Project Bipolar Man! (working title)

Alex Ball and Daniel Ball
Idea

Ned is a man with more mental disorders than his brain can handle. Ned’s brain finally goes on strike and makes Ned fall into a coma. His brain gives him an ultimatum: Fix your problems yourself or stay in this coma forever. To wake up, Ned needs to delve into the mysterious reaches of his own mind and clean up the mess one disorder at a time.
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 powers to defeat the enemy
- The player has gear and their powers have mods that improve various attributes.
Interesting Mechanics

- There is no mana, instead powers become more “manic” when used and can cause Ned to have a mental breakdown if he gets too manic.
- Powers “depress” over time.
- There are several basic strategies, for example…
  - Depress powers as fast as possible.
  - Keep powers manic but don’t cause a breakdown.
  - Cause a breakdown with certain powers.
- The game will need a variety of powers, 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 mods and gear create many ways to optimize his/her favorite strategy
- There is no clear choice of “best” gear or mods, 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 powers
- This project could be done with 4-6 people: 3-4 programmers, 1-2 artists, 1 composer
Penumbra

- Demons have entered your world and taken refuge in the shadows.
- A puzzle game where shadows are manipulated by the rotation of the level.
- Theme explored: dynamics of shadows
- Creates a literacy relating object shape with the shadow it casts.

Monday, October 12, 2009
Level Progression – Intro

- First level of the game.
- Introduced to the concept of shadows as an obstacle.
- Does not require use of the game play mechanic.
Level Progression - Rotation

- Given the knowledge just gained, level appears impossible.
- Rotating the world, which changes the lighting, reveals the solution.
Level Progression - Windows

- Similar to the previous level.
- Player needs to move with the rotation of the world.
- Introduces the idea that shadows are dynamic to the player.
Level Progression - Boxes

- Normal puzzle challenge of pushing boxes around with a twist.
- Player must be aware of sources of light.
- Shadow can come around and hit the player.
Level Progression

- Moving portions of the level cast moving shadows.
- Reenforces the idea of dynamic shadows.
Team Structure - Roles

Lead Designer
- Leads design meetings
- Primary Author on Design Document
- Enforces coding style
- Final word on level design
- Branching/Merging (SVN) Czar

Programmers (3 – 4 people)
- Implements architecture of lead designer
- Designs and tests levels
- Fixes bugs/issues
- Verified bug fixes

Producer
- Leads bugs/issues meetings
- Maintains Bugs/Issues Tracking
- Updates/Enforces schedule
- Main communication to artist

Artist – Joshua Palacios
- Character modeling & texturing.
- Skybox Matte painting
- Additional minor model needs
Target Platform / Key Technology

Development Tools

- **Languages:** C/C++
- **PlayStation 3 SDK – Real-world programming experience**
  - Graphics Pipeline
    - Collada (Models & Levels assets)
    - Edge (Geometry & Animations)
    - PSGL (Lighting, Shading, Shadows)
- **Team Management**
  - Subversion (SVN) – Revision control
  - Trac (SVN Integration) – Bug tracking
Technology Pipeline

Graphics
- Models and levels are created in Blender/3dsMax/Maya
- Texturing in GIMP/Photoshop
- Conversion into PS3 Preferred Formats
  - COLLADA for scene asset management
  - DDS (GPU accelerated format) for texture files
  - Converters are highly portable plugins.

Development
- Subversion (SVN) – atomic commits to single code base
- Branching for feature implementation
- Assignment of bugs and issues via Trac
Assets

- The minimalist style reduces the number of assets.
- **Character Models:** Joshua Palacios
  - Sol – female character
  - Demons – shadowy amorphous figures (perhaps particle system)
- **Levels:** Everyone!
  - Puzzles will be designed and created by the entire team.
  - Backgrounds may either be procedurally generated or outsourced to the artist.
- **Sounds**
  - Few sounds needed – Footsteps, wind, thunder
  - Public domain is a viable option. (Sources: Partners in Rhyme, PDSounds)
- **Music**
  - Public domain has may ambient music tracks (Possible Sources: Archive.org)
    - Two team members have musical experience.
Individual Contributions

- John Bowidowicz
- Raymond Calderon
- Raul Galvan
- Diego Villafana
- Samuel Wolf
Nowhere is farther from home then a nightmare.

