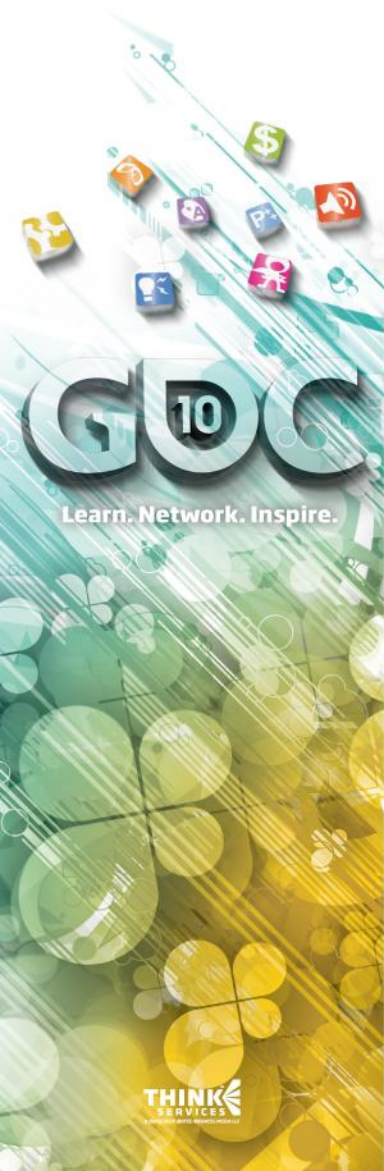


GD10C

Learn. Network. Inspire.

www.GDConf.com

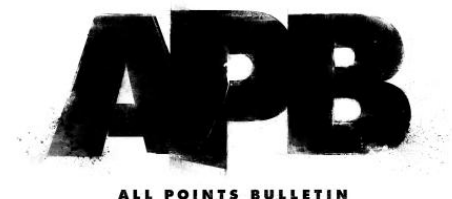


APB: Creating a Powerful Customisation System for a Persistent Online Action Game

Maurizio Sciglio

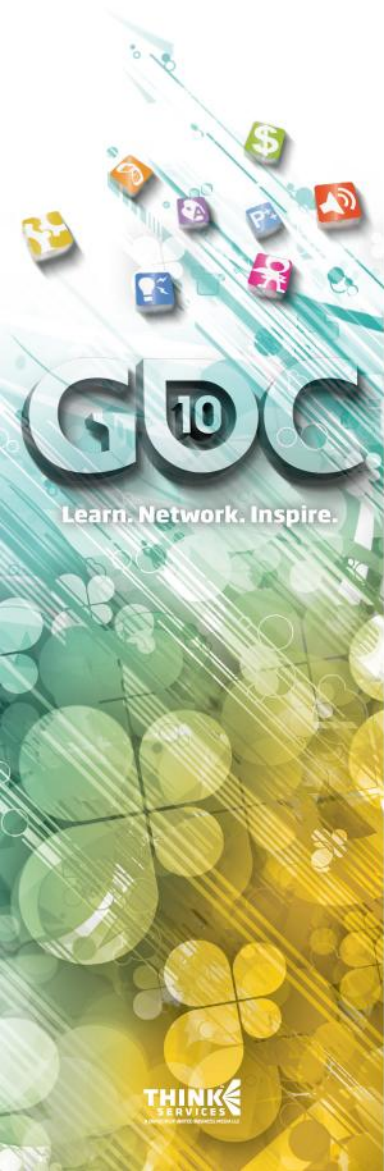
Simon Taylor

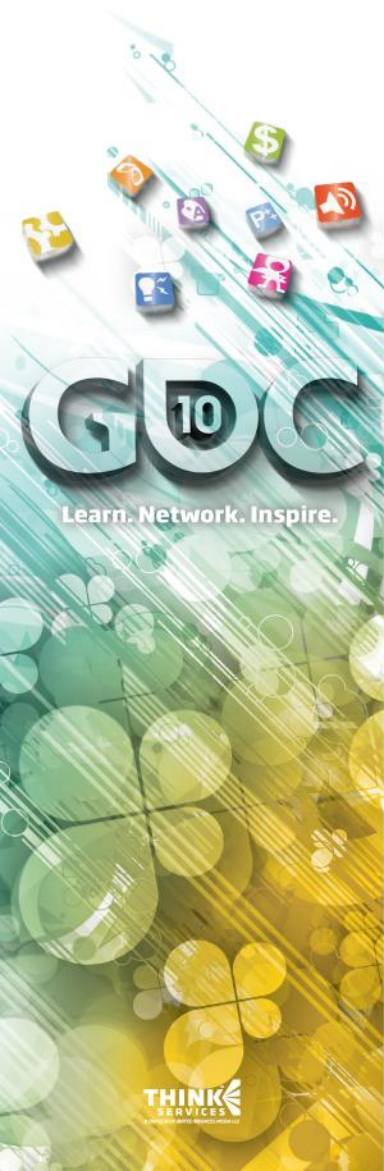
realtime
worlds



Outline

- ⌘ Goals
- ⌘ Initial Design
- ⌘ Refinements
- ⌘ Demo
- ⌘ Scalability
- ⌘ Conclusions

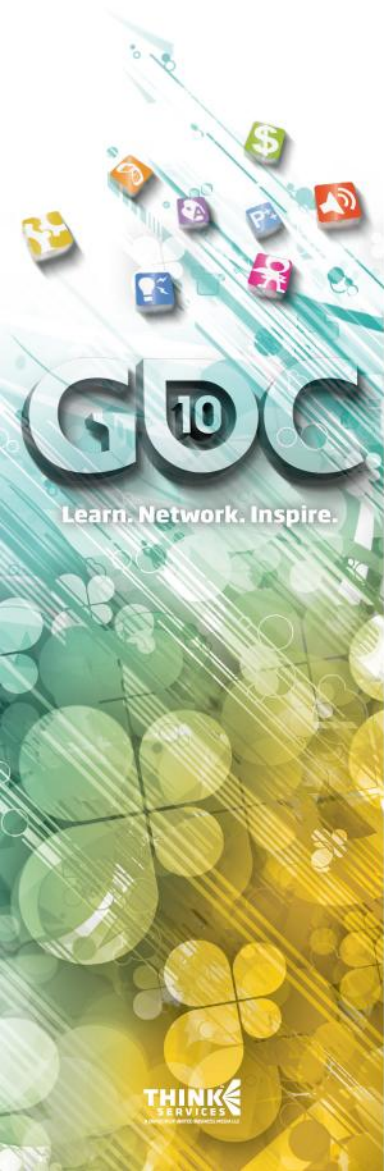




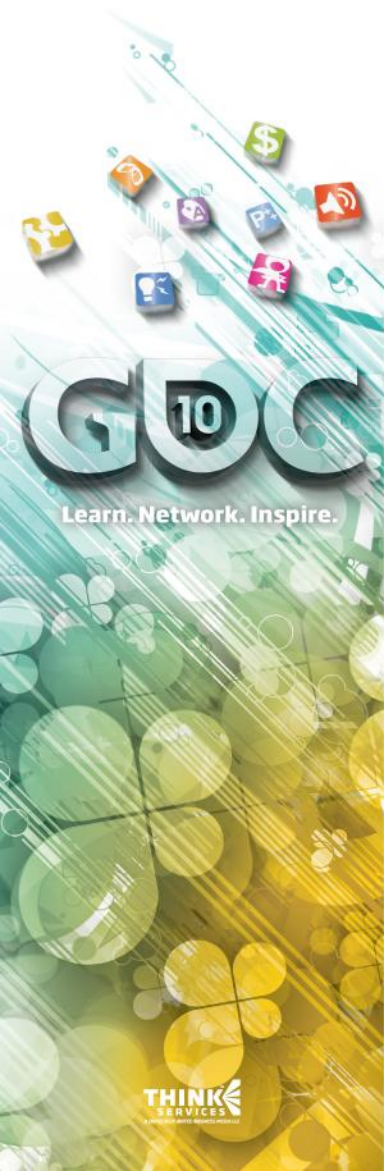
GOALS

What is APB?

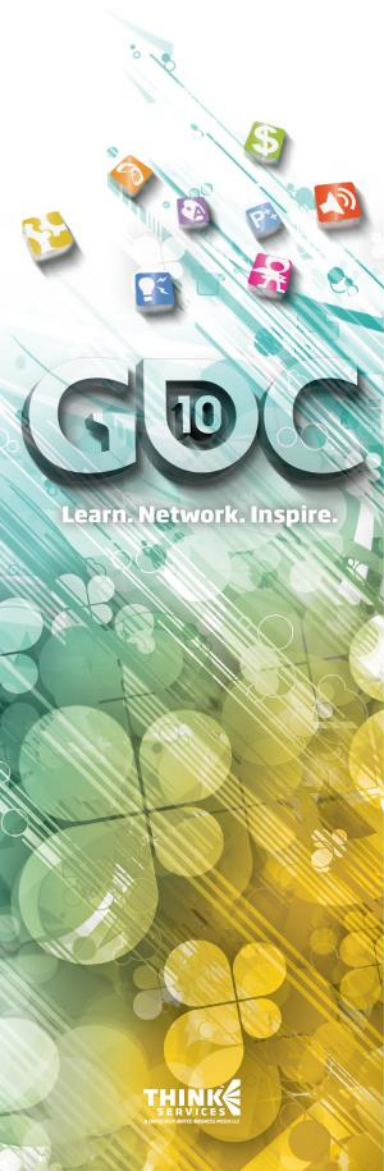
- ⌚ “Persistent Online Action Game”
- ⌚ Shared player-space
- ⌚ Fast-paced 3rd-person gameplay
- ⌚ RTW-run backend
- ⌚ Customisation a key selling point
- ⌚ PC initially
- ⌚ Unreal Engine 3



