Work Breakdown

Multimedia

[Overview](#h.c0b906q97cxv)

[Team](#h.31d0yk8p4uhc)

[Tasks](#h.7u6xq1xbtu5s)

[[> Design <]](#h.7bckfjo6vs1h)

[[ - Formatting - ]](#h.ulz8f6kwwuxj)

[[ - Conceptual - ]](#h.o9jqdaw5n9e8)

[[ - Flow - ]](#h.gk3n33vlceb3)

[[ - Pipeline - ]](#h.zg67ehrrm4vw)

[[ - Prefabrication - ]](#h.31g8xjf9nr9g)

[[ - Asset requirements - ]](#h.l2mtqt988wp7)

[[> Programming <]](#h.7xlsainhewfn)

[[- Graphics -]](#h.fz3042yg3ety)

[Scene graph / graphical hierarchy using Game Objects [GO]](#h.72iryguy3lt6)

[9-patch image support](#h.nvamzmsmbsfv)

[Parallax scrolling](#h.q8gjh53ul5i9)

[[- Graphical Engines -]](#h.ujs5zm7ecnf4)

[AppWindow - Scenes relationship](#h.ny7qqeq3expw)

[Particle system](#h.aujfluprtyao)

[TileMap engine](#h.2m68wpw9w3c)

[[- Rendering -]](#h.ccvtgx47e54r)

[Sprite batching](#h.53sy13ugf5y7)

[Sprite animation rendering](#h.c9q3ep12j5or)

[GLSL shading effects](#h.d37ijndcfchn)

[[- Resource Management (Internal & External) -]](#h.3zo5g0ggp8sq)

[Asset resource management](#h.txuwfbihx1xs)

[Asset storage](#h.1f5jo75juu2o)

[[- Audio -]](#h.ko4vtxvhp3ba)

[Sound Manager](#h.sfahf9su1x9b)

[[ > Sound < ]](#h.270wechb3aht)

[[ - Player - ]](#h.ekbyh8qufdki)

[Player Effects](#h.f19ueysz2rm8)

[[ - Weapons - ]](#h.884b8a7khuuu)

[[ - Monsters - ]](#h.1o6pc36pv4su)

[Bosses](#h.ym4k06gf59bw)

[Mini-Bosses](#h.yqks161rvkwq)

[Minions](#h.pwepl9u0r5o3)

[[- Environment - ]](#h.ju4n9h8y7zd6)

[[ - Deities - ]](#h.gmbfol7n7qx)

[[ - Interactables & Objects - ]](#h.aecttk3o0gei)

[[ - Particles - ]](#h.des9ieqqgmtj)

[[ - Miscellaneous - ]](#h.q3xdmxeoky2d)

[[ > Art < ]](#h.mpxrn6whyznk)

[[ - Player - ]](#h.vigatlgd0suy)

[Player Effects](#h.mn1vxq6ip1xz)

[[ - Weapons - ]](#h.92txc2e7r996)

[[ - Monsters - ]](#h.ukxj8hhsgbdm)

[Bosses](#h.uz79t3kcksi9)

[Mini-Bosses](#h.rtk1fxpcwd8n)

[Minions](#h.3ii98hgzyyw2)

[[- Environment - ]](#h.dv5ids357x29)

[[ - Deities - ]](#h.vogd8iop2t3u)

[[ - Interactables & Objects - ]](#h.1sykfbbsjyyg)

[[ - Particles - ]](#h.lur0yjyo0ema)

[[ - Miscellaneous - ]](#h.311nuehs7l93)

# 

# Overview

This document identifies and explains all tasks that the Multimedia team will be expected to complete for the project. These tasks are broken into four categories; design, sound, programming, and art. The bulk of the work will come from asset creation and integrating the assets within the game.

