmare

3D platformer – fantasy/horror
Adrien Young
Eric Osugi
Chris Ward
Michael Tierney
Daniel Ball
Andrew Ball
Sam Goldberg
Tommy Hamala
Summary

- 3D platformer fantasy horror game
- Integrated platforming and simple combat
- Elemental magic system
- Scalable scope
- Powerful technical tools and engine
The Story

- Angeline, a young demon hunter, is sucked into a world of nightmares while searching for her mother's soul.
- To escape, she must destroy the foundations of the nightmare world.
Mechanics

- Mario64 style platforming mechanics
- Jumping, dashing, and sliding
- Simple melee and ranged combat with an emphasis on magic
The Magic System

- Either weapon can absorb magic from the environment.
- There are ten elements: Fire, Water, Earth, Wind, Light, Dark, Moon, Ice, Lightning, and Magic
- Each weapon has a spell that can be cast based on what element it contains
- While an element is active, it has an continuous effect
Scope

- 3 major levels, a final levels, and a hub world
- 15 enemies and 4 bosses
- Plans for 4 more levels
- 3-5 objectives per level
- 30 spells
Content Team

- 3 composers/musicians
- 3 3D artists
- 4 concept artists
- Particle effect assets and other procedural assets
- Motion capture studio and actors reserved
Technology

- C#
- Visual3D.Net 3D engine
- Lua scripting language
Project Spell Blaster Man! (working title)

Alex Ball and Daniel Ball
Basic Framework

- The game is an RPG with a real-time battle system.
- The player is on the left and the enemies are on the right. Each move freely in their area.
- The player uses a group of spells to defeat the enemy.
- The player has gear and the spells have mods called sub-runes that improve various spell attributes.
Selected spell

Player side

Enemy side
Interesting Mechanics

- There is no mana, instead spells heat up when cast and can “burn” if they get too hot.
- Burned spells need to be “unburned” to be cast again.
- There are several basic strategies, for example…
  - Cool spells as fast as possible
  - Keep spells hot but don’t burn them
  - Burn certain spells
- The game will need a variety of spells, spells mods, and gear to allow each strategy to be interesting but the number can shrink or expand.
Mix of RPG and Skill

- RPG’s require strategic planning, but not much skill during battle
- Action games require quick reactions and well tuned motor skills
- This game would mix the two, so while you cannot completely prevent all damage from enemies with skill, you can reduce it
Customization

- Player-chosen spell sub-runes and gear create many ways to optimize their favorite strategy.
- There is no clear choice of "best" gear or sub-runes, since they each have a use in a different strategy.
Scope

- The core of the game is the battle system
- Much of the game is easily scalable. For instance, there can be more/less monsters, levels, gear, or spells
- This project could be done with 4-6 people: 1-2 artists, 1 composer and 2-3 programmers
Penumbra – Detailed Pitch

- **Game**
  - Penumbra is a puzzle game where shadows house the enemy. You play as Sol, a young girl, who wants to rid the world of demons which have taken refuge in these shadows.
  - With each completion of a floor, the player descends, venturing farther down the tower, and encountering darker and more challenging levels.

- **Plan**
  - Penumbra is a game that can be made with a small team, with few art assets, while having a very thoroughly playtested and tweaked game mechanic.
Gameplay & Levels Progression

- **Core Mechanic**
  - The core mechanic is the rotation of the levels & the movement of lights around the scene.
  - The flashlight causes demons to disappear. A short time later, the demons reappear in the shadows.

