

Art and Design for VR

Chandan Singh



About me

2008

Game Dev
Industry

Trine Games

BlueGiant
Interactive

Lakshya Digital

PSP, PS2, PS3,
PS4, Xbox 360,
Xbox One, Wii,
PC & Mobile

SmartVizX
Head of VR

Outline

Why VR?

Challenges with
VR

Designing for VR

Art creation for
VR

Optimizing for
performance

Q & A

Why VR?



Why VR?

Best sense of
immersion -
presence

No 'killer'
game yet

Easy to get
started

Instant VR Game

Metacritic

95%

JUST ADD WATER!

Challenges with VR



VR is difficult

Human eyes
are very
sensitive

Illusion of
presence is
easily broken

Motion
sickness!

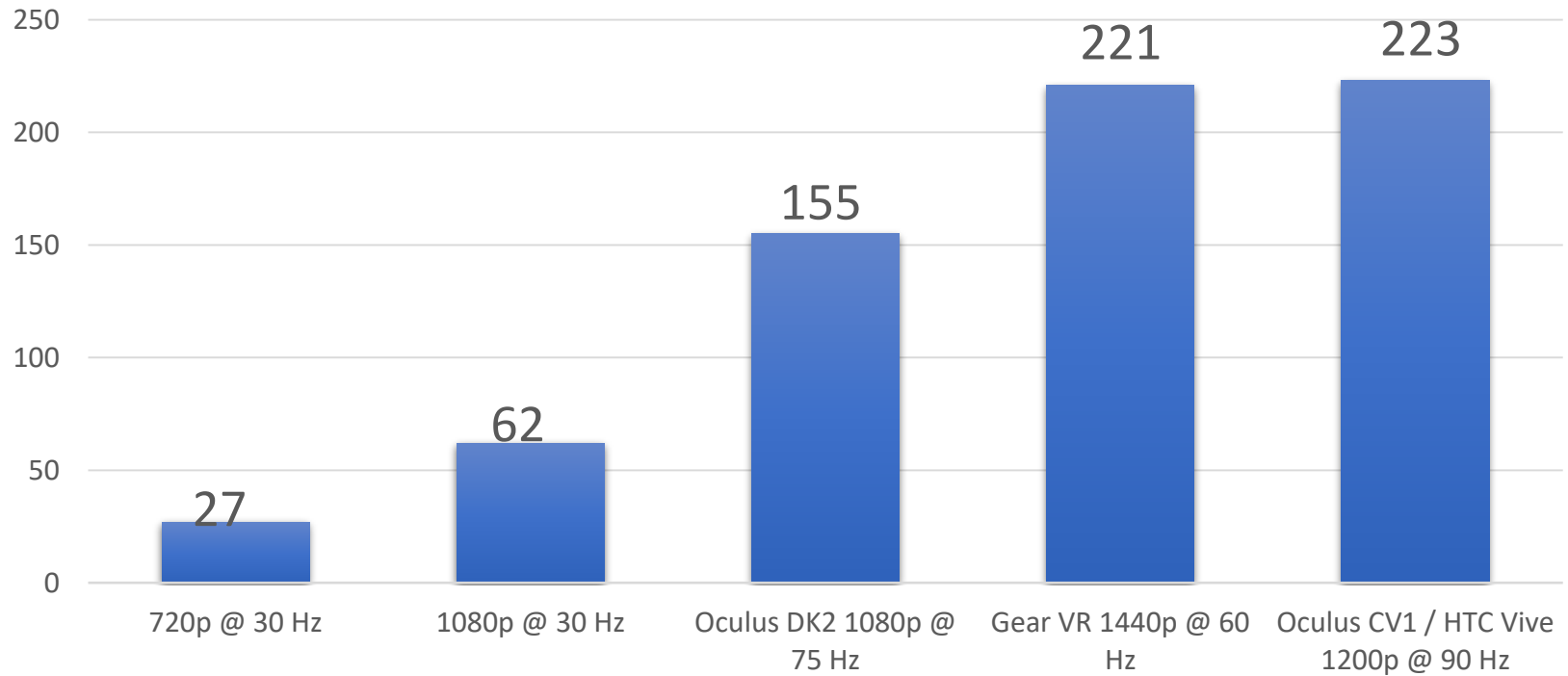
Too many pixels to render

VR needs high
resolution
displays

VR needs high
frame rate

Upscaling to
negate lens
magnification

Pixels to render per second (millions)



It's a LOT of pixels!



