

Game Design Document

Project Name: Arcana Game

Team Name: Arcana Team

Team Members:

Haneen Mandurah – Background Illustrator

Jessie Gao - Animator

José Romero – Game Designer / Programmer

Kenneth Andersen – Lead Programmer

Marcel Tau – Project Manager / Animator

Neeveen Bhadur – User Interface Designer

Randy Ruan – Asset Manager / Render Wrangler

Game Design Document Version Control

Version	Date	Author	Change Description
1.0	Feb/09	Marcel Tau	Document created.
1.1	Feb/13	Marcel Tau	Document updated.
1.2	Feb/22	José Romero	Game description and Functional Specifications updated. Sound & Music/Story bible added.
1.3	Feb/23	José Romero	Sound & Music assets added.
1.4	Mar/19	Marcel Tau	General document updated.
1.5	Mar/22	Marcel Tau	Functional Specifications and Art and Video sections updated.
1.6	Mar/29	Marcel Tau	Level Requirements updated.
1.7	Mar/30	Marcel Tau	Production Schedule updated.
1.8	Apr/08	Kenneth Andersen	Technical Specifications updated.
1.9	Apr/10	Randy Ruan	Assets Pipeline added.
2.0	Apr/12	José Romero	Technical Specifications updated.

TABLE OF CONTENTS

GAME DESCRIPTION	5
Design Goals.....	5
Influences & Sources.....	5
Target Market	5
FUNCTIONAL SPECIFICATIONS	6
Game Mechanics	6
Core Game Play.....	6
Game Flow	6
Characters / Units	6
Game Play Elements.....	7
Game Physics and Statistics.....	8
Artificial Intelligence.....	8
Multiplayer	8
USER INTERFACE	9
Flowchart.....	9
GUI Objects	14
ART AND VIDEO	14
Overall Goals.....	14
2D Art & Animation	15
GUI	15
Marketing and Packaging Art.....	16
Terrain	17
Game Play Elements.....	18
Special Effects	20
3D Art & Animation	20
Cinematics	21
Assets Pipeline.....	21
SOUND AND MUSIC	22
Overall Goals.....	22
Sound FX.....	22
STORY	24
Player Characters	24
Secondary Characters.....	24

Enemy Characters	24
Story theme.....	25
Visual theme	25
Story Outline.....	25
LEVEL REQUIREMENTS.....	26
Level Diagram	26
Asset Revelation Schedule.....	26
Level Design Seeds.....	26
TECHNICAL SPECIFICATIONS	27
Game Mechanics	27
Game engine	27
Platform and OS	27
External Code	27
Code Objects	27
Control Loop.....	28
Game Object Data	31
Data Flow	31
Artificial Intelligence.....	31
PRODUCTION SCHEDULE	33
Scope.....	33
Scheduling	33
Dependencies	33
Cost Estimate.....	34

Game Description

The Intrinsic Game is a 2D brawler style game for the iPad based on the comic book series of the same name by Arcana Comics, taking place in an epic battle for the future of the universe, where a powerful team of heroes face off the hordes of the apocalypse.

Design Goals

The game aims to achieve the following goals:

1. Establish the Arcanaverse and familiarize players with the characters and events of The Intrinsic, as well as with other characters from books published by Arcana.
2. Serve as a prototype and first level of an ongoing serialized comic book video game.
3. Develop an achievements system that rewards the player with Arcana content (comic book covers, videos, comic books and artwork).
4. Develop a fun gameplay, based on:
 - a. Discovery – players discover the storyline, new levels, as well as the several achievements and prizes.
 - b. Advancement – the playable characters will be able to advance in levels and become stronger.
 - c. Power – the game is about fighting your enemies and emerging victorious through several battles.
5. Have art and design that are faithful to the Arcana style, as developed in the comic books.

Influences & Sources

We have extensively researched the Arcana books and characters in order to get an idea of the right look and feel for this game, as well as to develop ideas for the gameplay, story, and interactions. We paid particular attention to the books of Kade, Kore, Philosopher Rex, Candice Crow, as well as the resources that were provided by Arcana.

Additionally, the team researched other comic books and comic book based video games by publishers such as DC and Marvel. In particular the gameplay and story development of Marvel's War of the Gems and its corresponding game, and DC's Infinite Crisis.

We reviewed several games in the side scroller 2d brawler action genre or related ones: Marvel War of the Gems, Battle Heroes, Wolfboy, Colosseum, and many others.

Finally, we looked at modern iPad apps with some relation to our development, such as digital comics, social games and e-books.

Target Market

Our ideal user has the following characteristics:

1. Male;
2. 13 to 16 years old;
3. Has an iPad;
4. Likes games;
5. Likes comics;
6. Plays mobile games;
7. Consumes many types of content on the iPad;
8. Purchases content through apps, or online;

Although young males are the primary targeted user, the game at its fundamental level has a more universal appeal.

Functional Specifications

Game Mechanics

Core Game Play

The PC (Playable Character) moves in a 2D (two dimensional) plane, exploring side scrolling city levels that extend from the left to the right of the screen. Enemies appear along the way as the characters try to fulfill quests, and the players must defeat them to move along. The player may also encounter obstacles that can be destroyed by attacks, and secondary characters that are being held hostage by several enemies, that must be defeated in order to free them.

As the PC advances through the levels, specific actions and events will trigger achievements that are rewarded with Arcana content, which can be retrieved at any time through the achievements menu screen. The PC's actions will also be rewarded with experience points which will allow it to "level up" and receive stats points, which can then be applied to increase the three different stats that govern the PC's strength.

Gameplay related dialogue scenes take place and are displayed for the player during gameplay through the HUD, and cut scenes are displayed between levels/stages, advancing an overarching storyline that corresponds to the events in The Intrinsic book.

Three stages make up a level, and the PC may face a mid-stage boss at the end of each level and a final boss at the end of the third level.

Game Flow

Actions that the PC can perform are:

1. Move left and right.
2. Jump up and jump while moving left and right.
3. Attack once.
4. Attack repeatedly in a combination of at least 2 moves, ending in a more powerful attack.
5. Perform a special attack move.
6. Destroy obstacles.
7. Jump up to platforms and hidden areas.

Characters / Units

1. *Kade*: Sword-wielding white skinned warrior that feels no pain and can withstand tremendous punishment. Stats should be very high health, average strength, and low special, and he should have average speed.
2. *Kore*: Human Alex Crane was turned into the Mantikore, a very powerful giant red beast. Stats should be very high strength, high health, and average special, and he should have very low speed.
3. *Tori*: Apparently a regular human, Tori is discovering her newfound psychic abilities on the go. Stats should be very high special, low health, and low strength, and she should have very high speed.
4. *Minions*: Possessed humans that disappear into dark smoke once defeated. The enemies overall health, strength, and speed will vary and increment as the player advances in stats and levels.
5. *Apollyon*: Boss and Commander of the horde of demons. If the player encounters him more than once, his stats should increment with each appearance. He's a very large character who will have at least three different attacks of varying speed and power, which should require skill for the PC to evade.

Game Play Elements

These are elements present in the game level that the PC can interact with:

- *Health recovery item*: Items that recover the player's health bar.
- *Power recovery item*: Items that fills the player's power bar.
- *Destructible obstacles*: Obstacles that hinder the player's progress and must be attacked until destroyed.
- *Platform obstacles*: Obstacles that hinder the player's progress and can be surpassed by jumping over them.

Additionally we have elements that communicate the PC's game state:

- *Health bar*: Bar that represents the PC's health, which remains the same size but increases/decreases differently depending on the health stats of the character. Once all of their health is depleted the PC dies.
- *Power bar*: Bar that represents the PC's special power move. Once the bar is completely filled that character's special move is available for the player to use.
- *Dialog scenes*: These will be small panels presenting dialogue in the form of comic book panels with text that will be displayed for the player on the HUD during gameplay. The player will receive relevant information through these dialogues.

