

# Game Design For Development

From Top to Bottom and Back Up

# Again, Who Are You?

- You are

- NOT just an Artist
- NOT just a Programmer

Millions of these

- You are learning to

- **Design**
- **AND Develop**
- Games

Very few Understand  
and Can Do this

## **NOT JUST GAMES**

Process for Games  
transfers to design and development  
of other things

*-- remember this !!!*

# Emphasis for Today

- Designing and Developing a game
  - requires understanding many aspects of the Design AND Development process
- You should be learning
  - to think at all levels of the process
- When designing a game you should consider
  - all aspects of creating and implementing it
    - *plus all the other things that make a good design*

# Motivation Summary

- Designing a game FOR Development
  - requires consideration and understanding of all the aspects of creating and implementing it
- Lots of people have game Ideas
- Some can put their idea into a Design description
- Very few see their designs be Developed into real games
- **Increase your odds**
  - **Design so it is easy to Develop**

# Design Versus Development

- Designing a game is easy
  - if you stop at the high level description
  - and do not care about the details
    - time travel is also easy if done this way
- A good design
  - takes into consideration the feasibility, measurability, and organization of development

# Previously

- Designing a game is designing a system
- Game design and development is iterative
- Games can be described as
  - the successive layering of constraints
- Games have elements

# Another Set of Parts

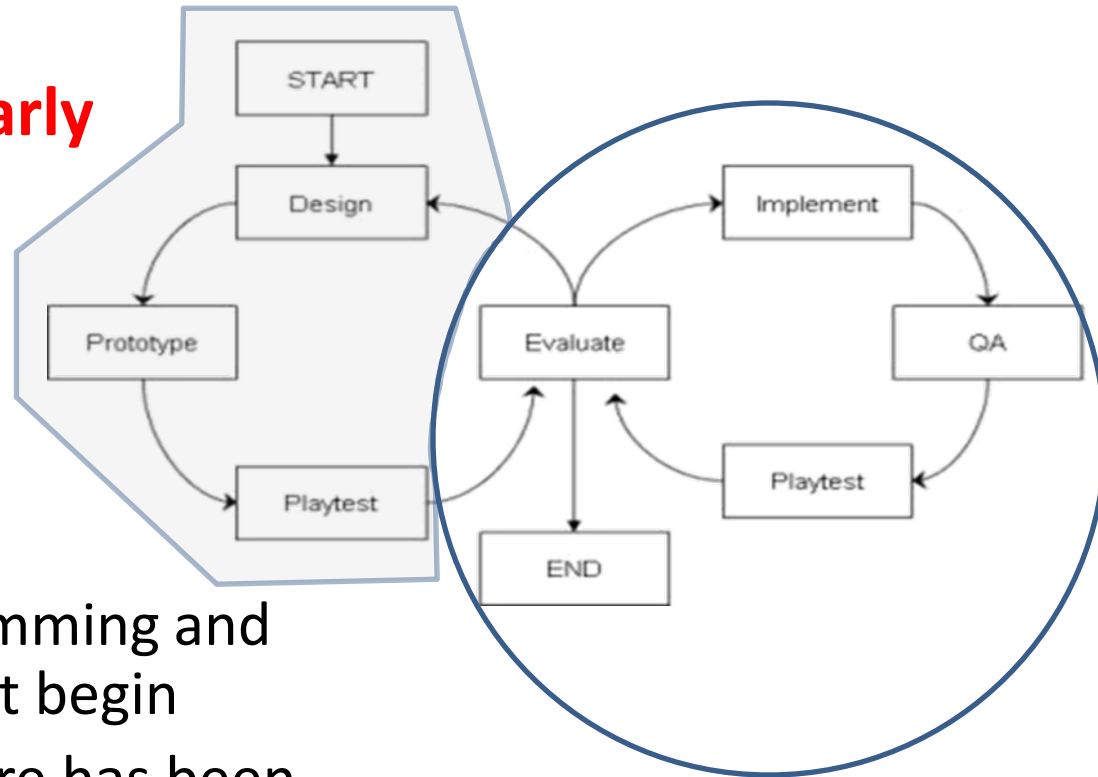
- A **game** (design)  
can be **decomposed** to parts  
that are **useful** in  
**planning** and **measuring**  
**development**