720p @ 30
FPS

9X



1080p @ 30
FPS

6X



1080p @ 60
FPS

3X

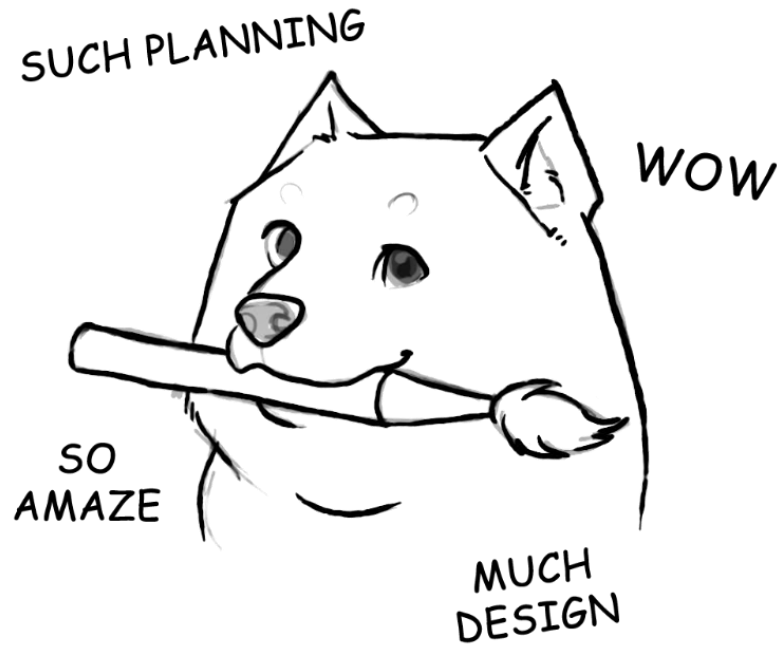
Latency

Motion to
photon time
<20 ms

60 FPS
minimum

Everyone
hates lag

Needs well designed and planned content



Designing for VR

Goal: Make the user feel comfortable



Goal: Make the user feel comfortable

Visual and
auditory senses
VS
rest of the
senses

The disconnect
causes 'motion
sickness' &
'nausea'

Drunk people &
really young kids
seem immune

VR is not a camera

VR is not the camera, it's the player's head

User always controls the camera

Give clues to look, never force it

No head bobbing

No camera shakes

No DoF or motion blur

It's an immersive world

Invite users to
look all around

People tend to
forget they are in
a 360 world

Level / world
design can be
more vertical
and diverse

Movement

Avoid rapid
movement

Don't require to
move head /
body a lot

Keep the horizon
line steady

If you must
move the
camera, don't
accelerate

Teleport / Blink /
Vehicles

Allow time to
settle into new
environments

Interaction Design



Interaction Design

Use real-world
cues when
appropriate

Audio cues, NPC
reactions

Have a proper
feedback
mechanism

Hands and Body

Have something
to denote the
body

In game limbs
and body parts

Even the nose
mesh!

Space, perspective and scale



Space, perspective and scale

Anything that
gives a sense of
scale is very
effective

The user can be
really small, or
really big

Really adds to
the WOW factor

UI Design

Keep the UI elements in the world, not stuck to the screen

Ideally further than 3 meters

Make sure words are easily readable

UI Design



Cutscenes



Cutscenes

Don't take
camera control
away, it's
nauseating

Hint to see in the
direction, hold
the action till
then

Half Life 2 is a
very good
example

Cutscenes



Audio

Really important,
really
understated

Positional audio
is easy to
implement in 3D
games

Ask the user to
put on
headphones

Field of View (FOV)