Decal Application – Forza

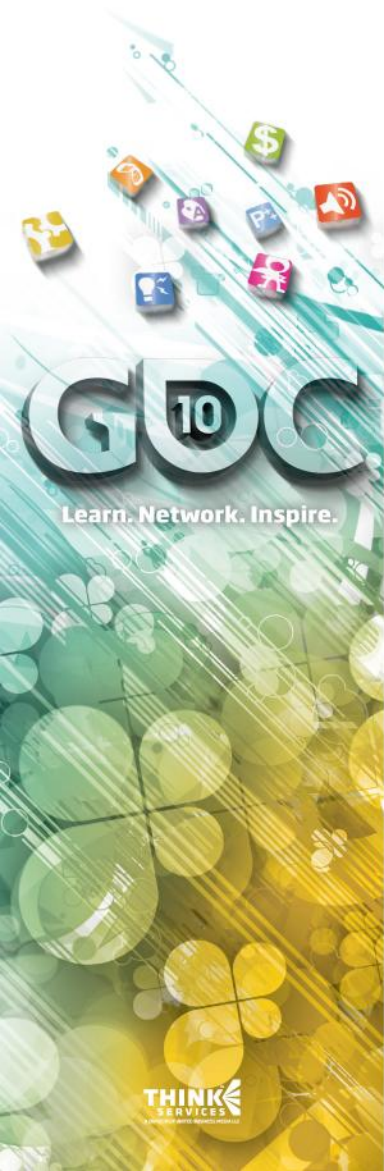


Decal Application – Forza



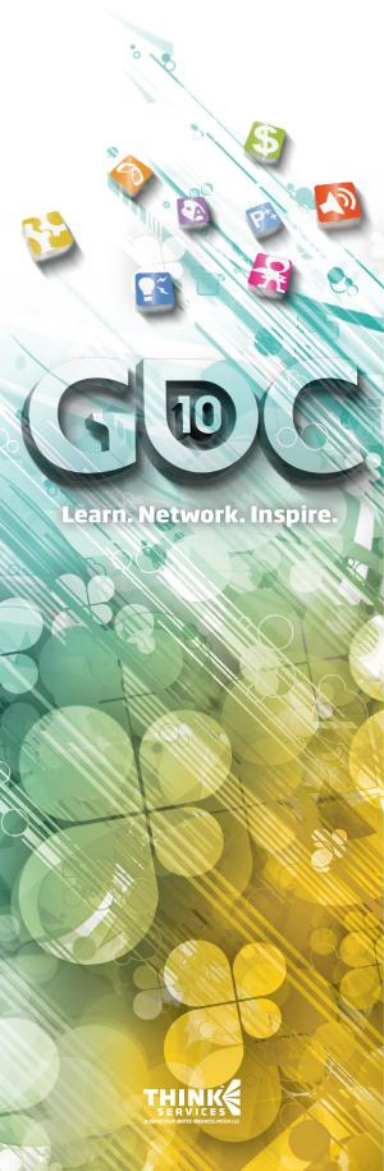
Decal Application

- ⌚ Not just vehicles; characters too
- ⌚ Characters are harder:
 - More complex shape/UV mapping
 - Heterogeneous surface (clothing, hair, etc.)
 - Complex masking for clothing
- ⌚ Therefore, simple UV-space composition not possible



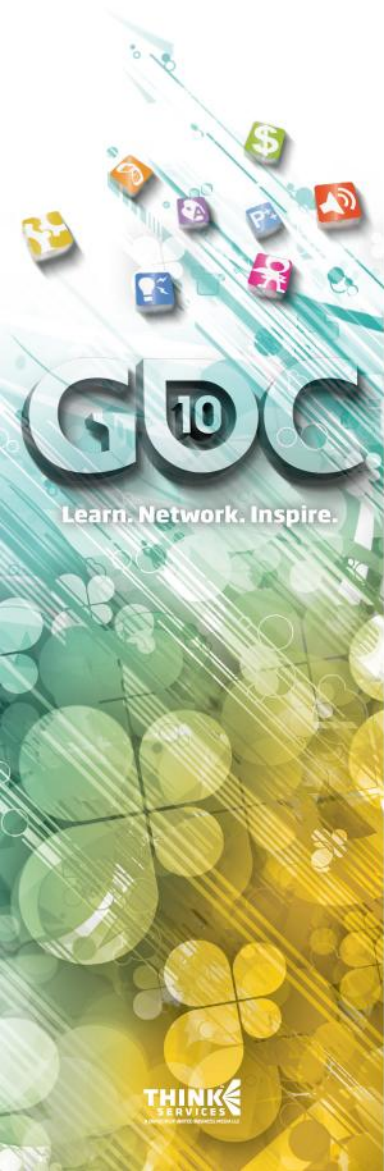
Other Requirements

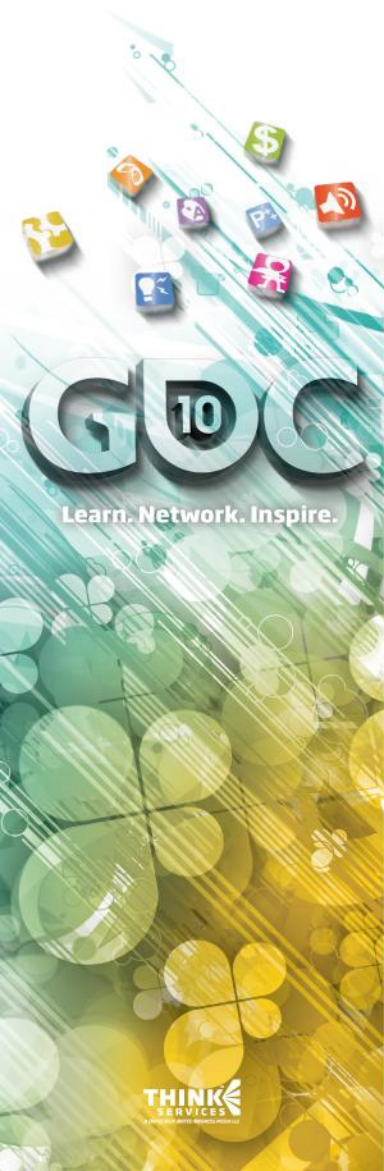
- ⌚ Face & body customisation
- ⌚ Hair styles
- ⌚ Clothing application with large variety of available styles
- ⌚ Decal application
 - Tattoos on skin
 - Print on clothes



MMO Concerns

- ⌚ Large player count
- ⌚ Limited bandwidth
- ⌚ No upfront “Lobby”
Players enter and leave at any time

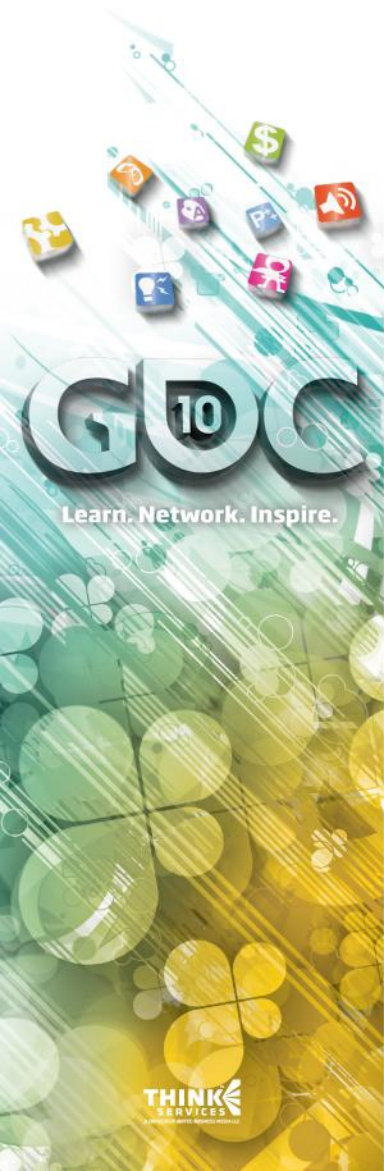




INITIAL DESIGN

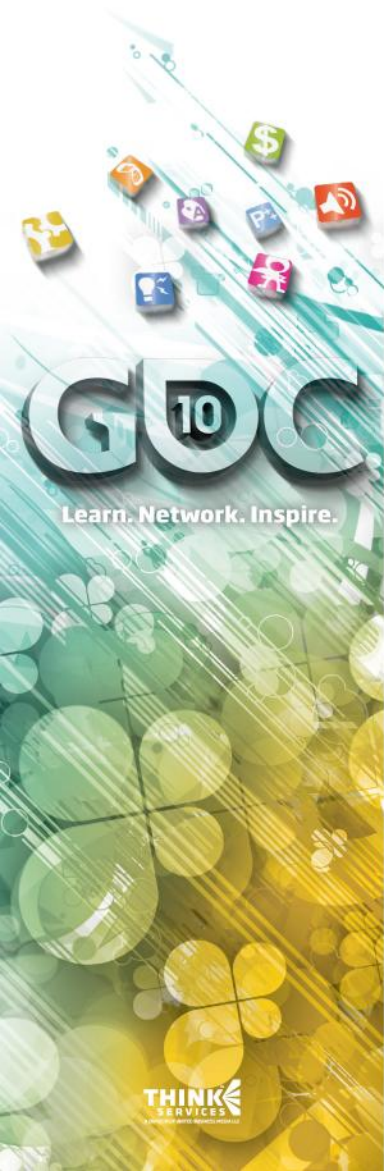
Parametric Description

- ④ Final assets (meshes & textures) are big
E.g. 1 MB for a 1024x1024 DXT5 texture
- ④ Bandwidth is expensive
Can't send final assets over the network
- ④ Therefore, describe assets parametrically
- ④ Build final assets on clients
- ④ Fundamental principle of the system