# How this Fits

- The following is more applicable to the stages

**after mockups and early prototype iterations**

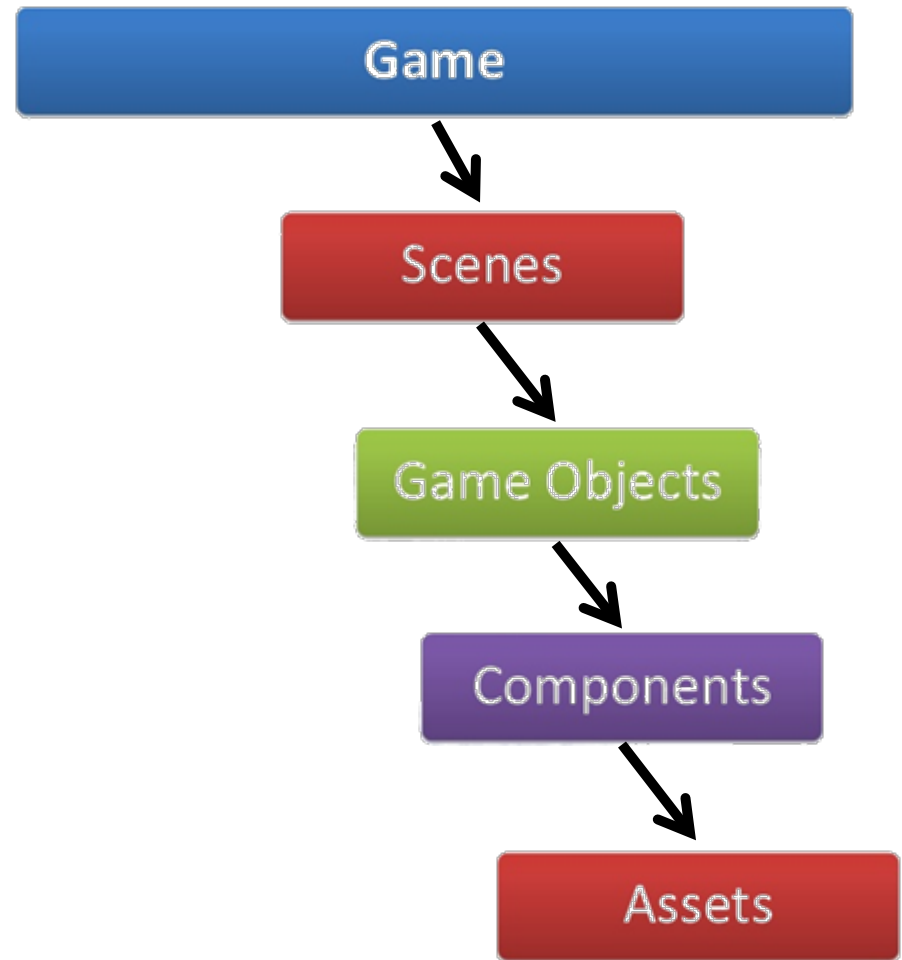
- This decomposition should be considered
  - before major programming and artwork development begin
  - after a solid game core has been established