Don't override
FOV manually or
expose to the
user to edit

Needs to match
physical
geometry of
headset and
lenses

Should be
automatically set
through the
device's SDK and
internal
configuration

Split content in manageable chunks

Long intervals of VR can be fatiguing

15-20 minutes is a good enough duration

This will get better as the technology improves

Test early, test often



Test early, test often

Developers make
the WORST test
subjects

Start testing as
early as possible

Test on as many
people as
possible

Maintaining immersion is the top priority

GREAT

Making you feel like you are really in a place

Letting you touch and manipulate objects

Face-to-face confrontations

NOT SO GREAT

First person virtual motion

Screen-relative HUD interfaces

Variable framerate

Art creation for VR

Review your artwork in the headset

Colorspace is locked on the hardware

Defects in VR are more pronounced

Start testing as early as possible

Make art fit for purpose

High level of detail is required for the first two meters

Allocate 40-70% resources for it

Test in-headset, you'll know where to spend resources

Colors

Avoid high
contrast
elements next to
each other

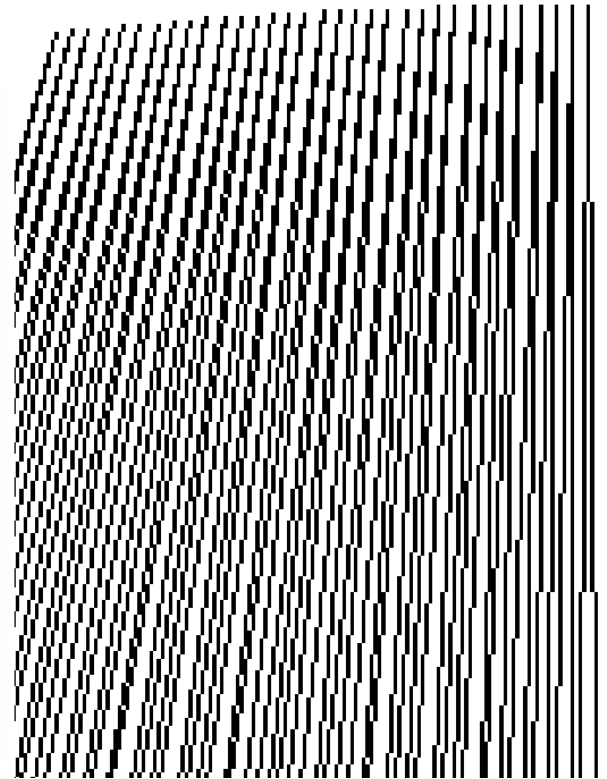
Avoid really
shiny chrome if
possible

Bright scenes are
fatiguing

Thin geometry looks bad

Aliasing is highly pronounced in VR

Apply as much anti-aliasing as performance allows



Accurate scale is vital

You judge your
own size based
on the world
around you

Applies to
objects and
characters too

Really helps to
the immersion

Large shapes > high frequency detail



Large shapes > high frequency detail

Human eyes read shapes first, then details

Silhouettes are very important

Spend your resources on shapes first, then details

Stylised can be better than photo-realistic



Stylised can be better than photo-realistic

Humans are better at finding defects in things they can relate with

The 'Uncanny Valley' effect is lower with stylized things

It's usually cheaper to render stylized art

Accurate materials

Go for physically based materials

It depends on art direction, but if you're going slightly styled, PBR saves loads of time

Get rid of the noise, make it clear to be easy on the eyes

Normal maps

Normal maps do not account for a binocular display or motion parallax

Use for objects that are far, otherwise add more polygons

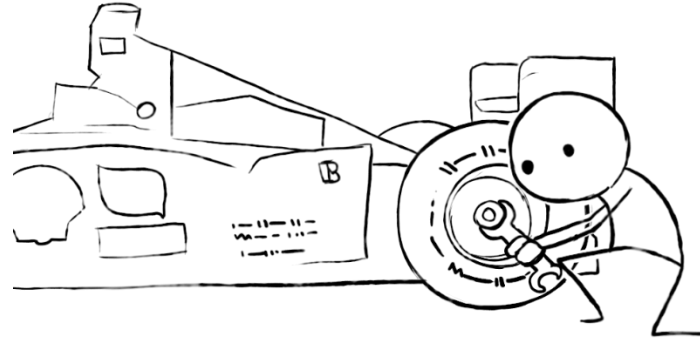
If really necessary, use Parallax mapping or Tessellation

Particles / VFX

Large particles
look flat, use
smaller particles

Interactive
particles have
more value

FX on the
camera can work



Optimizing for performance

Always hit the target framerate

60 FPS on mobile

75/90 FPS on
desktop

No framerate
spikes/drops

Hardware resources are finite



CPU

GPU

IO
(Size / latency /
bandwidth)

The usual suspects

Real time
shadows and
reflections
(CPU/GPU)

Transparency,
multi-pass
shaders, per
pixel lighting
(GPU/IO)

Large texture
loads, skinned
animation
(IO/CPU)

Unity Setup guidelines

Ensure project settings are set to maximum performance

Static batching

Dynamic batching

GPU skinning

Multithreaded rendering

Lock default orientation to left (for mobiles)

