# Art Bible


Heriberto Calderon, Maxime Gautier, Eva Khoury, Reuben Heatley Mulhall, Andrew Rogers, Tori Rossini, Trisha Surve

# Main Comps

Race the Sun is an endless, lane-less runner in which the player character is solar-powered. In addition to dodging obstacles, the player extends their run time by collecting pickups that reverse the sunset.


Neon Drive is ***aesthetically*** what we are going for. Disregard the actual gameplay and cityscapes, but take a look at this game for the colors and the aesthetic. The way the cars are designed in this game.


Super Sonic Surge is a mobile spinoff of Race the Sun, without the sun energy mechanic. Instead, it favors constantly high speeds and automatic shooting. It also added cosmetics, missions, and leaderboards.

# Look/Genre Inspiration

The look for CyberDrive is synthwave/outrun, with cyberpunk themes. It takes a selective, retro futuristic take on 80's sci-fi, with inspirations in movies like Tron and Blade Runner.

Specifically, we are leaning towards a red/blue contrast in dark settings. The colors should be high contrast and high intensity, like a nighttime getaway chase within a virtual world.


# Story see: Full Story

**Logline:**

- Set in a future dystopia, the Hacker (player) is uploaded into the System, a sentient and hostile computer superintelligence, with the mission to destroy as much of it as possible, sector by sector.
  - Between each sector, the player destroys a “firewall” (With the possibility for some shooting, pickups/rewards, fx, and a brief rest period)
  - The Hacker is represented by a digital vehicle/avatar that is powered by the electricity running through the System
    - The electricity constantly accumulates as the Hacker moves through the System. However, the Hacker needs to periodically release this electricity at certain points to avoid overcharging, (and thus, losing control of the vehicle)
  - Everything shown in gameplay is within The System
    - Menus, title screens, etc. may hint at the world outside of the system/allude to computer or TV monitors

# Mood, Theme, and Target Emotions

- **Tension, adrenaline, high stakes**
  - We want to keep people on their toes, as the story involves some high stakes: (life-or-death of the player, rescuing a loved one, and potentially saving the world! No biggie…)
  - The environment/levels are set within a hostile, digital world. The style should reflect this
    - The mood has to match a good balance of **dark/gritty**, glitzy, and **intense**
  - See: story
- "Flow"
  - However, nothing should be too flashy or distracting so that people can zone into a focused, feel-good state
    - A balance should be struck between monotony and variety- test for art!
- Nostalgia/Retrofuturistic coolness
  - A new, stylish spin on the 80's projections of the future/sci-fi
  - Aiming to please the cool-kid, vaporwave-listening, thrift-store-shopping youth with popular retrowave aesthetics, VHS distortion, pinky color palettes, etc.
    - It's all about that good kitsch
  - Aiming to please the people who were actually there for the 80's by pulling references from actual products and media popular during the time
    - Make it corny in familiar ways
- Cyberpunk is inherently **gritty, political**, tied to societal and technological **fears**
  - See: cyberpunk ramblings at the end of this document

# Color Palette

| rgb: 255, 34, 51 | rgb: 221, 0, 204 | rgb: 34, 0, 68 | rgb: 17, 17, 238 | rgb: 0, 136, 255 |
|---|---|---|---|---|

Dark background with bright, vibrant colors and their lighter/darker versions- taking advantage of color vibration. Neon, nighttime, digital. Generally high saturation colors, high contrast.

Red/magenta/Blue main,
cyan/magenta secondary,
seafoam/orange yellow tertiary

# Composition/ design

Guidelines

Mockups

# Video/Music Inspiration



# UI Design

SEE: UI

Theme/mood/style:

- Semi-minimalist/small/thin- lined
- 80's interface
  - Car dashboards
  - VHS tapes
  - Old video/TV logos, animations, etc.






WWF

SEGA GENESIS

TERMINATOR

KRULL

VAN HALEN

XEVIOUS

LAZER TAG

TRON

MAD MAX

MASTERS OF THE UNIVERSE

THUNDERCATS

THE TRANSFORMERS GENERATION ONE

MORE THAN MEETS THE EYE!

BACK TO THE FUTURE

STAR WARS RETURN OF THE JEDI

NINTENDO POWER

VCA Pictures

SPY HUNTER

SUNSOFT

LICENSED BY

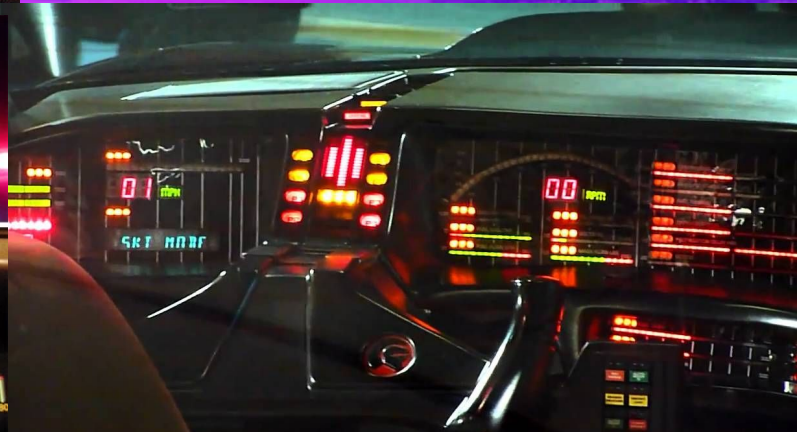
NINTENDO OF AMERICA INC.





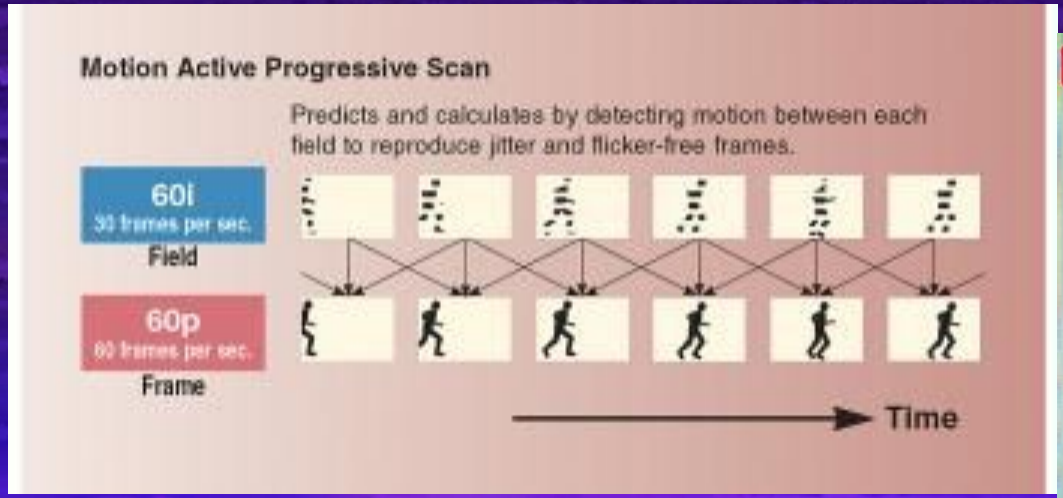




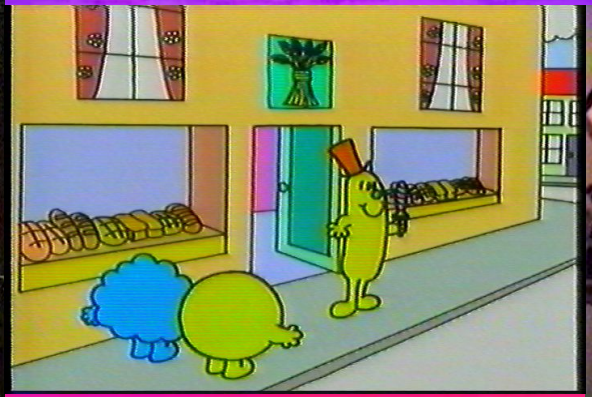



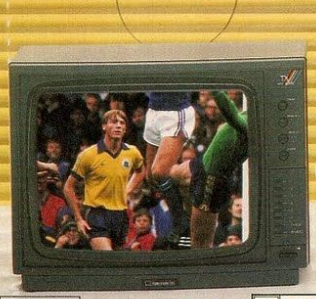

UI Inspiration for in-game HUD

UI/VFX for
title screen,
main menu,
pause screen,
end screen







14 INCH MODEL

COLOUR TV/COMPUTER MONITOR RGB

FERGUSON TX

14 INCH MODEL WITH REMOTE CONTROL

**6. Ferguson MC-01 Monitor Colour T.V.** 14" (V34cm) model. 8 pre-selectable channel buttons. RGB and composite video input sockets (both with audio) for improved sound and picture performance with home computers and video recorders. Also fitted with standard co-axial aerial socket and loop aerial for normal T.V. reception. Automatic switching allows home computer etc., to be permanently connected. Headphone socket. 240V A.C. mains.

**Cat. No.** 530/3159 — **Our Price** £219.95

**7. Ferguson 14C2 Remote Control Colour T.V.** 14" (V34cm) model. Up to date monitor styling. Infra-red remote control for 8 channel direct access. LED channel indication, electronic tuning and headphone socket. Loop aerial. 240V A.C. mains. Remote control unit supplied with batteries.

