Metrics & Technical Art Considerations

Project2 - Fall 2023
CSD2400, CSD2401, UXG2400, DAA2402

Week 06

Overview

Metrics & Technical Art Considerations

These are highly recommended practises and considerations to make your life easier when developing.

As we are only scratching the surface here, we encourage you to go deeper so:

DO CONSULT ON THESE POINTS!

Overview

Metrics & Technical Art Considerations

1. Metrics in games
2. Art Specifications
3. Target Resolutions and Aspect Ratios
4. Importing Texture Assets
5. Animations
6. Particle Systems
7. Communication: Best Practises

Establishing Metrics for Development

1. Metrics in games

Metrics usually refers collecting User Analytics. In our case we will aim metrics at run time game optimisations:

- Why is the game suddenly dropping frames?
- Why are loading times suddenly longer than usual?
- Why are some sprites blurry?
- Why are colliders or animations not working on certain assets?
- Etc…

Tracking and Gating in Custom Engine

1. Metrics in games

- Create texture rules for format, sizes and transparency and check on import.
- Checks for missing asset components: pivot points, colliders, expected triggers etc… Log this!
- Set FPS targets and incorporate colour changes into counter: Green = Good, Red=Huh?
- Designers/BFA's: Performance graphs can show all breakdowns of realtime information! If available learn how to use them!
- Don't hide errors and logs! Force them to be visible!

Establishing Metrics for Development

1. Metrics in games

OR, we can do this manually, or by eye … but this is ‘guesswork’ having to inspect many files.

Automation and Graphs = Saves Time, Ensure Accuracy, Instant Results encourages bug fixes as they are detected.

Settings can be adjusted!

Consistant frame rates, Fast loading, and stutter free gameplay = Quality!

Don’t sabo your game, though guesswork or unawareness!

Units: “A Song of Art and Code”

2. Art Specifications

ASK: “What is a Unit?”

Techies: ‘Well…. It can be anything. It’s just a number!’

Designer/Artist: ‘What is anything?’

Techies: ‘Anything you like, it’s just a value!’

Designer/Artist: ‘…’

(Everyone shrugs shoulders and slowly moves away from the conversation!)

Units: Size

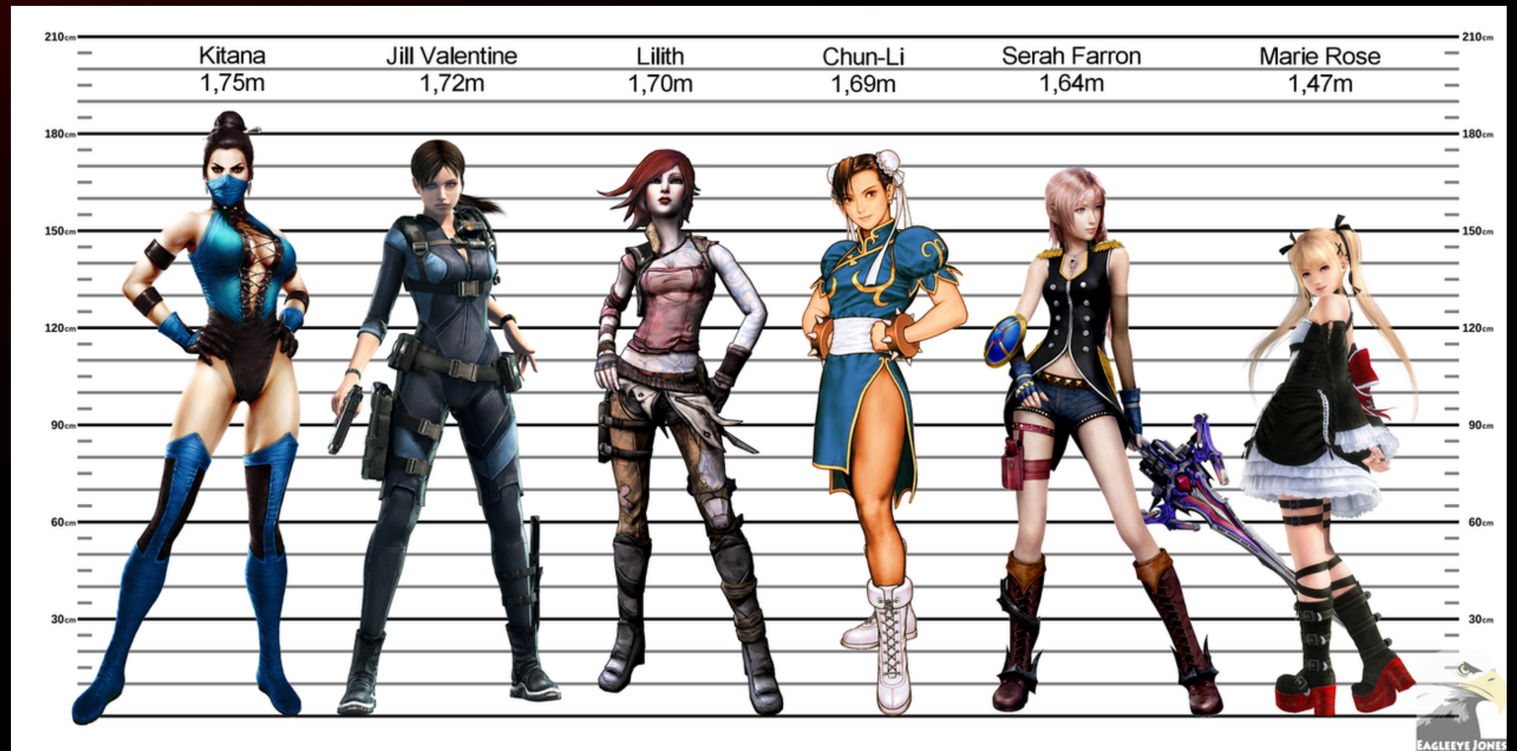
2. Art Specifications

Units usually refer to Physics: Size, Speed, Weight, Velocity, Acceleration, Force