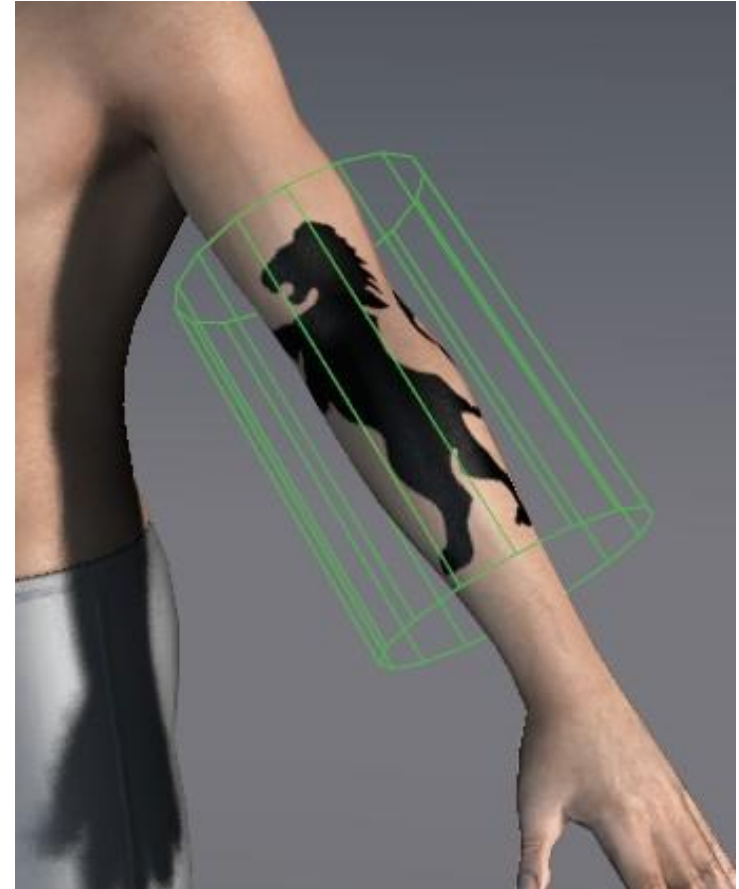
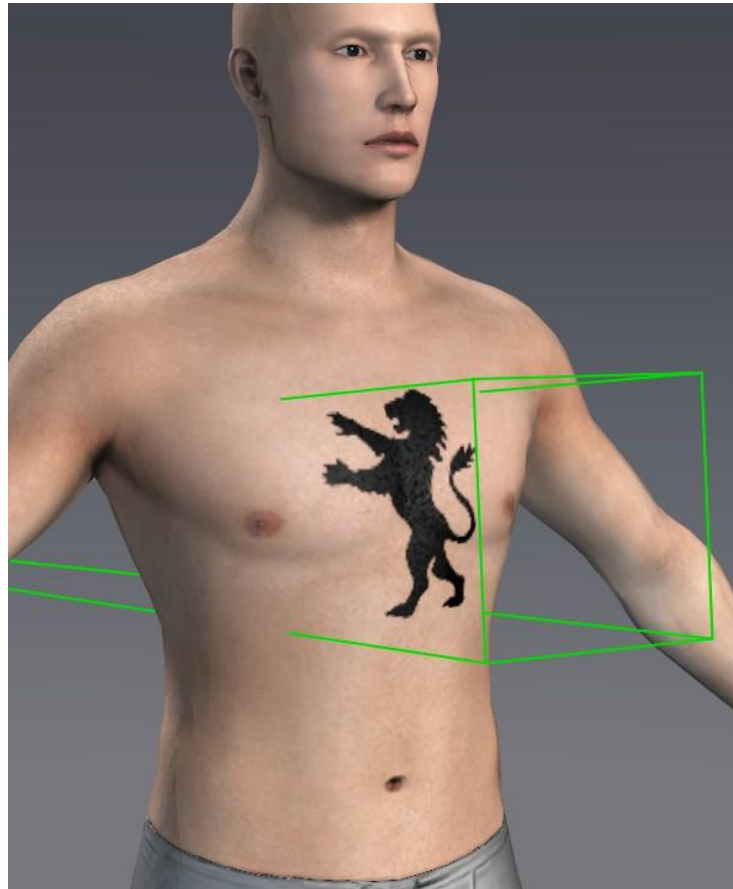
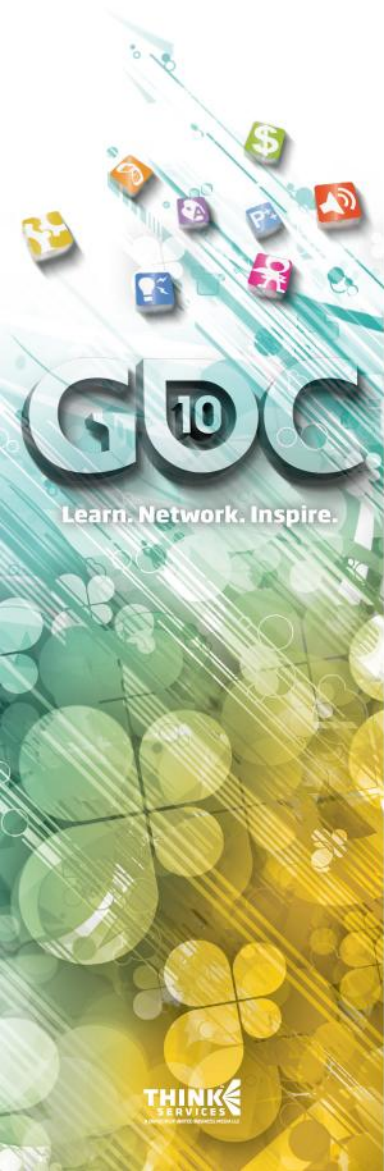


Morphing

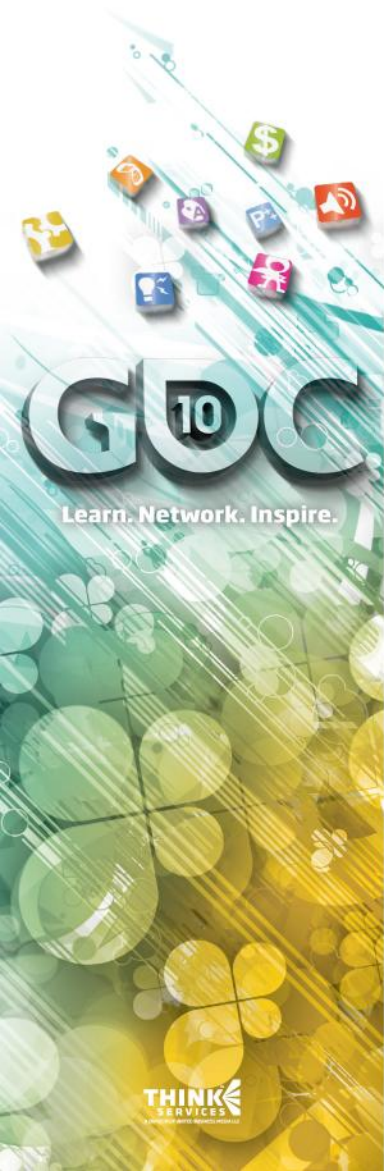
- ⌚ Fundamental mesh operation
 - Standard additive morphing
- ⌚ Used for Body & Facial modification
- ⌚ Sliders control individual target weights