**Cat. No.** 530/3290 — **Our Price** £199.95

**8. Fidelity F14R Remote Control Colour T.V.** 14" (V34cm) model. Automatic station search. Slimline infra-red remote control for 8 channel access. Carrying handle. 240V A.C. mains. Remote control unit operates from 1 x PP3S battery (order 1 of 980/1042 at 95p).

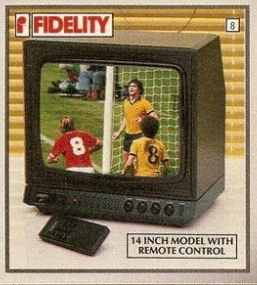
**Cat. No.** 530/3283 — **Our Price** £169.95

**9. Ferguson 37140 Portable Colour T.V.** 14" (V34cm) model. 8 channel selectors with VHS synchronisation. Loop aerial. Lightweight with carry grips in base. Headphone socket. 240V A.C. mains.

**Cat. No.** 530/2813 — **Our Price** £167.95

**10. Ferguson 16A2C Remote Control Portable Colour T.V.** 16" (V38cm) model. Slimline infra-red remote control for 8 channel direct access, volume control, sound mute and standby on/off. VHS synchronisation. LED channel indicator. Integral carry grips. Built-in aerial. Remote control unit supplied with batteries. 240V A.C. mains.

**Cat. No.** 530/3056 — **Our Price** £229.95



FIDELITY

14 INCH MODEL WITH REMOTE CONTROL

14 INCH MODEL

FERGUSON TX

16 INCH MODEL WITH REMOTE CONTROL



16 INCH MODEL

Ford Fiestas with Philips Stereo

PHILIPS

16 INCH MODEL WITH REMOTE CONTROL

**1. Philips 2016 Portable Colour T.V.** 16" (V38cm) model. 12 pre-selectable channel buttons. Integral loop aerial and handy side grips for easy movement. Headphone/tape recorder socket. 240V A.C. mains.

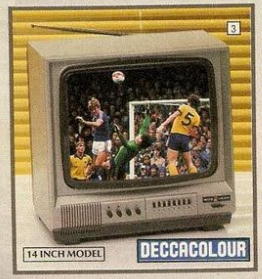
**Cat. No.** 530/3049 — **Our Price** £209.95

**2. Philips 2216 Remote Control Colour T.V.** 16" (V38cm) model. Remote control operates volume, standby on/off and 20 pre-selectable channels. Integral loop aerial, handy side grips for easy movement. Headphone/tape recorder socket. Operates from 240V A.C. mains. Remote control unit operates from 4 x LR03 batteries (not supplied, order 4 of 980/0153 at 50p each).

**Cat. No.** 530/3063 — **Our Price** £239.95

**Television Sizes:**

**Television sets have always been described by the nominal size of the glass tube across its greatest diagonal. The figures shown in brackets e.g. (V34cm) refer to the actual viewable picture diagonal.**



14 INCH MODEL

DECCACOLOUR

**3. Deccacolour DN16715 Portable Colour T.V.** 14" (V34cm) model. Monitor style cabinet. 8 pre-selectable channel buttons. Rotary controls for volume, colour, brightness and contrast. Telescopic aerial. 240V A.C. mains.

**Cat. No.** 530/3221 — **Our Price** £159.95

**4. Philips 1014 Portable Colour TV.** 14" (V34cm) model. Monitor styling with 12 pre-selectable channel buttons. Headphone/tape recorder socket. Loop aerial. 240V A.C. mains.

**Cat. No.** 530/3269 — **Our Price** £177.95

**5. Philips 2206 Remote Control Colour T.V.** 14" (V34cm) model. Features 20 pre-selectable channels, LED channel indicator, loop aerial, retractable carrying handle. Operates from 240V A.C. mains. Remote control unit operates from 4 x LR03 batteries (not supplied, order 4 of 980/0153 at 50p each).

**Cat. No.** 530/3018 — **Our Price** £204.95

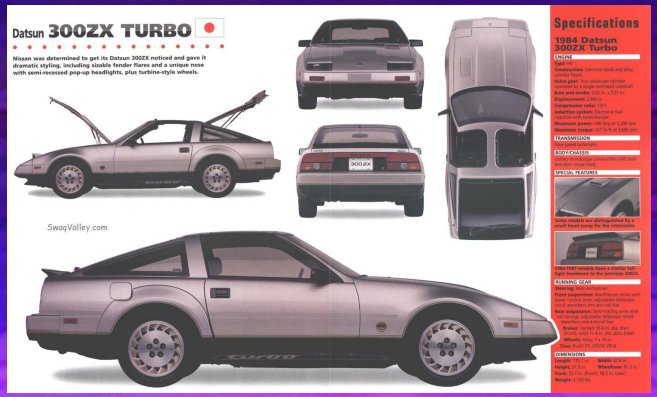
14 INCH MODEL

PHILIPS

14 INCH MODEL WITH REMOTE CONTROL

# RIVAL GEARS

## Main Comps Part 2
## Car Aesthetic.

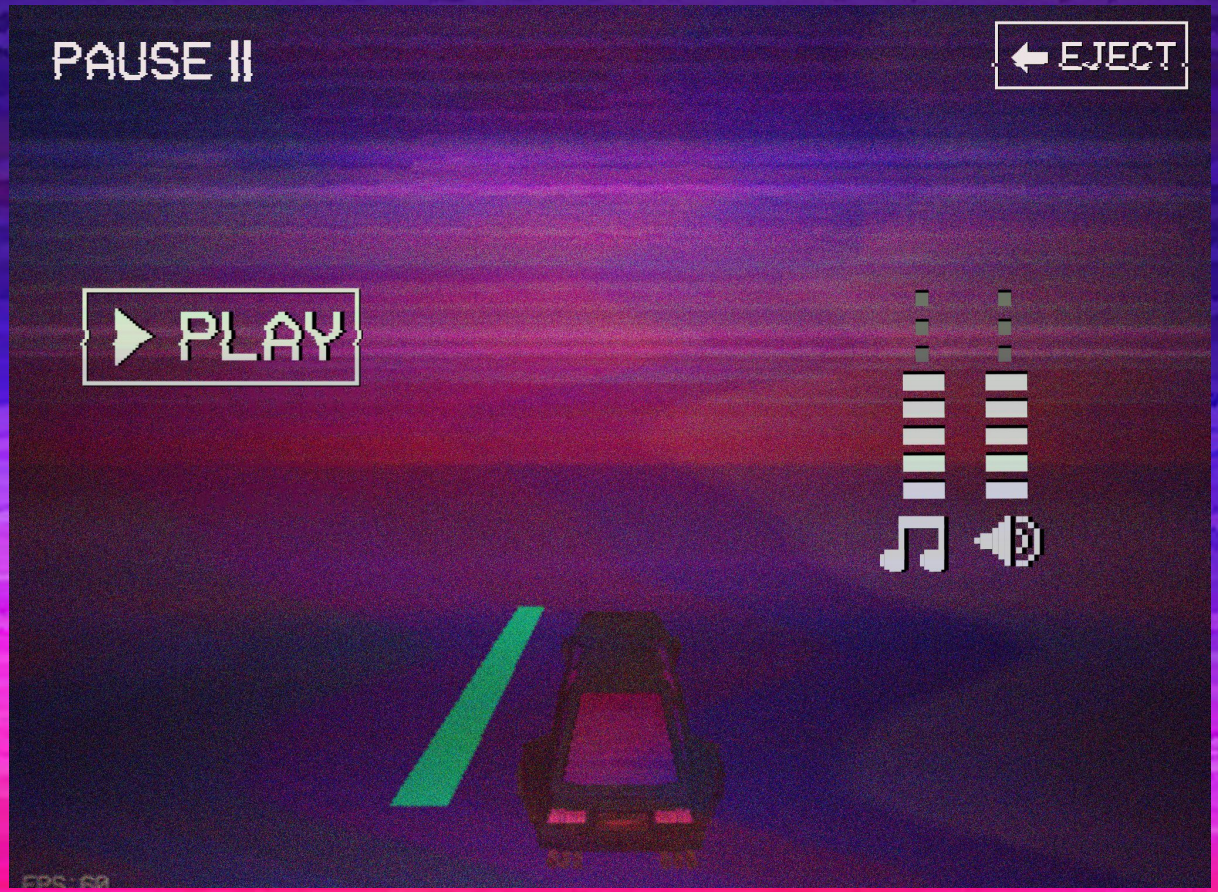




# UI: Pause Screen

Dark and glitched overlay on gameplay, eject/menu button on top (to avoid accidental clicks), buttons and sliders enlarged and placed on sides for easy touch screen use with a two-thumb hold

VHS font and icons
Can apply VHS shader with more exaggerated distortion and animation



# Pipelines & Processes

# Naming Conventions & Workflow Requirements

Naming Convention: **AssetType/Category_AssetName_Modifiers_v.###.fileformat**

File Format: FBX (Filmbox) Only
no

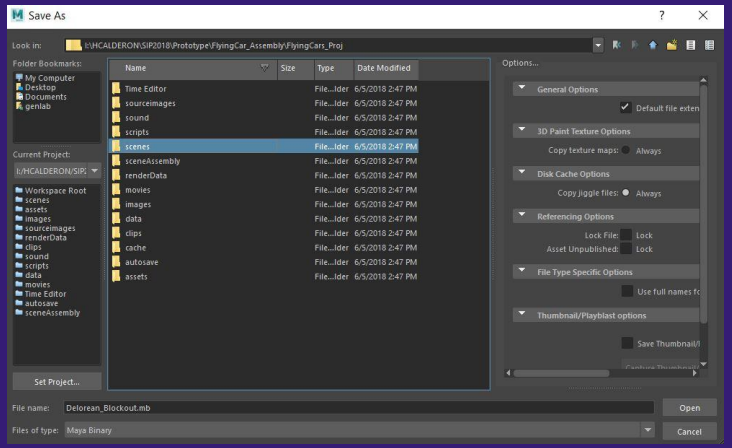
Maya Project Folder:

All maya scenes need to be properly formatted. When working on platforms, characters, assets, etc.  You need to make sure you have the project window set up.