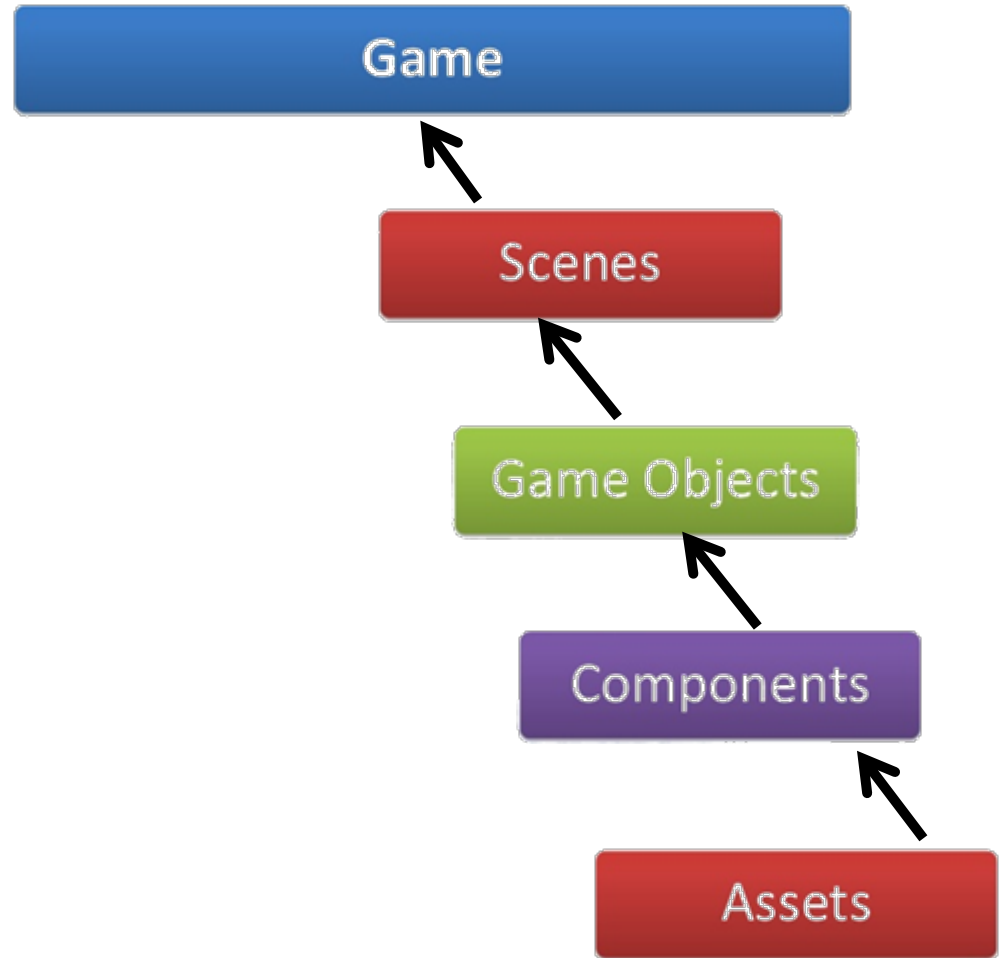
# Design: Top-Down

- Design
  - starts at a high level
  - and works down into the details
- This is TOP-DOWN



# Development: Bottom-Up

- Development
  - is often a Bottom-Up process
- Start Small
  - Assemble into Big



# Design FOR Development

- Design is Top-Down
- Development is Bottom-Up
- If the **Design** can **reduce** things into the **low level asset** requirements
- Then **Development** becomes the **process** of **creating** and **assembling** those **assets** into the game
  - the **success** of which can be **tracked** and **measured**
  - there is a **clear picture** of **how** and **what** needs to be developed and by **who** to achieve the game as designed
    - organized chaos

# And

- So What?
  - Design = top-down
  - Development = bottom-up
- How do we use this information?
- **Start with the high level design**
  - **Can you decompose it for development?**

# From the Top

- **Decompose the game into manageable pieces**
  - Assets, or **game elements**, that can be assigned to people to create
- How?
  - **Categorize the assets**
  - Similar to how people skills might be categorized

# Asset Categories

**Assets**



# Asset Categories

## Assets

```
public class MediaPlayer {  
    01  
    public Color colorColor = Color.green;  
    private void definition() {  
        02  
        this.audioSource.audio = clipColor;  
        03  
    }  
}
```

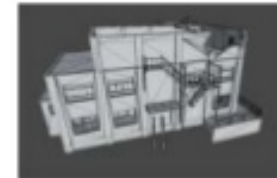
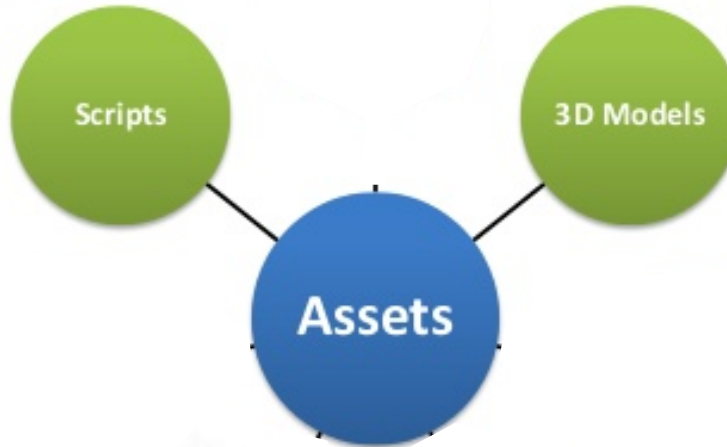
Scripts

Assets

# Asset Categories

## Assets

```
public class SphereScript : MonoBehaviour  
{  
    public Color color = Color.green;  
    private void Awake()  
    {  
        this.enabled.enabled_color = color;  
    }  
}
```

