Week 8: Digital Asset Management

DIGITAL ASSET DEVELOPMENT

Contents

- Project structure
- Organising digital assets
- Production pipelines

Managing Assets

- So far we have largely looked at the technical detail of asset development
- How we organise and structure assets within a project is also important
- Many software tools require a specific project structure
- Digital asset management is a major issue for any studio
  - Determined by production pipeline

Building a Project

- Creating individual assets can be done in a fairly unstructured way
- Projects involving significant numbers of assets require a systematic approach
  - Enables smoother workflow
  - Minimises errors and misuse of assets
  - Allows projects to be more portable
  - Essential for team working
  - Allows material to be reused efficiently

Project Structures

- Different software tools have very varied approaches
- Some are relatively non-prescriptive about project structure
  - Often leave you to handle it yourself!
- Some require all items to be in a defined location
- In general, more complex tools will have greater demands

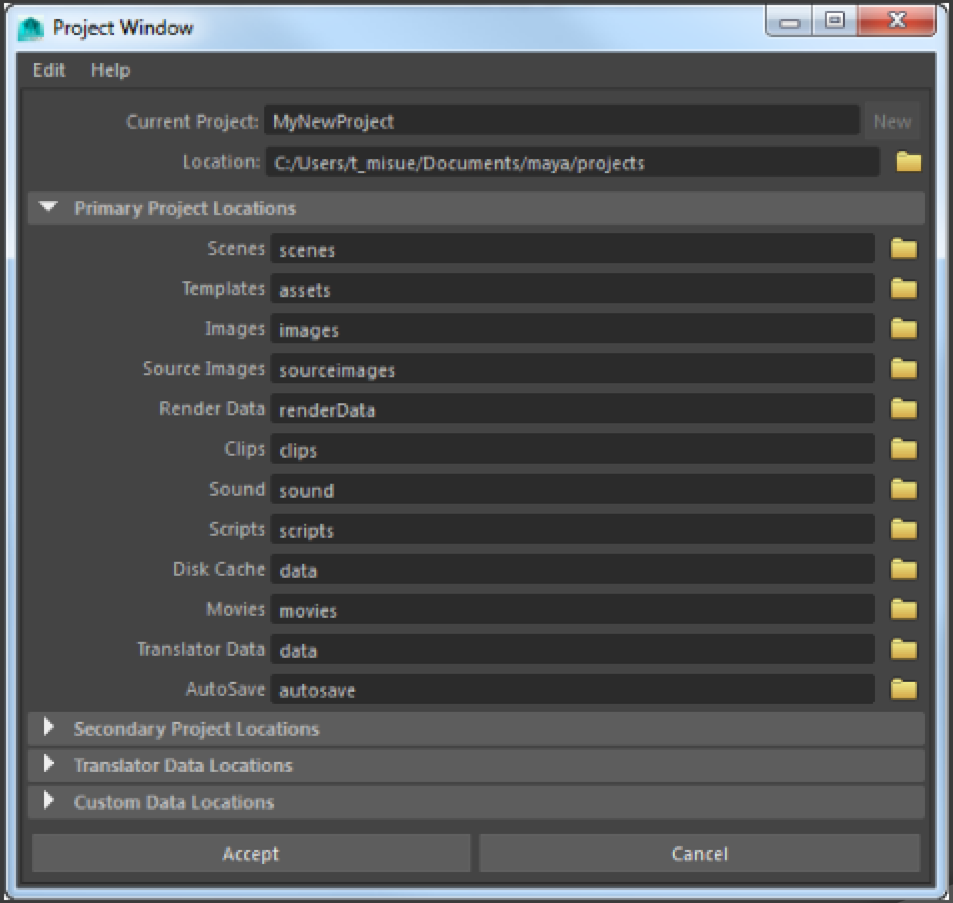
Adobe Tools

- Adobe products tend to be relatively "relaxed" about asset management
  - Can use Bridge for this purpose
- Photoshop effectively bundles all assets used into the main document
  - Can lead to very large files
- Premiere uses bins in the Project panel to organise assets used
  - This leads to problems if assets are moved relative to the project file