File > Project Window > Select a location > Set project > save scene in the scene folder
** Every time you load up maya you need to SET the project to the project window. For example File > Set Project > Select Platform_Proj

Textures are saved in the “source images” folder

FBXs and OBJs are saved in the “data” folder



# Handing Off Your Work

## Good Practices

- Texture packing, if you are working with many maps such as Metallic, AO, and Normal Maps, texture packing can be very helpful.
- Texture atlas' are crucial for performance. Most of the UI will be all on one texture atlas.
- Plastic integration, make sure as an artist you use plastic to add your artwork.
- Asset packages
    - Package, plug in textures, materials, etc.  sort into the appropriate folder(s), bring the art into Unity before handing the assets to designers/programmers.

TEAMWORK!

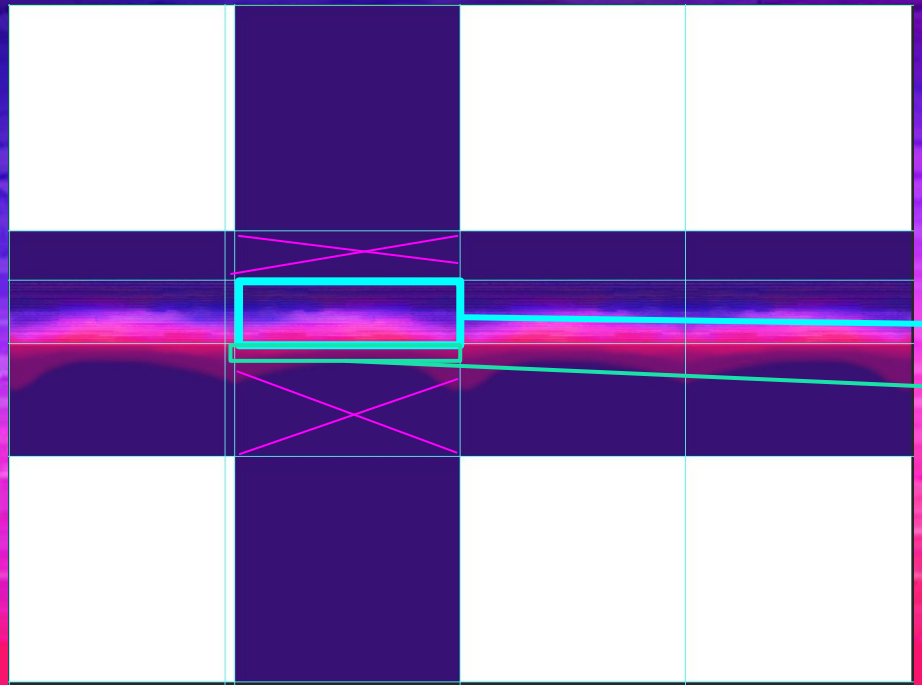
A good artist makes art, a great artist can implement it and prepare it for a production pipeline.

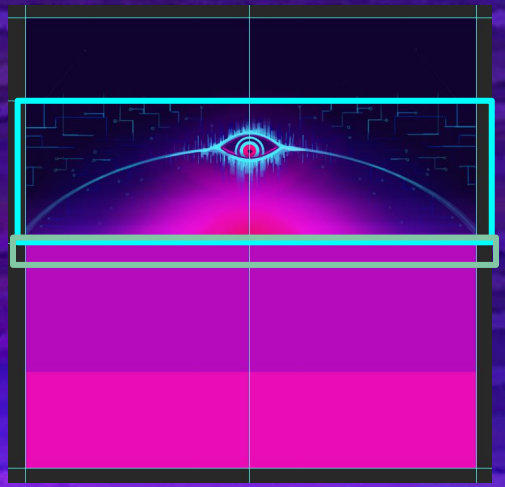
# Icon

- Apple res = 1024 square
  - Automatically rounds corners
- Android res = 512 square
  - Must manually round corners
  - Supports custom transparent shapes
- Use of red and simple, readable graphics encouraged
- Useful in creating a brand for the game
  - Since we don't have a character depicted within game, the eye symbol can work as a recognizable motif

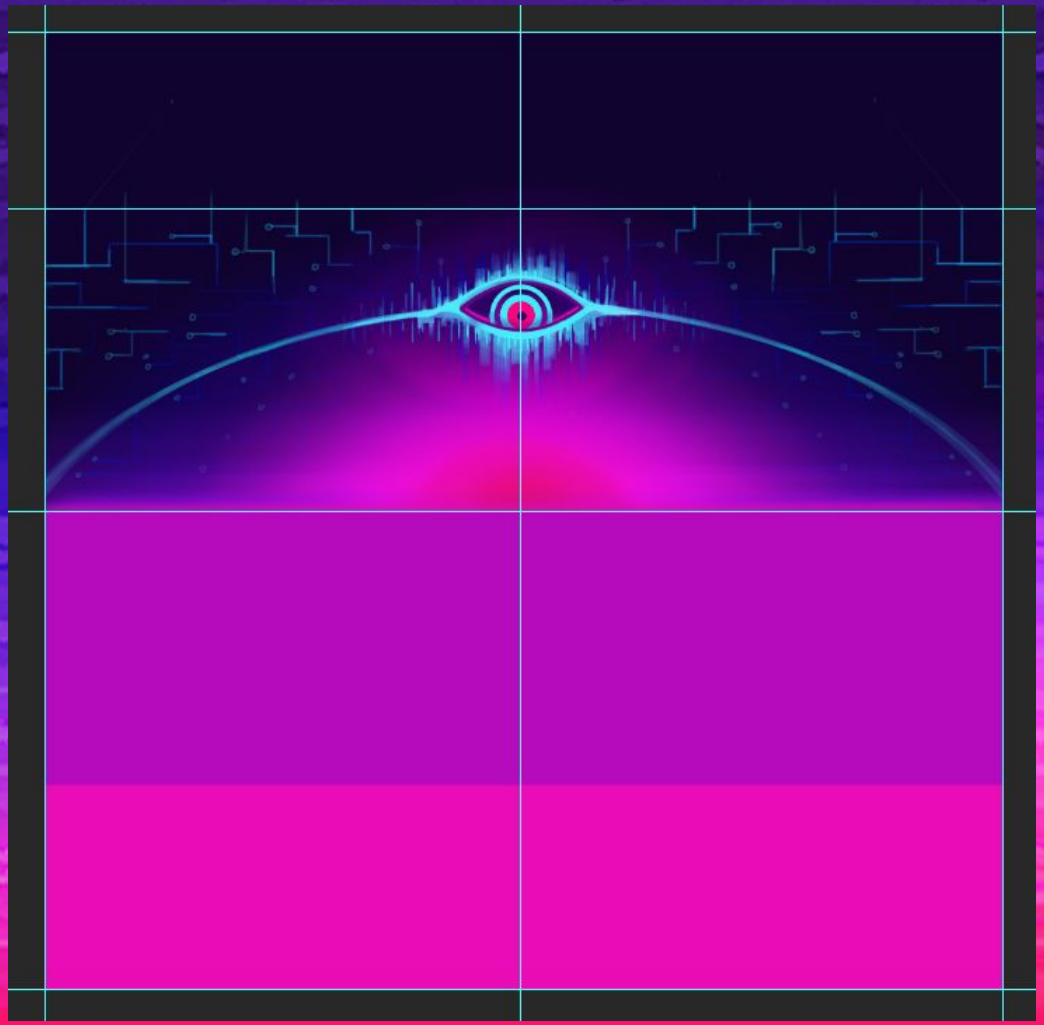
# Skybox

- A skybox is a 6-sided, infinitely large cube around the game world within Unity. In **[GAME NAME]**, we hand painted our own skybox textures in Photoshop:
  - TUTORIAL HERE
- However, that results in hard corners in the textures. To only show the front (+Z) texture, set the FOV of the skybox camera to 58.7. Make sure to test with the aspect ratios **4:3**, **16:10**, and **16:9**





The ground plane and FOV obstruct most of the skybox, so only paint within the designated section. Fill a buffer zone at the bottom with color so that it shows during turns. The other sides of the skyboxes aren't shown directly. They just contribute to light color. Copying and pasting the skybox or filling w/ solid colors will work just fine.