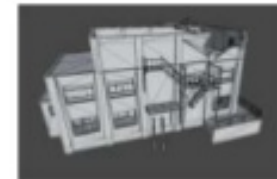
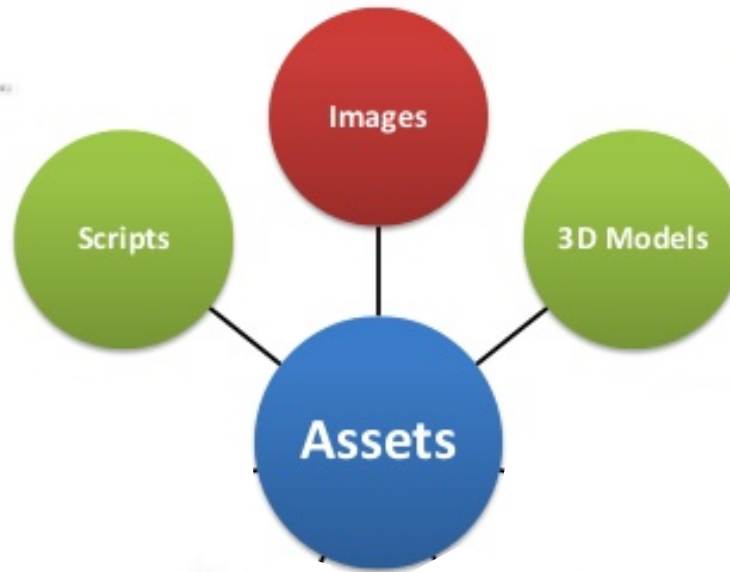




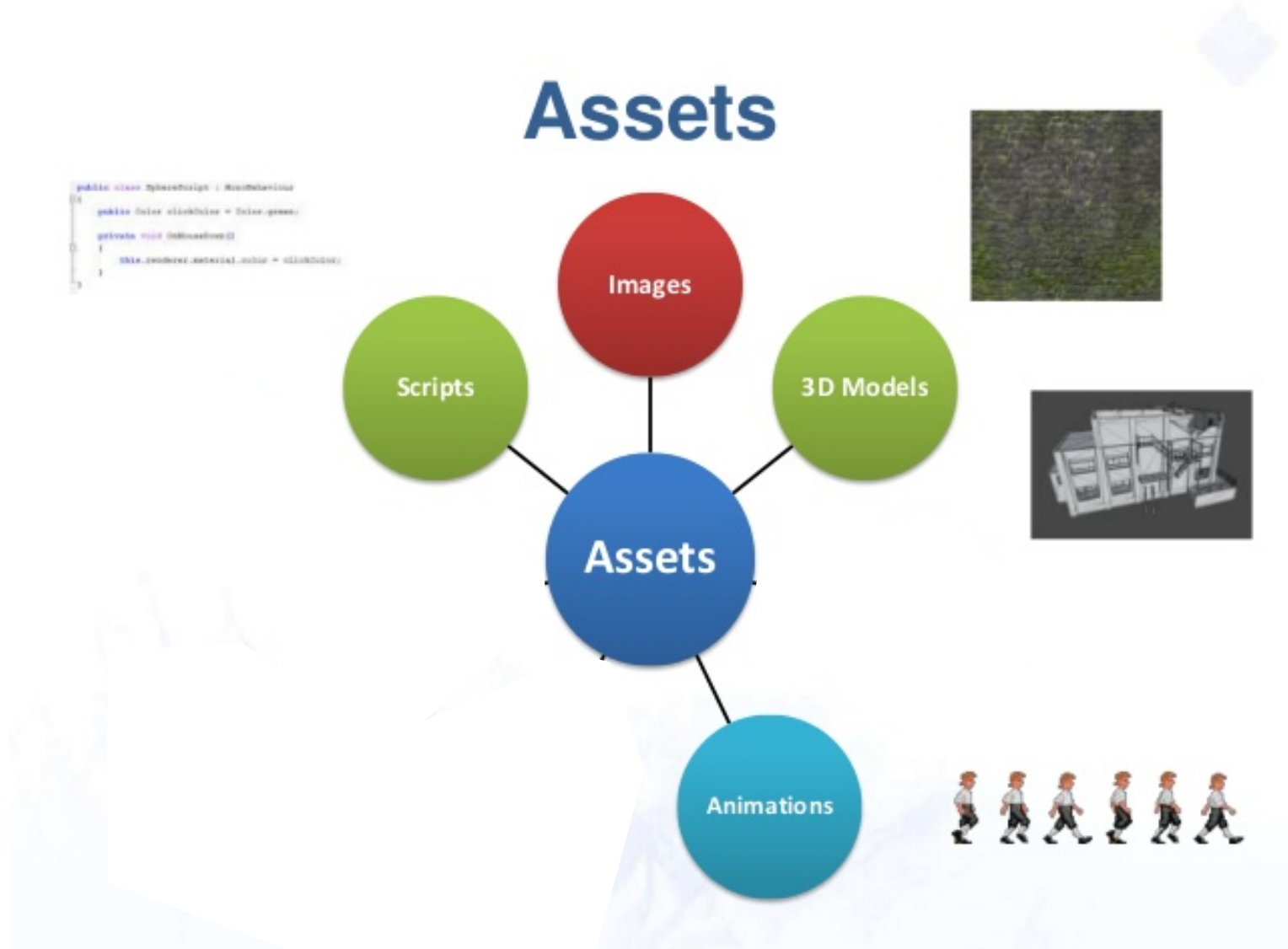
# Asset Categories

```
public class SphereScript : MonoBehaviour  
{  
    public Color color = Color.green;  
    private void Awake()  
    {  
        this.enabled.enabled = color;  
    }  
}
```