There also exist elements that determine the PC's stats:

- *Experience Points*: Points that the character earns every time he defeats an enemy and/or performs some special task. Once the character has acquired enough experience points the character "levels up".
- *Level*: This shows the progress that a character has made, as he gains a level once he acquires enough experience points, and for every level that the character gains, the player receives a certain amount of Stat Points that can be used to improve the character's attributes.
- *Stat Points*: Points that the player earns every time he gains a level that can be used to increase the character's attributes.
- *Attack attribute*: This attribute determines the damage caused by the PC's attack. Each character starts with a set number, but this number can be improved as the character gains levels, by using the Stat Points.
- *Health attribute*: This attribute governs the number of health units the character has in its' health progress bar. Each of the characters starts with a set number in this attribute, but this number can be improved as the character gains levels, by using the Stat Points.
- *Power attribute*: This attribute governs the damage caused by the PC's special power. Each character starts with a set number, but this number can be improved as the character gains levels, by using the Stat Points.

Game Physics and Statistics

Physics in the game work as such:

- PCs move in a 2D side scrolling plane, either from left to right, or from right to left.
- PCs can jump either straight up and down or in a parabola going left or right.
- PCs can cross paths with enemy characters unless it is hit by an attack, which pushes it back.
- Minions are pushed back when struck, but recover quickly and in correlation to their overall speed stats.
- Boss characters should be mostly impervious to attacks, except when they establish a weakness and present an opportunity for the PC.
- When an obstacle is in front of the playable character it cannot proceed further in that direction.
- Gravity should be present.

Artificial Intelligence

Regular enemies:

- Enemies appear on screen either from the right or left side.
- Some enemies will be located in specific positions in a level.
- Enemies may pursue the PC on sight or wait until it is within a certain range.
- Enemies attack the PC as soon as they are in range. The frequency of their attacks will vary, but they will be able to attack repeatedly.
- Once the enemy's health bar is in low he starts using attacks with better range, or stepping away from the PC to the edge of the screen.

Mid-level boss:

- The mid-level boss will have a similar behavior to regular enemies, but will have better range, and more powerful attacks, and may also have a special attack.

Boss:

- The final boss is faced at the end of a level and has a variety of special attack moves with a lot of range, and that have varying speeds and strength.
- The Boss attacks with his long-range special power whenever the player is away from him.
- When the player comes closer to him the boss charges and attacks the player with his melee attack.
- The time the boss is most vulnerable is while he is using his special power.

Multiplayer

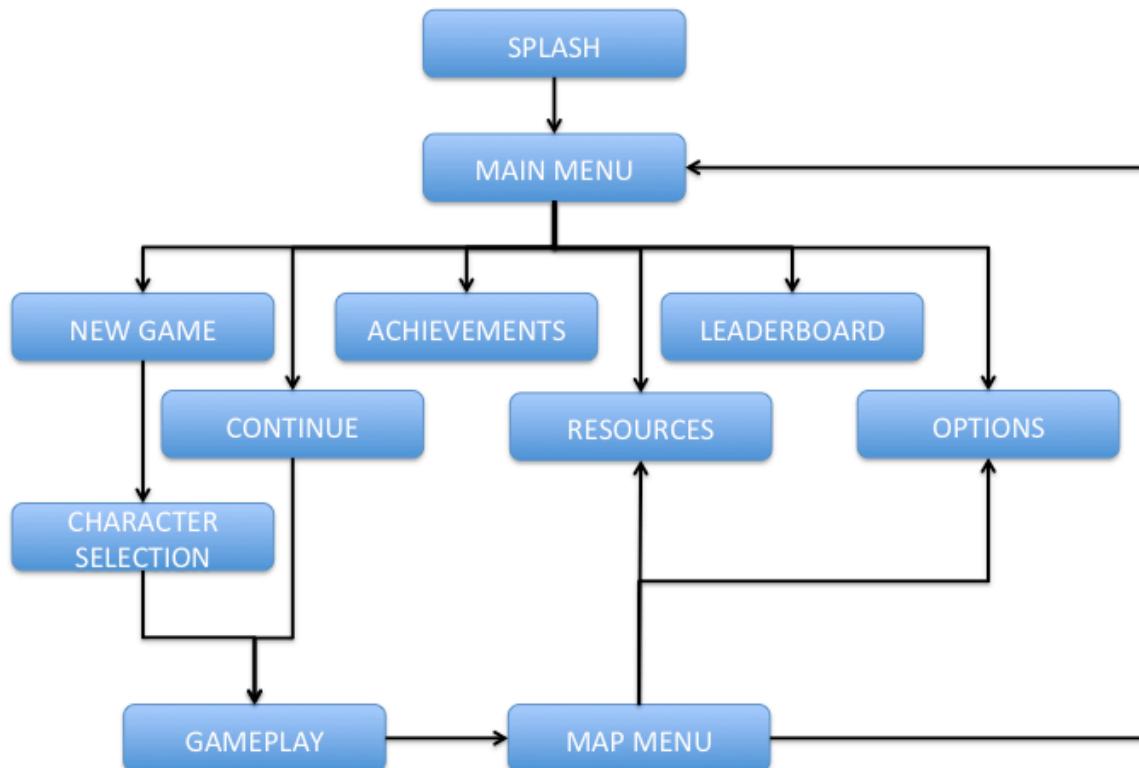
The game includes a leaderboard to display the best results obtained by the player and compare them to others.

If we were to incorporate a multiplayer versus battle system, it could work like this:

1. Choose the character you want to battle with.
2. Choose an opponent from the list of connected opponents.
3. The game goes to the versus battle screen that looks much like other fighting games such as Street Fighter. The two characters are located on opposite sides of the screens in a background related to the game, with health bars and special bars on top of the screens.
4. On the bottom of the screen you have a bar of available attack motions, they tell you what kind of attack the player can perform and what finger motion they must do on the iPad in order to use it.
5. Once one player depletes the others' health bar he is the winner.
6. Every time the player battles in a multiplayer battle his multiplayer stats change, for battles fought and battles won.

User Interface

Flowchart



Functional Requirements

Splash: This is the initial screen and introduces the Arcana logo.

Main Menu: Presents all of the possible choices for the user as well as a captivating graphic image that introduces the look and feel of the game and of the series.

New Game / Character Selection: Shows the character selection screen where the user can choose the character he wants to play with as well as brief information on the character's bio and stats.

Continue: Displays all of the user's games with each character and gives the option to continue from where he last left off.

Achievements: Gallery where the user can check all of the prizes he has achieved as well as the ones he can still unlock. This section will first present a gallery of thumbnail images, and you can see a larger version of the image by interacting with the thumbnail.

Resources: Page in which the user can check further information on different characters, this information is unlocked during gameplay.

Leaderboard: Ranks players in order according to total points acquired.

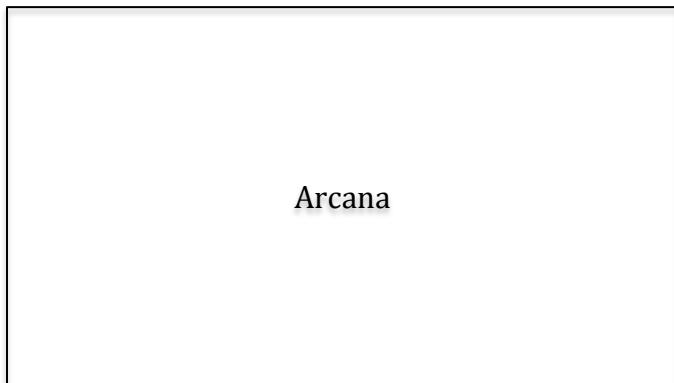
Options: Gives the user the ability to control some of the game's attributes, such as sound volume and music volume.