- **Levels**
  - The goal of each level is to reach the end point and progress to the next level.
  - There is a time limit on the level, which gets stricter as the player progresses.
  - A literacy about the shape of objects and the shadows they cast is learned throughout the game.
Implementation

- **Graphics**
  - Shadowing mapping & stencil shadows are well explored techniques for producing shadows and can be implemented quickly and cheaply.
  - These techniques can be used in the prototype to tune gameplay, while more sophisticated and aesthetic techniques such as deferred lighting and radiosity can be implemented later with no effect on gameplay, but a large effect on the look of the game. These effects can be developed in parallel.
Resources

- **Development Platform/Tools**
  - PlayStation 3 SDK
    - Gameplay & Tools - C/C++
    - Graphics Pipeline
      - Collada (Models & Levels assets)
      - Edge (Animations)
      - PSGL (Lighting, Shading, Shadows)
  - Team Management
    - Subversion (SVN) – Revision control
    - Bugzilla (Feature Rich) / Trac (SVN Integration) – Bug tracking

- **Team Members**
  - 3 Programmers – AI, Gameplay, Graphics (Lighting), Tools
  - 1 Artist – Character Artist / Concept Art / Music
Resources - Continued

- **Art Assets**
  - Minimalist style will reduce to number of assets needed.
  - Required assets:
    - Female character w/ flashlight
    - Demons – Shadowy amorphous figures
    - Music – Transitions music from ambient and peaceful at the top of the tower to electronic & metal at the bottom of the tower.
    - Sound effects (Foot steps) – Achieved by public domain sounds.
CMPS170 Second Pitch

Jarrett Tierney

Email: jtierney @ ucsc

Roles: Team Leader, Gameplay Programmer, Game Engine Programmer
Tetryon

- Game Type: First person 3D Platformer.
- Game Style: puzzle platformer like *Portal* with aspects of a psycho-thriller
- Game mechanics: The player has to utilize his ability to create gravitational distortions and subspace effects to progress through the levels.
- Player also needs to maintain his focus during creepy situations
Game Innovation and Scope

- Innovation through the use of the gravitational distortions and subspace effects.
- Players become their own enemy. Distortions could pick up turrets and complicate puzzles even further.
- Game Scope: 4 levels each of them with multiple puzzles. Each level will also have a psychologically annoying or disturbing feature.
The Distortions

- Gravity Well: A rotational orb that rotates objects caught in its path around it and eventually inwards to an epicenter.
- Subspace Fold: Folds space to travel over large gaps and hallways.
- Black and White Holes: Black holes suck objects into it and smashes them. White holes do the opposite.
- Gravity burst: A subspace explosion that launches objects or players through the air.
The World

- Post-apocalyptic world in a partially destroyed mental asylum. But a mental asylum that makes its patients go from sane to insane.

- Hence sub-puzzles through left over experiments including a level featuring never ending elevator music and a creepy cat. Sub-puzzle: how do you turn off the elevator music
Raymond Calderon

Journey South

sasquatch@soe.ucsc.edu

General Programmer, Level Design, Musician
Game Concept and Mechanics

- Android system based 2D parallax side scrolling game.
- You’ve died and you must traverse the pre-underworld/underworld to get to your final resting place. The game’s story takes from all sorts of folklore: Nordic, Greek, Mexica, Mayan.
- Game will be a side scrolling game but like many games will have at least a few super secret “right” ways of finishing the level and then the regular boring get to the end way.
- Simple mechanics: Move back and forth and jump.
Game Innovation

- Game about death that exhibits many ideas from many people about what happens to the self after physical bodily death.

- The accelerometer or motion sensor will work completely work as the controller.

- To jump you can use the browser button or the main button, but to move back and forth you’ll need to tilt the phone.
Raul Galvan

g64993(a)ucsc
Programmer and Photoshop
Game Concept/Mechanics

- 2D side-scroller
- Main character gets sent 1 month before war began. Tries to figure out how to stop the war from starting.
- Many weapons
- Puzzles
Game Innovation/Scope

- Butterflies change next level.
- Can make different enemies/weapons appear or make level easier.
- “The Butterfly Effect” happens when killing butterflies.
Sam Wolf

(really likes math)
Bob From Accounting Saves the Universe

• The Universe is in peril and destiny has chosen Bob as its Savior.
• Yes, Earth is being invaded by mysterious alien conquerors, and only Bob is in the position to do anything about it.
This sounds a lot like Half-Life
But...

• The object of the game is to shirk responsibility, and run as far away from heroic situations as possible.

• Also, Bob is asthmatic. If things around him get too intense, he will have an attack.
Format

• 2-D sidescroller.