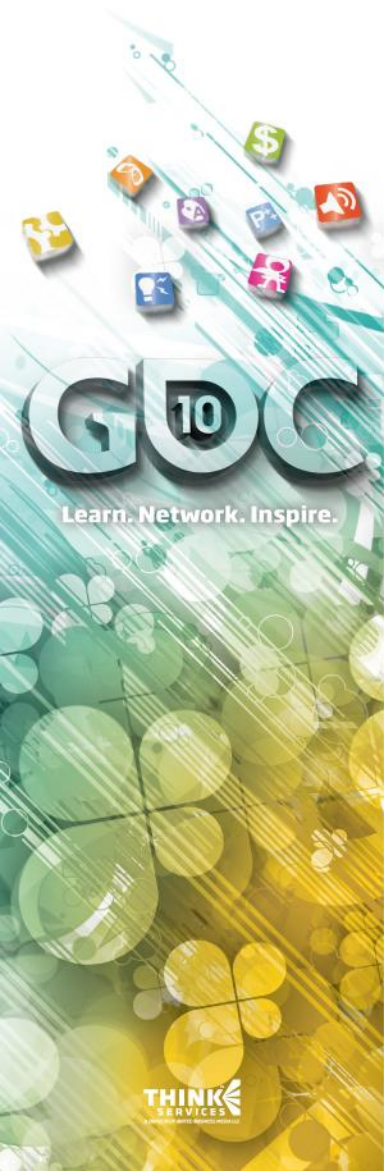
Decals / Tattoos



Skin Tone



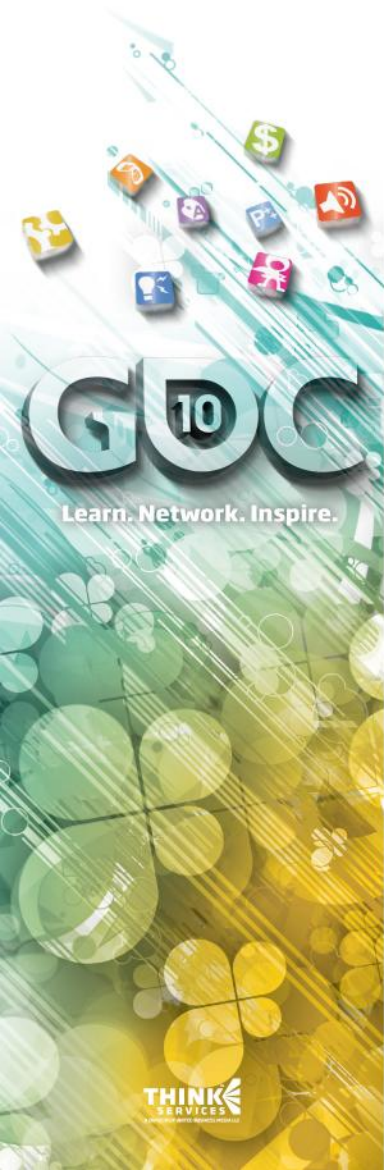
Skin Pigment



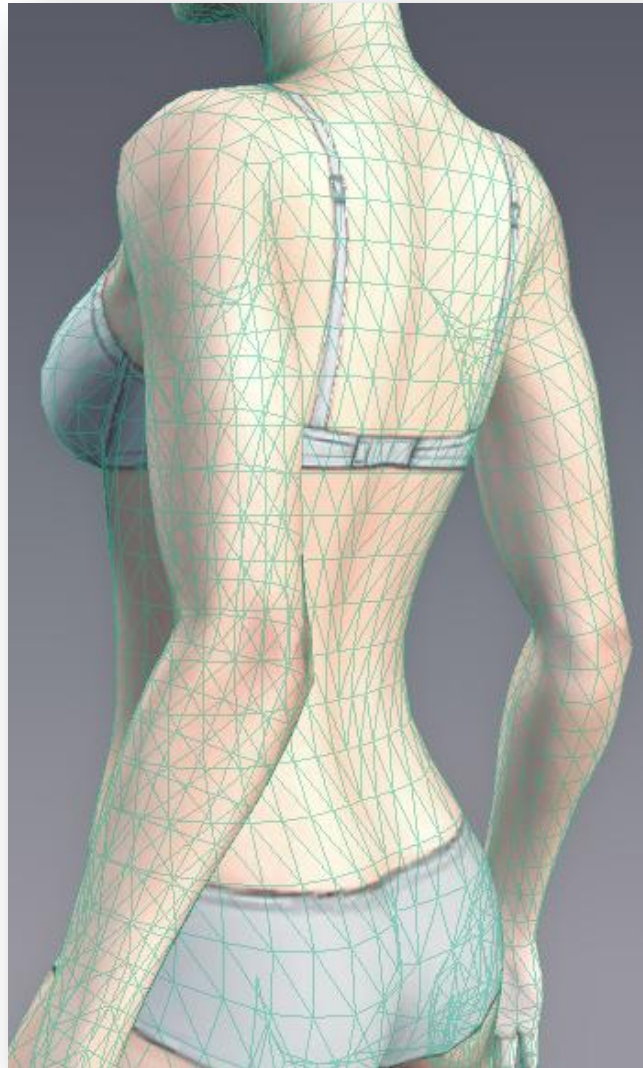
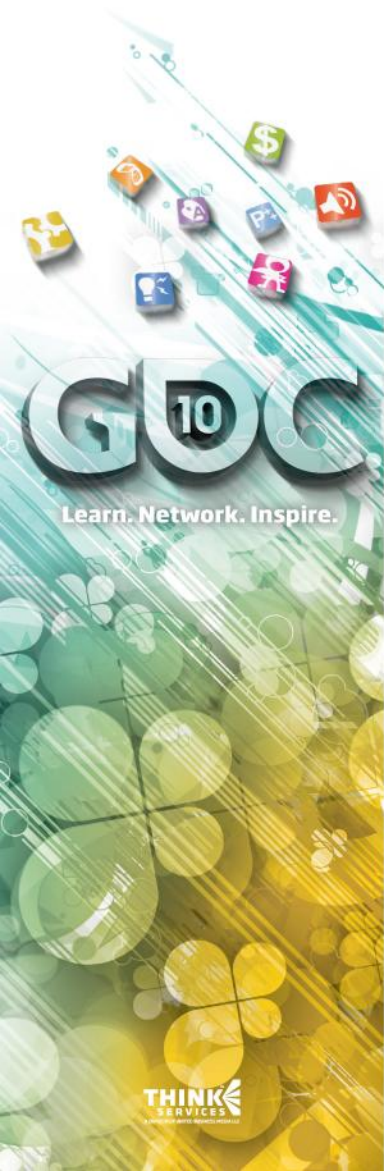


Clothing Customisation

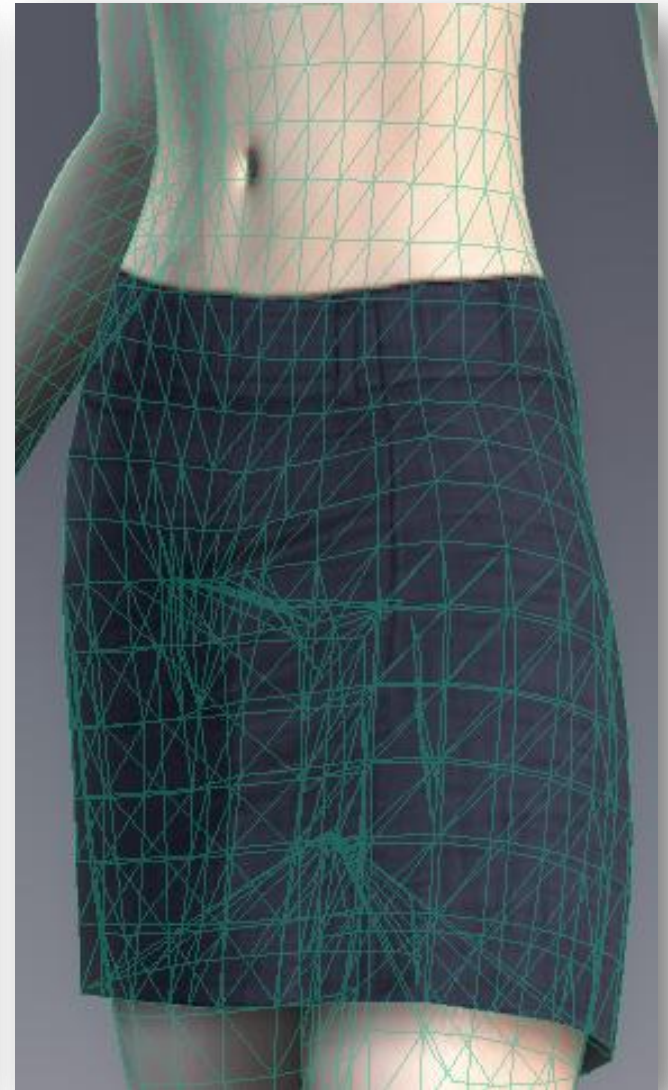
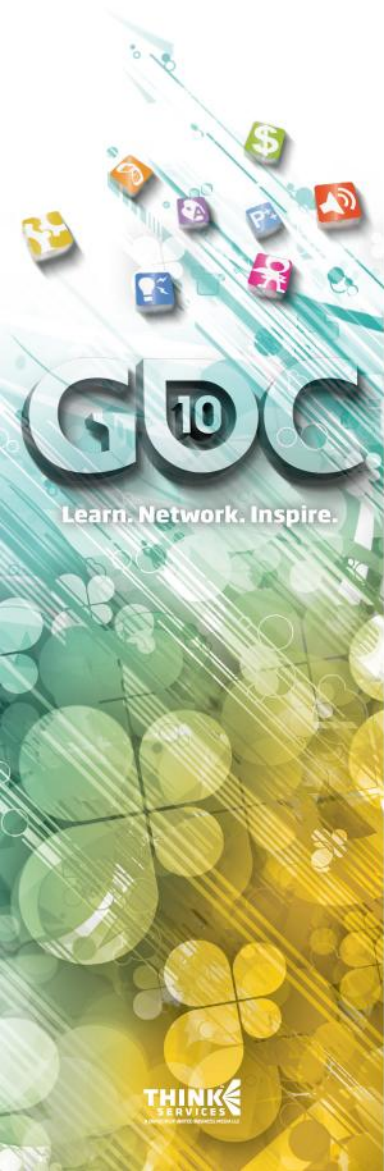
- ⌚ Rendering / memory cost independent from the number of items
- ⌚ One mesh to rule them all!
 - Clothing mesh must be merged with the base mesh
 - Texture must be combined with the base texture



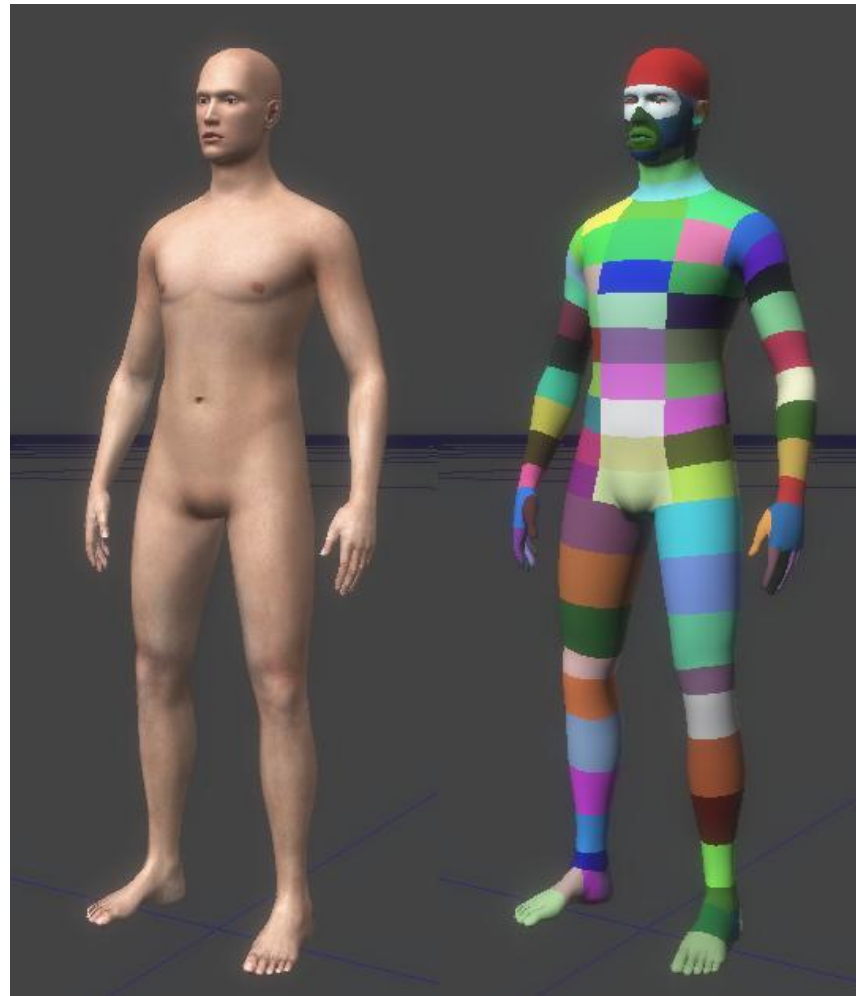
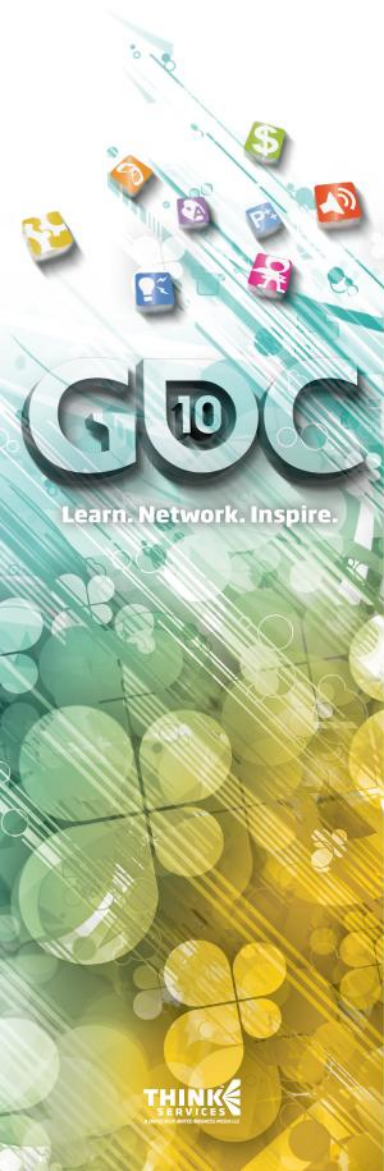
Clothing – Morph Item