Map Menu: Can be accessed during gameplay and works as a sort of pause menu, from this page the user can also see which mission he is currently in or he can quit the game.

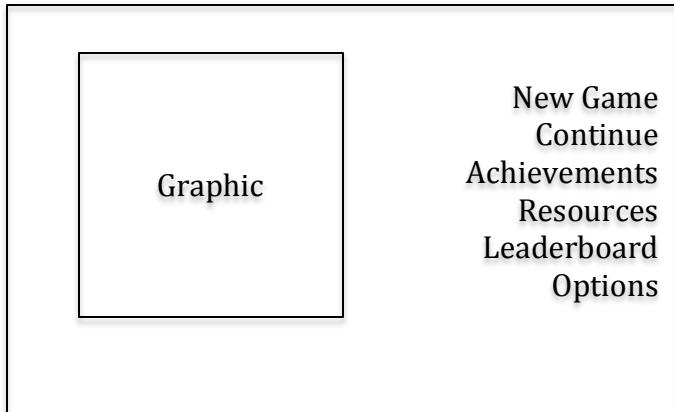
Gameplay: Where the game happens, has a fixed Heads Up Display that contains a character portrait, the total points, a health and a power bar as well as a level display. Also has the gamepad with the interaction buttons (Left, Right, Jump, Defend, Attack).

Mockups

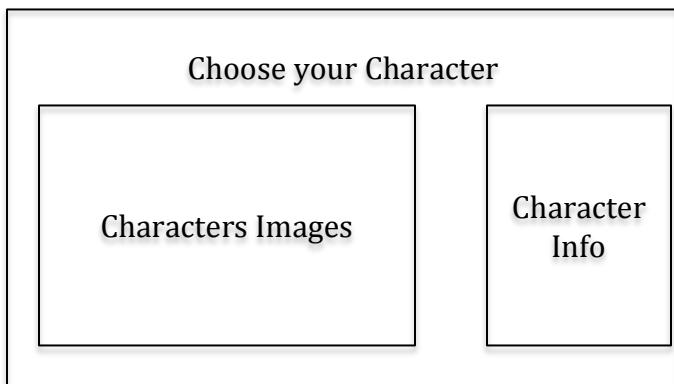
1. Splash Page



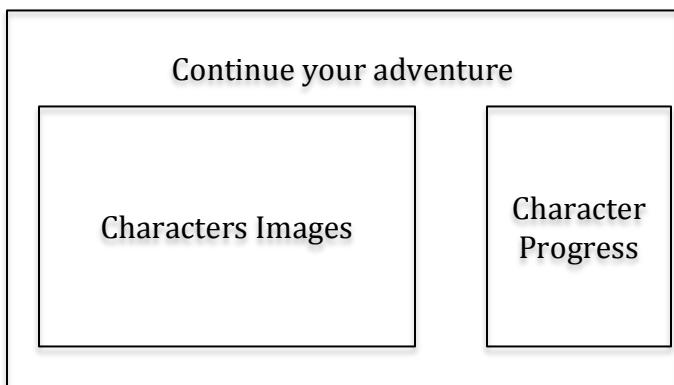
2. Main Menu



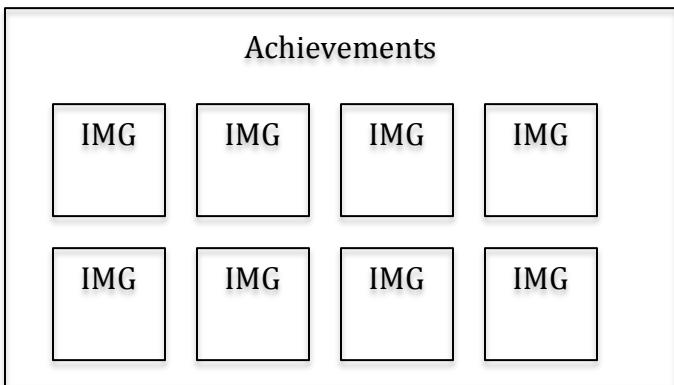
3. Character Selection



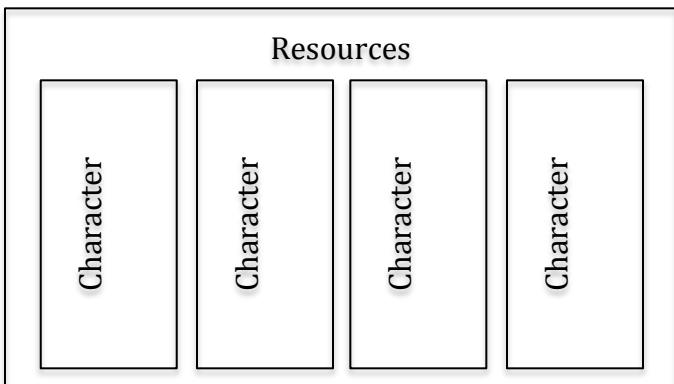
4. Continue



5. Achievements



6. Resources

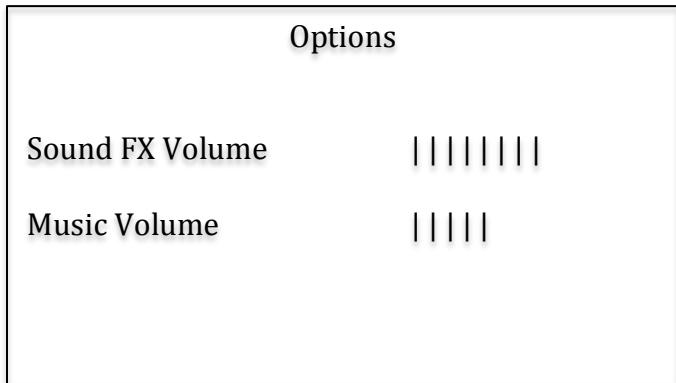


7. Leaderboard

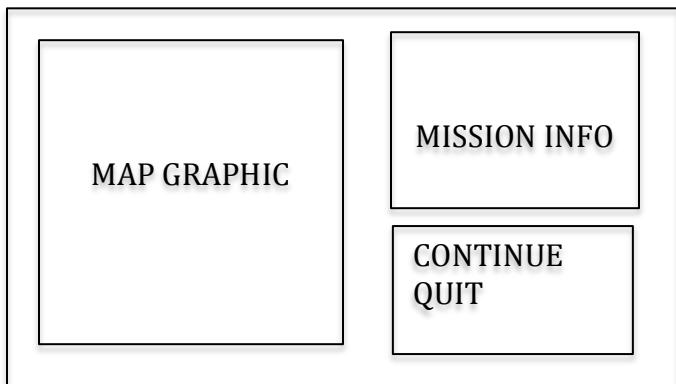
Rank	Player Name	Points

A rectangular container labeled "Leaderboard" at the top. Inside is a table with three columns: "Rank", "Player Name", and "Points". There are three rows of data, each consisting of three empty boxes.

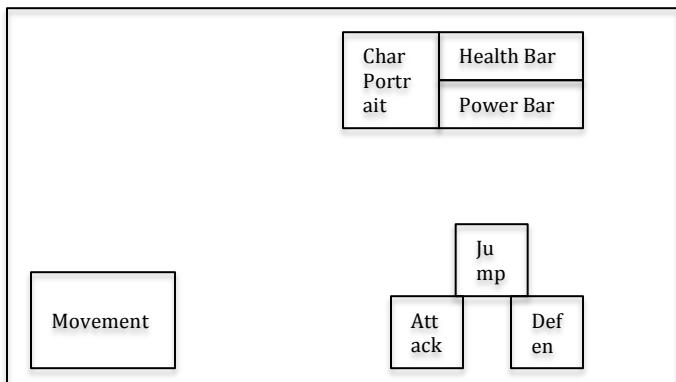
8. Options



9. Map Menu



10. Gameplay



GUI Objects

The GUI of the game is divided in a few sections. First we have the HUD (Heads Up Display). This is the information that appears on screen when the user is actually playing the game, the HUD contains:



HUD Image

- Character image – displays the character currently in use.
- Health bar – represents the total health the character has and the current amount of health.
- Power bar – shows the current level of the power bar,
- Special bar – it is a special button that becomes available only when the bar is completely filled, the user interacts by tapping on it in order to use the special power.
- Left and Right arrows – The user taps the arrows to move the character in that direction.
- Attack – It is an attack button and the user must tap it.
- Defend – The user taps this button to make the character defend.
- Enemy health bar – small health bar that appears on top of the enemy characters showing how much health they have.