Guide the Little Witch, the Blood Rose Angeline, in her quest to escape.
Team

Adrien Young
Eric Osugi
Chris Ward
Jarret Tierney
Daniel Ball
Andrew Ball
Sam Goldberg
Summary

- 3D platformer fantasy horror game
- Nightmare vs Steampunk visual style
- Integrated platforming and 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.
- Mario64 style platforming mechanics
- Jumping, dashing, and sliding
- Melee and ranged combat with an emphasis on magic
Verb Types

Combat (melee)
- Downward Slash
- Hack
- Stab
- Horizontal Slash

Combat (ranged)
- Fire Forward
- Hold Release

Camera
- Adjust Camera
- Default Camera

Platforming
- Jump
- Run
- Dive
- Wall Jump
- Crouch
- Evade
- Backflip
- Long Jump
- Roll
- Drop Off
The Magic System

- Either weapon can absorb magic from the environment.
- There are six elements: Fire, Water, Earth, Wind, 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
Innovation

- Player is given magic based on what element is nearby
- Each element has its own play style and platforming mechanic.
- Dark, bizarre, nightmarish levels contrast with the steam-punk, slightly gothic main character.
Why 3D?

- 3D allows for exploration
- 3D gives the player more options
- 3D combat has significantly more depth
- Ranged attacks make more sense in 3D
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
- 18 spells
Technology

- C#
- Visual3D.Net 3D engine
- Lua scripting language
- CharacterFX animation studio
Primary Character
Model: Contracted to Girish Kare, enforceable under US contract law
Animation: Arrangements made with on-campus motion capture studio
Animation Actor: Arrangements in process with Ashoka McCormick

General Modeling
Adrien Young, Sam Goldberg, Eric Olund (Cogswell Polytechnic)
Additional Resources: Cogswell Polytechnic, Vancouver School of Art

Music and Sound Design
Stephen Hawkins, Jacob Pernel, Marc Wolterbeek

Concept Art
Teaghe Yalon, Pricilla Truong (paid), Daniel Ball, Chris Ward
Animation Assets

- Motion capture studio available, as well as actors
- CharacterFX comes with animations, and can be used for touching up motion capture
- Visual3D.Net supports animation out of box, including transitional animation
Stack 'n Deploy

VINIT AGARWAL
DEXTER LOHNES
MIGUEL VIDAURE
JOSHUA SCORCA
SAM ZUBLIN
ANDREW BROZEK
JOHN WILSON
JIMMY FISHEL
HUNTER FRANCIS
Game Concept

- Roughly 5-10 minute matches (short and focused)
- fast paced competitive digital trading card game
- customizable ships
- command with the objective of destroying the enemy base
- commander: you must understand your ships, learn your pilots' personalities, and lead them to victory

- Accessible

- Innovative autonomous units
- Emergent strategy (chess style)
S.P.E.C. System

Card Types:

Ships - this is the vessel that the other two types of cards stack on. This is the "base unit".

Pilots - create the "personality" for the ship, and give passive bonuses to the ship.

Equipment - add new abilities, both active and passive, to the ship.
Personalities

- Governs the manner in which a unit completes the commander-issued objective.

- Strategic decisions rely on knowledge of pilot personalities and how to employ them -- not micromanagement.
Game Match Flow

- Map consists of nodes that the player moves units between
- Both players have their own base, located at opposite ends of the map.
- Players take simultaneous turns on a frozen game state
- Upon committing moves, game updates.
- Game ends when a player destroys the enemy player's base.
Player Turn Flow

DRAWING

- Turn begins with the player drawing new cards into their hand.

QUEUING

- The player takes cards from their hand and sends them to production

STACK N' DEPLOY

- Cards that have completed production are available to combine to create a single customized ship.