Maya Projects

- “organisation is the key to creating a successful animation” (*Mastering Maya*)
- Creating a new project
  - File > Project Window, and click New
  - Window shows list of folder locations
  - These define where Maya expects to find assets of each type relative to main folder
  - Can reset these locations if necessary
- Not doing this leads to serious problems

Maya Project Window


Project Window
Edit
Help
New

Current Project: MyNewProject

Location: C:/Users/t_misue/Documents/maya/projects

Primary Project Locations

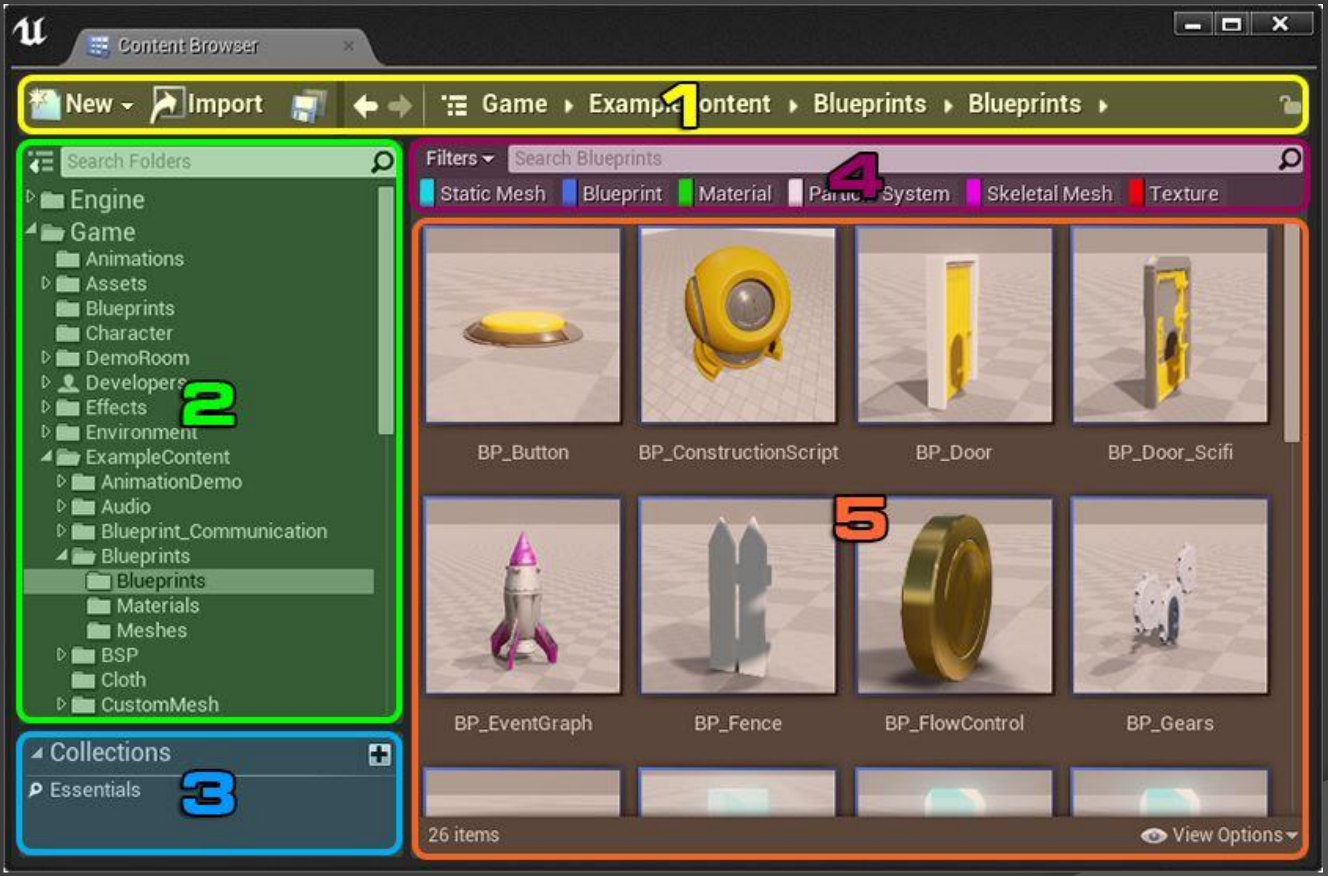
Scenes	scenes
Templates	assets
Images	images
Source Images	sourceimages
Render Data	renderData
Clips	clips
Sound	sound
Scripts	scripts
Disk Cache	data
Movies	movies
Translator Data	data
AutoSave	autosave