- The skybox is a 2048 square, with only a small part showing.
- Use colors from the color palette
- The horizon and center area should be a medium to light value and have a hue that matches the fog color
    - It should also not feature any detail so the obstacles on the horizon are clear and readable
- More detail can be placed in the blue ring area
- Some detail can be placed in the upper corners, but keep it subtle enough as not to interfere with UI
    - The top center above the eye should be kept clear for popup UI
- Keep the atmosphere dark and foreboding. This eye is hostile and watching for the protagonist.
    - Use imagery reminiscent of digital technology and glitches
- Above all, design for readability. Nothing should be too distracting.

# Alternate Color Palettes

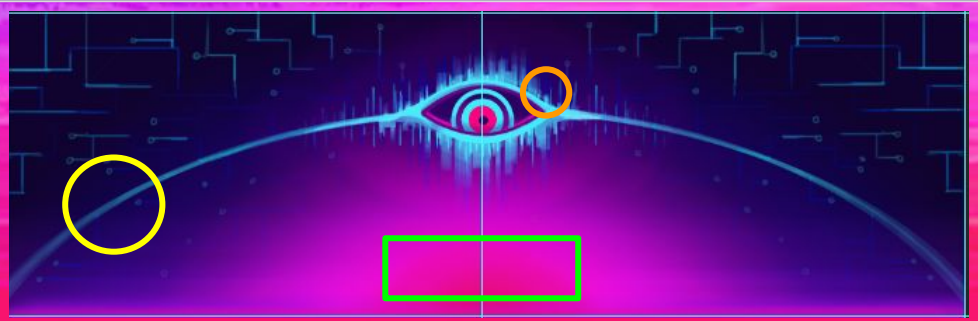
- Since our game is endless, and we can't hand-design endless content (models, level designs, etc.), we need a little help to make the sectors distinct.
- By changing the color palette of the world, we can make levels more unique. The world aspects that change are:
  - Skybox shader
    - Color: Tint
      - hexcode
    - Exposure
      - # value
  - Ground plane color (unlit color shader)
    - Hexcode
  - Glow shader (obstacle glow) color
    - Hexcode
  - Fog color
    - Hexcode

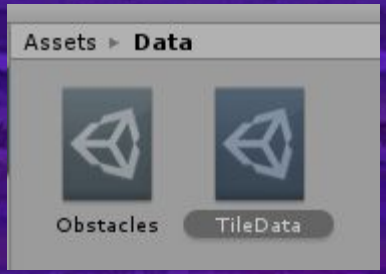
# PALETTE SPREADSHEET

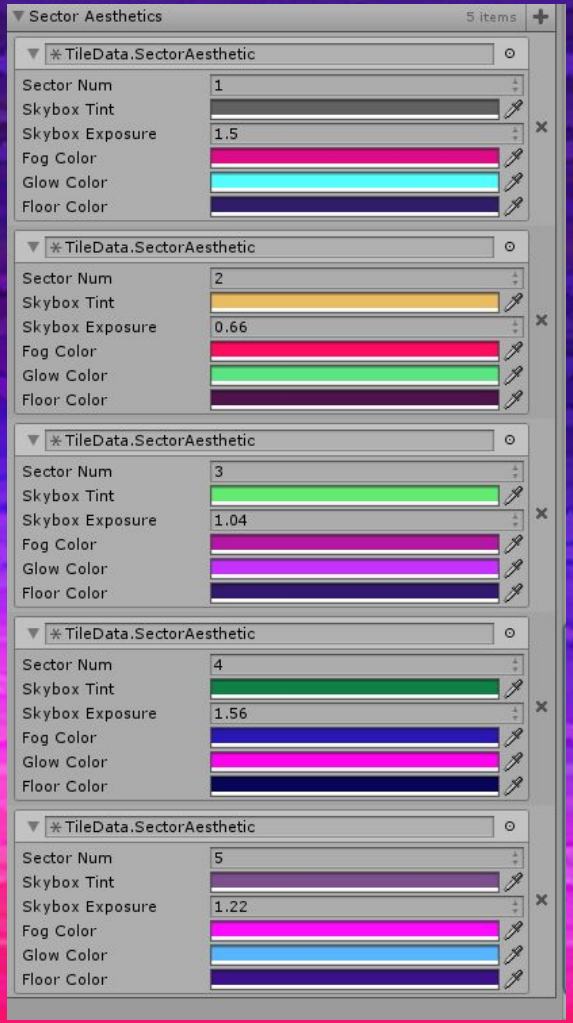
- Above is the spreadsheet for the values needed to achieve a new color palette. These should be implemented in the code.

- We need some RULES for determining this color:
  - Determine the skybox color/exposure first, then the other values by colorpicking the skybox
    - Use the unity color picker tool in the inspector as an easy way to get the hexadecimals. Click on the color box to show the color information.

  - The skybox should have a pleasing color palette through the tint
  - The skybox should be readable and not over- or under- exposed
  - The ground plane color should generally be color picked from the mid-dark background of the central ring area of the skybox.
  - The fog color should match the central horizon area.
  - The glow shader color should match the brightest part of the eye outline.
  - See below and in the spreadsheet/game for examples






The colors we picked for the nodes are all matching the color palette. We plan on testing out what colors people enjoy the most and keeping them consistent. We may also add more worlds with more colors to make the game feel truly endless. It would also be wise to change the hue/saturation of the nodes slightly even after the cycle through. So if you hit Node 2 which is purple, then the same purple node appears on Node 6, then you can up the saturation and or hue to make it “seem different” but in reality it’s the same color, just slightly varied.

# The Eye ( AI, The System)

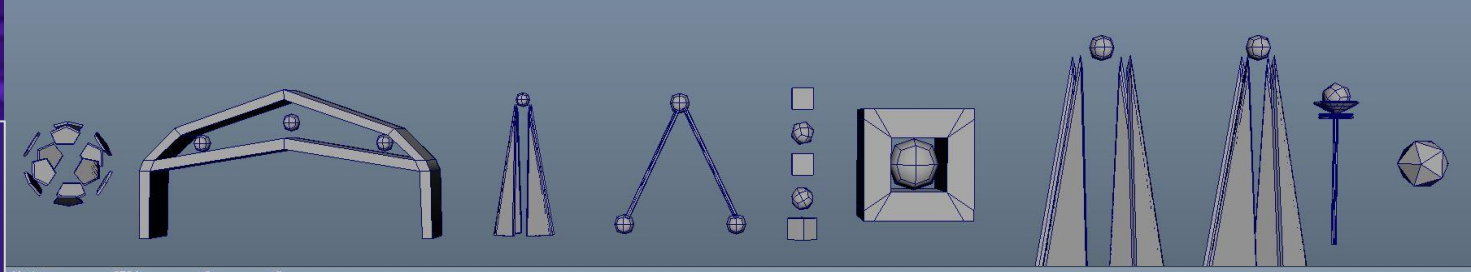
- The eye is a 3D object with the skybox mapped onto it as a texture.  It is animated in maya, baked, and exported. The animations are then added to the animation controller for the eye in unity.
- The look animation plays during jumps and between sectors. The angry vibrating animation plays during soft hits/collisions. There is plenty of room here for new animations and situations where they might be played.

# PolyCount

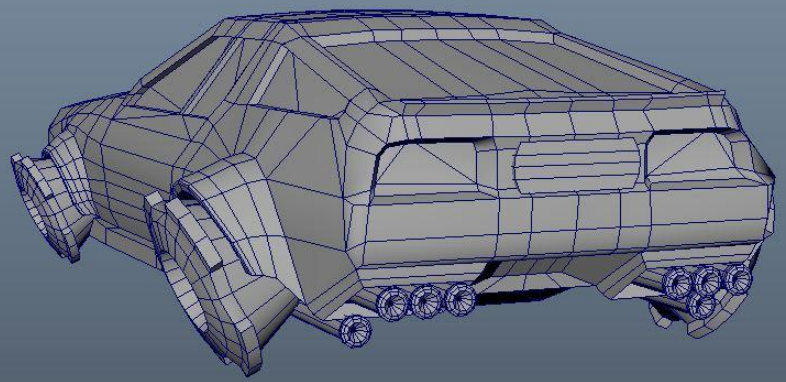
In order to keep the poly count low, you need to delete unnecessary geometry. Since most obstacles are grounded on a plane in Unity, you can chop off the bottom of the obstacles.