Setting Units to Real World Unit Measurements: Meters, M/s, etc… Everyone will understand better:

Size: 1 Unit = 1 Meter

Speed: 1 Unit = 1 M/s

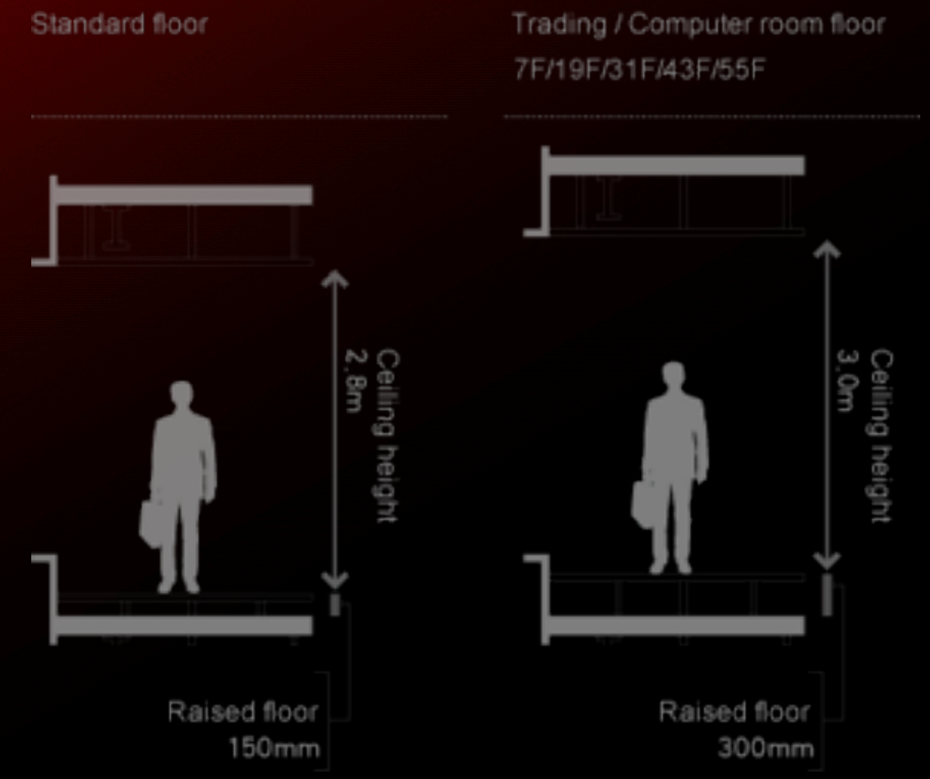


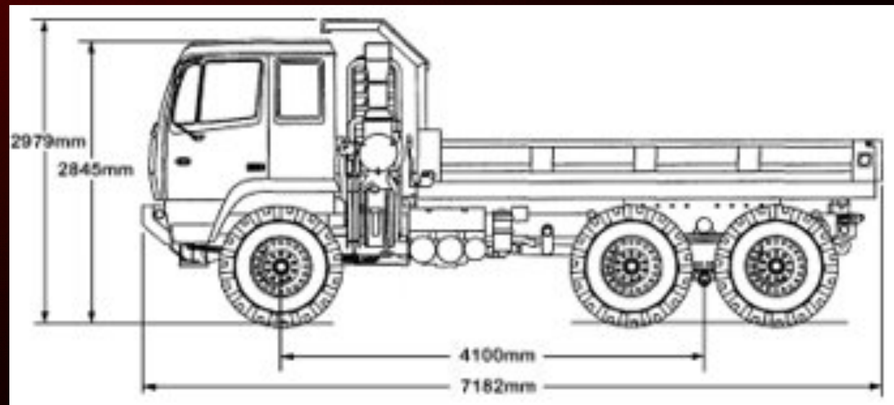
Units: Size

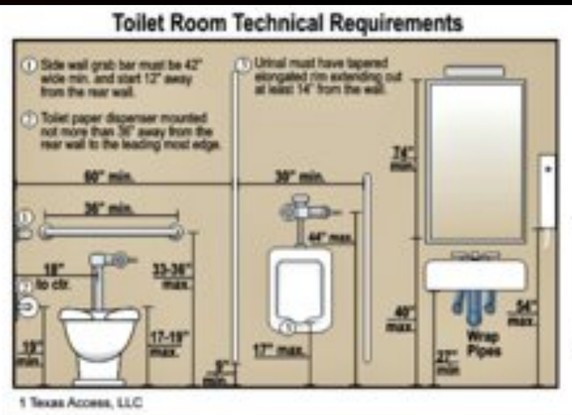
2. Art Specifications

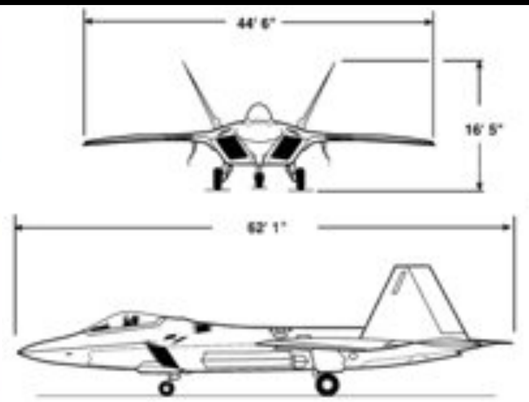
Project 2: this will help with Basic proportions.