There are other GUI objects located in the menus and are comprised of different buttons to navigate through the screens, in order to interact with them the user must simply tap the desired text button. Finally in the Option page we have a slider that the user can use to change the volume in the game, by dragging their finger in the slider the player changes the values.

Art and Video

Overall Goals

From the onset we have sought to achieve an art style that resembles a comic book. Our art style is slightly darker yet maintains a cartoony mood.

We have extensively used original artwork from the novel in the game interface in order to create a strong bond with the original Arcana material. By using new background art and 3D pose-able models for the characters we intended to achieve an art style that reflected the comics nature of the source material but still provided a fresh and unique experience for the player.



Main menu image

2D Art & Animation

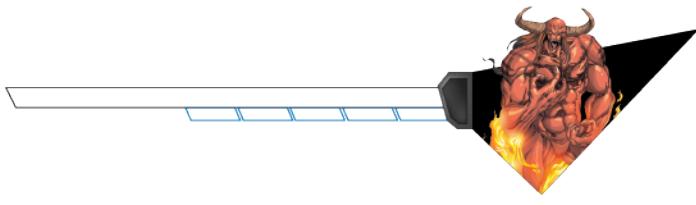
GUI



Achievement Unlocked Alert



Level Up Alert



Character Health and Power bar.

- Character portrait;
- Health bar;
- Power bar;
- Points system;
- Level display;
- Enemy health display;
- Boss health display;
- Power attack button;
- Left and Right arrows
- Attack button;
- Defend button;
- Jump button;
- Map menu button;
- Level Up alert
- Achievement Unlocked alert

Marketing and Packaging Art



Splash Image



iPad Icon

The packaging art for our game consists of having the iPad icon reminiscent of Arcana Comics, and the very first image when one opens the game (splash screen) has the Arcana logo on it.

- Splash screen;
- iPad icon.

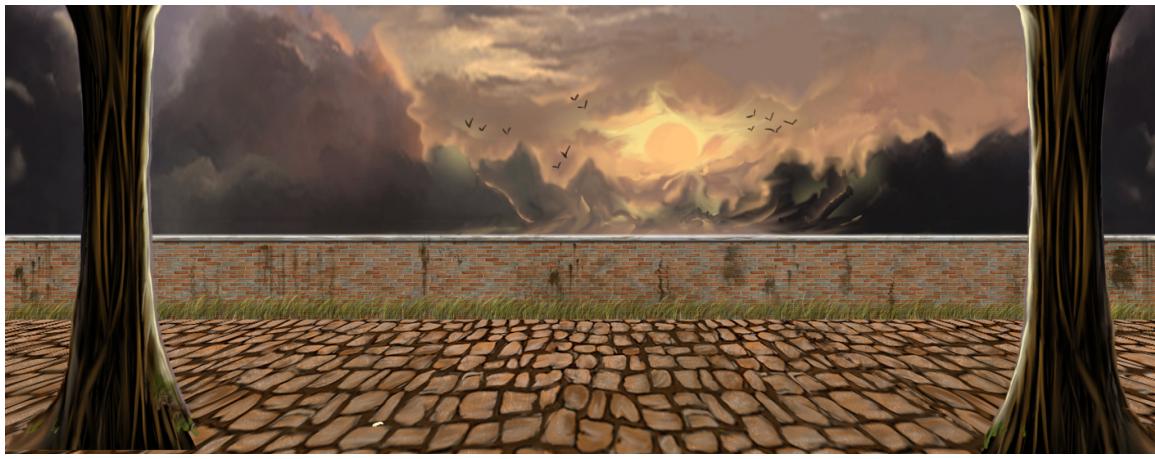
Terrain



Background art for destructed city.



Background art for regular city.

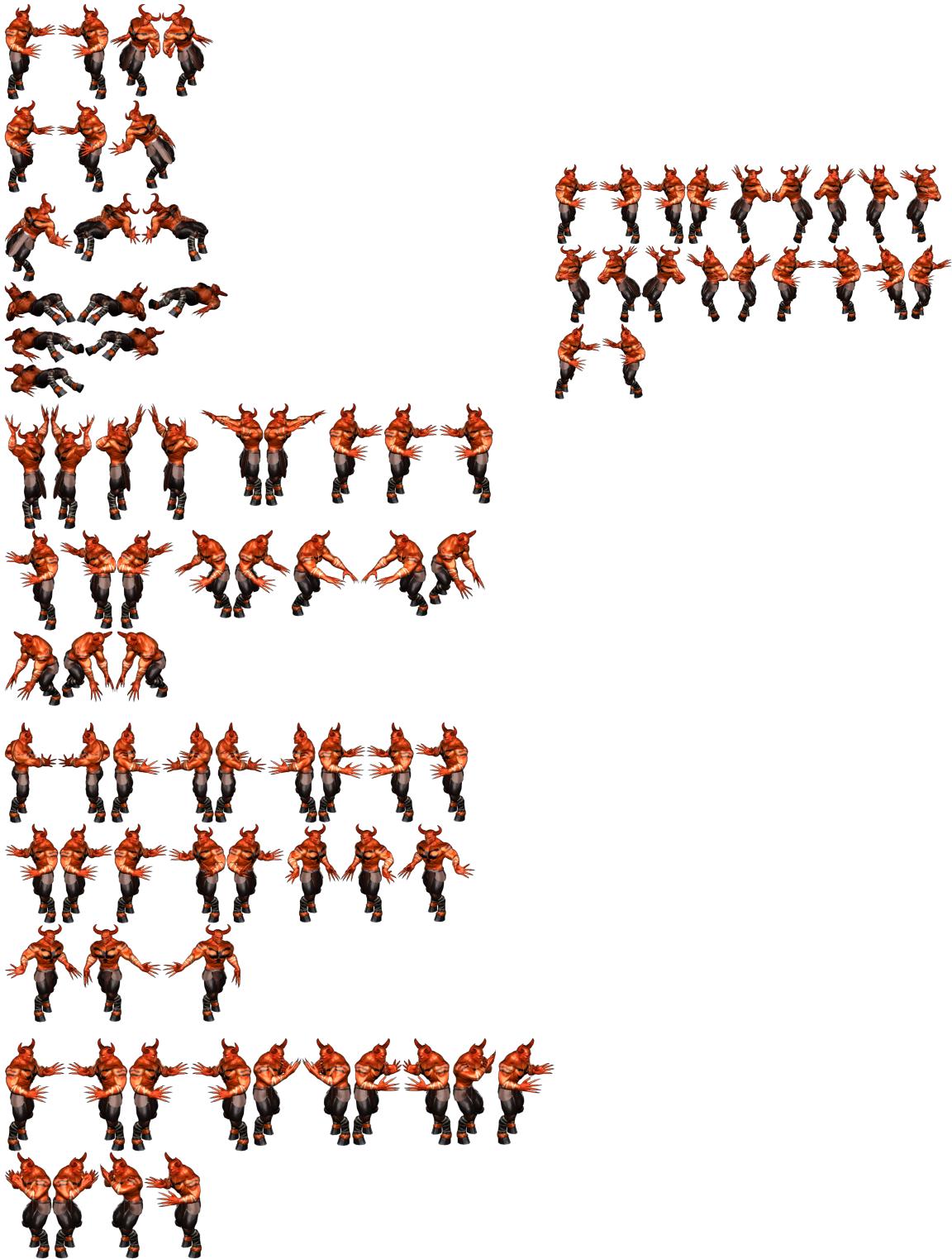


Background art for field.