Secondary Project Locations
Translator Data Locations
Custom Data Locations
Accept
Cancel

Unreal Engine

- Unreal Engine effectively takes over asset control via the Content Browser
- Provides a variety of mechanisms for managing game assets
  - Sets up and manages asset folders
  - Lists assets through filters to specify certain types of data
  - Enables export of assets or migration between projects

Unreal Content Browser



navigation bar

sources panel

collections panel

asset management area

asset view

Research

- Research is vital for most projects
  - Images, video, sounds
  - Background and technical information
- However, research material is of little use if you can’t find it when required
  - A systematic approach is crucial
- For a modelling or texturing project create separate folders for each item
  - Folder holds all research data for that item

Categorising Research

- Any project will generate a variety of research data with varying uses
  - Inspiration: no direct contribution, but influences the final output
  - Information: guidance for production, such as tutorials or relevant facts
  - Photo reference: use to ensure correct representation of some detail
  - Blueprint: template for asset creation, such as floor plan for a building

Naming Conventions

- It is important to have a clear and logical approach to naming asset data
  - Saves endless hunting through folders
- File naming conventions may refer to:
  - The project element to which they belong
  - The type of data involved
  - The attribute being defined
  - Quality or resolution of the asset
  - The version number or date created

Examples

- Texture image for environment model:
  - wall_brickTexture_bump_1024_v3.jpg
  - Bump map for a brick wall with pixel resolution of 1024x1024
- Audio for game character:
  - orc_vocal_grunt_16bit_20151005.mp3
- The terms used are less important than having a consistent convention
  - Results group naturally in a file browser

Version Control

- One use of this approach is that it aids version control
  - This involves tracking assets as a project evolves and keeping them synchronised
- Especially important when assets can reference each other
  - eg. 3D models referencing texture files
  - Any situation where code is involved
- Thus it is crucial for game engines

Metadata

- Some environments encourage the use of metadata for assets
  - Descriptive data that identifies certain asset properties – this can be searchable
- MP3 files have metadata embedded in the format via use of ID3 tags
- Digital photos often include metadata for camera type and resolution
  - Other tags can usually be added

Backing Up

- Project backup is **essential**
- We may give you an extension (if you're lucky) – employers or clients won't!
- In developing a backup strategy:
  - Back up files to an independent file system
  - Use multiple backups where feasible
  - Keep backups in separate locations
  - Online storage is reliable – use it!
  - Keep a systematic archive

Larger Projects

- Projects with large or multiple teams are much harder to manage effectively
- Different teams creating and using assets must coordinate with one another
- Most large studios use a dedicated asset management system
  - Database driven
  - Assets tagged and tracked
  - Accessible via browsing system