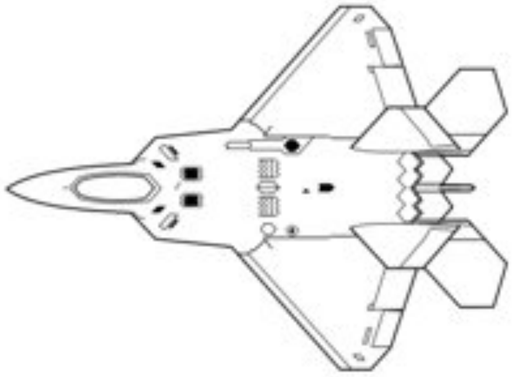
3D: Unit scales are essential!











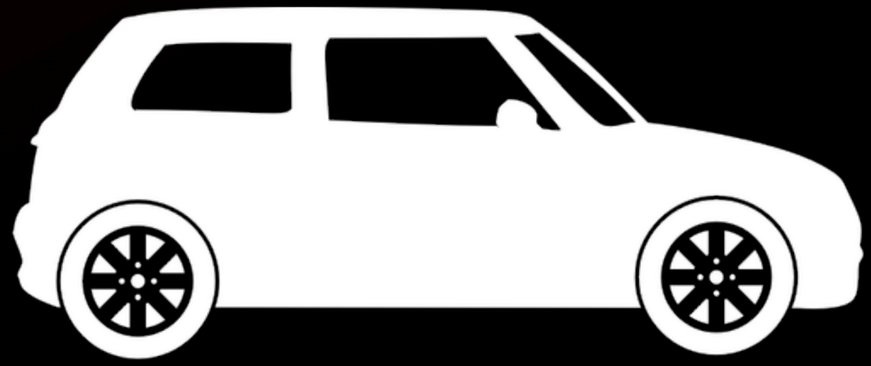
Units: Speed

2. Art Specifications

Why? Sizes and speeds for everything can be googled, thus better imagined!

1.4 units (M/s)

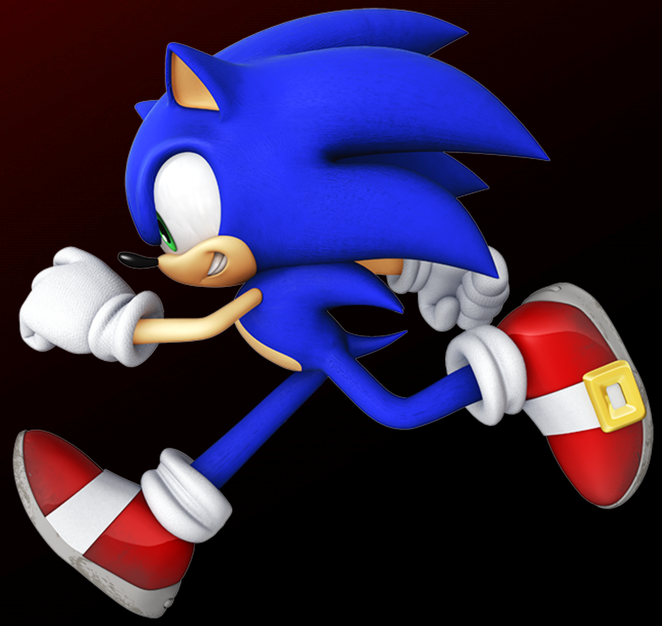
8 units (M/s)



55 units (M/s)

Units

2. Art Specifications



Do I have to follow Real world measurements?

NO!

But it’s a good starting reference!

Texture Sizes

3. Target Resolutions and Aspect Ratios

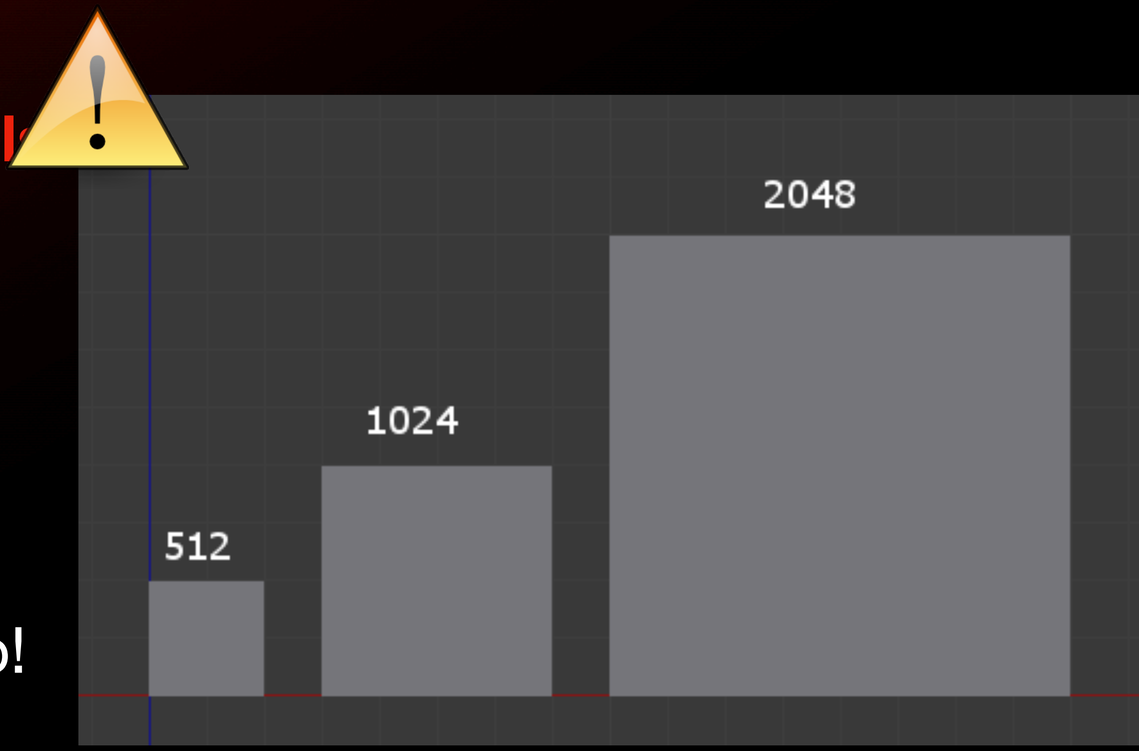
Author your textures to the power of 2 Pixels