**Obstacles - 300 verts MAX**

**Player Type - 4000-5000 tris MAX**

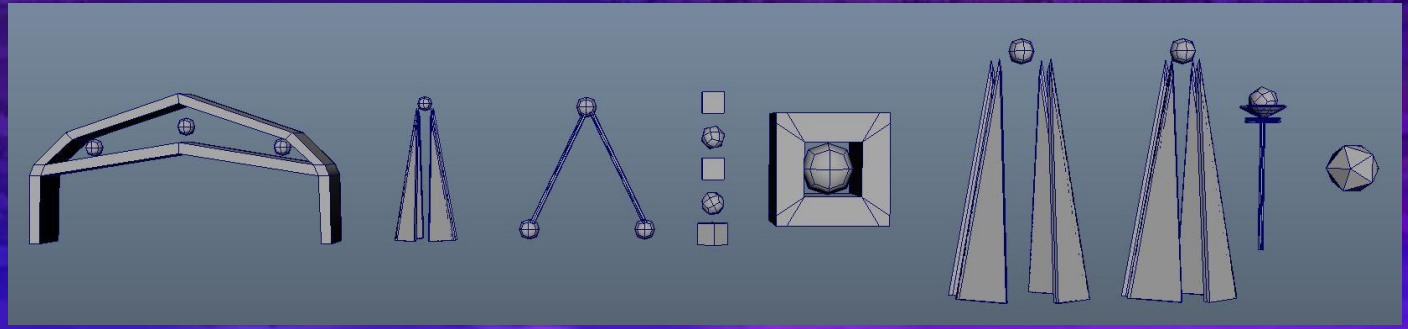


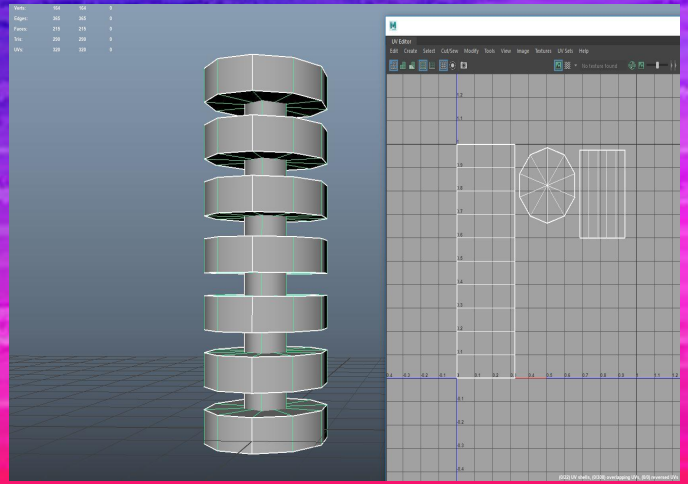
| | | | |
|---|---|---|---|
| Verts: | 2784 | 0 | 0 |
| Edges: | 5404 | 0 | 0 |
| Faces: | 2626 | 0 | 0 |
| Tris: | 5078 | 0 | 0 |
| UVs: | 3370 | 0 | 0 |



# 3D Pipeline

# IMPORTANT: design guidelines





It is important to remember that when creating obstacles you are efficient and optimizing as much as possible. Those platforms on the left are all UV'ed for recycling. You can create 3 different variation of the same obstacle. This makes it so you can texture just one of them, and apply them to any of those three obstacles.

Obstacles all **MUST** be **less than 300 vertices** in order for them to be batched. (Here's why)

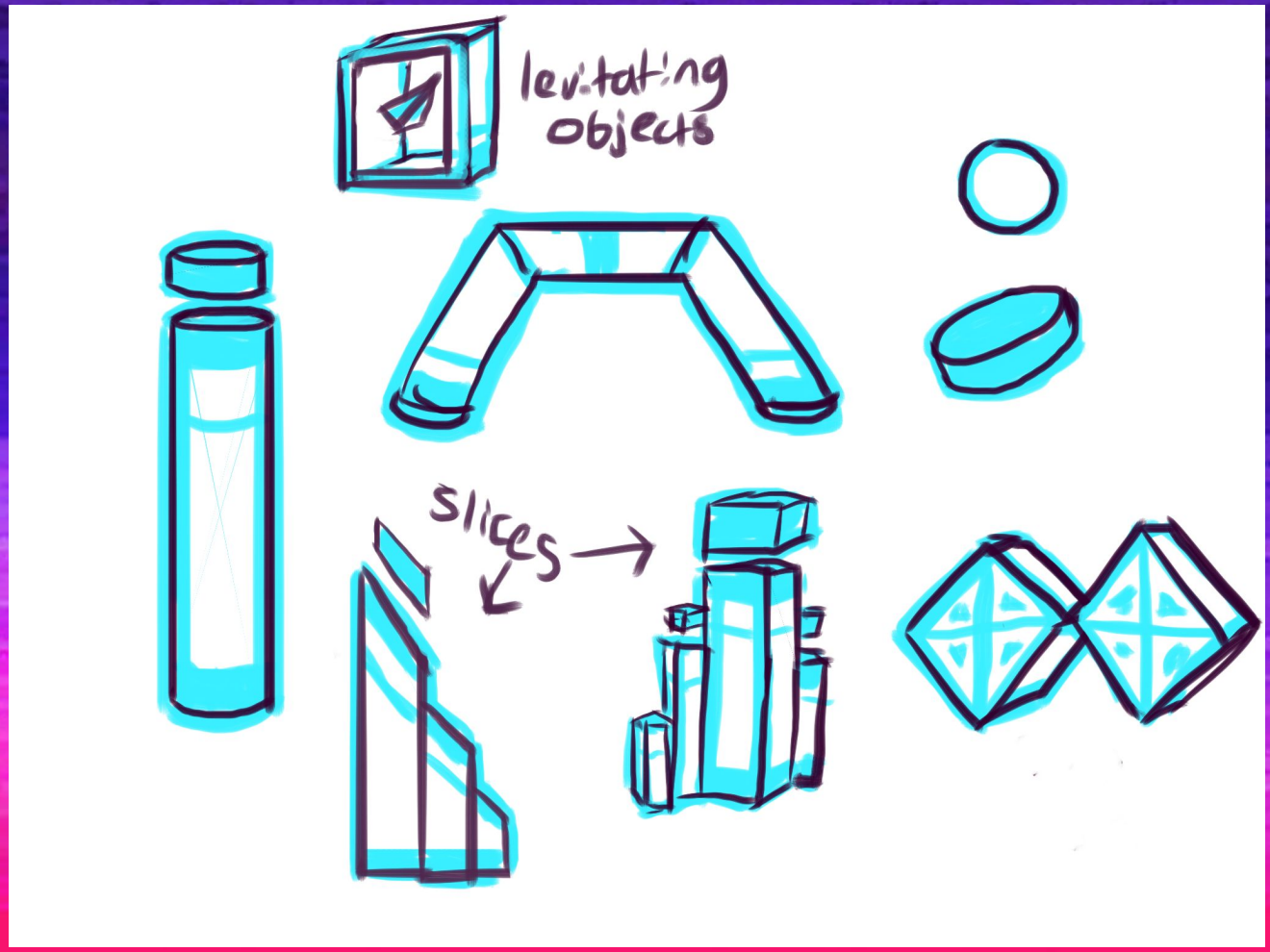
**Why this draw call can't be batched with t**

A submesh we are trying to dynamic-batch has more than 300 vertices.

# Obstacle Design

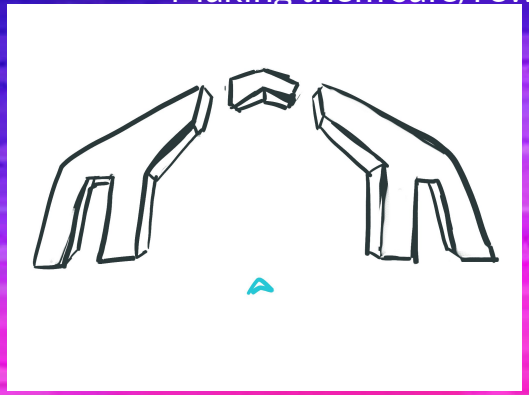
- Obstacles should be **very simple** and have **instantly readable silhouettes**
  - The most important part is the **base**, which is the area the player can collide with
    - Bases should be very emissive, static unless designed otherwise, modeled/rotated in a readable way (no large corners pointing forwards), and be a medium to large thickness for visibility
- Obstacles should be mostly geometric, with a few twists. Common shapes used in designs include rectangles, cylinders, cubes, diamonds, and pyramids.
  - Low poly spheres can be used as levitating ornaments
- All parts of obstacles should maintain a relatively large thickness for readability
  - Only exceptions are very tall objects coming to a point at the top
- Obstacles generally will have a “sliced” effect near the top, or alternatively all the way through

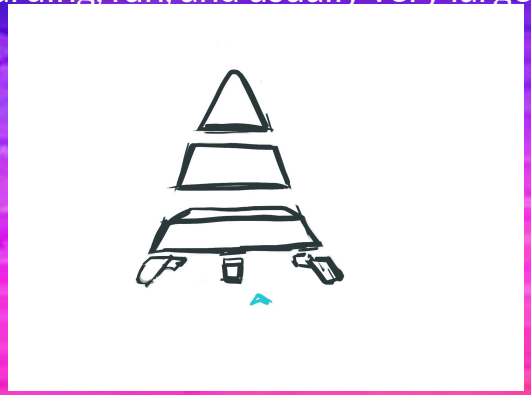
# Obstacle Design(part 1)

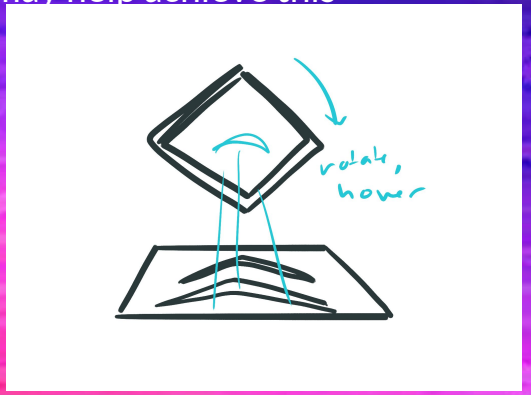


# Obstacle Design (cont.)

- Large tiles should include one or more “Landmark” obstacles
  - These should be simple, but iconic. It should have a recognizable “wow” factor
  - Making them safe/rewarding, fun, and usually very large may help achieve this