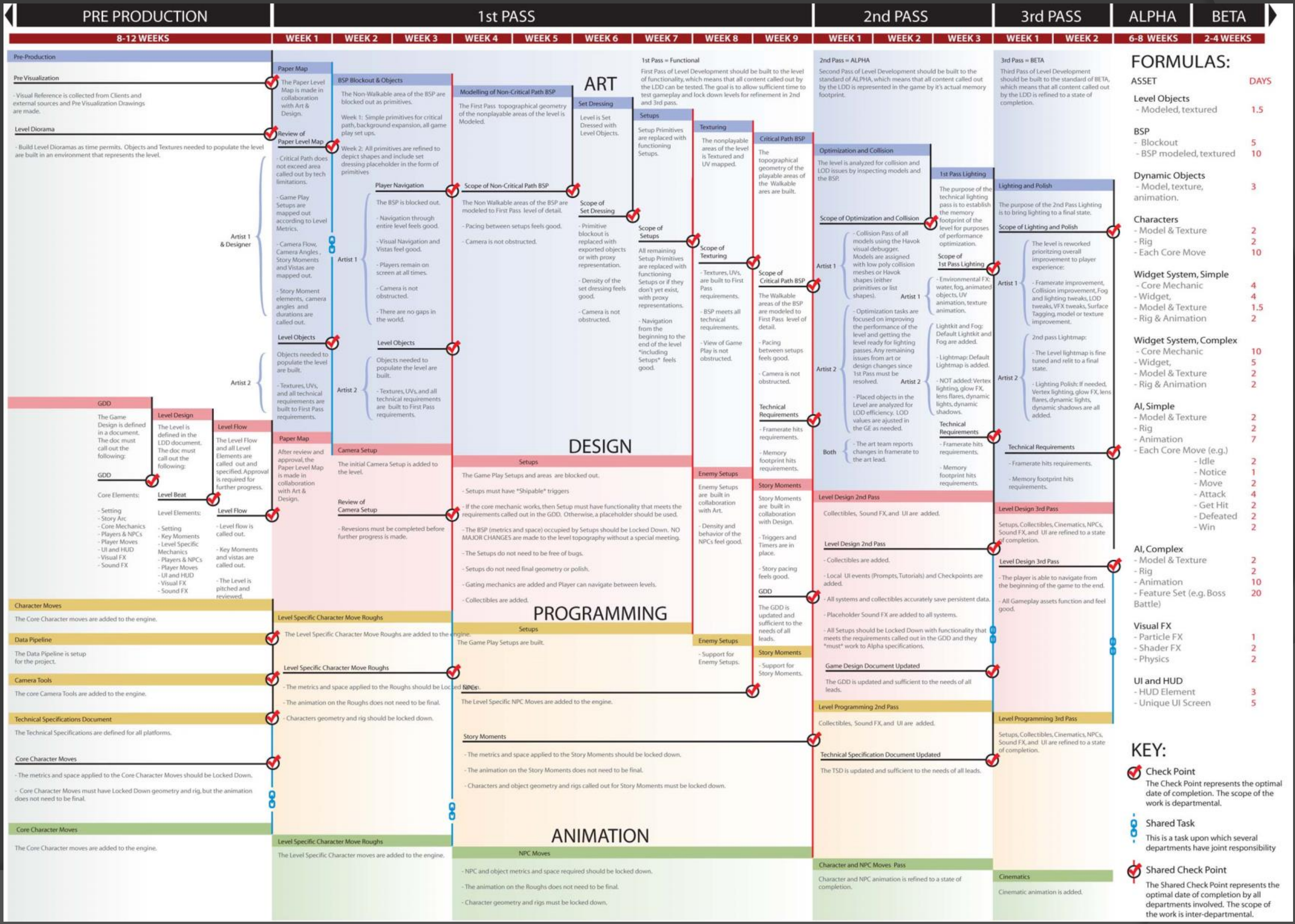
Industry Practice

- There are many systems available for managing digital assets
  - Most are aimed at conventional businesses
  - Some studios have much tougher demands
- Volume of data is a major issue
  - Very large storage (and backup) needs
  - Transferring data offsite can be a problem
- For animation, rendering is another challenge for IT infrastructure