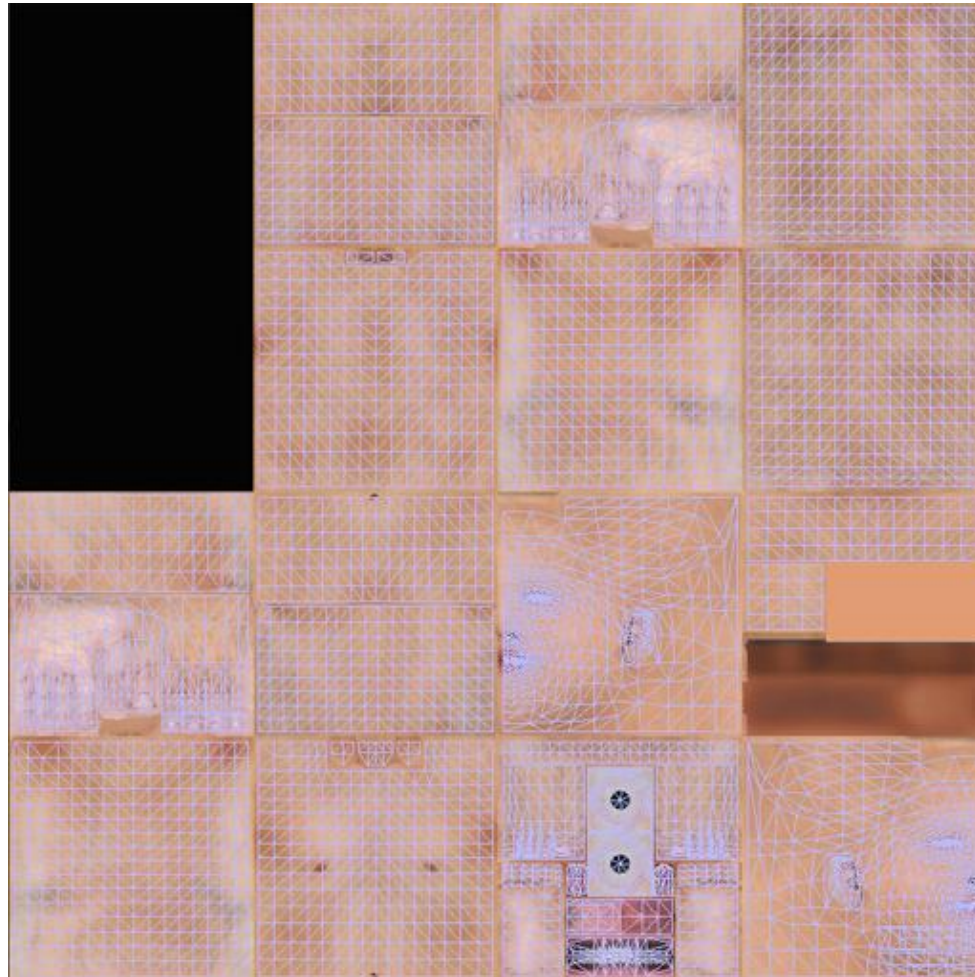
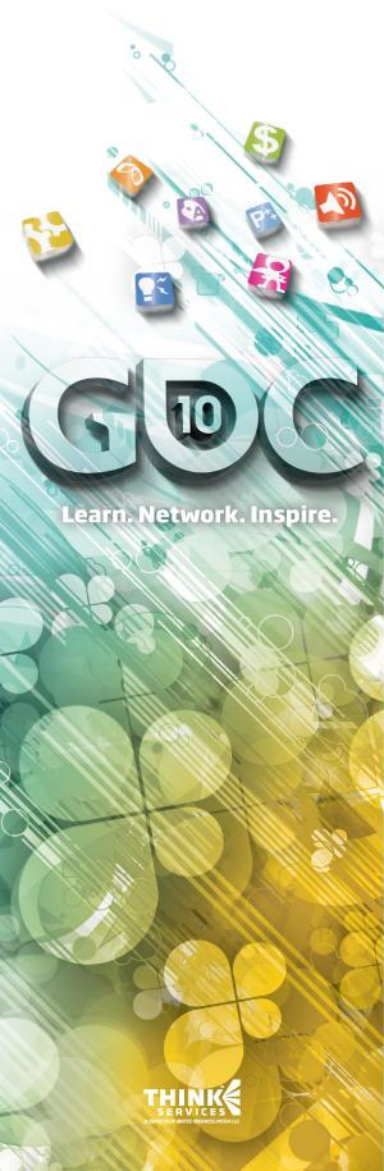
Clothing – Extra Mesh Item



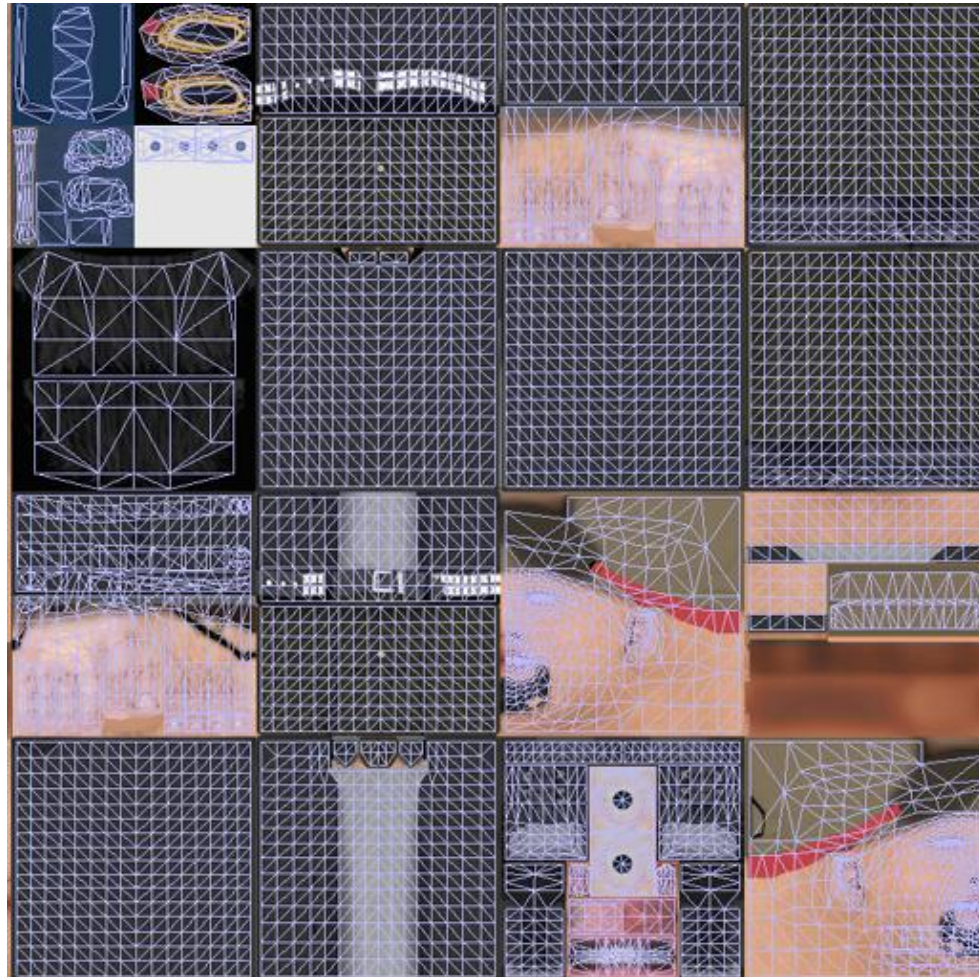
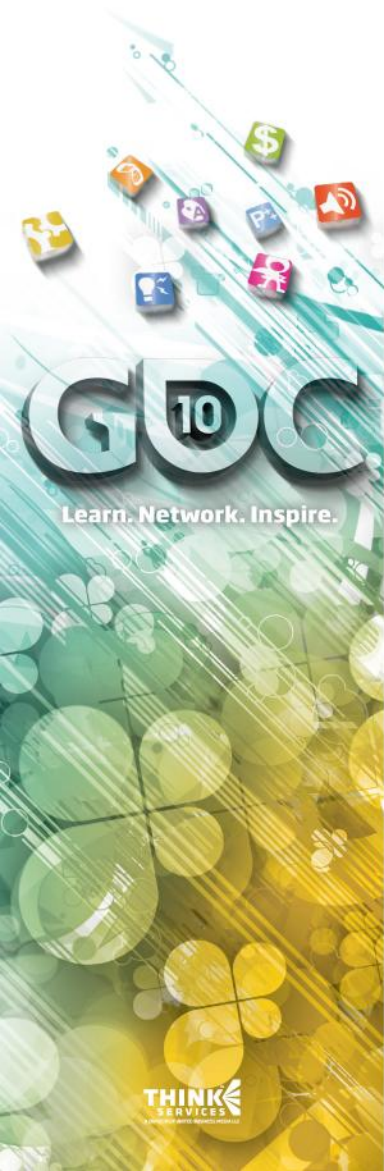
Mesh Culling