2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048

Textures do not need to be square:

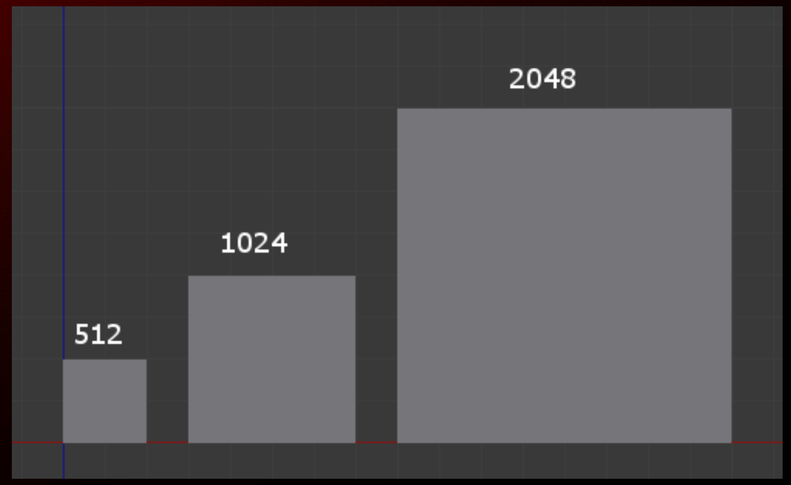
Rectangular is Okay!

512x128, 256x1024, etc.. Is also power of two!



Texture Sizes

3. Target Resolutions and Aspect Ratios



DO NOT!

Arbitrary sizes!

487x678, 198x198, 568x1207, etc…

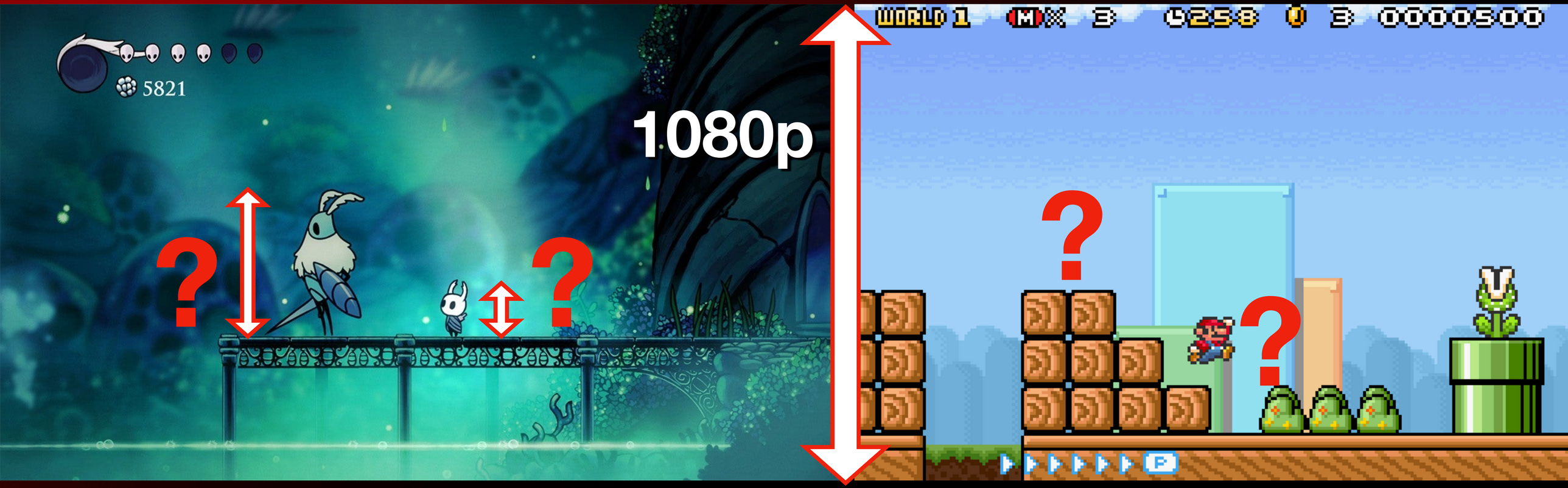
WHY?

[Memory will automatically allocated to the next Power of two]

Computers will hate you … and summon Elie :O

Sizing your Art

3. Target Resolutions and Aspect Ratios



Define your max-screen resolution (eg.1080p / 1920x1080 for P2)

Author and import your art at this resolution. (Vector export to target resolution)

Sizing your Art

3. Target Resolutions and Aspect Ratios

WHY?

Creating Oversized Art: invites detail work!

Wastes your time, Memory, Load Times and Gpu time!

Up or Downscaling 2d art creates artefacts: Blur and loss of detail!

Important for Fonts as well!