COMMANDING

- A deployed unit is movable across the map. The objective is to eventually destroy the enemy base.
Scope/Design Goals

- Multiplayer-only game, we hope to make it playable over the internet.
- A variety of cards (scalable)
- 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. Lack of dependency on art.
  - Want to create a game we can release to the public.
Summary of Innovation

- Card game to iPhone interface.

- S.P.E.C. system to combine cards for final unit.

- Personalities (A.I.) to allow for autonomous units, reducing micromanagement in a typically micromanage heavy strategy genre.

- Personality system enforces importance of story/characterization as an element of gameplay.

- Ultimately, we don't want to make another Mafia Wars clone. A.K.A., we want to make a game that actually means something on the iPhone.
Dyson Game : "Let's get back to our roots."

→ Units with automated behavior.
→ No tiles, simple map scheme.
→ Procedural graphics.
→ Fluid control scheme.
Final Pitch: Team ARC

Galactic Arms Race

→ Morphing technology items
→ Procedural Graphics
→ AI manipulated technology items

Monday, October 12, 2009
They missed a playable mechanic.

Instead of using AI to procedurally generate cool visuals and an "online learning weapon" (GAR), or just use procedural graphics generation for aesthetics (Dyson), we can make the procedural elements interactive to the player.

ARC will use the tech morphing as an interactive device, and AI behaviors to simulate character units.
ARC Infinitum

A Fleet Command game with procedural technology items, set in a galactic Sci Fi Simulation.
High Concept Statement

The player, who assumes the role of fleet commander, explores a simulated galaxy full of conflict and discovery.
Emergent Real Time SANDBOX using strategy game mechanics.

**Audience**: RTS players who are sick of ”the micromanagement grind”, RTS players who want more than ”war strategies”, gamers who want a ”strategy immersion experience”

**Platform**: PlayStation 3, developers will be under the PS3 academic program, using Sony's SDK. 2D logic over 3D logic.

**Game Flow**: Players run a ”Saga”, with a fleet. Players form fleets and deploys them to locations, manipulates technology items and manipulates Aibo behavior to explore a simulated galaxy.

**Storyline and Campaigns**: Unique races in a Space Opera saga, written by Andy Lucas, a local science fiction writer and veteran gamer.
Scope
This year:
Single Player,
PvE sandbox,
1 storyline "tutorial" campaign.

Ambitions for Spring Quarter:
simple 4 Player LAN,
complete storyline campaigns,
"Blind Objective Mode": Each player is assigned an objective, and everyone tries to achieve their objective (which might collide with someone elses object). Allows players to form alliances (and then betray them) or turn enemies into allies.
Key Innovations

A.I. Boid units with distinct behaviors
Interactive Technology Morphing
Fleet Formation Tool
Role of the Player
She is the primary fleet commander of her civilization. She has control over "fleet high command" and she guides the star fleet to explore the galaxy.

Primary Gameplay
The Player morphs her civilizations technology. The Player manipulates formations and fleet movement. The Player interacts with a simulated galaxy by observing and reacting to layered dimensions of strategy.
Core Game Modules

2D logic map based on orbital discs.
AI Boids as character units.
2D logic, procedural weapons/items.
Formation based unit control.
2D Discworld

Provides different modes of view; Planetary Orbit, Star Orbit, Galactic Orbit.
Final Pitch : Team ARC
( I.Te.M )

- Achieves the following modes:
  - 1. Dynamic level progression.
  - 3. Extensive replay value.
- All in one code base!
Step 1. Reduce character actions to basic geometric components.
Step 2. Use functions to manipulate these geometries.
Affordance: Player manipulates stack function placement and order to simulate a technology upgrade.

Affordance: Allows for emergent gameplay with respect to level progression AND regression.

Agile and Scalable code base: stack size of 8~128, define 8~32 primary functions.