Production Pipelines

- Each studio has its own production process, referred to as the pipeline
- Pipelines vary between industries, but have many similarities
- Important to know how your skill set fits into the overall pipeline
- In a small studio, individuals may be involved in different parts of the pipeline
  - Larger studios are more compartmentalised

Example Game Asset Pipeline



PRE PRODUCTION: 8-12 WEEKS | 1st PASS: WEEK 1, WEEK 2, WEEK 3, WEEK 4, WEEK 5, WEEK 6, WEEK 7, WEEK 8, WEEK 9 | 2nd PASS: WEEK 1, WEEK 2, WEEK 3 | 3rd PASS: WEEK 1, WEEK 2 | ALPHA: 6-8 WEEKS | BETA: 2-4 WEEKS

1st Pass = Functional

First Pass of Level Development should be built to the level of functionality, which means that all content called out by the LDD can be tested. The goal is to allow sufficient time to test gameplay and lock down levels for refinement in 2nd and 3rd pass.

2nd Pass = ALPHA

Second Pass of Level Development should be built to the standard of ALPHA, which means that all content called out by the LDD is represented in the game by it's actual memory footprint.

3rd Pass = BETA

Third Pass of Level Development should be built to the standard of BETA, which means that all content called out by the LDD is refined to a state of completion.

ART

Pre-Production

Pre Visualization

- Visual Reference is collected from Clients and external sources and Pre Visualization Drawings are made.

Level Diorama

- Build Level Dioramas as time permits. Objects and Textures needed to populate the level are built in an environment that represents the level.

Paper Map

The Paper Level Map is made in collaboration with Art & Design.

Review of Paper Level Map

Artist 1 & Designer

- Critical Path does not exceed area called out by tech limitations.

- Game Play Setups are mapped out according to Level Metrics.

- Camera Flow, Camera Angles , Story Moments and Vistas are mapped out.

- Story Moment elements, camera angles and durations are called out.

Level Objects

Artist 2

Objects needed to populate the level are built.

- Textures, UVs, and all technical requirements are built to First Pass requirements.

BSP Blockout & Objects

The Non-Walkable area of the BSP are blocked out as primitives.

Week 1: Simple primitives for critical path, background expansion, all game play set ups.

Week 2: All primitives are refined to depict shapes and include set dressing placeholder in the form of primitives

Player Navigation

Artist 1

The BSP is blocked out.

- Navigation through entire level feels good.

- Visual Navigation and Vistas feel good.

- Players remain on screen at all times.

- Camera is not obstructed.

- There are no gaps in the world.

Level Objects

Artist 2

Objects needed to populate the level are built.

- Textures, UVs, and all technical requirements are built to First Pass requirements.

Modelling of Non-Critical Path BSP

The First Pass topographical geometry of the nonplayable areas of the level is Modeled.

Scope of Non-Critical Path BSP

The Non Walkable areas of the BSP are modeled to First Pass level of detail.

- Pacing between setups feels good.

- Camera is not obstructed.

Set Dressing

Level is Set Dressed with Level Objects.

Scope of Set Dressing

- Primitive blockout is replaced with exported objects or with proxy representation.

- Density of the set dressing feels good.

- Camera is not obstructed.

Setups

Setup Primitives are replaced with functioning Setups.

Scope of Setups

All remaining Setup Primitives are replaced with functioning Setups or if they don't yet exist, with proxy representations.

- Navigation from the beginning to the end of the level *including Setups* feels good.

Texturing

The nonplayable areas of the level is Textured and UV mapped.

Scope of Texturing

- Textures, UVs, are built to First Pass requirements.

- BSP meets all technical requirements.

- View of Game Play is not obstructed.

Critical Path BSP

The topographical geometry of the playable areas of the Walkable ares are built.

Scope of Critical Path BSP

The Walkable areas of the BSP are modeled to First Pass level of detail.