- Background sky art;
 - Neutral;
 - Night;
 - Destruction;
- Background landscape
 - Neutral;
 - City;
 - Destrukted;
 - Industrial;
- Foreground path
 - Road;
 - Dirt;
 - Broken path;
- Foreground details
 - Trees;
 - Boulders;
 - Grass;
 - Wall;
 - Ironworks;
- Destructables;
 - Trees;
 - Debris;
 - Cars;
 - Walls;

Game Play Elements

- Health recovery item;
- Power filling item;
- Sprites;
 - Idle;
 - Defend;
 - Attack 1;
 - Attack 2;
 - Attack 3;
 - Power attack;
 - Walk;
 - Jump;
 - Damage;
 - Dying;





Kore sprite sheet

Special Effects

- Special power attacks:
 - a. Visual effects for certain special power attacks that character can perform.
- Tori
 - Defense
 - Attack 1, 2 and 3
 - Special Attack
- Kade
 - Special Attack
- Apollyon
 - Special Attack



3D Art & Animation

We received from the client 3D models with textures and animation rigging. The team was responsible for all character animation in the game.

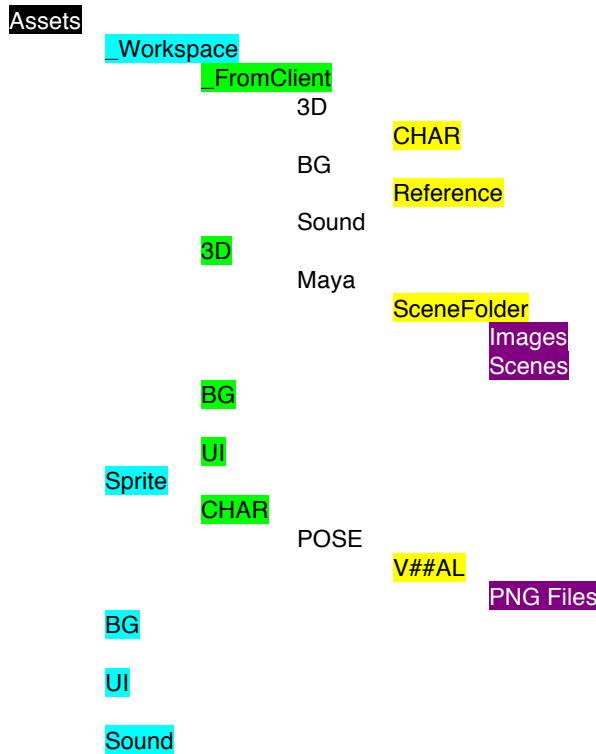
Cinematics

We received comic book style panels from the client to be used for the cinematics. We have approximately 12 different images that tell an entire story from beginning to end as the game levels progress.

Assets Pipeline

In order to organize of the assets we have for the game we created a pipeline for creating, saving and naming the files, as well as a file structure that determines where everything is put and how the files and folders are organized. Here is a breakdown on how the files and naming structures work.

Folder Structure:



The Assets Folder keeps all the game related files. Workspace folder keeps all the files from the client and the project files we are working on. Sprite, BG, UI and Sound folders keep that assets used in the game.

In the Maya scene folder, _CHAR_V## is the master file that has the model, textures, rig, lights and camera. The _CameraAndLight file contains all the cameras and lights for the model. CHAR is the master file that has everything referenced from _CHAR_V## files. The animators deals with the CHAR.mb file, when doing the poses they duplicate this file and rename it as CHAR_POSE_V##A.mb format.

When rendering we used the Camera04 option with 1k Square preset and PNG image format, later with flipped the images using Photoshop or compositing softwares.

We used a naming convention for all of our sprite creating that follows this method:

CHAR_POSE_V##AL_###.png

CHAR: Using 4 digits it stands for the name of the character: KADE, KORE, TORI, BS01, EM01, etc. Although had only one boss character this allows for further expansions in the future.

POSE: Using 4 digits it stands for the name of the pose:

IDLE: Idle position
ATK1: Attack number 1
ATK2: Attack number 2
ATK3: Attack number 3
POWR: Power
WALK: Walking
JUMP: Jumping
DEFD: Defending
DAMG: Taking damage
DEAD: Dying

V##A: V stands for version. ## is an internal version number: 01, 02..99. A is client version: A, B, C...Z.

L: sprite direction: L (left) or R (right)

###: Frame number

Example: KADE_ATK2_V02AL_005.PNG EM01_DEAD_V03CR_033.PNG

Sound and Music

Overall Goals

The goal of the music within the game is to set the mood of the world as well as indicate a call to action for the player during gameplay events. There are two clearly different music soundtracks. A basic gameplay music loop that's not too fast, that is very minimal and subtly layered with synthesizer textures and digital samples (Limbo, Quake); and a 'Call to arms' song loop that should be faster, and could have an industrial rock feel (Quake, TMNT: the Arcade Game, Descent). The music has been normalized, mixed, and compressed as MP3 with a sampling rate that we should aspire to be 192kbps, and should be no lower than 128kbps, and should consist of loops that can be cycled repeatedly.

Sound FX

The GUI interactions are indicated with sounds for Menu button, Stats button, Reward button, and Play Game button.

The game uses special effects to indicate powerful actions and to establish the game's ambiance, such as distant explosions and screams, but no background city sounds. The sound FX within the game must enhance the gameplay and clearly identify each character. Even though the actions that PCs can perform are the same, their specific moves are different, and require different sounds.

What takes place in the game is indicated with sound effects, such as the appearance of the sidekick characters or of the quest giver during gameplay has an accompanying sound. For instance, a tinkle indicates the presence of Molli when playing Kore, a radio crackle indicates Alex Rand's communications coming in when playing Tori, and an ethereal sound indicates the telepathic communications of Ezra, received when playing Kade, and an impactful sound could serve to highlight Dr. Ishmael's messages. When a PC recovers health it is also indicated with a sound effect.

Kade's attacks are made with his sword, so the sounds required for his moves is a sword swing combined with a Kade shout, several of which can be chained together and end in a deeper swoosh and a battle cry to go along with hit combos and their ending. His special attack also has a particular sound also related to the sound of his swinging blade and the effort of the moves he performs.

Kore's attacks are made with his body, so the sounds required for his moves are powerful punching and kicking sounds, and the special attack has a particular sound accompanied by a powerful battle cry from Kore. Kore's footsteps are accentuated with a thump to impress the feeling of weight, size, and power.

Tori's attacks are made with her body, but take place without her actually having to touch any enemies through psychic energy and psychosomatic power. Therefore her attacks sounds are based on powerful punch and kicks sounds but have a thundering or interesting sound that precedes them. Her special attack also has a loud and big thundering sound.

Sound FXs also indicate the characters' attacks, when they receive damage, and when they die. There are sounds for when a character picks up an item, receives an achievement, finds a hidden area, or rescues a hostage.

The basic enemy characters are able to cycle randomly through a few grunt sounds that will be heard when they are struck the first time in a combo, and have a particular sound to indicate when they die. They also have a basic grunt sound for when they are not engaged in combat, and another grunt for when launching an attack.

The boss character has very deep and reverberating sounds to accompany his attacks. The character is also identified with distinctive evil grunts and such that sound muffled since the character is inside of armor, as well as a demonic howl for when the character loses a battle. We also have a number of different sound effects, such as explosions and fire that are used in conjunction with other sound effects in a boss battle scenario.

