Game Asset Manager

Ferdinand Majerech

Univerzita Pavla Jozefa Šafárika v Košiciach
UPJŠ

December 10, 2012

What GAM is (going to be)

- System to manage game assets
- Web frontend
- Cross-platform backend (but not in Java)
- Built on top of time-tested tools

What GAM is not

- A full-featured VCS
- An issue tracker
- A project management system
- However, it might use these.

Focus

- Should be used by artists, level designers, etc.
- Not used by programmers.
- Lightweight server; customize for modder use.
- Not specific for one project/engine/studio
- Integrate (plugin) with an asset store (Unity asset store)

Workflow

- An asset has an owner (not a title)
  - Specifies dependencies, milestones
  - Constraints for an asset (triangle count, texture resolution. . .)
  - Assigns an author for an asset
  - Reviews an asset
- Asset author submits asset file/s.
- Any user can view/download assets.

Backend

- Built on a DVCS (at first just Git)
- One central repo
- Local branch repos for local changes
- Might integrate with an issue tracker (?)

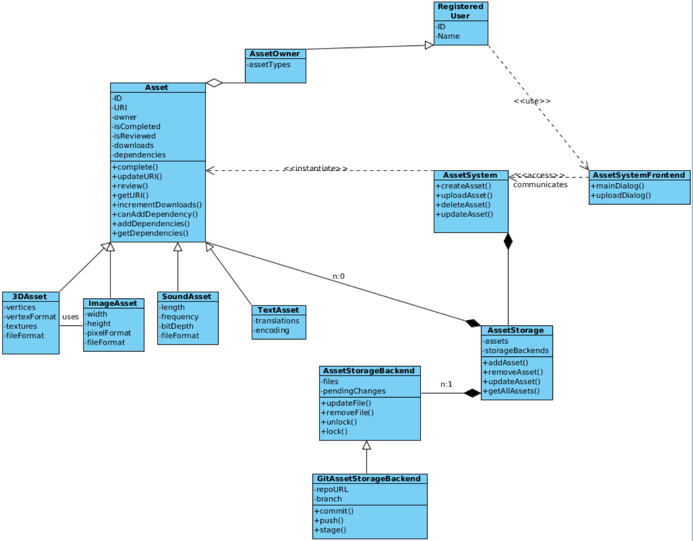
Asset dependencies

- Issue tracker on the backend?
- An asset can depend on other assets
- Some types of dependencies are not mandatory for testing
    - Model without textures in a test build
- Milestones (e.g. a DLC)

Packaging

- Packing into game data archives.
- Extend (plugin) for store packages (e.g. Desura).
- Game/mod archives from milestones.
- Packages for test builds (non-essential dependencies)
- Preprocessing (textures to GPU compressed formats)
- Package review (except testing packages)

Class diagram



Registered User
-ID
-Name

AssetOwner
-assetTypes

Asset
-ID
-URI
-owner
-isCompleted
-isReviewed
-downloads
-dependencies
+complete()
+updateURI()
+review()
+getURI()
+incrementDownloads()
+canAddDependency()
+addDependencies()
+getDependencies()

AssetSystem
+createAsset()
+uploadAsset()
+deleteAsset()
+updateAsset()

AssetSystemFrontend
+mainDialog()
+uploadDialog()

3DAsset
-vertices
-vertexFormat
-textures
-fileFormat

ImageAsset
-width
-height
-pixelFormat
-fileFormat

SoundAsset
-length
-frequency
-bitDepth
-fileFormat

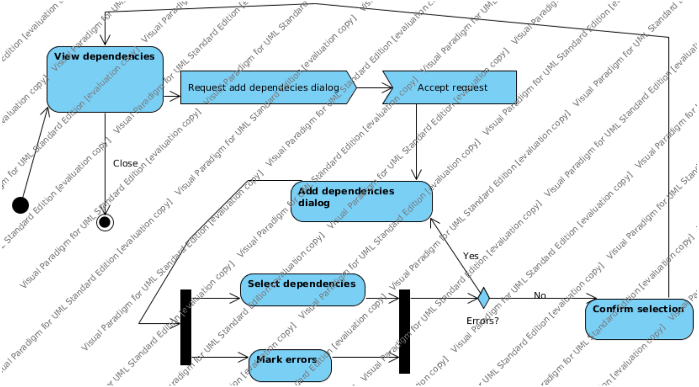
TextAsset
-translations
-encoding

AssetStorage
-assets
-storageBackends
+addAsset()
+removeAsset()
+updateAsset()
+getAllAssets()

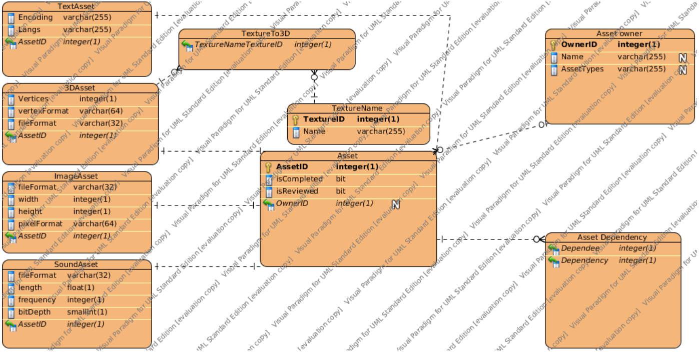
AssetStorageBackend
-files
-pendingChanges
+updateFile()
+removeFile()
+unlock()
+lock()