UV Layout

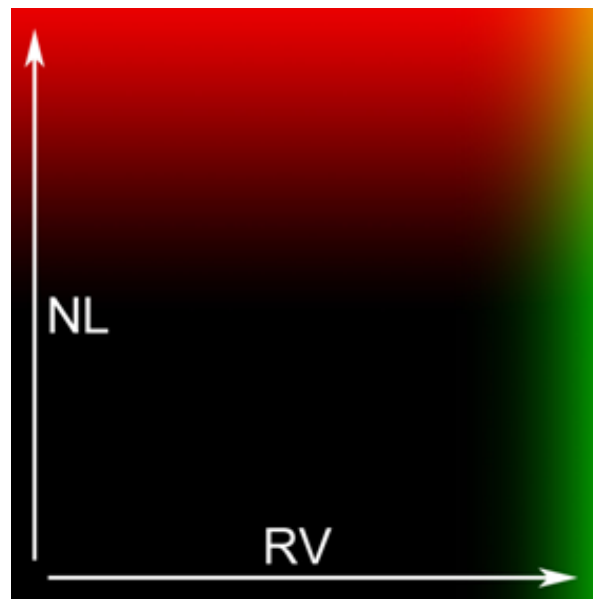


UV Layout

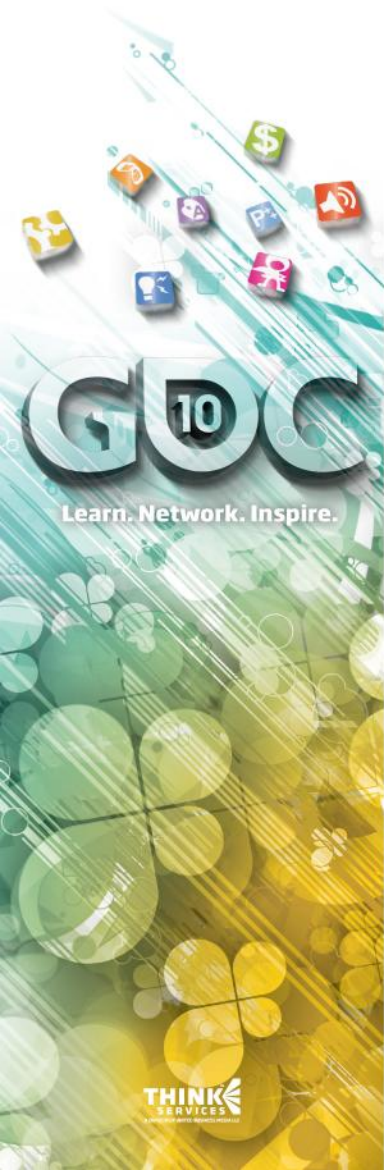


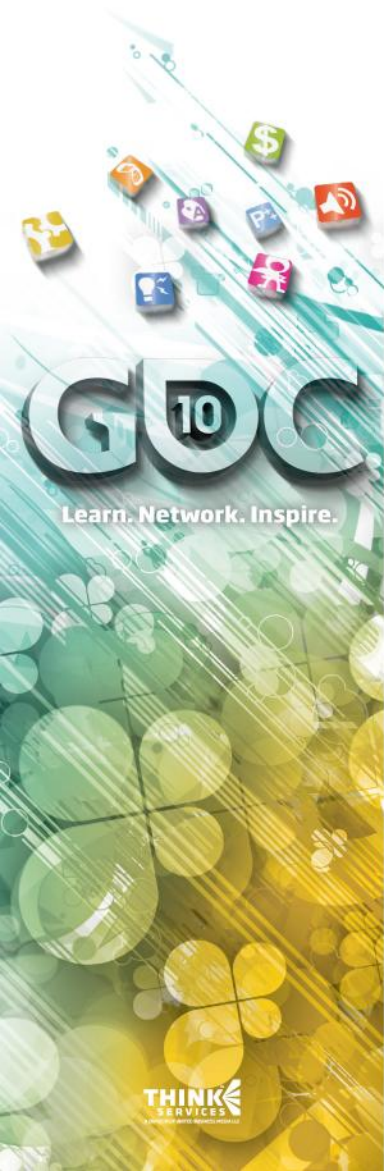
Character Materials

- ④ Main goals:
 - Single draw call per character
 - Support multiple materials
- ④ BRDF decompositions stored in a 3D texture
 - $N \cdot L$
 - $R \cdot V$
- ④ Per-pixel material selection



Phong decomposition

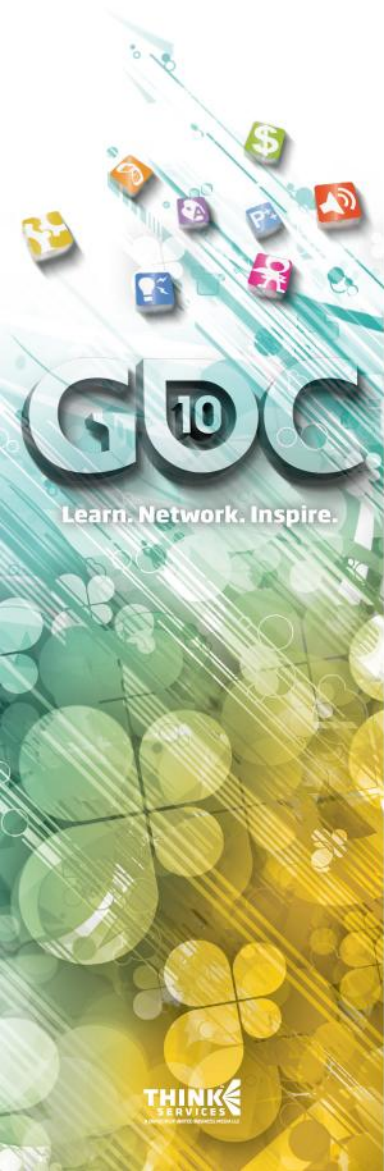




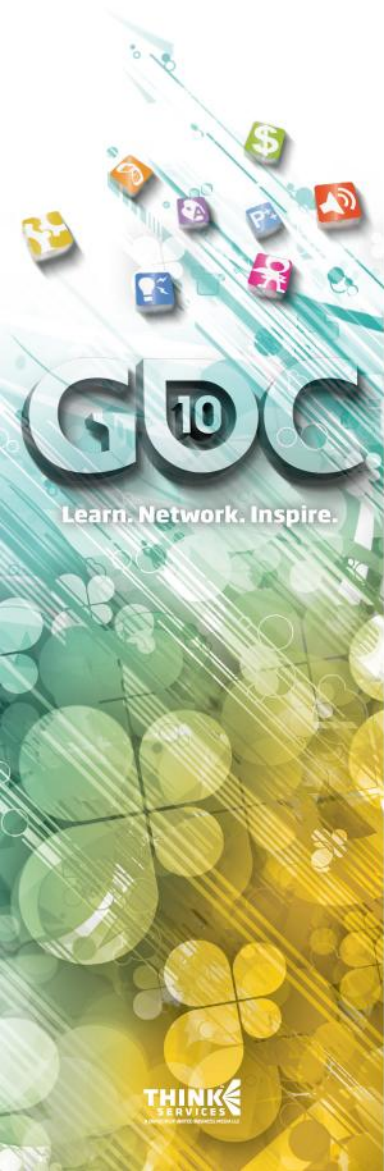
REFINEMENTS

Decal Projection Issues

- ⌚ Slow to construct
- ⌚ Large data size
- ⌚ Layer count had to be restricted
Artists didn't like this
- ⌚ In most cases, not actually required
- ⌚ Solution: Symbols



Symbols



Distance-Field Encoding

Regular Texture



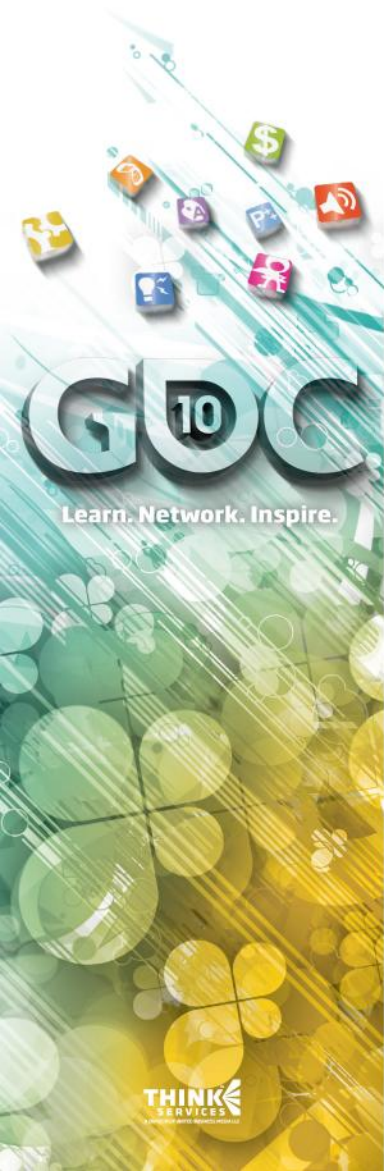
**Distance-field
encoded**



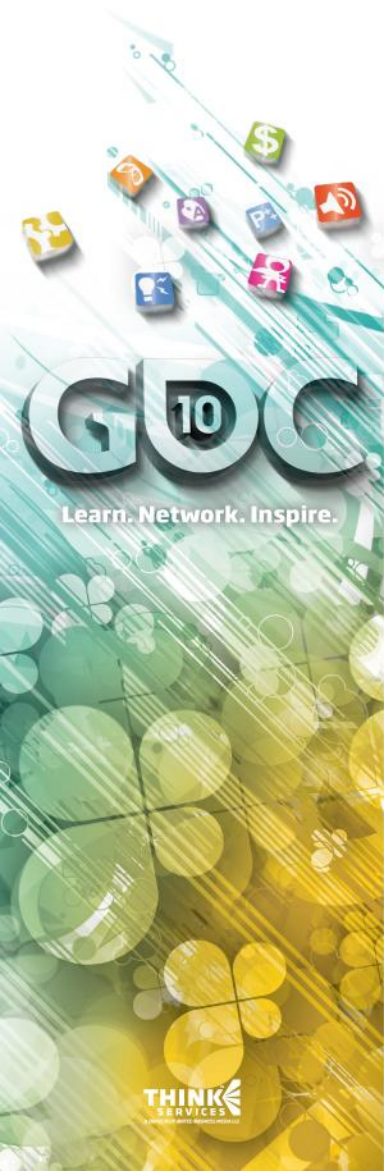
[http://www.valvesoftware.com/publications/2007/SIGGRAPH2007_AlphaTestedMagnification.pdf]

Height Scaling

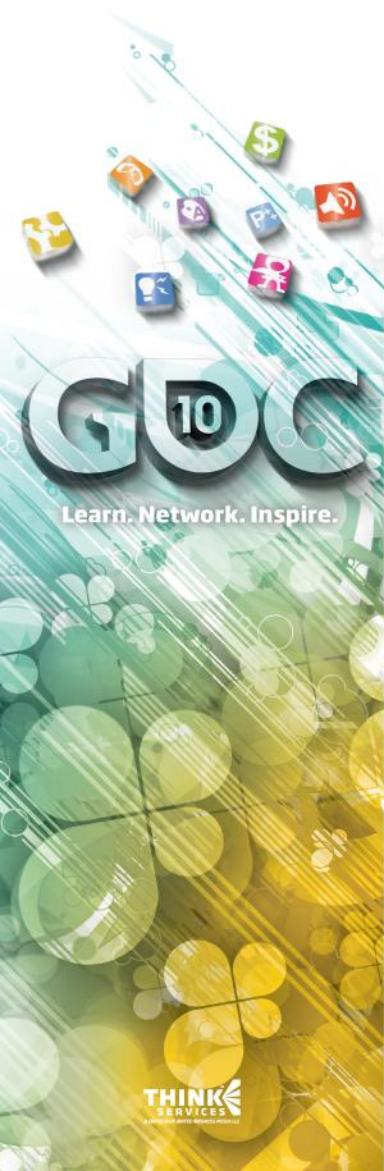
- ⌚ Customisable character height
- ⌚ Per-bone non-uniform scaling
- ⌚ Many issues
 - Artist time
 - Animation issues
 - Collision implications
 - Fairness
 - Bugs!