Music and sound FX assets

Minimum of 2 song loops:

- A slow ambient song loop for regular gameplay
- A faster rocking industrial song loop for facing off bosses or many enemies

Ambient sounds:

- Wind
- Distant explosions
- Distant shouting
- Distant crying

Kade:

- Shout 1
- Shout 2
- Pained scream
- Death scream
- Sword attack 1
- Sword attack 2
- Sword attack 3
- More powerful sword attack
- Special fast swinging sword attack
- Ezra telepathic ethereal sound

Kore:

- Growl 1
- Growl 2
- Pained roar
- Death roar
- Loud footsteps
- Punch attack 1
- Punch attack 2
- Punch attack 3
- More powerful punch attack
- Special super powerful striking attack
- Molli tinkle

Tori:

- Shout 1
- Shout 2
- Pained scream
- Death scream
- Psychic attack 1
- Psychic attack 2
- Psychic attack 3
- More powerful psychic attack
- Special thundering psychic attack
- Alex Rand intercom crackle

Apollyon:

- Demonic laughter
- Demonic grunt 1
- Demonic grunt 2
- Powerful strike 1
- Powerful strike 2
- Sweeping strike 1
- Sweeping strike 2
- Demonic howl
- Explosion
- Fire

Minion:

- Attack grunt 1
- Attack grunt 2
- Pained grunt
- Death grunt

GUI/HUD:

- Alert
- Power up
- Achievement
- Regular menu button
- Stats increase button
- Reward button
- Play game button

Story

Horrors from different worlds have assembled to bring about a global threat known only as "The End". A Philosopher has recognized the signs in time to gather his own forces for Good. Only together can they avert disaster on a biblical scale. Every last one of them... is Intrinsic.

Player Characters

Tori Newton: This young woman is one half of an Intrinsic, which means she's a human born with uncanny abilities. She has the power of Telekinesis, which has only recently begun to manifest, and she is shouldered with the responsibility of joining a team of other super powerful beings to avoid an oncoming apocalypse. She's the book's main character.

Kade: He is child of the black sun, which means he can't feel anything, he is mortal but doesn't age, and has been battling a tortured life as a demon hunter since the dark ages, surviving thanks to his ability in combat, strength, endurance, and powerful healing ability. Kade can't control his anger, and goes into fits of rage.

Kore: The Mantikore is an immensely powerful creature from a world called Abaddon, a place where fantastic creatures of all kinds live together in a mirror of contemporary society. Formerly irresponsible human Alex Crane accidentally became bonded to this creature by the power of uncreation, and they are now Kore.

Other potential PCs: Dr. Ishmael Stone, Candice Crow, Jazz Manral.

Secondary Characters

Dr. Ishmael Stone: Stone is the leader of the Intrinsic. He foresees the coming danger and assembles the team. He's also the most powerful of the Philosophers, a network of supernatural guardians sworn to shelter the world from the infernal orders. He has many supernatural abilities and devices and has been alive for over a century. He will give the quests to all PCs.

Molli: Molli is Kore's sprite (fairy) friend. She is a sassy and brave companion to him and delivers great lines to go along with Kore's fighting.

Alex Rand: Alex is Tori's partner, and both of them are required to form an Intrinsic. He's a security guard with firearm skills and a strong protective instinct.

Ezra: Ezra is Kade's long lost love. She was a child of the black sun who hunted demons with him for centuries but was then taken by Abaddon.

Civilians: These are anonymous civilians with generic features who are running from or being attacked by the dark horde.

Enemy Characters

Abaddon aka Apollyon aka Kamric: He is essentially a very powerful ancient demon who has been banished to the Abyss by Kade before, and has now assembled a horde of demons, death gods, and possessed minions, with the intent of tearing down the wall between the abyss and reality. The different names of this creature serve to distinguish between its different forms: Abaddon has a human like appearance, Kamric is completely encased in a suit of armor, and within it is Apollyon, which is his original demonic form, displaying huge wings, horns, and hellfire.

Monster Minions: These are possessed warriors who follow every command from their demonic overlords. They are empty shells of former humans, and they crumble to black dust when destroyed. There are minions of different types, with different strengths, who may be recognized by their markings or their colors. They

have fast but weak attacks and their strength lies in numbers, which is why they attack in packs. They are simplistic creatures with no real will of their own other than the will to destroy everything, which consumes them, and the command to defend their evil masters. A special kind of minion can be developed to function as a mid-level boss.

Cizin: This is an ancient underworld deity of death and earthquakes from Mayan mythology, and lives in the subterranean purgatory called Land of the Dead. He has a death collar, which features eyes dangling by their nerve cords among other fascinating things, and he is also referred to as “the Stinking one”.

Other possible enemy characters: Lord Chamiabac (Bone Staff) and Lord Chamiaholom

Story theme

One against all, the obstacles faced by one struggling to overcome a huge opposition, and the power and ability they must demonstrate.

Visual theme

A shattered version of the world as we know it. Where evidence of powerful evil forces and an ongoing battle is everywhere.

Story Outline

- The wall between Earth and the Abyss has been torn down and the Apocalypse is upon us as soon as we begin the level.
- The Intrinsic arrive on the scene and Dr. Ishmael Stone assigns the PC an initial quest.
- Each stage will consist of four separate quest based levels.
- The PC will go through the streets of a collapsing city where the population is under attack by Abaddon's Monster Minions, who will be setting obstacles and attacking the PC to stop him from reaching the objective.
- A sidekick, who will not be present in the scene, but will maintain constant communication, will accompany the PC.
- While the PC is fighting, there is an implicit understanding that other members of the team are fighting along in other parts of the city, and we may encounter evidence of their combat.
- The PC may explore some hidden rooms where he will find special items to pick up, for unlocking achievements and content.
- The PC is trying to stop Abaddon but he can't defeat him because he's too powerful. When this happens, we may become aware that this is all currently happening in Dr. Stone's dream, and the adventure is only starting.

Tori

- Dr. Stone explains to Tori that she's more powerful than she ever imagined, and that he can't tell her what to do but only to fight for her life, and that somehow she needs to be there to have a chance of defeating Apollyon. He says he can't say anymore, but that when the time comes she'll know what to do. She's very scared.
- Alex contacts Tori through his security guard radio communication equipment, and tells her that he'll be giving her advice and encouragement along the way. Along the way Tori will be able to share her confused thoughts and emotions with Alex, as they are both caught in the whirlwind of what's going on.

Kade

- Dr. Stone simply points the way for Kade to go after his ancient nemesis, and Kade runs after him lusting for revenge.
- Ezra, who can only reach him since the wall between Earth and the Abyss has been torn down, contacts Kade telepathically. She doesn't immediately explain what has happened to her since she last saw Kade, she only encourages him to stop Apollyon. As Kade advances, she gives him advice, and also tells him about where she is, why Apollyon must be stopped, and how/if they may once again be reunited.

Kore

- Dr. Stone has recruited Kore because of his legendary power, and he is a major part of the offensive capabilities of the Intrinsic. Kore goes in with the mission of causing the most damage on the horde.
- Molli goes into battle along with Kore, and punctuates the game events with witty one liners and hints for Kore delivered with spunk.

Level Requirements

Level Diagram

LEVEL 1 – LOS ANGELES		
Stage A – Field day	Stage B – City night	Stage C – Destruction day
Medium numbers lvl 1 enemies	High number of lvl1 enemies	Boss Battle + lvl 1 enemies
LEVEL 2 – NEW YORK CITY		
Stage A – City day	Stage B – City night	Stage C – Destruction night
High number of lvl 1 enemies + medium number of lvl 2 enemies	Low number of lvl 1 enemies + high number of lvl 2 enemies	Boss Battle 2 + lvl 2 enemies
LEVEL 3 - VANCOUVER		
Stage A – Field night	Stage B – Destruction day	Stage C – Destruction night
High number of lvl 2 enemies + low number of lvl 3 enemies	High number of lvl 3 enemies + medium number of lvl 2 enemies	Final Boss Battle + lvl 3 enemies

Asset Revelation Schedule

The game has two different kinds of items, the Health Recovery Item, that recovers the player's health, and the Power Recovery Item that fills the player's power bar.