We want the player to feel that they ARE discovering and manipulating new technology, not just navigating an arbitrary menu.
Boid System

Autonomous character units.
Archetypes to represent different kinds of boids.
Affinity actions; positive, negative, neutral.
Affinity actions mapped to a tech item.
What are these boids?

- Each boid represents a faction of that civilization.
- The players fleet is a composition of these different faction units.
- Boids are the main point of interaction with the player.
- An NPC civilization is a composition of different boids.
Example : Warlord Boid

[+] DEFEND → uses Shield
[-] ATTACK → uses Beam Cannon
[/] PATROL → uses Command Center
Proposed Faction Archetypes

WARLORDS → warfare
ENGINEER → structures
CLERGY → allegiances
SCIENCE → technology
PHILOSOPHER → behavior
EXPLORER → space

These factions manipulate specific RTS dimensions, and place them into the context of a sandbox galaxy, populated by autonomous units.
Combinations of Archetypes

Discovering a new civilization means interacting with the primary composition of their fleets.

Each civilization will also have an abstracted affinity, which simulates their behavior to the player.
Affinity
[+] Positive
[/] Neutral
[-] Negative

A system where a player can express a particular style of play, that is mapped to a boid behavior and its respective technology item.

Gives the boids a "modus operandi", and simulates their intentions and actions.
Cognitive Innovation

- Player has multiple dimensions of unit interaction to observe, and plot strategies.
- Instead of micro-managing, you have automated actions that make room for the player to plot and decide.
- Added with dynamic tech items, the player experiences a modular and interactive level progression.
Narrative Innovations:

- Each type of boid manipulates with a unique strategy dimension, not just "war".
- By defining an affinity system, we can map actions to a play style; making this game accessible to everyone; from carebears to griefers.
- Makes all types of players interact in the same game multiverse (for multiplayer).
<table>
<thead>
<tr>
<th>Team ARC</th>
<th>WARLORD</th>
<th>Assault ( - )</th>
<th>Patrol ( / )</th>
<th>Fortify ( + )</th>
<th>CLERGY</th>
<th>Sacrifice ( - )</th>
<th>Bless ( / )</th>
<th>Convert ( + )</th>
<th>EXPLORER</th>
<th>Piracy ( - )</th>
<th>Scout ( / )</th>
<th>Survey ( + )</th>
<th>Default</th>
<th>ORBIT</th>
<th>FOLLOW</th>
<th>SEARCH</th>
<th>MOVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENGINEER</td>
<td>Fabricate ( + )</td>
<td>Trade ( / )</td>
<td>Sabotage ( - )</td>
<td>SCIENTIST</td>
<td>Experiment ( + )</td>
<td>Observe ( / )</td>
<td>Deconstruct ( - )</td>
<td>PHILOSOPHER</td>
<td>Pacify ( + )</td>
<td>Mediate ( / )</td>
<td>Argue ( - )</td>
<td>ubuntu</td>
<td>Monday, October 12, 2009</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
- **Fleet Formations**

  - Create a social structure (expressed as a flocking behavior) between different boids.
  - Within the context of affinity, archetype, and technology, the player can make a wide range of different fleets: trade based, exploration or saboteurs.
  - From a template formation, the player can gives them missions, allowing them the affordance of "high command".
What the heck is ARC?

Ascension

Ragnarok

Chaos
Possible "Sandbox Objective"

Ascend

The player must succeed in a progression of positive affinity actions, so her civilization can ascend to a higher plane of existence.

Rewards: the civilization leaves behind their technology, and the player can start a new race!
So now, when the player starts a new civilization, you can discover this on your planet!
Production Capabilities

- We have artists who work on "hobbyist schedule", 3d models aren't essential but makes us look cool.
- Our coders have all been exposed to boids and AI coding, and our code base is very modular and scalable.
- The PS3 devkits provide us with support, documentation, tutorials and code examples for every coding module needed.
- We have been researching the PS3 and actively recruiting artists/musicians over the summer.
**Core Members**

- Slade Villena (OpsChief)
- Laker Sparks (Lead Design)
- John Bowidowicz (Graphics Lead)
- Greg Fuller (Programmer)
- Daryl Wong (Programmer)
- Alan Richardson (Programmer)
- Off Campus Members:
  - Rustan Melville (Art Director)
  - Andy Lucas (Storyline, Campaigns)
  - Bradley Burr (Music Lead)

**We have the following core members, and a full roster of concept artists, musicians and 3d modelers working on “hobbyist hours”**.