## Assets



# Asset Categories



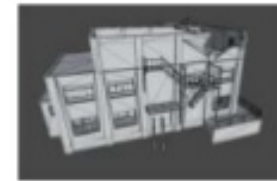
# Asset Categories



# Asset Categories

## Assets

```
public class SphereScript : MonoBehaviour  
{  
    public Color color = Color.green;  
    private void Awake()  
    {  
        this.enabled.enabled_color = color;  
    }  
}
```



# Asset Categories



# Decompose

- Have categories of people skills and assets
- Now Decompose the game – From the Top

# Top Down

- Since we have only a high level (top) view
  - **We start with a TOP-DOWN approach**
    - *Later*
    - *When we have lots of basic pieces/assets*
      - » *We may add extra stuff using a Bottom-Up Approach*
        - *Re-using assets already created (crazy isn't it?)*

*Reiterating:*

*Design and Planning go Top-Down*

*Development and Implementation go Bottom-Up*

# Asset List Creation – Top Down

- From Big to Small

Game

We have a description of a game  
*Should include scenes/levels/menus/...*



# Asset List Creation – Top Down

- From Big to Small



Game

We have a description of a game  
*Should include scenes/levels/menus/...*



Scenes

Decide on a scene to create

# Asset List Creation – Top Down

- From Big to Small



Game

We have a description of a game

*Should include scenes/levels/menus/...*



Scenes

Decide on a scene to create



Game Objects

Determine what type of objects are needed to create that scene

# Asset List Creation – Top Down

- From Big to Small



Game

We have a description of a game

*Should include scenes/levels/menus/...*



Scenes

Decide on a scene to create



Game Objects

Determine what type of objects are needed to create that scene



Components