- Pacing between setups feels good.

- Camera is not obstructed.

Technical Requirements

- Framerate hits requirements.

- Memory footprint hits requirements.

Optimization and Collision

The level is analyzed for collision and LOD issues by inspecting models and the BSP.

Scope of Optimization and Collision

Artist 1: - Collision Pass of all models using the Havok visual debugger. Models are assigned with low poly collision meshes or Havok shapes (either primitives or list shapes).

Artist 2: - Optimization tasks are focused on improving the performance of the level and getting the level ready for lighting passes. Any remaining issues from art or design changes since 1st Pass must be resolved.

- Placed objects in the Level are analyzed for LOD efficiency. LOD values are ajusted in the GE as needed.

Both: - The art team reports changes in framerate to the art lead.

1st Pass Lighting

The purpose of the technical lighting pass is to establish the memory footprint of the level for purposes of performance optimization.

Scope of 1st Pass Lighting

Artist 1: - Environmental FX: water, fog, animated objects, UV animation, texture animation.

Artist 2: Lightkit and Fog: Default Lightkit and Fog are added.

- Lightmap: Default Lightmap is added.

- NOT added: Vertex lighting, glow FX, lens flares, dynamic lights, dynamic shadows.

Technical Requirements

- Framerate hits requirements.

- Memory footprint hits requirements.

Lighting and Polish

The purpose of the 2nd Pass Lighting is to bring lighting to a final state.

Scope of Lighting and Polish

Artist 1: The level is reworked prioritzing overall improvement to player experience:

- Framerate improvement, Collision improvement, Fog and lighting tweaks, LOD tweaks, VFX tweaks, Surface Tagging, model or texture improvement.

Artist 2: 2nd pass Lightmap:

- The Level lightmap is fine tuned and relit to a final state.

- Lighting Polish: If needed, Vertex lighting, glow FX, lens flares, dynamic lights, dynamic shadows are all added.

Technical Requirements

- Framerate hits requirements.

- Memory footprint hits requirements.

DESIGN

GDD

The Game Design is defined in a document. The doc must call out the following:

GDD

Core Elements:

- Setting
- Story Arc
- Core Mechanics
- Players & NPCs
- Player Moves
- UI and HUD
- Visual FX
- Sound FX

Level Design

The Level is defined in the LDD document. The doc must call out the following:

Level Beat

Level Elements:

- Setting
- Key Moments
- Level Specific Mechanics
- Players & NPCs
- Player Moves
- UI and HUD
- Visual FX
- Sound FX

Level Flow

The Level Flow and all Level Elements are called out and specified. Approval is required for further progress.

Level Flow

- Level flow is called out.

- Key Moments and vistas are called out.

- The Level is pitched and reviewed.

Paper Map

After review and approval, the Paper Level Map is made in collaboration with Art & Design.

Camera Setup

The initial Camera Setup is added to the level.

Review of Camera Setup

- Revesions must be completed before further progress is made.

Setups

The Game Play Setups and areas are blocked out.

- Setups must have *Shipable* triggers

- If the core mechanic works, then Setup must have functionality that meets the requirements called out in the GDD. Otherwise, a placeholder should be used.

- The BSP (metrics and space) occupied by Setups should be Locked Down. NO MAJOR CHANGES are made to the level topography without a special meeting.

- The Setups do not need to be free of bugs.

- Setups do not need final geometry or polish.

- Gating mechanics are added and Player can navigate between levels.

- Collectibles are added.

Enemy Setups

Enemy Setups are built in collaboration with Art.

- Density and behavior of the NPCs feel good.

Story Moments

Story Moments are built in collaboration with Design.

- Triggers and Timers are in place.

- Story pacing feels good.

GDD

The GDD is updated and sufficient to the needs of all leads.

Level Design 2nd Pass

Collectibles, Sound FX, and UI are added.