Vector vs Pixel Art

3. Target Resolutions and Aspect Ratios

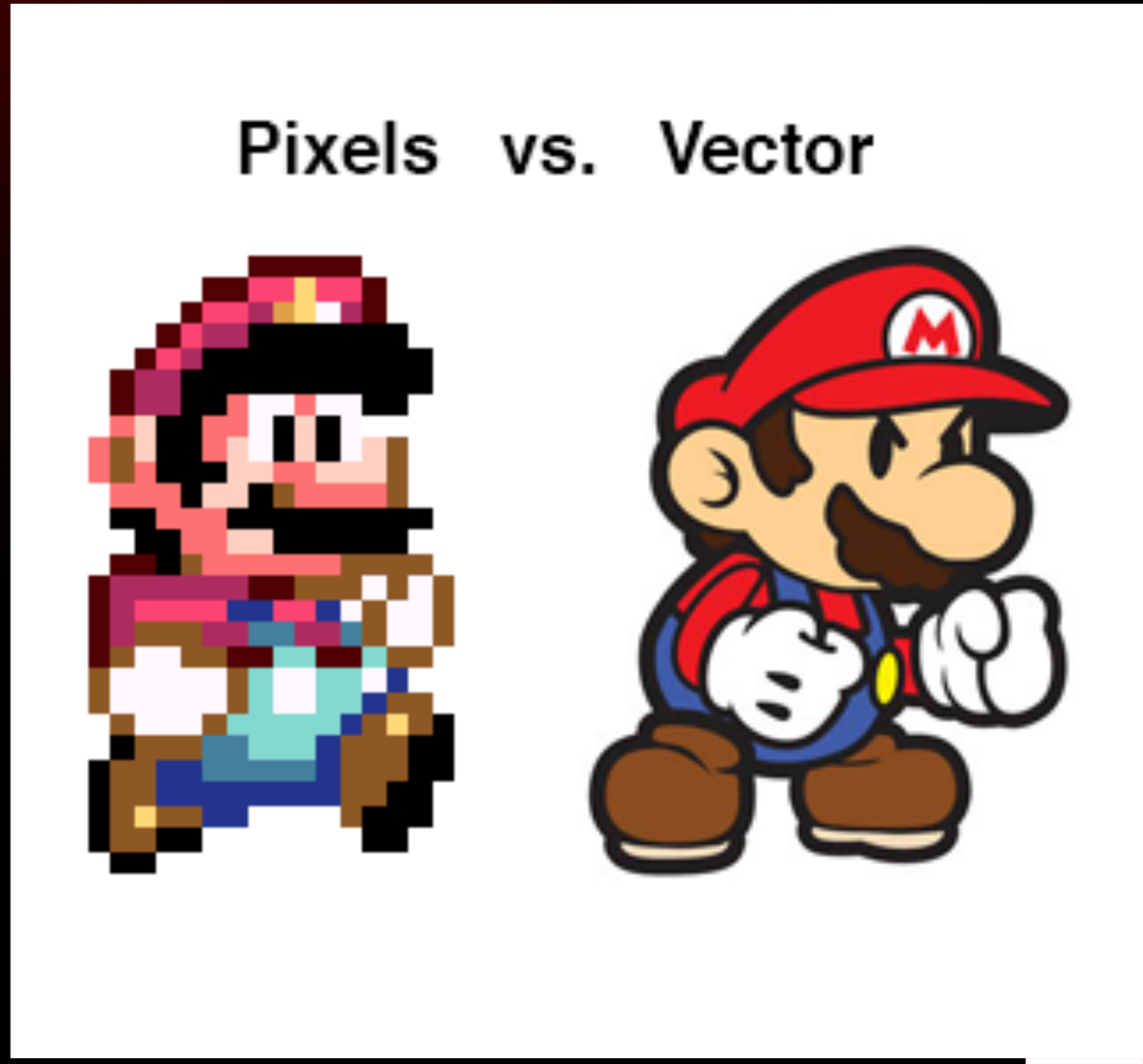
Consider!

Vector art is resolution independent, hence popular.

Vector Art has a *steeper leaning curve but many advantages!*

Pixel Art is popular and easy and fast to get into!

Changes or fixes to Pixel Art are *time consuming and painful!*



Level and Spritesheets

4. Importing Texture Assets

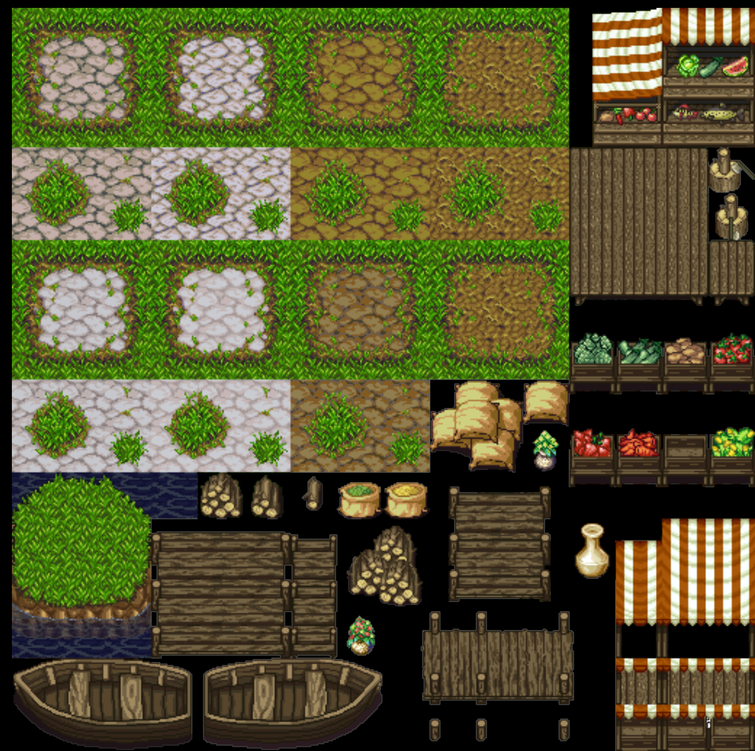
Try to create Level specific sheets to batch into texture atlases!

Pack your Textures! Do not Leave empty spaces!

Sheets can share different Characters etc… Pack them up!