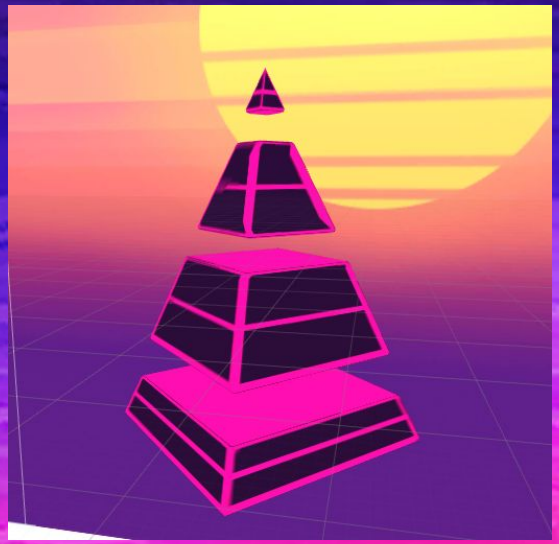






Made w/ Shader Graph

# Obstacle Texture Pipeline



- We have a custom glow shader.
  The colors can be edited per sector
  in tiledata (shown in slides 23 & 24)

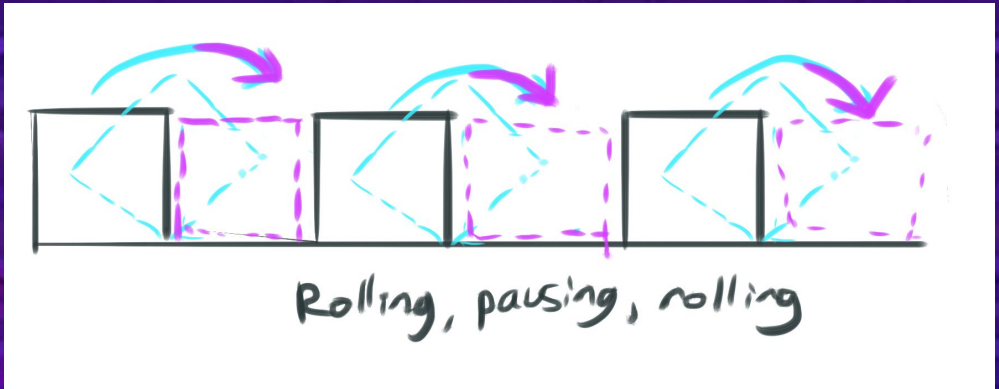
- Obstacles only use 1 texture:
  a transparent mask
  - Solid areas represent
    the outlines/emissive
    sections

# Animations

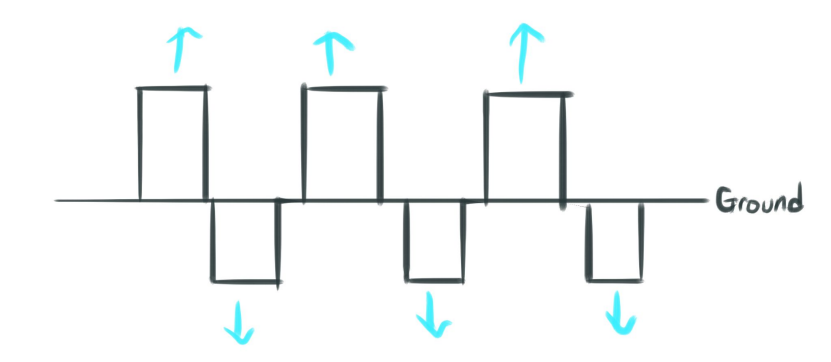
- Animations should be generally slow and predictable. They should have constant rhythms.
- Movements should be generally
  - Reminiscent of rubik's cube movements, fitting, interlocking
  - Levitating
  - Slow, large, with weight

EXAMPLES →

FULL ANIMATION LIST




Rolling cubes in a row




Alternating pop-up obstacles

# Obstacle & Animation Testing Guidelines

- As stated before, objects should be **instantly readable/visible, simple, visually pleasing, and consistent with the design of other objects.**
- The animations should be **easily understood, predictable, rhythmic, and encourage risk-taking behavior and exciting near-death moments.**
- In order to determine if your obstacles and/or animations have achieved these parameters, it is important to playtest with them.
    - Test objects within the game before committing to them
        - This can sometimes be done without texturing. However, keep in mind that textures can impact the visibility/readability. Test according to your discretion.
    - First, immediately test within your team/with yourself. You and other devs are readily available, and can sometimes spot whether obstacles work or not.
    - However, it is most important that you playtest with other people. Here are some questions to ask:
        - How did you feel when you encountered [specific obstacle]?
        - How did this level look?
        - Walk me through your experience when you played this level/encountered this area.
        - Are there any areas/obstacles that felt unfair?
        - What was your favorite part/obstacle?
        - What was your least favorite part/obstacle?
        - If you could change anything about the game/obstacles/levels, what would you change?
- Take feedback seriously. It can be easy to become attached to ideas. Some things simply don't work as planned. Always axe or edit an obstacle/animation if it is problematic in any way. It will happen.

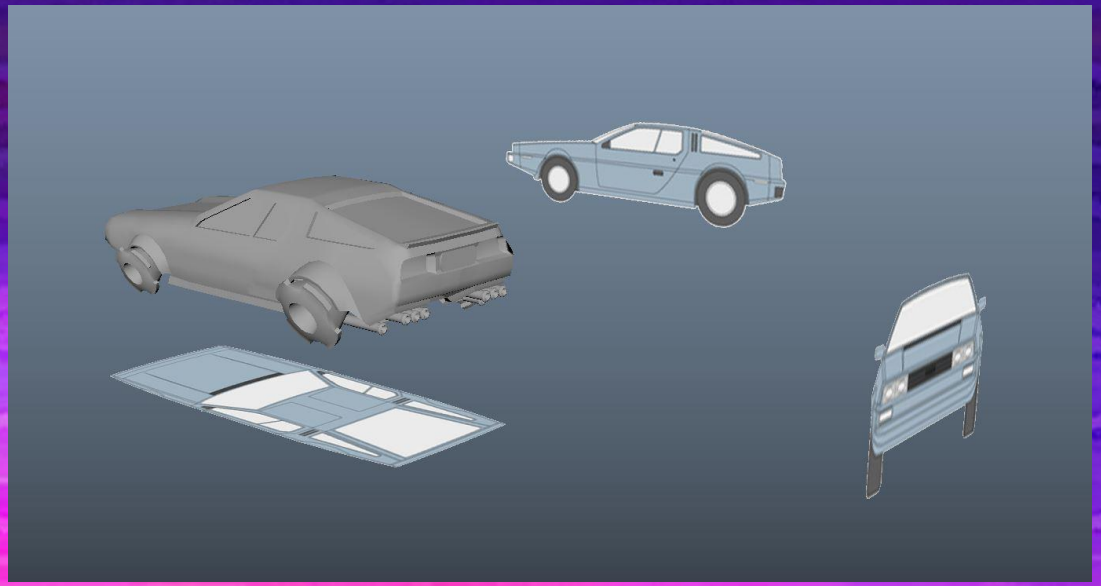
# Car Creation

When creating cars it is important to get exact blueprints of vehicles then once you block out the car and get the general design down, then you can analyze the concept art and customize that car into your own vision. Flying cars don't exist, yet, so it is important to use real blueprints.

This is also amazing. LINK

But wait…. There's more.

This is an example of how we use the blueprints. For the prototype build I used these blueprints to create the "delorean". It is important to assemble these in photoshop properly.



# Car Creation PT 2

In photoshop you need to make sure that the blueprints are properly assembled. You need to use the ruler to measure the blueprints. If you have a concept of a car that you made, you need to do this exact thing in order to build it in Maya.

Follow the step by step process here.

It would be too difficult to explain how to create vehicles in powerpoint form and or a word document so I suggest going through this tutorial series.

Keep in mind this tutorial series is to get you started. There are many things done in this video that wont help the mobile game development process, but it is to get an idea on how to get started on car creation.  Keep it LOW POLY. Around 5000 tris.

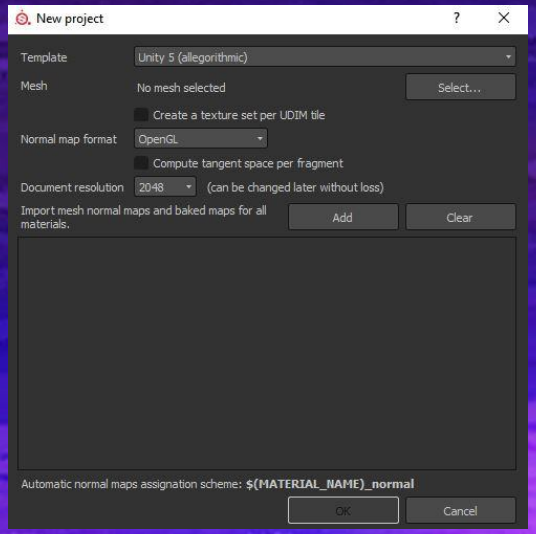
# Car Creation PT 3

Okay, now that you have the car finished. You are ready for texturing.