• Levels are Bob’s daily commute to and from work through a city. He must avoid both battles and the resistance’s attempts to recruit him for his advantage over the aliens.

• That advantage is that the aliens see him as the least threatening thing on the planet and as such, don’t notice him.
Time Limit

• Battles and recruiters are wasting Bob’s time and he needs to get to work! If he’s late, the stress will give him an asthma attack.

• The consequences for getting home late are less severe, but Bob will be sleepy for his morning commute the next day and will move slower.
Items

• Inhalers: Prevent one asthma attack each. If any are left when (and if) the morning time limit runs out, they are automatically used and the time limit is slightly extended.

• Coffee: Negates any negative effects of Bob staying out too late. MUST AVOID DURING EVENING COMMUTE.
Weapons

• Bob can pick up weapons to clear obstacles.
• Remember, if he gets too excited, he will have an asthma attack.
• Rather than employing an ammo system, each shot will have an impact on Bob’s nerves, and flashier weapons will have a bigger impact.
• The preferred way to use weapons is to convince an NPC to do it for you.
Innovations

• A game that has its main character avoid heroics at all costs and lead a normal, boring life.

• Players need to decide which routes are less stressful to Bob. Is it less stressful to jump over chasms than walk through a battle? Players will discover what stresses out Bob as they play.
Scope

• A few levels showcasing Bob trying (and failing) to find different routes to work that are less problematic.
2:35 Time Limit

Level of Excitement

Bob

Battle
Stack n' Deploy

VINIT AGARWAL
DEXTER LOHNES
MIGUEL VIDAURE
JOSH SCORCA
SAM ZUBLIN
ANDREW BROZEK
JOHN WILSON
JIMMY FISHEL
JOHN HUNTER FRANCIS
Multiplayer Digital Trading Card Game

- Unified TCG community, not limited by location.

- Cards govern the range of moves, with customizable decks
Stacking

- Equipment and crew are stacked on ships to create a single deployable unit.

- Card types: Crew, Equipment. Equipment can be weapons (missiles), passive bonuses (additional shields), or ability-enabling devices (radar jammer)
Deploying (Turn-Based Strategy)

Pros:
- Deployed cards become units that the player directly controls
- Players give new commands per unit per turn
- Easier to paper prototype
- More low-level micromanagement
- Mix of table-top strategy and card-game

Cons:
- Less approachable
- Slower games
Deploying (Card-Battle)

- More accessible
- High-level tactics
- Shorter/Faster game
- Innovative autonomous units
- Less about cunning tactics, more about high-level strategy
  - Focus is on playing cards, rather than specific movements on the battlefield.
  - Goal-based AI, rather than giving specific controls to the player.
Personalities

- In the turn-based strategy game, personalities control stats and response to enemy movement.
- In the Card-Battle game, the personalities govern the way that unit will approach a given goal.
Scope/Design Goals

- Multiplayer-only game, we hope to make it playable over the internet.
- A variety of cards
- We have an iPhone developer's license and lots of libraries. Great GUI elements already available.
  - Prototyping and RID (rapid iterative development.)
  - Basic traditional art. 2D sprites, no special tools.
Name: Tommy Hamala

E-Mail: thamala (at) ucsc (dot) edu

Desired positions: Level design, enemy planning, simple AI, non-graphics/music-related code
Types of Experience

Generic Experience - After every battle. +1 to all stats upon level up.
Physical Attack - Defeating an enemy with physical attacks. Atk+ Int-
Magic Attack - Defeat an enemy with magic. Int+ All Elmt+1 Res+ Atk-
Allied KO - Defeating an ally. Atk++ Int++ Def-- Res-- Spd++ Ail--
KOed finish - Finishing a battle while being KOed. Def++ Atk- Spd- Ail-
Status Effect - Inflicting an enemy with a status ailment. Int+ Ail+ Def-
Elmt Attack - Defeating an enemy with an elemental attack. There are 8 different kinds. Same Element+ Opposite Element-
Unarmed attack - Defeating an enemy while unarmed. - Atk++ Spd+ Int--
Cowardly Exp - Gained when fleeing a battle. Spd++ Others--
Defensive Exp - Gained for getting hit while using "Defend" or blocking an attack with a shield. Def+ Res+ HP+ Spd--
Dungeons

Can be taken in any order. Different strategies are needed for each. There will be numerous floors, with a boss at the end of each Dungeon. Beating all four will unlock the end-game bonus dungeon. (Might be randomly generated)
Bosses will be recruitable in the end-game.