# Team

One team will be working on multimedia throughout the project in order to have consistency and harmony between components. This team will be broken down as follows:

***Design -***  Team

***Programming -*** Melvin Loho

***Art -*** Rhea Lauzon (Team Leader)

***Sound -*** Lewis Scott

# Tasks

The list of tasks to be performed by the multimedia have been broken down into sub-sections to identify the area of development they belong to. Any tasks that are unclear or more vague have been given a short description for clarification.

Tasks marked with a **(?)** signify tasks or features that are uncertain or potentially developed by other teams.

## [> Design <]

This section contains a list of general and miscellaneous tasks that will need to be completed by the multimedia team. Most of these tasks involve drafting various aspects needed by the multimedia team.

### [ - Formatting - ]

* Decide on sound format
* Decide on graphics format
* Decide on art and sound style(meet with design team)

### [ - Conceptual - ]

* Concept art and sound
* Wireframe menu and lobby guis
* Wireframe game gui

### [ - Flow - ]

* Design rendering classes
* Design sound manager
* Design resource manager
* Design abstract particle system class
* Define asset workflow

### [ - Pipeline - ]

* Scene hierarchy
* Viewport system
* GO Inheritance and components

### [ - Prefabrication - ]

* Particle systems
* Shaders

### [ - Asset requirements - ]

* List of monsters
* List of characters / classes
* List of tiles

List of weapons / spells / abilities

## [> Programming <]

This section will list and explain all of the ideas / concepts / classes / engines required to store, manage, represent, render and output the assets we produce.

### [- Graphics -]

#### Scene graph / graphical hierarchy using Game Objects [GO]

Every game entity, whether it is seen as a sprite, a text object, a combination of both or by something else, needs to be encapsulated in a Game Object. GOs will serve the purpose of helping the things that they hold to exist in the environment correctly (visibility/transformations) and hierarchically. A GO can be added into another GO, thus forming a parent-child bond.

#### 9-patch image support

The backgrounds of GUI elements can either be made of plain colors or a texture. Using a texture that can be scaled infinitely will be very beneficial as it will introduce a lot of flexibility and ease. One way to do this is to use the 9-patch image concept, where an image will be split into 9 pieces in order to stretch the areas that can be stretched without making any undesirable stretchings or pixelations.

#### Parallax scrolling

A pseudo-3D effect where background elements scroll slower than foreground elements. We can manipulate the scene graph’s transformations hierarchy logic in order to give us this effect.

### [- Graphical Engines -]

#### AppWindow - Scenes relationship

An instance of a game will have an instance of an AppWindow. It is an extension to SFML’s sf::RenderWindow that will provide easy access to manipulate settings of the window. It holds a vector of Scenes that can be added and removed at any time. The main game loop will reside in the AppWindow and there are 5 main parts that a Scene will be able to override and customize:

* loading
* unloading
* event handling
* updating
* rendering

A Scene will act as a part of the game that you can see and interact with (equivalent to Android’s Activities).

#### Particle system

Particle systems are containers in which particles are made, manipulated, built, and rendered. They should be extended for uses like making fireballs or smoke emitters.

A particle system holds many variables to control its particles with:

* lifetime
* colour over life
* size over life
* emitter position
* gravity
* texturing
* particle rotation
* particle velocity
* collisions

#### TileMap engine

This will be able to read/parse in data in order to build a map that uses a single texture (tileset) for creating the environment with.

### [- Rendering -]

#### Sprite batching

A rendering technique that increases rendering performance by batching similar sprites together (through storage and sorting) and thus decreasing the amount of draw calls.

Should be used especially for rendering tilemaps.

#### Sprite animation rendering

Used by sprites that will need to be animated. The texture could be composed of multiple sub-rects that will be scrolled through as the rendering is done.

#### GLSL shading effects

Utilizing the GLSL high level shading language, we can manipulate each vertex / plane to get the desired effect, such as

* Glow
* Haze
* Additive colour
* Blur
* Shock/waves

### [- Resource Management (Internal & External) -]

#### Asset resource management

This will be used to load, cache, destroy and free our assets. This will decrease memory usage (storing only 1 copy of each resource) and increase performance (referencing the already cached resource instead of making a new one on the fly).