ITEM	CHANGE OF DROP	RECOVERS
HEALTH RECOVERY ITEM	10% chance of drop	25% of total
POWER RECOVERY ITEM	25% chance of drop	25% of total

Level Design Seeds

First level is Los Angeles and with three different stages. This level should be an introduction to the game for the player and initially have an easy degree of difficulty and allows the player to gain 1 or 2 levels and receive some achievements. At the last stage of this level we should have a first boss battle that introduces to the player the concept of strategies necessary to defeat the boss.

Second level is in New York City, which also has three different stages. The New York level should ramp up the difficulty for the player and introduce the second kind of enemies. At the third stage we will again have a boss battle except this time it is more challenging and complicated.

The third and final level happens in Vancouver and again has three different stages to it. By now the player should be at a fairly high level and with established playing skills so the difficulty should be at maximum. It is now harder and slower to unlock level progressions and achievements. At the final stage we have the final boss battle that should be the most difficult battle in the game.

Technical Specifications

Game Mechanics

Game engine

We have used a game engine to develop our game called Unity3D developed by Unity Technologies. A game engine is a system designed to develop games for various platforms like consoles, computers and handheld devices like smartphones.

Platform and OS

Unity gives the possibility to export the game to various platforms, these includes: iOS, Mac Standalone, Windows Standalone, Web, Nintendo Wii, Xbox 360 PS3 and Android.

This game specifically is developed for the iPad, which are a series of tablets developed by Apple Inc., which means that the game is developed for iOS5. The game can easily be ported to android and iPhone with a few adjustments.

External Code

As mentioned earlier we used Unity3D Game Engine to develop the game. Besides the game engine we used an editor extension called ex2D. This is a 2D sprite solution, which provides the functionality of creating sprite atlases from a number of sprites. After the atlas has been created the sprites can be combined as an animation to be used in the game. We used the ex2D to handle all our animations this includes animations for the playable characters, enemies, the boss and special effects.

Code Objects

Player Object:

The player object is named after the current character being played and can be identified by the “Player” tag. The object has the Player.cs script on it, which is used to act according to the input from the user, this is done by listening to events from the ButtonManage.cs. This object is also connected to the ex2D editor extension so that the player can be animated according to his actions.

Enemy Objects:

The enemy objects come in 3 different variations and is identified by the tag “Enemy”. Its type is decided by a Boolean value in the Enemy.cs script that is attached to the enemy object. The difference between the 3 types of enemies is the amount of experience they provide the user with after getting killed, the amount of health they have and the amount of damage they deal. This means that enemyOne is the weakest and enemyThree is the strongest. The Enemy.cs script also provides the enemy with an AI, which will be explained later.

Boss Object:

This object is identified by the tag “Boss” and is named Apollyon. The boss has a very simple AI just like the enemy.

Camera Object:

The camera Object is an orthographic camera that exists in every scene of the game. This object has attached 2 scripts to it. The first is called rayCastScript.cs this script manages all the raycasts made when the user is touching the screen. The scripts purpose is to let the ButtonManager.cs script, which is also attached to the camera object, know if the user is pressing a button or not.

Button Objects:

The game has 6 in game button objects, these are: LeftButton, RightButton, Attack, Defend, Power, IngameMenu. All these objects are linked to the buttonManager script on the camera object.

Power Up Objects:

The game has 2 different types of Power Up objects, one recovers the player's lost health, and the other fills up the player's special power bar that allows him to use the special power attack. Both Power Ups have a 25% chance to spawn each time the player kills an enemy. The Power Ups have the PowerUp.cs script attached to them where a Boolean value decides its type.

GameManager:

This object is named GameManage and has the GameManager.cs script attached to it. The Object manages things such as enemy spawning and the points the player is able to spend on stats after leveling up. It handles the enemies by making a list with all the enemies in the current scene. Then the Trigger objects can ask the game manager for an enemy on the list when the trigger needs to spawn one.

Trigger Objects:

There are 3 different kinds of these object, they are StartTrigger, SpawnTrigger and WinTrigger. The start trigger makes sure that the player can't leave the defined level when he is colliding with it; the WinTrigger ends the level when the player collides with it. The spawn trigger spawns an enemy the first time the player collides with it. The trigger will ask the GameManager for an enemy with the stats defined on the specific Trigger the player is colliding with, if the enemy is available it will spawn it.

Control Loop

Unity has some predefined Functions that are run on different times.

When a game is executed the first function that will be run is called Awake() this is usually used to set the standard values of variables, after this function is run it will call the Start() function it can be used as Awake, but it's called later. When start has been run it will call the Update() function every frame. There is also an OnGUI() function that is used to make calls to Unitys own GUI system. This function is called every frame just like Update(). Here is a breakdown of how the enemy an the player uses these functions:

Player.cs

```
Awake()
{
    The player doesn't do anything in this function
}

Start ()
{
    //Creates a ref to the GameManager object and get the GameManager script component
    gm = GameObject.Find("GameManager").GetComponent<GameManager>();

    //Stores the current char name to be used later by the GameManager
    charName = gameObject.name;

    //Sets the default stats according to the current char
    SetDefaultStats();

    //Caches the players transform
    myTransform = transform;

    //Ref to the exSpriteAnimation
    anim = GetComponent<exSpriteAnimation>();

    //Ref to the raycast object taken from the MainCamera
    rayCast =
    GameObject.FindGameObjectWithTag("MainCamera").GetComponent<RayCastScript>();
```

```

//Makes a reference to the boss
apolluon = GameObject.Find("Apolluon").GetComponent<Apolluon>();

//Sets the default facing direction to right
facingDirection = Direction.right;

//Set the last action to idle right as default
lastAct = LastAction.IdleRight;

//Sets the current animation type as idle as default
currentAnimationType = AnimationType.idle;

//Sets the default attack to 1
attackNumber = 1;

//Makes sure that the player is alive when starting a new level
allive = true;

//Fills the array to calculate the players stats
FillArrays();

//Calculated the players stats according to his current points
CalculateStats();

//Creates a start position for the player
startPos = myTransform.localPosition;

//Sets the players current health to his max health, to fill his health up
health = maxHealth;

//Makes sure that the player has 0 when he starts a new level
currentAbility = 0;

}

void OnGUI()
{
//If player is dead it will show a GUI box and give him the possibility to restart
the level
if (!allive)
{
    if (GUI.Button(new Rect(Screen.width / 2 - 25, Screen.height / 2 - 10,
500, 200), "Retry"))
    {
        Application.LoadLevel("lvl1"+charName);
        gm.ClearList();
    }
}
}

Enemy.cs
Start ()
{
    //Gets the player script component on the PlayerGameObject
player = GameObject.FindGameObjectWithTag("Player").GetComponent<Player>();

//Gets the GameManager script on the GameManagerObject
gm = GameObject.Find("GameManager").GetComponent<GameManager>();

```

```

//Gets the exSpriteAnimation component
anim = GetComponent<exSpriteAnimation>();

//Sets the default value for enemySpeed
enemySpeed = 45;

//Makes sure that the enemy is alive
alive = true;

//Sets the enemies inrange to false
inRange = false;

//Makes sure that the enemy isn't on the list
isOnList = false;

//Makes sure that the enemy isn't in action when spawning
inAction = false;

//Stores the enemies start position
startPos = transform.localPosition;

//Defines the stats for each enemy type
if (enemyOne)
{
    exp = 2;
    startHealth = 15;
    enemyAttackPower = 3;
}
if (enemyTwo)
{
    exp = 5;
    startHealth = 35;
    enemyAttackPower = 5;
}
if (enemyThree)
{
    exp = 10;
    startHealth = 50;
    enemyAttackPower = 8;
}

//set the enemy's health
health = startHealth;

//Adds all enemies to
FillList();

//Caches the enemy's transform
myTransform = transform;
Update ()
{
//Calculates the amount to move per frame
amountToMove = enemySpeed * Time.deltaTime;
amountToFollow = mimicSpeed * Time.deltaTime;

if (inAction)
{

```

```

//Makes the enemy follow the player
FollowPlayer();

//Animates the enemy
AnimateEnemy();

//Makes the enemy able to attack the player
Attack();

//Locates the enemy
LocateEnemy();

//Checks if the enemy dies or not
DidIDie();

//Makes the enemy follow the background when standing still
FollowBackground();

//Despawns the enemy after death
Despawn();
}

}

```

Game Object Data

I'm documenting all the scripts which will be transformed into a doxygen document, but I can't finish it before the game is done.