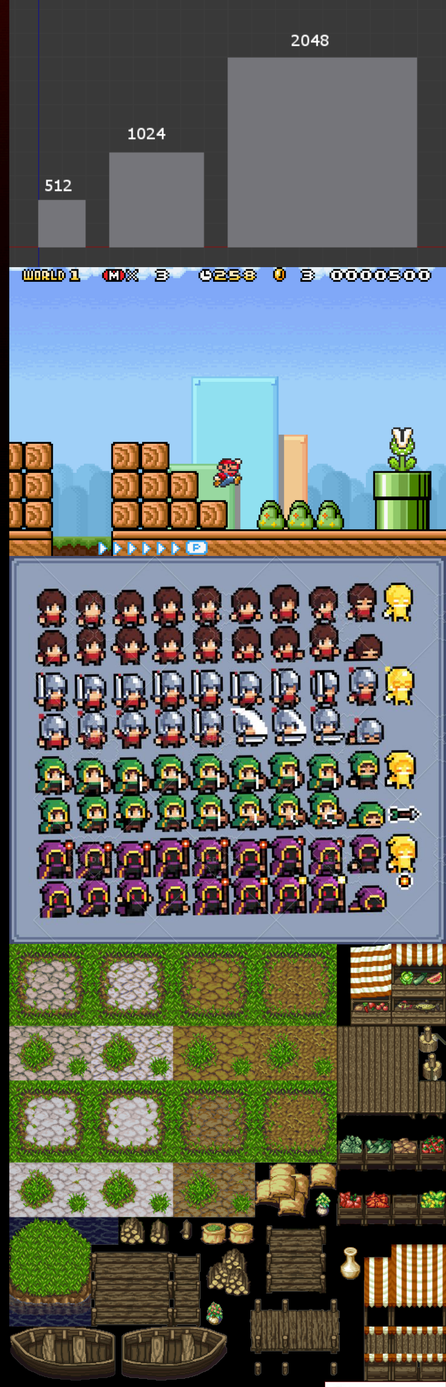
Again, ***import*** at intended target resolution! Pixel Perfect!

These do not have to be square, as long as Power of 2 pixel dimensions: 512x1024 etc…!



Texture Summary

4. Importing Texture Assets



- Size up Tiles and Sprites based on target screen resolution. Decide how large the texture should be.
- Power of 2!
- For Tiles and Monsters create level specific sheets as they are not always shared between levels.
- For Sprite Sheets remove all translations. (Jumping etc. See 5. Animations)

First things First!

5. Animations

SPECIFY YOUR FRAMERATES !!!

Agree with your Techies on the preferred FPS the engine will render your animations at.

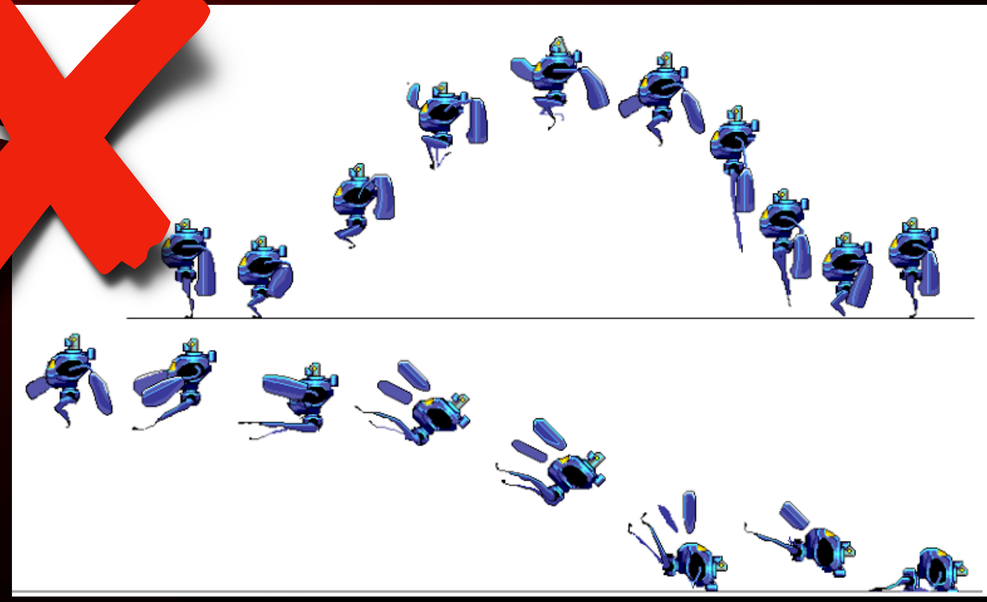
ENSURE to setting your TOOL to the same FPS.

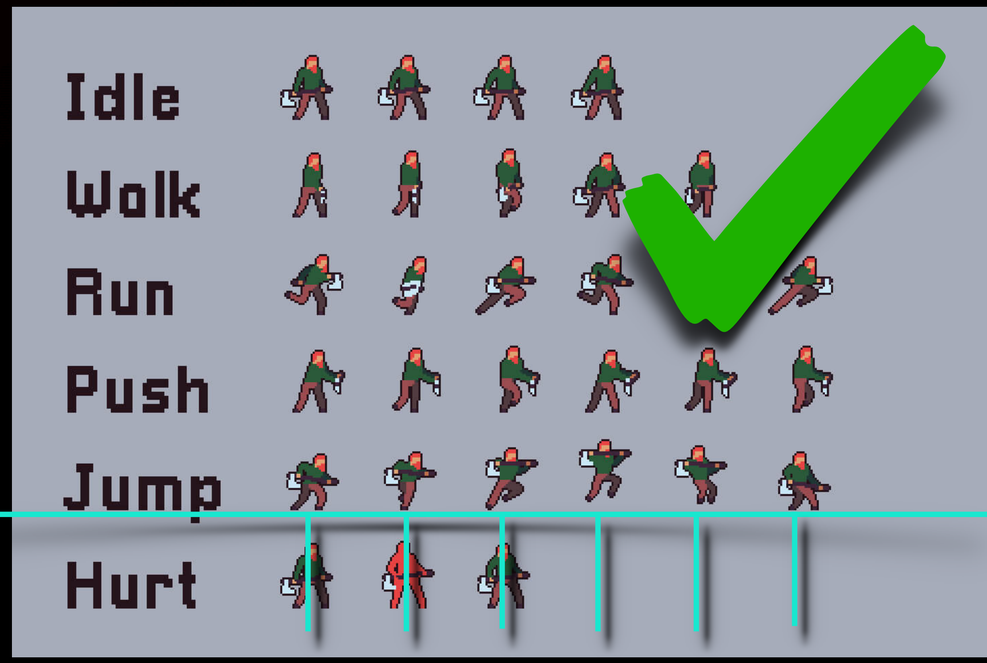
Recommended animation rate is 30. OR: multiples of 10 for less smooth animations but less frames to author.