Level Design 2nd Pass

- Collectibles are added.

- Local UI events (Prompts, Tutorials) and Checkpoints are added.

- All systems and collectibles accurately save persistent data.

- Placeholder Sound FX are added to all systems.

- All Setups should be Locked Down with functionality that meets the requirements called out in the GDD and they *must* work to Alpha specifications.

Game Design Document Updated

The GDD is updated and sufficient to the needs of all leads.

Level Design 3rd Pass

Setups, Collectibles, Cinematics, NPCs, Sound FX, and UI are refined to a state of completion.

Level Design 3rd Pass

- The player is able to navigate from the beginning of the game to the end.

- All Gameplay assets function and feel good.

PROGRAMMING

Character Moves

The Core Character moves are added to the engine.

Data Pipeline

The Data Pipeline is setup for the project.

Camera Tools

The core Camera Tools are added to the engine.

Technical Specifications Document

The Technical Specifications are defined for all platforms.

Core Character Moves

- The metrics and space applied to the Core Character Moves should be Locked Down.

- Core Character Moves must have Locked Down geometry and rig, but the animation does not need to be final.

Level Specific Character Move Roughs

The Level Specific Character Move Roughs are added to the engine.

Level Specific Character Move Roughs

- The metrics and space applied to the Roughs should be Locked Down.

- The animation on the Roughs does not need to be final.

- Characters geometry and rig should be locked down.

Setups

The Game Play Setups are built.

NPCs

The Level Specific NPC Moves are added to the engine.

Story Moments

- The metrics and space applied to the Story Moments should be locked down.

- The animation on the Story Moments does not need to be final.

- Characters and object geometry and rigs called out for Story Moments must be locked down.

Enemy Setups

- Support for Enemy Setups.

Story Moments

- Support for Story Moments.

Level Programming 2nd Pass

Collectibles, Sound FX, and UI are added.

Technical Specification Document Updated

The TSD is updated and sufficient to the needs of all leads.

Level Programming 3rd Pass

Setups, Collectibles, Cinematics, NPCs, Sound FX, and UI are refined to a state of completion.

ANIMATION

Core Character Moves

The Core Character moves are added to the engine.

Level Specific Character Move Roughs

The Level Specific Character moves are added to the engine.

NPC Moves

- NPC and object metrics and space required should be locked down.

- The animation on the Roughs does not need to be final.

- Character geometry and rigs must be locked down.

Character and NPC Moves Pass

Character and NPC animation is refined to a state of completion.

Cinematics

Cinematic animation is added.

FORMULAS:

ASSET	DAYS
Level Objects	
- Modeled, textured	1.5
BSP	
- Blockout	5
- BSP modeled, textured	10
Dynamic Objects	
- Model, texture, animation.	3
Characters	
- Model & Texture	2
- Rig	2
- Each Core Move	10
Widget System, Simple	
- Core Mechanic	4
- Widget,	4
- Model & Texture	1.5
- Rig & Animation	2
Widget System, Complex	
- Core Mechanic	10
- Widget,	5
- Model & Texture	2
- Rig & Animation	2
AI, Simple	
- Model & Texture	2
- Rig	2
- Animation	7
- Each Core Move (e.g.)	
- Idle	2
- Notice	1
- Move	2
- Attack	4
- Get Hit	2
- Defeated	2
- Win	2
AI, Complex	
- Model & Texture	2
- Rig	2
- Animation	10
- Feature Set (e.g. Boss Battle)	20
Visual FX	
- Particle FX	1
- Shader FX	2
- Physics	2
UI and HUD	
- HUD Element	3
- Unique UI Screen	5

KEY:

Check Point

The Check Point represents the optimal date of completion. The scope of the work is departmental.

Shared Task

This is a task upon which several departments have joint responsibility

Shared Check Point

The Shared Check Point represents the optimal date of completion by all departments involved. The scope of the work is inter-departmental.