Data Flow

We are using Unity PlayerPrefs to store and access our game data between game sessions. The game data we store includes the player stats such as, level, health, attack power and ability power. We also store which achievements the player has earned.

On Mac OS X PlayerPrefs are stored in ~/Library/Preferences folder, in a file named unity.[company name].[product name].plist, where company and product names are the names set up in Project Settings. The same .plist file is used for both Projects run in the Editor and standalone players.

Here is an example of how the players health is stored and loaded:

```

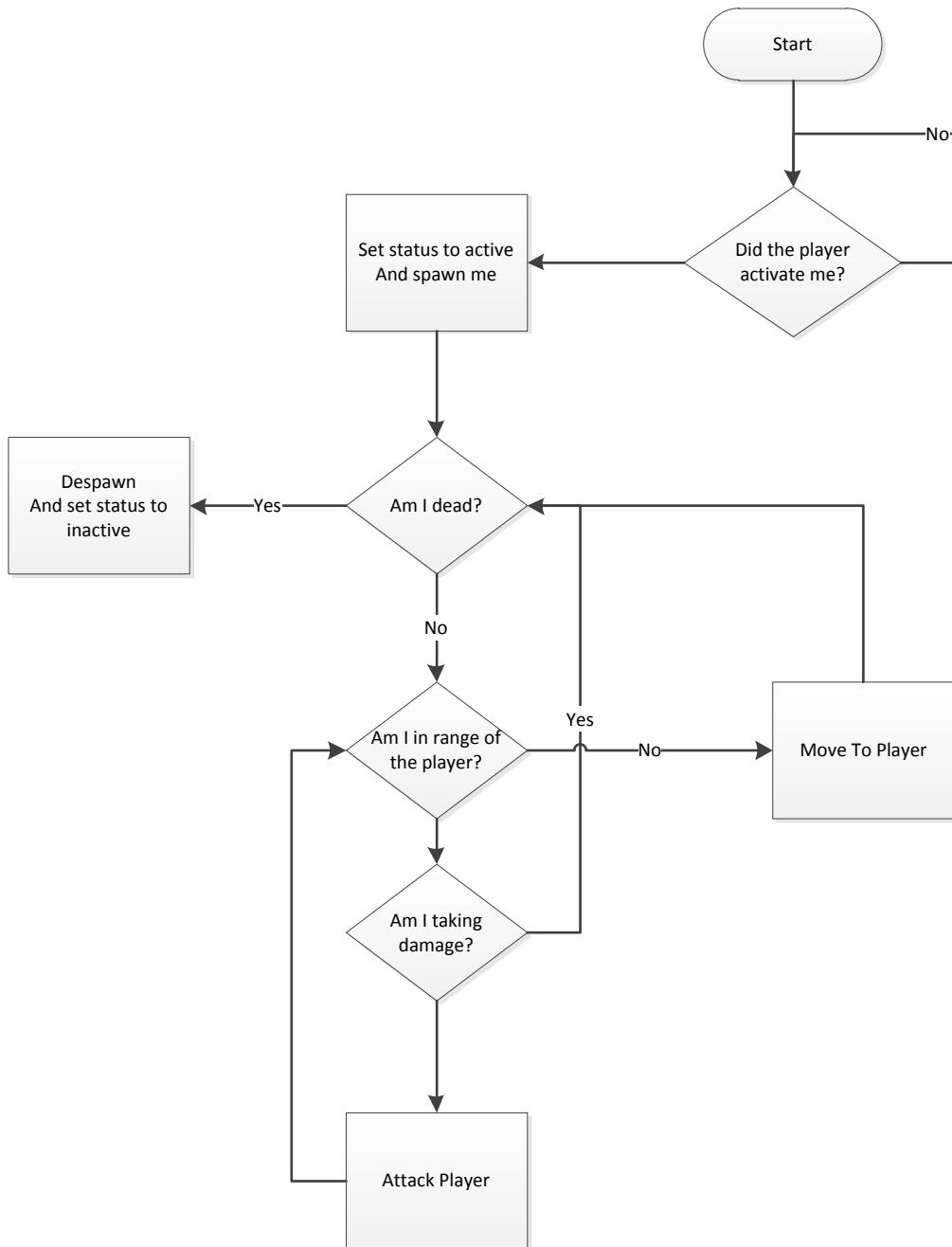
Save health
PlayerPrefs.SetInt("Kade Health", healthPoints);

Store health
healthPoints = PlayerPrefs.GetInt("Kade Health", 3);

```

Artificial Intelligence

Here is a flowchart of the enemy AI to explain how it works. If the player runs through a trigger the trigger will ask the game manager for an enemy. If there is an enemy that fulfills the triggers request the enemy will spawn. The enemy will then run through the cycle seen below until it's killed.



Production Schedule

Scope

Our project Scope includes the following:

1. Designing and development of an iPad Game;
2. Game is to have a Brawler style of gameplay;
3. Minimum of three playable characters;
4. One type of regular enemy character;
5. One boss type enemy character;
6. Three slightly different background images that can be expanded in several possible game levels;
7. Prize giving mechanic with several loaded prizes;

Scheduling

The team has divided the production schedule into the following major points:

- Pre-Production / Discovery – Jan 5th, 2012 to Jan 27th, 2012
- Concurrent Development of Art and Code Assets – Jan 30th, 2012 to Mar 23rd, 2012
- Integration and Design Balancing – March 23rd, 2012 to April 6th, 2012
- Alpha – March 23rd, 2012
- QA – March 30th, 2012
- Beta – April 6th, 2012
- Deployment and Presentation – April 13th, 2012

Dependencies

The schedule and scope outlined in the above sections can only be followed and fulfilled when the necessary assets are delivered by the client on schedule, below is a initial list of the necessary assets and times that we will be dependent upon:

1. 3D Character Models
 - a. PC 1 – Feb 13th, 2012
 - b. PC 2 – Feb 20th, 2012
 - c. PC 3 – Mar 12th, 2012
 - d. Enemy – Feb 27th, 2012
 - e. Boss – Mar 5th, 2012
2. Art Assets
 - a. Character Images – Feb 13th, 2012
 - b. Cut scenes Images – Mar 22nd, 2012
 - c. Gameplay art – Mar 12th, 2012
 - d. Prizes – Mar 05th, 2012
3. Sound Assets
 - a. Complete list of sounds – Mar 19th, 2012

Cost Estimate

Once we have completed and delivered the original project scope the client will be able to change, update and/or insert any new features. The following chart is an estimation of the tasks and time required for future modifications.

TASK	TIME (in man-hours)
1 Background	35 hours
Modeling, texturing and rigging of 1 new character	(Delivered by the client, must be calculated)
1 full set of animation sprites	60 hours
Programming for 1 character and 1 level	25 hours

The estimates above consider only the work necessary to implement 1 new playable character, 1 new background, 1 game level, and program the necessary codes for these elements to function in the game.