Batching

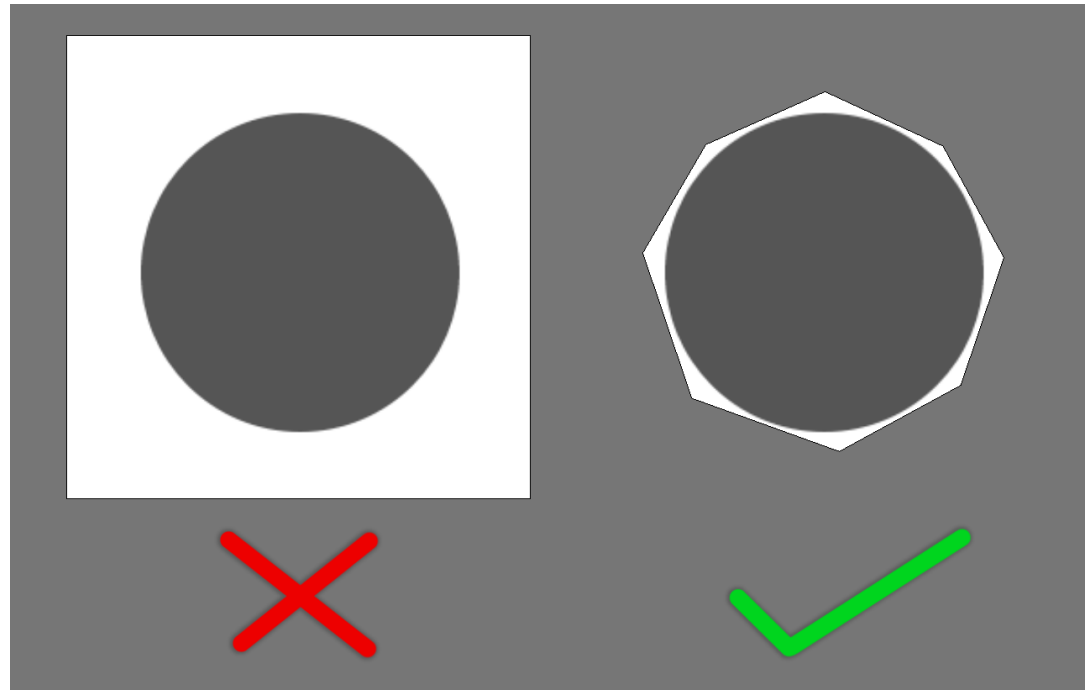
Use texture atlases wherever possible

Use Renderer.sharedMaterial instead of Renderer.material

Use StaticBatchingUtility.Combine at runtime to generate a batched mesh

Transparency, Alpha Test, and Overdraw

Stick with
opaque
geometry or
alpha-to-
coverage for
cutouts



Targets for Mobile VR in Unity

50-100k
polygons, 50-100
draw calls

As few textures
as possible (but
they can be
large)

1 ~ 3 ms for
script execution

Unity Update()

Unreal Engine 4 setup guidelines

- In project settings, set frame rate to 75 fps for both min and max
- Enable low persistence mode for Desktop VR
- Temporal AA causes Texture Blurring / Vibrating
 - Disable Temporal AA/switch to FXAA
- Disable Mobile HDR if working for Gear VR
- Use static meshes instead of BSPs
 - Turn off gravity and collision (against statics) on static meshes
- Use baked lighting as much as possible, avoid dynamic lighting
 - Set 'Cast Dynamic Shadow' to off for all static objects
- Use the console command hmd mirror off

Unreal Engine 4 setup guidelines (contd.)

- Go easy on movable lights, GPU particles, scene collision for GPU particles, particle lights, and extremely high-instruction-count materials
- Transparency and translucency are super expensive
- Screen space reflections are very expensive, use reflection capture probes instead
- Tone down engine scalability settings, you can go down to medium or high without seeing much rendering fidelity changes
- Set up a post process volume and disable all post effects you don't need
- Make sure you are using Precomputed Visibility in your levels

Profiler is your best friend!

Both UE4 and
Unity 5 come
with great
profiling tools

Identify and fix
the bottlenecks
ASAP

Make sure you
are always
hitting the target
framerate

Wrapping up

Well done VR is
an unmatched
experience

It's really easy to
get started

VR experiences
have to be
designed
carefully

User comfort is
super important

Consistent
performance!

We still are
waiting for the
killer VR game!

Questions?

Thanks!

chandan@smartvizx.com