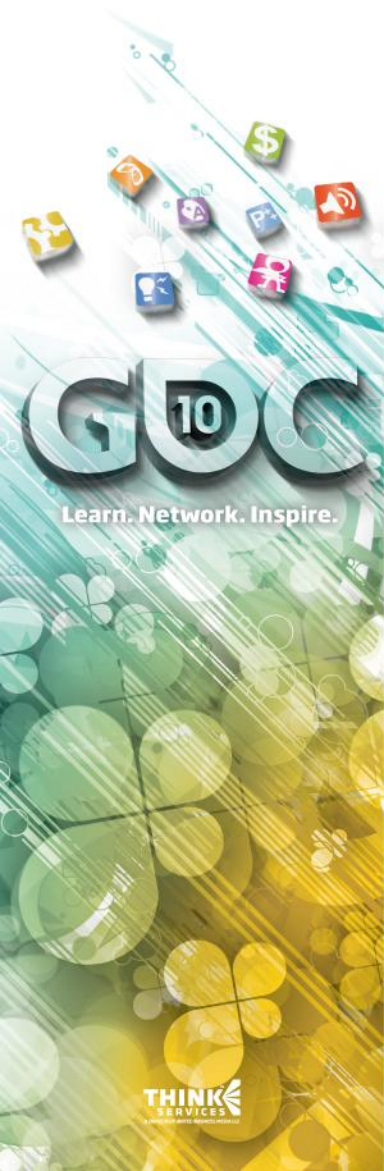
Height Scaling



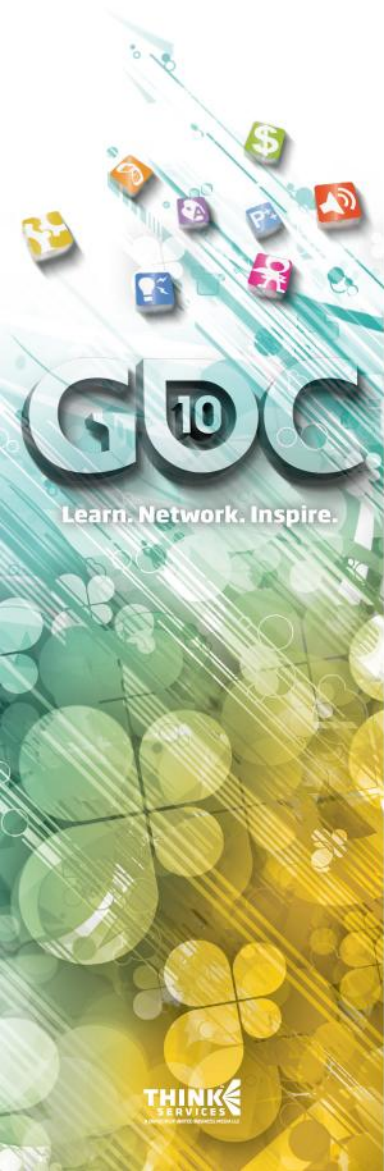
Projection Seams



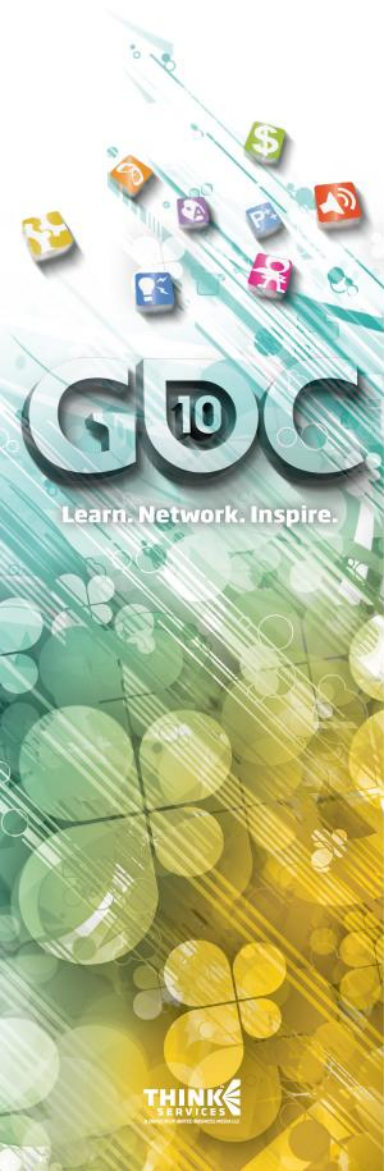
Projection Seams



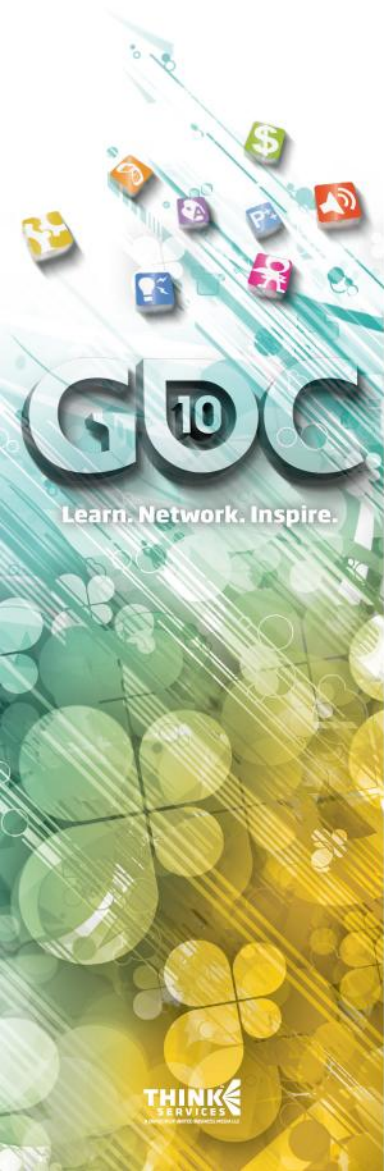
Hair-Clothing Interactions



Original

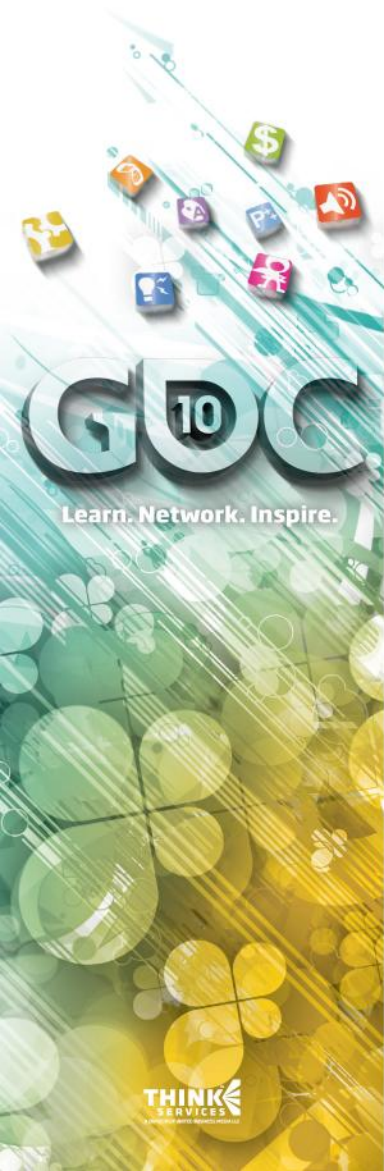


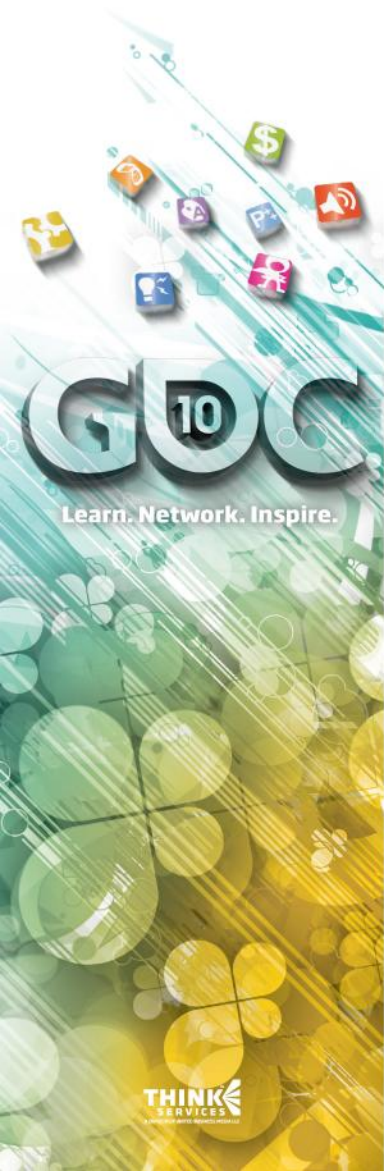
With Fitting Morphs



**Game Developers
Conference®**
March 9-13, 2010
Moscone Center
San Francisco, CA
www.GDConf.com

Demo

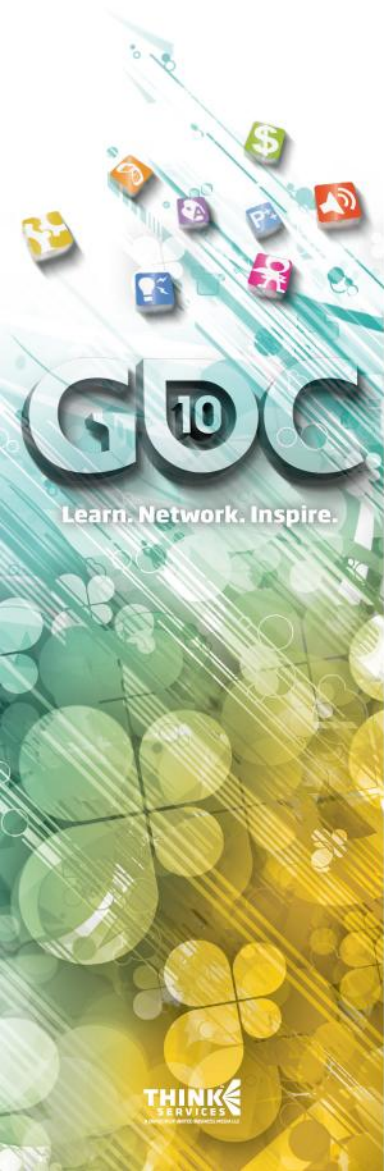




SCALABILITY

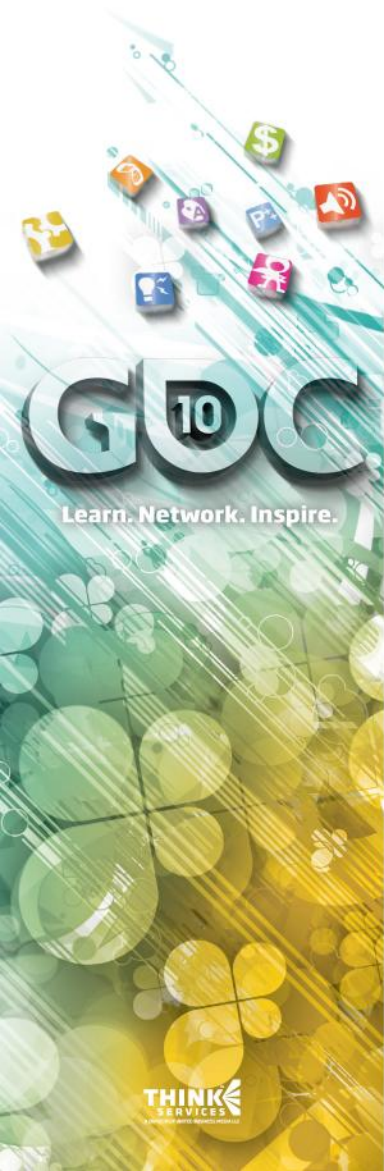
Bandwidth

- ⌚ ...is expensive
- ⌚ Even parametric representation is big
 - Bandwidth is n^2 in player count
- ⌚ Must minimise data size
 - Quantise parameters
 - zlib compress all data
- ⌚ Average ~ 4 kB per character
- ⌚ Limit complexity user can create



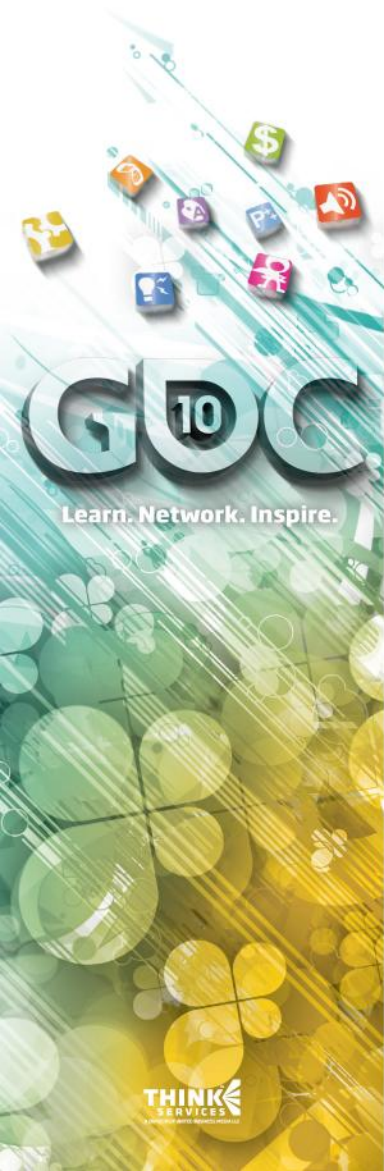
Background build

- ⌚ Assets built during gameplay
- ⌚ Two options