Here is what you need to get started on “texturing”

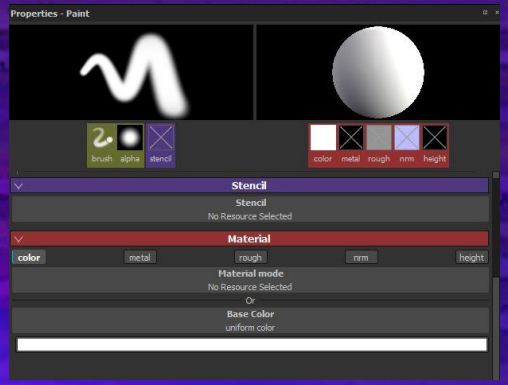
1. Car needs to be modeled, optimized (no Ngons, etc) and approved by the lead artist.
2. Once the car is approved, it needs to be UV cut. When UV mapping cars be sure to use planar mapping for UVs.
3. If the UV’s are done and approved, you need to make sure you right click on the car and assign a lambert material.
4. The big three ( Edit > Delete by Type History.. Modify > Freeze Transforms.. Modify > Center Pivot )
5. Center the car on the zero axis of Maya.
6. Export into the DATA folder within the Maya project folder.
7. Now you are ready for texturing.

# Car Creation PT 4 - Texturing in Substance Painter




No PBR textures. No normal maps. Etc

1. Import mesh
2. Make sure the textures 2048 when you work, but when you export it is 1024 x 1024
3. Open GL is the format we are using since Unity uses Open GL
4. Now that you are in Substance Painter world take a look at the channels in the fill layer. Notice you have all of them checked on. Go ahead and check everything off except for Color and emissive in some cases.



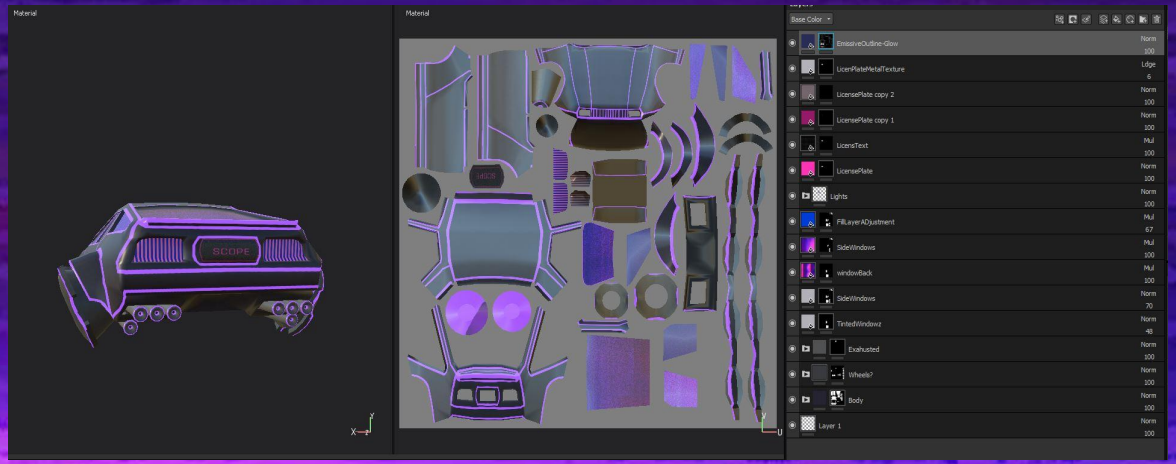
# Car Creation PT 5 – Texturing in Substance Painter

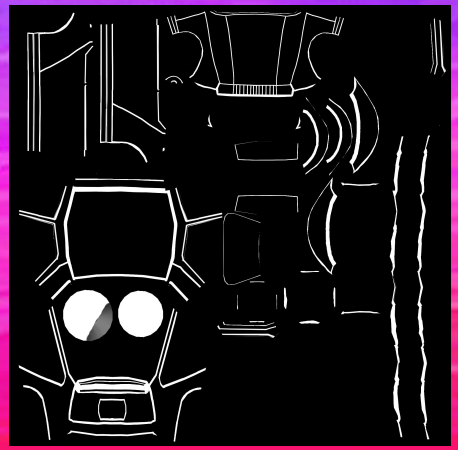
No PBR textures. No normal maps. Etc

1. The image below shows the layers for this vehicle. You want to make sure you have the emissive channel activated. Even though you want be using the emissive channel in game, it is good to use it for visualization purposes.
2. The first layer is that purple outline that you see.
3. When you are finished with the outline, right click on the black box that contains the mask information, then export mask to file.

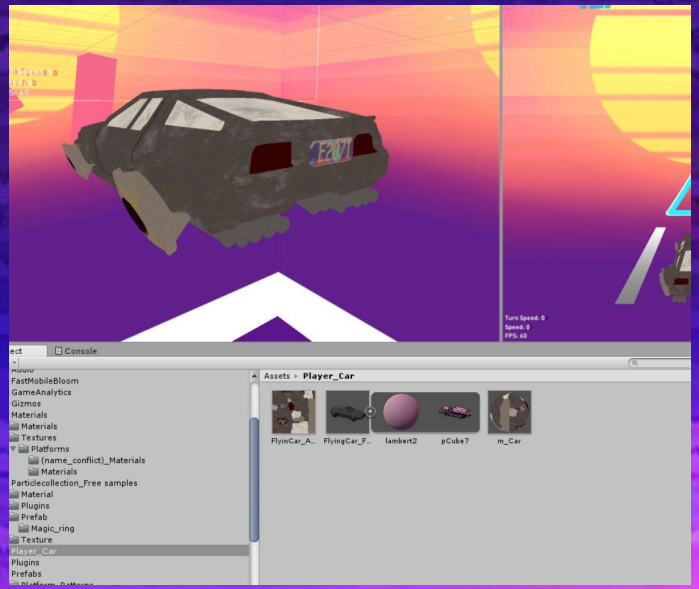
*** This will allow you to mask out that outline in game and make it glow with shaders in Unity. All the vehicles will look like this. It is important to keep things consistent.

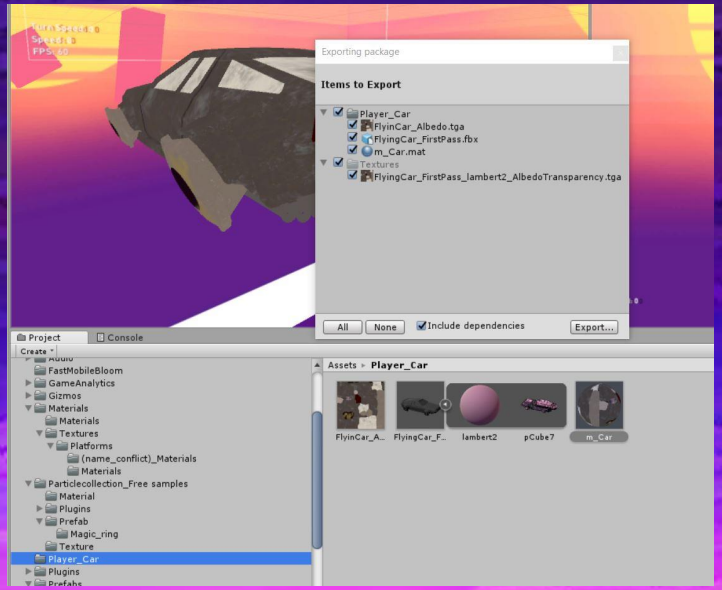
The alpha that you are exporting can be layered on top of the existing cars textures in Unity and you can then get that shader effect only on the white parts of the alpha.

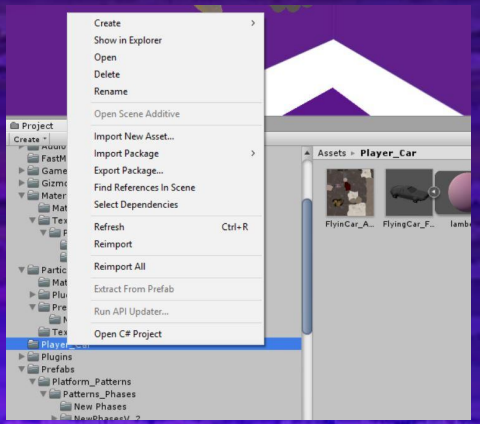


# Asset Packages

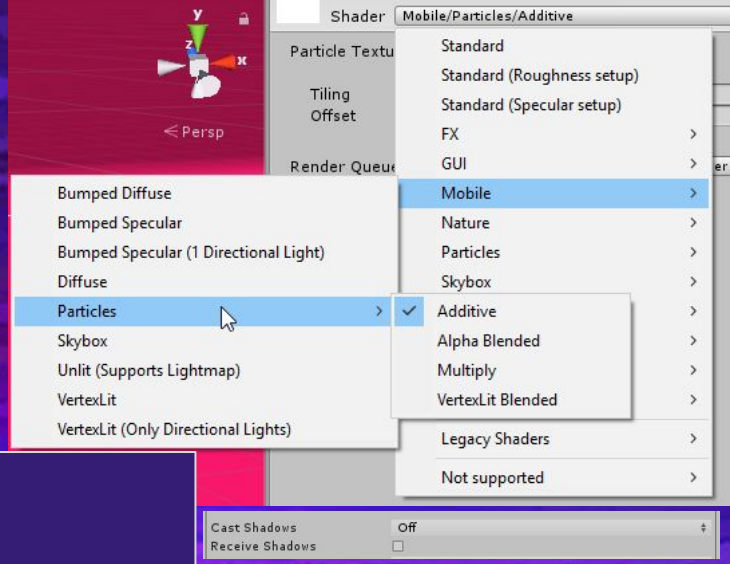





Asset packages are industry standard. When you want to send a designer your work, it is best to assemble it inside an asset package. Refer to this link for more information.
Create a folder in Unity > assemble art > Right click on folder in the project tab > export package