GitAssetStorageBackend
-repoURL
-branch
+commit()
+push()
+stage()

Activity diagram



ERD diagram (physical)



TextAsset

Encoding	varchar(255)	
Langs	varchar(255)	
AssetID	*integer(1)*	

3DAsset

Vertices	integer(1)	
vertexFormat	varchar(64)	
fileFormat	varchar(32)	
AssetID	*integer(1)*	

ImageAsset

fileFormat	varchar(32)	
width	integer(1)	
height	integer(1)	
pixelFormat	varchar(64)	
AssetID	*integer(1)*	

SoundAsset

fileFormat	varchar(32)	
length	float(1)	
frequency	integer(1)	
bitDepth	smallint(1)	
AssetID	*integer(1)*	

TextureTo3D

TextureNameTextureID	*integer(1)*	

TextureName

TextureID	**integer(1)**	
Name	varchar(255)	

Asset

AssetID	**integer(1)**	
isCompleted	bit	
isReviewed	bit	
OwnerID	*integer(1)*	N

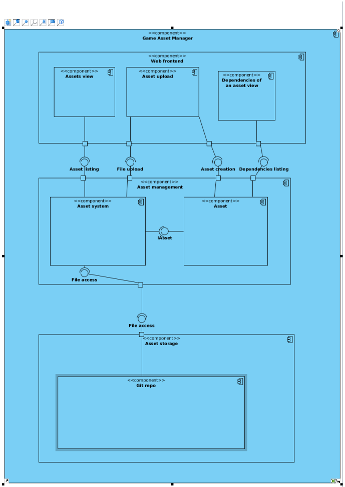
Asset owner

OwnerID	**integer(1)**	
Name	varchar(255)	N
AssetTypes	varchar(255)	N

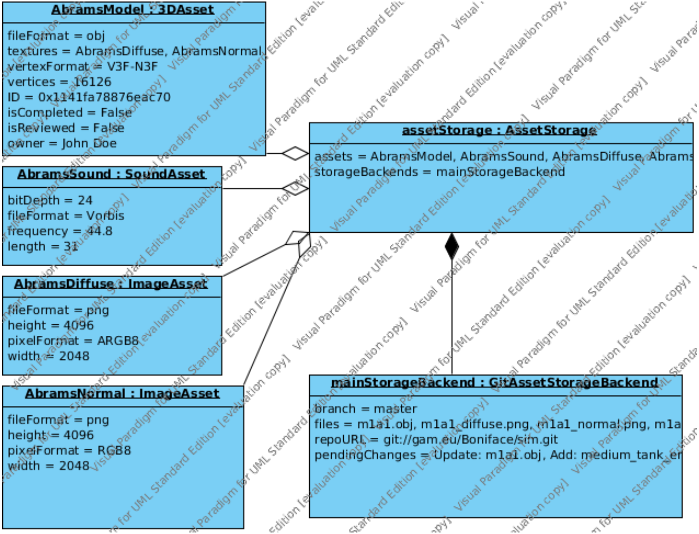
Asset Dependency

Dependee	*integer(1)*	
Dependency	*integer(1)*	

Component diagram



Object diagram



AbramsModel : 3DAsset

fileFormat = obj
textures = AbramsDiffuse, AbramsNormal
vertexFormat = V3F-N3F
vertices = 16126
ID = 0x1141fa78876eac70
isCompleted = False
isReviewed = False
owner = John Doe

AbramsSound : SoundAsset

bitDepth = 24
fileFormat = Vorbis
frequency = 44.8
length = 31

AbramsDiffuse : ImageAsset

fileFormat = png
height = 4096
pixelFormat = ARGB8
width = 2048

AbramsNormal : ImageAsset

fileFormat = png
height = 4096
pixelFormat = RGB8
width = 2048

assetStorage : AssetStorage

assets = AbramsModel, AbramsSound, AbramsDiffuse, Abrams
storageBackends = mainStorageBackend

mainStorageBackend : GitAssetStorageBackend

branch = master
files = m1a1.obj, m1a1_diffuse.png, m1a1_normal.png, m1a
repoURL = git://gam.eu/Boniface/sim.git
pendingChanges = Update: m1a1.obj, Add: medium_tank_er

SWOT

- Strengths:
  - Engine agnostic
  - Non-tech centric
- Weaknesses:
  - Engine agnostic
  - Monolithic
- Opportunities:
  - Modders
  - Open source + support
  - Kickstarter
- Threats:
  - Collaboration in infrastructure
  - Github, Bitbucket. . .
  - Engine-specific toolchains

Development costs

- Can vary drastically
- Small scale, FLOSS style development:
  - $5000 1 person on noodles for a year
  - Kickstarter: ask $5000, hype, get $20000 and an extra person. - (Or fail).
  - Support by open source (depends on interest)...
- Small-medium, more traditional (but still on noodles):
  - $50000+ for a (very) basic release
  - $20000+/yr for 1 outsorced person to support it
    - If anyone in gamedev cares to use an obsolete bloated tool
- Realistically:
  - Don't estimate the costs before spending a half

Thank you for your attention!