#### Asset storage

We need a way to organize all of our assets in a neat and easy to navigate folder structure and possibly have them be in a custom made ZIP-format resource file.

### [- Audio -]

#### Sound Manager

A static class that will manage all of the audio resources. It will provide easy ways to play sounds and music appropriately. Features: 2D Spatialization which can take mono sounds and place them in the stereo image at a point relative to the observer. Pre-loads short and/or common sounds into memory for instant access. Streams music from the disk. Performs automatic resource clean-up.

## 

## [ > Sound < ]

This section encompasses the aural elements of the game. This includes sound effects(collisions, events), music(menu, lobby, in-game) and entity emission(footsteps, attack sounds, death sounds, idle sounds, etc.).

The music will follow the 'tribal' theme of the game, with soft percussion including timbales, djembe, congas and bongos. It will also include some traditional woodwind and light brass tones. 'Creepier' or more discordant areas will use a more synthetic style with an ominous timbre to create tension.

Sound effects will match in feel with the graphics of the respective entity that they originate from(e.g: the sounds for a particular monster should match the ‘feel’ of the graphics for that monster). Many monsters might be ‘slimy’ or have ‘creepy’ or ‘disgusting’ look to make them obviously the enemy. Many monster sounds may reflect this with ‘moist’ or ‘wet’ sounds. Most monster sounds will be ‘vocalisations’ from the respective monsters.

Event sounds should be satisfying or disappointing, depending on the rectitude of the respective event. Character sounds should be short, simple, and most of all, **not** annoying or grating when played repetitively. These are sounds that will be heard often, so they need to be palatable when heard repeatedly, potentially having variants from which a random choice can be made a playback time.

### [ - Player - ]

* Walk/run sounds (including footsteps on different tile types)
* Attack sounds (unarmed and each weapon)
* Death sounds
* Hurt sounds
* Class specific sounds (taunts, abilities, etc.)
* Ghostly noises for ghosts or spectres? (theremin?)

#### Player Effects

* Status Effects
  + Bubbling or other effect noises
* Level sound
* Spawn sound
* Victory music
* Buff sounds (shimmering, etc.)

### [ - Weapons - ]

For each weapon:

* Hit sound
* Miss sound
* Crit sound?
* Projectile launch
* Projectile collision(each tile)
* Projectile collision(entity)

### [ - Monsters - ]

#### Bosses

* Main attack charge
* Main attack hit
* Main attack miss
* Secondary attack sounds
* Movement sounds
* Death sounds
* Hurt sounds
* Idle sounds
* Aggro / taunt sounds
* Boss music

#### Mini-Bosses

Mini bosses will be placed on the map in small quantities as well as summoned by deities. To start there will be ***three*** mini-bosses available. Each mini- boss will be composed of:

* Movement sounds
* Attack sounds
* Death sound
* Hurt sound
* Idle sounds
* Chase / aggro sounds
* Summon sound(poof or shimmer)

#### Minions

* Movement sounds
* Attack sounds
* Death sounds
* Hurt sounds
* Summon sound(lesser poof or shimmer)

### [- Environment - ]

* Ambient sounds (dripping, creaking, etc.)

### [ - Deities - ]

* Taunt / catchphrase for selection
* Aspect / ability sounds

### [ - Interactables & Objects - ]

* Rustling leaves
* Pots breaking
* Crates breaking / damage
* Trap setting, activation
* Torches / lamp crackling

### [ - Particles - ]

* Sparkles
* Grinding / sparks
* Splatter
* Fire / Crackling

**[ - Menu & GUI - ]**

* Menu select sound
* Menu movement sound
* Slider / checkbox tick
* Chat send / receive sound
* Low HP sounds?
* Exp gain sound?
* Timer sounds?
* Splash audio / music
* Menu music

### [ - Miscellaneous - ]

* Audio / music for trailer
* Announcements (ex: “PLAYER DEFEATED”, “TIME’S UP!”, “GAME OVER”)

## 

## [ > Art < ]

This section encompasses the visual elements of the game. This includes sprites, logo work, box art, icons, etc. All assets are organized into categories they belong to.