**Full Roster at:**

http://forums.soe.ucsc.edu/viewtopic.php?f=79&t=6395
Tetryon
Team Roster

- Jarrett Michael Tierney
- Chris Ward
- Tommy Hamala
- Eric Osugi
- Alan Richardson
What is Tetryon?

- Tetryon is a 3D puzzle platformer in a similar model of *Portal*.
- The player in the game has to use gravitational distortions and subspace effects to progress through the levels.
- The player is thrown into a psycho-thriller game in the setting of a post-apocalyptic mental asylum.
Tetryon Base Information

• Genre: First Person Puzzle Platformer, Psycho-thriller game
• Target Audiences: Portal fanbase, Puzzle game fanbase, psycho-thriller fanbase, open source gaming community (i.e. Nexuiz)
• Camera Model: First Person camera system like in first person shooters.
Game World and Play Aesthetic

- Psycho-Thriller aesthetic
- World will have full range of colors but normally muted. Except for light from the outside to emphasize the outside as freedom
- Post-apocalyptic partially destroyed asylum
- Asylum utilized to turn sane people insane
What are the Puzzles Like?

• Main Puzzles: main set of puzzles involve getting through the levels, repositioning objects utilizing the distortions
• Secondary Puzzles: Each level will also have a psycho-thriller aspect that will be connected in some way to a secondary puzzle that the player will receive “Sanity Achievements” for solving.
Sample Room with Puzzle

How do we get to the gap!?
Sample Room with Puzzle

Demonstrates the basic need to use the Gravity well to create a staircase
But wait? There’s no enemies?!!

- True in the standard sense of enemies there aren’t any.
- However, like in *Portal* there are turrets.
- The player will have to avoid or destroy turrets while also being careful not to over-complicate a puzzle with turrets.
The Distortions

• Core Distortions:
  – Gravity Well
  – Black Hole
  – White Hole
  – Gravity Burst

• Extra Distortions
  – Subspace Fold
  – Subspace Ripple
# Game Core, Mechanic Breakdown

<table>
<thead>
<tr>
<th>Core</th>
<th>Planned Expansions</th>
<th>Extras (if we have time)</th>
<th>Out of Scope for this year but planned for further development</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Gravity Well</td>
<td>• Psycho-Thriller moments including scripted events</td>
<td>• Subspace fold</td>
<td>• Multiplayer online</td>
</tr>
<tr>
<td>• Black Hole</td>
<td>• Secondary Puzzles</td>
<td>• Subspace ripple</td>
<td>• 14 more levels</td>
</tr>
<tr>
<td>• White Hole</td>
<td>• Ability to charge the intensity and size of the distortions</td>
<td>• Subspace psycho-thriller pockets</td>
<td>• More psycho-thriller aspects and fleshing it out even further</td>
</tr>
<tr>
<td>• Gravity Burst</td>
<td></td>
<td>• Changing so that the objects smashed into a wall mesh into a single object</td>
<td></td>
</tr>
<tr>
<td>• Basic scripted interaction with the levels</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• First Person mechanics and aiming mechanics that work smoothly with the distortions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Primary Puzzles</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Learning Curve

• The player undergoes a tutorial in the first two rooms of the first level where Heads Up Display info is presented to the player to teach him how to create the initial distortions.
Skill Progression

• The player starts out in the first level with only a single distortion and ends with a second. When the second distortion is acquired the HUD will give a tutorial on how to switch distortions.