*If you haven’t done so: it’s not too late!

Sprite Import considerations

5. Animations





Remove all Translations when placing sprites on a sheet.

Create image guides marking position 0,0 and center your sprites there.

Translations are handled by code!

Built in translations prohibit dynamic jump heights or lengths.

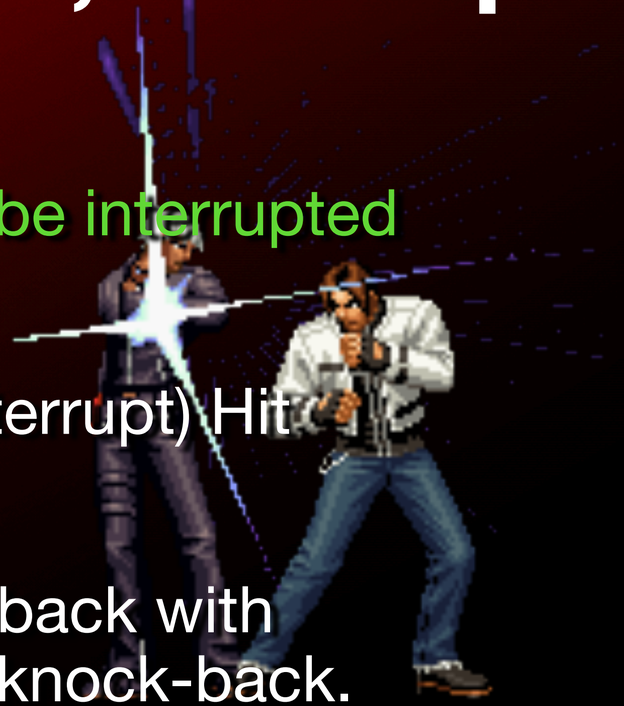
Animation tree, Pauses, Interrupts

5. Animations

Define ***which*** and ***how*** animations can be interrupted and paused.

Examples: Running along getting hit (interrupt) Hit animation follows.

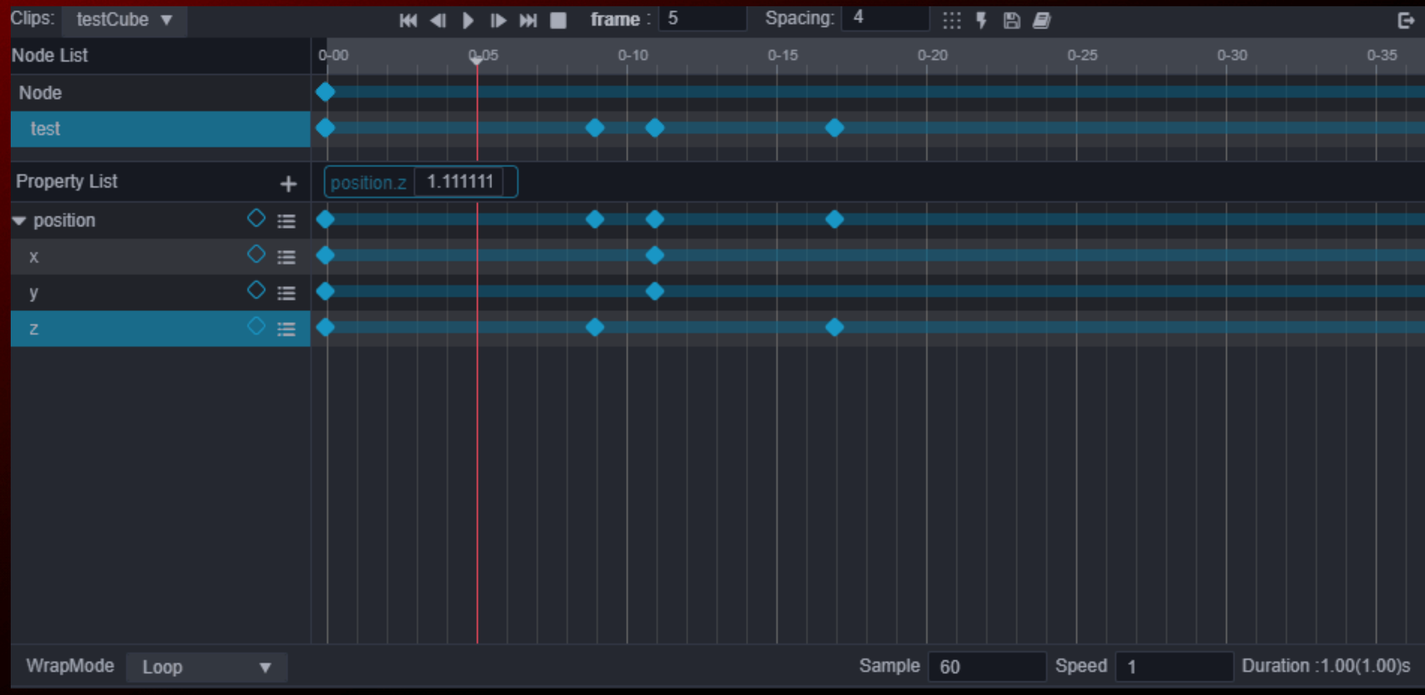
OR: Player character incurring a knock-back with animation pause for the duration of the knock-back. (Which frame to hold and for how long?)



Design, plan, and incorporate RULES specific to your game design into your engine.