Cave Dungeon - Enemies have low Atk/Int, but high Def/Res/HP.

Waterfall Cavern - Low Atk/Def, but high Int/Res. Elemental magic.

The Lavafeild - High Atk/Def, low Int/Res.

Heaven’s Tower - Low HP/Atk/Int. Very high Spd. Lots of status Magic.

Underground Catacombs - End-game. Maze with various themed areas. Possibly has floors with story/super-bosses.
The Town

Inn: Heal here for a nominal fee. Visitors stay here (changes each day)
Pub: Adventurers/Mercenaries are found here. Also, a barkeeper
    (who is recruitable) to manage your party.
Stores: Comes in Weapon, Armor, Sundries, and Black Market varieties.
    The Black Market is open at night and its stock changes daily.
Mind Science Institute: Reset all levels (except for generic) to base level.
    Item Synthesis (if we have time) done here.
Catacomb Entrance: Locked during main game.
Houses: Townspeople live here.
Main Street: Honest citizens and businesses can be found here.
Back Alleys: Dishonest citizens, the Black Market, and the Catacombs
can be found here.
Mansion: The Baron and his staff live here. He runs the town, and
doesn't trust you (or any other visitor). Hard to recruit.
Helios

Geon Lee  Art, Level Design
Ryan Dooley  Art, Level Design
Matvei Stefarov  Programming
“Tower Offense”

Tower defense
Slow-paced, repetitive, unforgiving

Shoot-‘em-up (shmup)
Fast-paced, hectic, reflexive

Both
Emphasis on survival
Defeating swarms of enemies
“Tower Offense”

We take best mechanics from both genres.

Tower Defense

Customization: construct your own ship.
Strategy: chose weapons and upgrades.
Cooperation: work together.

Shmup

Action: swarms of enemies everywhere.
Movement: manuever to survive.
Variety: many enemy types and bosses.
Constructing the Ship

Player builds a ship by installing towers into slots on one of the base ship frames.

Each base ship has limited number of slots and unique flight characteristics and features.

There is a vast assortment of towers that is gradually made available to the player:

- weapons
- shields
- engines
- radar
- power-ups
- etc
Movement

More “realistic” than most shmups: inertia! More towers = heavier ship = slower ship

Field-of-view and range limitations on some towers require player to decide not only the position, but also orientation of the ship.

For the most part, levels scroll (classic shmup). Players are free to move within the screen.
Level Design

Levels are organized around a storyline and are unlocked as the player progresses.

Each has alternating scrolling/special stages:

Scrolling:
Classic shmup - all about survival

Special:
Boss fights, traps, pursuits
Two-player coop is planned. Over LAN, internet, or locally (2 controllers).

Difficulty of normal levels will be adjusted, and there will be several coop-only levels with mechanics that rely on cooperation.
Co-op

Examples of coop mechanics:

Specialization:

e.g. Player in the front can get short-range weapons and shields, and the other can use slow and powerful long-range cannons.
Co-op

Examples of coop mechanics:

**Attack/defense:**
While one player has to halt to complete an objective, second has to defend both players.
Co-op

Examples of coop mechanics:

Siege:

Players split up to defend objective in the middle from attacks on both sides.
More customization

We don’t want the player to just move the ship around with the fire button held down.

Every weapon and tower can be tweaked to...

Constantly fire
Automatic trigger
Tied to a hotkey
Set to fire at weakest / strongest targets
- or closest / farthest
- or fastest / slowest
C#, XNA.
Targeting Windows and Xbox360.

2D gameplay (shooting/movement)
3D graphics (top-down view)

Tools
Visual level editor
Internal scripting language (Lua?)