Determine what components we need to make each object in the selected scene

# Asset List Creation – Top Down

- From Big to Small



Game

We have a description of a game

*Should include scenes/levels/menus/...*



Scenes

Decide on a scene to create




Game Objects

Determine what type of objects are needed to create that scene



Components

Determine what components we need to make each object in the selected scene



Assets

Determine what assets are needed to make each of those components

# Asset List Creation – Top Down

- From Big to Small



Game

We have a description of a game

*Should include scenes/levels/menus/...*



Scenes

Decide on a scene to create



Game Objects

Determine what type of objects are needed to create that scene



Components

Determine what components we need to make each object in the selected scene



Assets

Determine what assets are needed to make each of those components

And now we have a list of assets  
for ONE scene

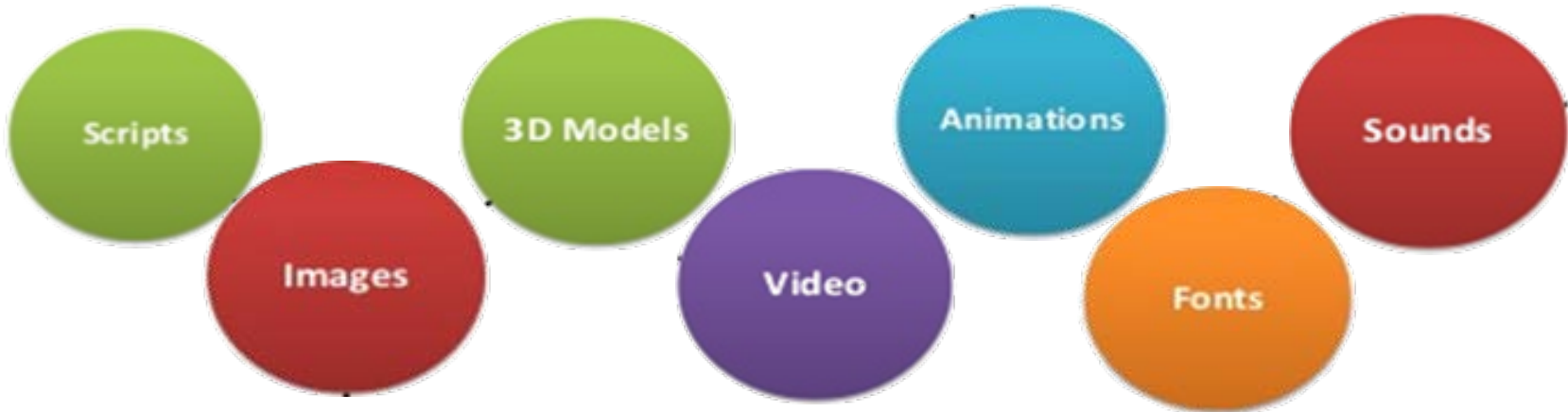
*Repeat for the other scenes*

# Decomposition Process

- Decomposing a Game into Scenes is ‘easy’
  - Scenes are as in theater → “set the scene”
    - A level is a scene (sometimes multiple scenes)
    - A secret area is a scene
    - A start menu can be a scene
    - A pause screen can be a scene
  - Objects are also ‘easy’
    - Things that are in the scene
    - Each thing is made of re-usable components
      - Car versus Truck... wheels, doors, headlights, engine

# Components Versus Assets

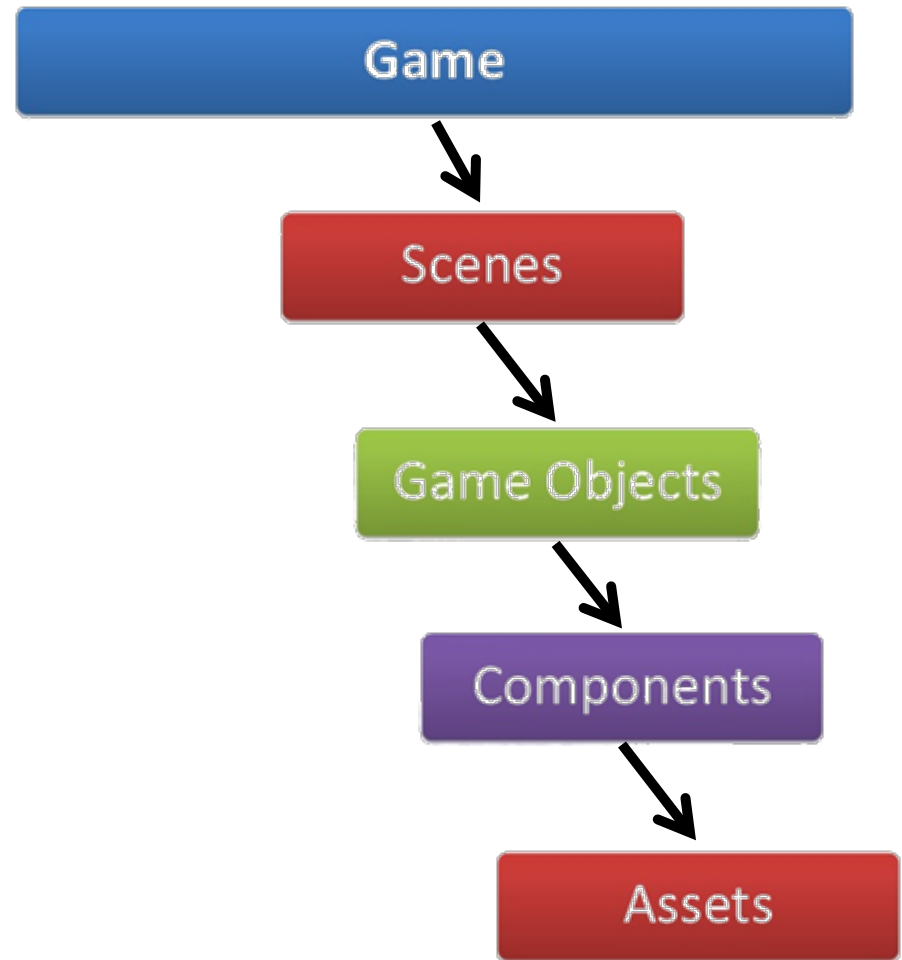
- **Assets** are typed/categorized
  - For example:



- A **component** is “composed of”
  - a small number of assets
    - of the same or different types