• The third and fourth core distortions will appear in the second level.
Engineering Team Roster

- Michael Tierney: Producer, Core Engine Programmer, Interfacing
- Chris Ward: Lead Designer, Gameplay programmer, UI Programmer, Scripter
- Tommy Hamala: Gameplay programmer, scripter
- Eric Osugi: Core Engine and gameplay programmer
- Alan Richardson: Core Engine Programmer, Gameplay programmer, UI programmer
Art Team Roster

• Matt Coleman: 3D Level Designer, Sound Effect Artist
• Sam Goldberg: 3D Modeler, 2D artist
• Animator, Rigger: Jason Moore
Project Plan

• Platform: Personal Computers: Windows, Linux
• Technologies and Engine: Ogre3D 1.7 + Bullet Physics + OgreOggSound (OpenAL) + OIS + my-GUI
• License: Open Source non-commercial game
Assets Plan

• 3D levels and Sound effects: Matt Coleman
• Fallback sound databases:
  – Findsounds.com
  – Freesound.org
  – PdSounds.org
Assets Plan continued

• 3D customized art created by Sam Goldberg
• Fallbacks:
  – Laser turrets and bullet turrets can be modeled by Michael Tierney
  – Creepy Cat can become a photo undergone photoshop
  – Zbrush + MakeHuman for hand models
Art Fallbacks

• 3D model databases:
  – 3dm3
  – Katsbits
  – Blender model repository
  – Low poly co-operative
  – WikiWorlds
  – ShareCG
Further Sources and Tools

• Help wanted posts at: cgsoiety.org, katsbits.com, and game-artist.net

• Free tools:
  – MakeHuman
  – Mapzone
  – OgreRadiant
  – Nvidia FX-Composer
  – Nvidia Melody
  – Blender
  – Character-FX
  – Scythe Physics Editor
For More information

• For more information on my team task split down and time table see my final pitch document
QUESTIONS?
“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 the best mechanics from both genres.

Tower Defense

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

Shmup

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

On the loadout screen, the player builds his/her ship by installing **towers** into **slots** on the base frame.

There will be **9 base ship frames**, each with a unique slot layout and flight characteristics.

New types of towers can be constructed out of the **debris** collected from killed enemies and the [gradually unlocked] tower **blueprints**.
Constructing the Ship

There will be many different types of towers:

- weapons
- engines
- shields
- radar
- power-ups
- rockets
Controls

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

**Action:**
- Movement
- Orientation
- Aiming
- Shooting

**KB/Mouse:**
- WASD
- Q/E
- Mouse
- Mouse Btts

**Controller:**
- Left Analog
- LB/RB
- Right Analog
- Triggers
Level Design

Nine levels are planned, spanning three distinct environments and organized around the story. Each has alternating scrolling/special stages.

Scrolling:
- Survival

Special:
- Boss fights
- Traps
- Pursuits
Co-op

Two-player coop is planned!

**PC:** over LAN and Internet

**Xbox Live Arcade:** two controllers

Enemy count is greatly increased in coop games, to maintain the overall difficulty.

All nine levels are available for coop, plus certain coop-only stages in each level.
Co-op Mechanics

Siege

Double Defense

Specialization
An experimental test for a new space engine goes awry, creating a temporary wormhole. Two ships are sucked in and sent halfway across the galaxy. Badly damaged ships start the journey home with limited defenses. Unbeknown to them, they have been taken behind the lines of a vicious, xenophobic race of aliens and their only way home is through enemy territory.
The Tech

2D Gameplay
Players, enemies, obstacles are on the same plane. Movement/combat simulation is 2D.

3D Models
3DSMax and Maya.

Textures
Photoshop.

Sound
Reason (sfx)
Ableton (music)
The Tech

XNA (Managed DirectX 9+, C#)
Easy to prototype, debug, and port.

Lua scripting
Allows for special, dynamic stages on levels.

Visual level editor
Level editor will share a codebase with the engine and use Windows Forms and GDI+.

Targeting Windows and Xbox360.
Scope Summary

3 environments
9 levels
  2-3 special stages per level
  1-2 co-op special stages per level
9 ship frames
15-20 tower types

Questions?