Keyframe Editors

5. Animations



It’s highly recommended that your engine has an Keyframe editor!

Keyframe editors **SAVE** lots of time!

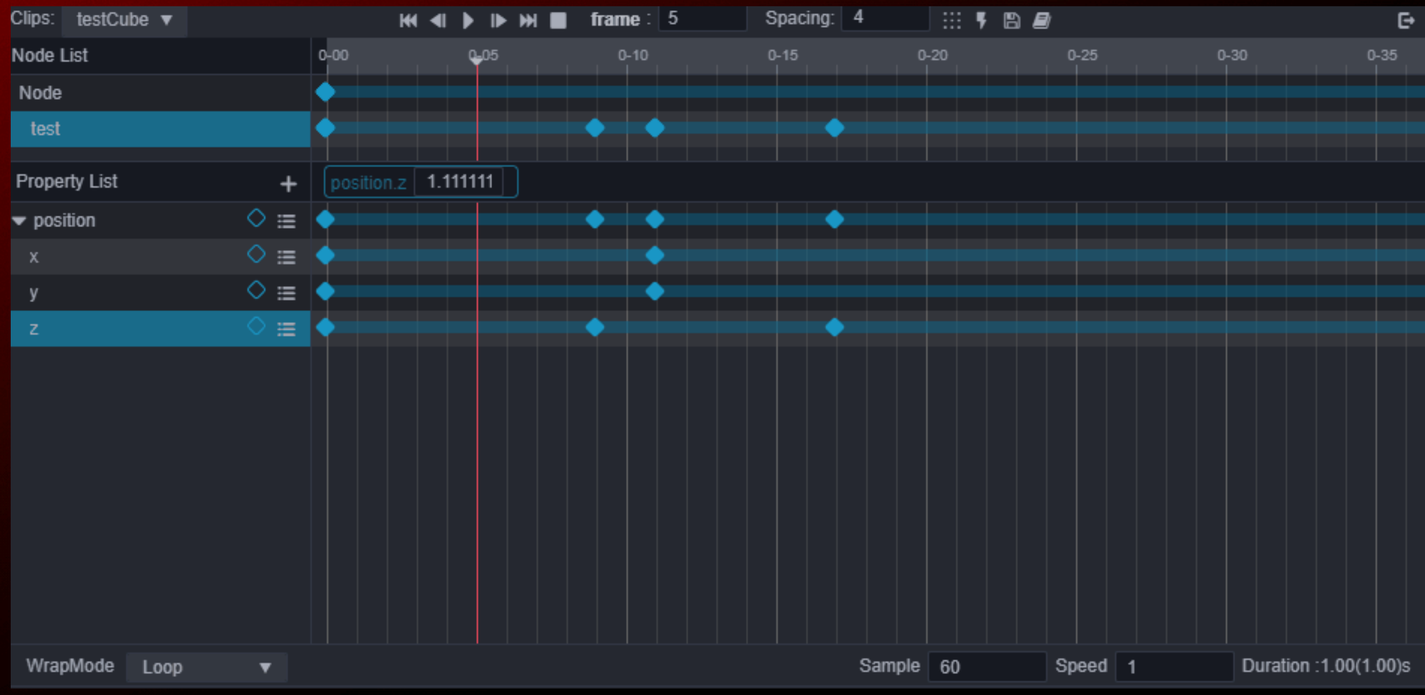
Anyone can use & offload tasks from the programmers!

Use these for Character & UI animations, SFX and Particle systems and Sound and VFX triggers within animations.

Be sure to have an option for your **animation rate settings** as well as switching between fps and seconds is quality of life! (Remember Unit scales?)

Keyframe Editors

5. Animations



Features to think about:

Tagging of loop-able animations

Fps/Seconds toggle

Clips for compound animations such as jumping or falling.

Physics triggers (variable knockmbacks etc..)

Checklist / Testing

5. Animations

- Designers/ Artists: Do check that your animation for timing and looks the same in game as in your animation tool.
- If your game requires jumps and falls, please consult as these animations *may* need to be broken up into sections!
- Ensure Interupts and compound animations are working correctly.
- Physics driven interrupts and jumps are working.
- Problems? Check your work first! Then go ask team and find out immediately!
- Problems will multiply !!!

When to start planning and integrating.

6. Particle Systems

Almost all games exhibit the use of particle systems. Particle systems crucial in visually underlining player feedback.

Although considered ‘polish’ at DigiPen. Do not start integration at the 11th hour. They are integral to player feedback and quality feeling of your creation.

FIRST: ***Create a ‘shopping list’ of what you need. Forecast Game feedback.***

List down VFX with asset sharing in mind! Not everything needs to be unique.

Do not Leave planning too late, there a quite a few considerations to be taken care of which can greatly enhance the feedback and feeling of your game

Components

6. Particle Systems

Emitters: Emitters control position, shape and direction.

Particle: Your sprite, or sprite page

Lifetime parameters: Colour Change, Translations and Transparency.

Time controller: Timing of Lifetime parameters. (Remember the keyframe editor)

*Physics: Do particles collide? Can they bounce?

***Mixing Physics with Particles is a Framerate Assassin!**

Wiki it!

7. Communication and setting priorities

Create a wiki for established Art and Import Rules! Not much work, Update as needed! **Beats Discord!**

Keep it up to date!

Tip: When updating, Email and link changes to your team.

Tip: Tag wiki entries to the person entering it. Any questions, you know how to ask!

Team Communication

7. Communication and setting priorities

Project managers, Coders and Designers/Artists **must** agree on feasibility!

Scary Can’t be done? Say so and kill the feature.

As new asset / feature updates go into engine, *vet* your changes. Designer/artist to programmer. Or vice versa.

Test, test. test. Play to break the game.

That’s All!

Questions?

If unsure about any of these points, please consult as you develop your Game!

Thank you.