The art that will be created will be in a **simplistic *¾ perspective pixel art*** with african inspired tribal horror elements matching the chosen theme of ***Dark Fantasy.*** This does not mean **dark colors.**

To elaborate on this concept; players will be given random heads upon game entry with various vibrant colors, patterns, feathers, radical facial expressions, beak faces, etc. Monsters will also follow this theme as well as the rest of the game.

### [ - Player - ]

Anything needed for the ‘champion’ that the players control

* Each character will have 4 frames min. in at least 2 directions (depends on art direction)
* run animations
* class selection
* attribute selection
* 8-16 random heads/masks to be selected on round entry (x4 viewing direction)
* attack animations
* death animation
* hurt frame
* on fire animation

#### Player Effects

* status effects
  + Effects that overcome the sprites such as poison, paralysis, and sleep.
* level up animations
* victory dance
* buffs from deities (ex: health regain)

### [ - Weapons - ]

Weapons will be a small portion of the game but includes:

* attack animations
* projectiles

### [ - Monsters - ]

In the game there are 3 tiers of monsters; bosses, mini-bosses, and minions. For a starting point the ratio of monsters created will be 1, 3, 5 respectively. Monsters are spawned on map creation and by deities.

#### Bosses

Each round will contain ***one*** boss at the center of the map. This boss will be a large creature yielding a high reward.The boss will require the following:

* 4 frames min. in at least 2 directions (depends on art direction)
* attack frames
* death animation
* hurt frame

#### Mini-Bosses

Mini bosses will be placed on the map in small quantities as well as summoned by deities. To start there will be ***three*** mini-bosses available. Each mini- boss will be composed of:

* 4 frames min. in at least 2 directions (depends on art direction)
* attack frames
* death animation
* hurt frame
* summon frames ( poof or glow )

#### Minions

Minions will appear on the map most frequently as they are easy little critters.The minimum number of minions will be ***five*.** Each minion requires:

* 4 frames min. in at least 2 directions (depends on art direction)
* attack frames
* death animation
* hurt frame
* summon frames ( poof or glow )

### [- Environment - ]

Environment composes all things outside of characters and enemies on map. The bulk of this work is tilesets.

* unknown amount of tile sets
* liquid tiles & animation ( ex: water, poison liquids, slime )(?)

### [ - Deities - ]

Although deities are not visible on the map there are still some assets needed as following:

* abilities icons
* aspects icons with cooldown animations

### [ - Interactables & Objects - ]

This is a general category anything on the map that is physical but non-living. Some of these objects may be breakable as well.

* shrines for each player starting point
* trees
* pots
* crates
* traps
* totems
* lamps & torches with their lighting
* sign posts

### [ - Particles - ]

* sparkles
* hit sparks
* blood
* fire sparks
* smoke

**[ - Menu & GUI - ]**

* Mini map
* mini map markers (ex: green arrow to show where player green is located; must move proportionate to player)
* menu art and icons
  + sliders
  + checkboxes
  + buttons
* chat UI
  + chat box
  + chat buttons
  + chat fields
* HP bar for player with hue changes as it decrement
* HP bar for monsters
* buff and status icons
* player stats
* exp bar
* level display
* scoreboard and timer
* scoreboard icons to show if player is deity, champion, or ghost
* transitions between lobby and game
* splash screen on launch
* victory screen
* enclosing walls

### [ - Miscellaneous - ]

A list of odds and ends that don’t fit in any specific area.

* logo
* game icon for desktop
* box art
* poster art
* trailer art
* announcements at the top of the screen (ex: “PLAYER X DEFEATED”, “TIMES UP!”, “GAME END”)
* tutorial images for controls
* ghost haunting sprite (only visible to ghosts); 2 dimension 3 frame animated