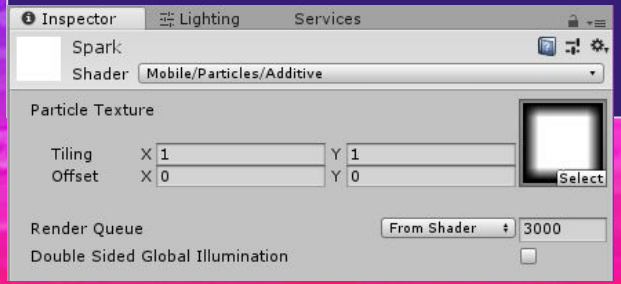
# VFX workflow: Particle Systems

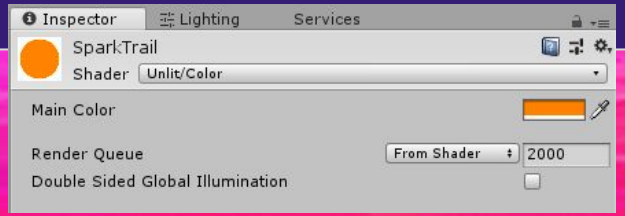



- Use Mobile or Unlit materials for all particles.
- Don’t set particle system to pre-warm!
  - Forces the particle system to simulate one entire cycle / loop every time it is made active.
  - Will cause a frame drop every time one is set active - which happens a lot if placed in the world (like a boost)
- If using noise, use the lowest quality (1D) noise wherever possible
- Particle systems shouldn’t cast shadows, shouldn’t receive shadows.
- Keep the textures used as small as possible.
- Do your best to keep automatic culling from being disabled
  - https://blogs.unity3d.com/2016/12/20/unitytips-particlesystem-performance-culling/
- And obviously - keep the number of particles down!

# VFX Workflow: Particle System Example: Sparks

- Only appears when a soft hit happens - this informs our performance concerns
- Two materials used:
  - The spark itself is an additive material. The texture is a 64x64 white square on black with motion blur applied in photoshop. White on black is used so that the particle can be colored in the particle system rather than in Photoshop. The additive material turns the black into alpha.
  - The spark trail, which is an unlit solid color.
- 200 particles per second at 0.1 second, so only 20 particles emitted.
- 1D noise is used.
- Automatic culling is disabled, because:
  - We use noise
  - We use trails
  - We use the orbital velocity over lifetime setting.
  - **However**, as this will only be on screen for 0.3s max, this isn't a concern of ours.



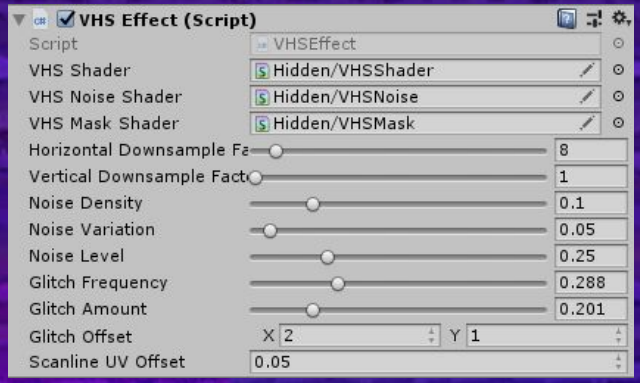

# Electric Speed Boosts

- Boosts  temporarily increase player speed, add vfx and sfx, and change the FOV.
- Boost should be made primarily of transparent light fx and hovering geometry
    - There are no collisions. Players should be encouraged to pick these up, and they should not be confused with obstacles
- There should be an infinitely tall light shaft through the center for visibility from afar
- Upon collecting/passing through, a 2D electric spreading animation should play from the character or point
- A bright ring on the ground as well as outer light fx should indicate the outer edges of the hitbox
- Emissive properties should be exaggerated. We want to encourage players to approach and collect these points.
- We want all interactable objects to be the same color

# Tutorial

- The tutorial should be simple and easy to understand. Ideally, it shouldn't have much text, or any text if you can get away with it. Many players like to click first, read later. 3 words is by far the maximum.
- Here are the goals of the tutorial:
  - Tell the player to hold the tablet sideways (landscape)
  - Tell the player to use both hands, and use thumbs (like a controller)
  - Tell the player to tap and hold to steer left and right
    - Show the player that no, this is not a gyro game
    - Show the player that no, swiping with 1 finger is not the easiest way to play this game
- Right now, the stand-in animation is a few frames showing hands tapping on the screen. It works for most of these goals, but players don't understand that they can tap and hold. Many just tap more often to move, which doesn't work with big turns. This is a problem that needs to be solved.
- Some ideas we had for the new tutorial:
  - Glowing fingerprints
  - "TAP AND HOLD"
  - Outlining and highlighting the left and right areas of the screen where the thumbs should be
  - Making players tap and hold both sides for a second- but this has to be very clear
- In terms of visual design, see the UI slides for in-game HUD. We want it to be as minimalistic as possible, using glowing outlines in a car dashboard-like style. Use curves that were used in 80's product and car design.

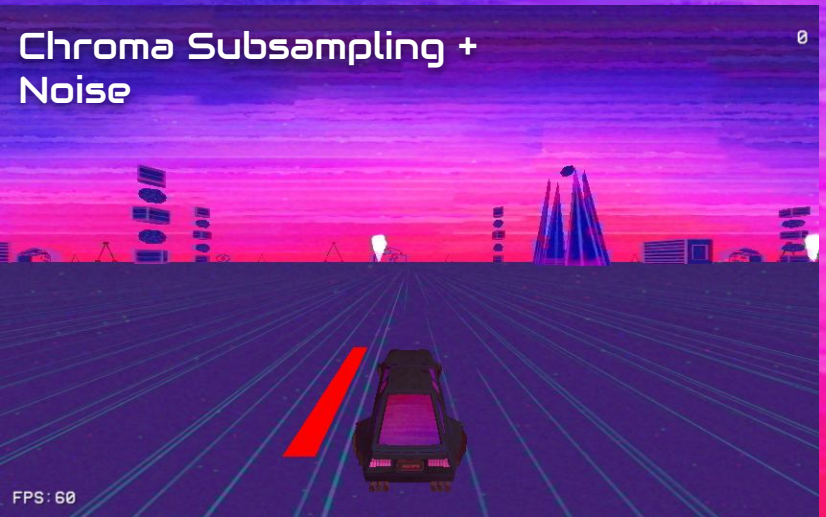
# VFX Workflow : Shaders




- VHS post-processing effect is a multi-pass custom shader.
- Multiple passes are used for performance
  - Entire effect could be done in a single shader, but a lot of work would be repeated
  - Render targets and multiple shaders are used to avoid this
- Low resolution render targets are used in several parts of the process, which when scaled up add to the fuzzy VHS style.
  - And smaller render targets = better performance
- #defines are used in the shader to allow parts to be switched on and off for performance
  - If we decide not to use a certain part of the effect, we can switch it off to avoid doing useless operations.


No Shader
0
FPS: 60


Chroma Subsampling
0
FPS: 59


Chroma Subsampling +
Noise
0
FPS: 60


Chroma Subsampling +
Noise + Scanline Noise
0
FPS: 60

# Glitch effect



# Trailer

For our goals/tips/philosophy on trailers, see:

TRAILER NOTES

- The trailer will need updated gameplay footage. To update the current trailer, download the ENTIRE TRAILER FOLDER. This includes:
  - The final AfterEffects project file
  - The referenced Premiere file, which holds audio and storyboards
    - The premiere file is linked within the AfterEffects project using Adobe Active Link
  - All footage, assets, sfx, music, etc.
  - *** it is a bit outdated, so you will need to move things around
- To relocate all the missing files to the project, you only have to reconnect one. If you imported the entire Trailer folder, it should then automatically re-link the media.
- **To get gameplay footage**, we used NVidia Shadowplay, with this reg file downloaded to hide the mouse. Make sure to follow the linked instructions and restart your computer after installing
  - In plastic, switch to the TrailerChanges branch
  - Apply the camera controller script to the world camera
  - Edit the slo mo amount in the player
  - In unity, hit “maximize on play” to get higher resolution footage
  - Make sure that the elements of the gamecanvas you want to hide are disabled
  - Make sure that the aspect ratio is 16:9
  - Turn off performance mode so the game looks as pretty as possible