# Ideal Decomposition

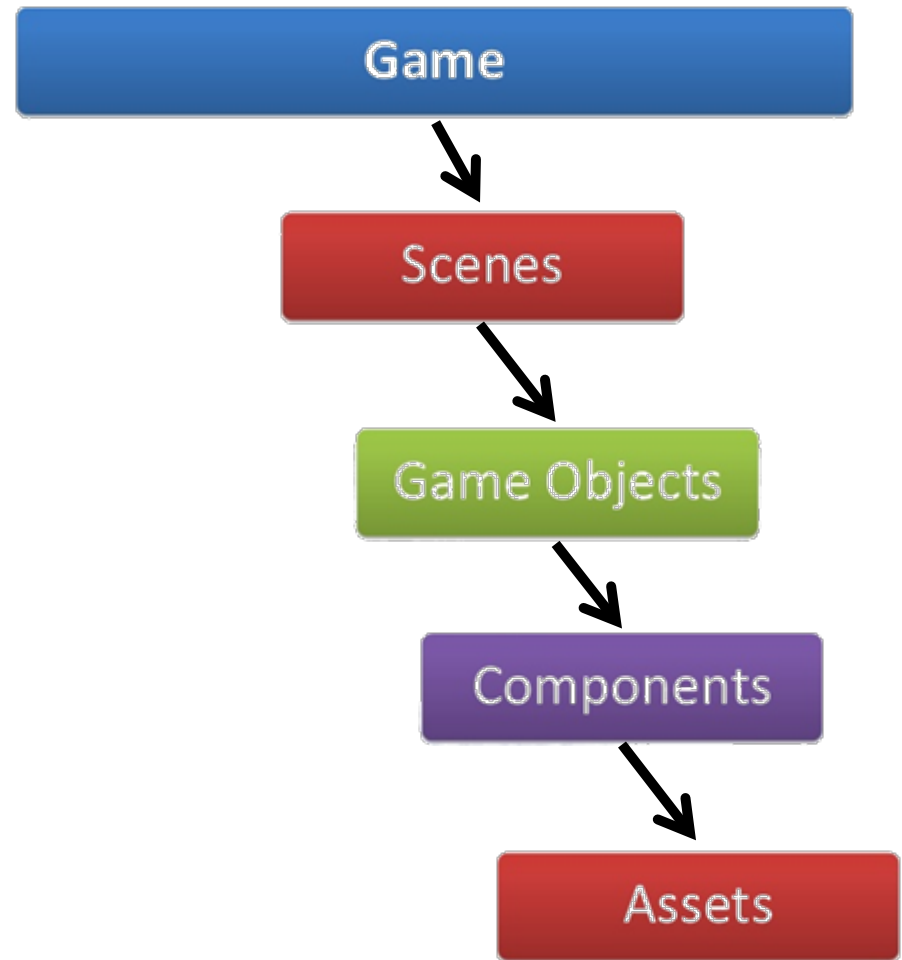
- Ideally the designer would decompose the entire game to assets





# Reality

- Ideally the designer would decompose the entire game to assets
- More often Design and Planning stop at the game object level



# OBJECT Asset Sheets

- Design stopping at the object level leads to the use of **OBJECT Asset Sheets**
- Typically these are **created by the development team**
  - **One for each object** the designer has specified to be in a scene

Asset Identifier	Fill in information here	
Asset Name	Fill in information here	
Asset Type	Fill in information here	
Script Description	Specification	Fill in information here
	Estimated Time Size (cost)	Fill in information here
Image Description		
	Specification	Fill in information here
	Estimated Time Size (cost)	Fill in information here
	Pixel Size	Fill in information here
3D Model Description	Specification	Fill in information here
	Estimated Time Size (cost)	Fill in information here
	Fits in Cube Size	Fill in information here
Video Description	Specification	Fill in information here
	Estimated Time Size (cost)	Fill in information here
	Size	Fill in information here
Animation Description	Specification	Fill in information here
	Estimated Time Size (cost)	Fill in information here
	Pixel Size	Fill in information here
	Number of Frames	Fill in information here
	Frames per Second	Fill in information here
Font Description	Specification	Fill in information here
	Estimated Time Size (cost)	Fill in information here
Sound Description	Specification	Fill in information here
	Estimated Time Size (cost)	Fill in information here

# Object Decomposition Example

- Example: **PAC-MAN Object**
  - Script Component
    - Asset: Script for user interface motion control
    - Asset: Script for collision with walls
      - with food and points update
      - with ghosts and death (if not powered up)
      - ...
    - ...
  - Image Component
    - Asset: Static Image for marketing
    - Asset: Static Image for menus
  - Animation Component
    - Asset: Image Sequence for Moving
    - Asset: Image Sequence for Dying
  - Sound Component
    - Asset: Sound for moving
    - Asset: Sound for eating
    - Asset: Sound for dying

# Object Asset Lists

- To assist the team in planning tasks
- The Team Lead creates Object Asset Lists
- Example:

Scene	Object Asset Name	Asset Identifier	Type	Scripts	Images	3D Models	Animations	Vid
Start Screen	Play Game Button	BTN_playGame_01	Button	x	x			
Start Screen	High Score Button	BTN_highScore_01	Button	x	x			
Start Screen	Settings Button	BTN_settings_01	Button	x	x			
Start Screen	Credits Button	BTN_credits_01	Button	x	x			
Start Screen	Game Logo	IMG_gameLogo_01	Image_Static		x			
Start Screen	Start Background Image	IMG_bckgrndStartScn_01	Image_Static		x			
Level 01a	Level 01a Background	IMG_bckgrndLvl01a_01	Image_Static		x			
Level 01a	Player Ship Idle	IMG_playerShipIdle_01	Image_Static		x			
Level 01a	Player Ship Bank Left	IMG_playerShipBankLeft_01	Image_Static		x			
Level 01a	Player Ship Bank Right	IMG_playerShipBankRight_01	Image_Static		x			
Level01a	Enemy Ship Fighter	IMG_enemyShipFighter_01	Image_Static		x			
Level01a	Enemy Ship Bomber	IMG_enemyShipBomber_01	Image_Static		x			
Level01a	Ship Explosion	ANM_shipExplosion_01	Image_Anim		x		x	

# (object) Asset Lists

- To assist the team in planning tasks
- The Team Lead creates Object Asset Lists
- Example:

Scene	Object Asset Name	Asset Identifier	Type	Scripts	Images	3D Models	Animations	Vid
Start Screen	Play Game Button	BTN_playGame_01	Button	x	x			
Start Screen	High Score Button	BTN_highScore_01	Button	x	x			
Start Screen	Settings Button	BTN_settings_01	Button	x	x			
Start Screen	Credits Button	BTN_credits_01	Button	x	x			
Start Screen	Game Logo	MG_gameLogo_01	Image_Static		x			
Start Screen	Start Background Image	MG_bckgrndStartScn_01	Image_Static		x			
Level 01a	Level 01a Background	MG_bckgrndLvl01a_01	Image_Static		x			
Level 01a	Player Ship Idle	MG_playerShipIdle_01	Image_Static		x			
Level 01a	Player Ship Bank Left	MG_playerShipBankLeft_01	Image_Static		x			
Level 01a	Player Ship Bank Right	MG_playerShipBankRight_01	Image_Static		x			
Level01a	Enemy Ship Fighter	MG_enemyShipFighter_01	Image_Static		x			
Level01a	Enemy Ship Bomber	MG_enemyShipBomber_01	Image_Static		x			
Level01a	Ship Explosion	ANM_shipExplosion_01	Image_Anim		x		x	

Objects

Are Composed of

Assets of Various Types

# (object) Asset Lists

What is important about this list?

- Example:

Scene	Asset Name	Asset Identifier	Type	Scripts	Images	3D Models	Animations	Videos
Start Screen	Play			x	x			
Start Screen	High			x	x			
Start Screen	Setti			x	x			
Start Screen	Cred			x	x			
Start Screen	Game				x			
Start Screen	Start				x			
Level 01a	Level				x			
Level 01a	Playe				x			
Level 01a	Playe				x			
Level 01a	Playe				x			
Level01a	Enem				x			
Level01a	Enem				x			
Level01a	Ship				x		x	

This ties your Design through Decomposition to People Skills

Making it Easier to Understand, Manage and Develop

Object Assets

Are Composed of

Assets of Various Types

# Object Asset Lists – Just ONE way

- You can then use Object Asset Lists to generate Team Task lists
  - is Just One Way
  - It is not perfect
- It **illustrates** how to
  - **Link the Architecture and Design of the Game to the** (late stage) **Development of the Game**
    - never forget about the people aspects
    - **make the development of the design easy to plan and implement**

# Summary

- Designing a game FOR Development
  - requires consideration and understanding of all the aspects of creating and implementing it
- Lots of people have game Ideas
- Some can put that idea into a Design description
- Very few see their designs be Developed into real games
- **Increase your odds**
  - **Design so it is easy to Develop**



# Applying This

- Your client has a high level game idea
- Take it, refine it
  - make sure it can be quickly decomposed to object level
    - if not, revise it until it can be
  - additional “to think about/plan for”
    - game mechanics/rules
    - user interface
    - data storage (game saves, game results, player stats)
    - clarity of game’s meta-purpose (serious/educational)

# The End

- Questions?