 - Threads
 - Time-slicing
- ⌚ Problems with threads
 - No multithreaded D3D access
 - Less control
- ⌚ We chose time-slicing



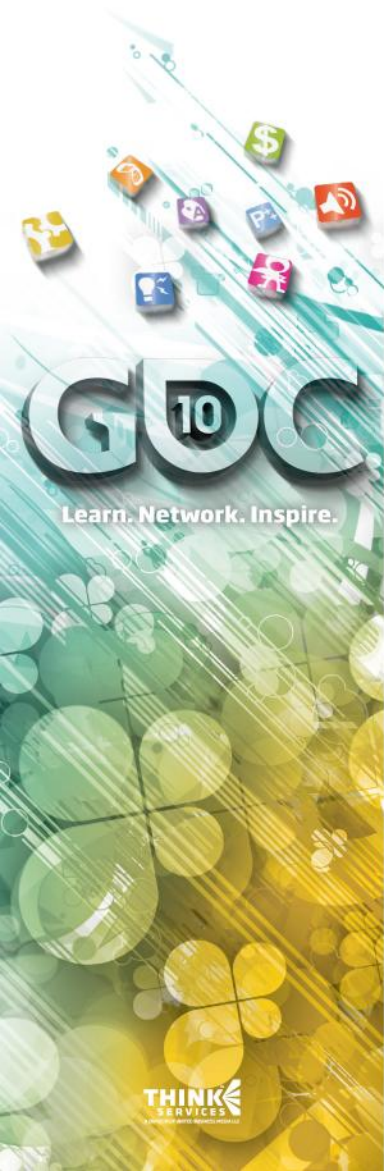
The Correct Choice?

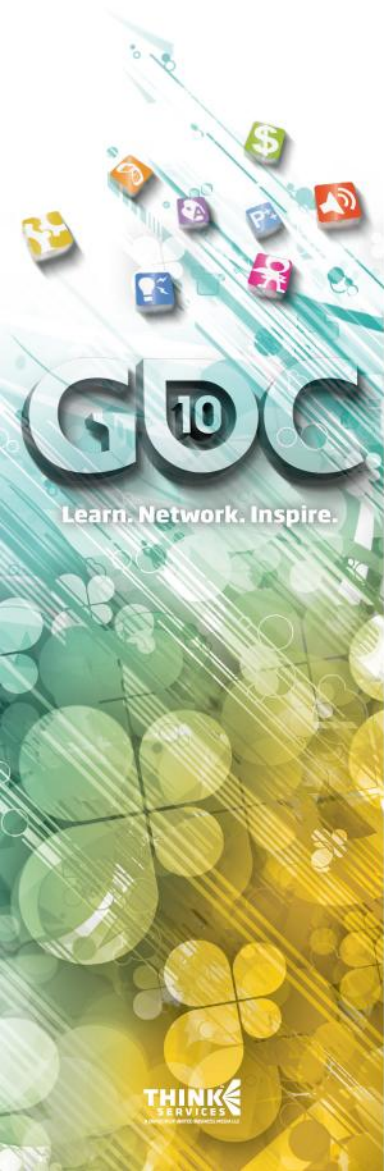
- ⌚ Time-slicing brings its own issues
 - Developer time
 - Frame-rate spikes
 - No true parallelism
- ⌚ In hindsight, threads were maybe the better option



Memory Management

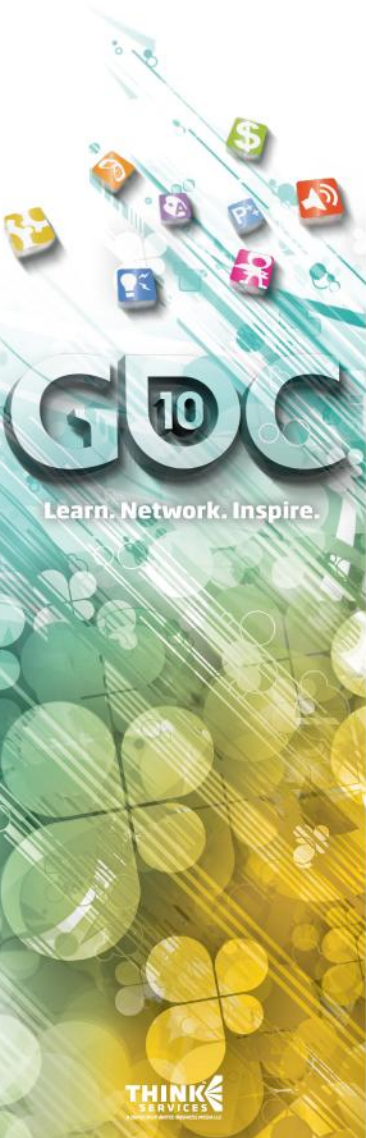
- ⌚ Memory use for customised assets is huge
- ⌚ Too slow to build on-demand
- ⌚ Therefore, must write out assets to *disk cache*
- ⌚ Textures and meshes streamed from cache files
- ⌚ Total memory usage capped
- ⌚ *Not* persistent





RESULTS & CONCLUSIONS

Game Developers
Conference®
March 9-13, 2010
Moscone Center
San Francisco, CA
www.GDConf.com



THINK
SERVICES
A MINDSPACE COMPANY



APB





APB
ALL POINTS BULLETIN

BLOODROSE CONTACT -- TIER 0 -- STREGA BLOODROSE

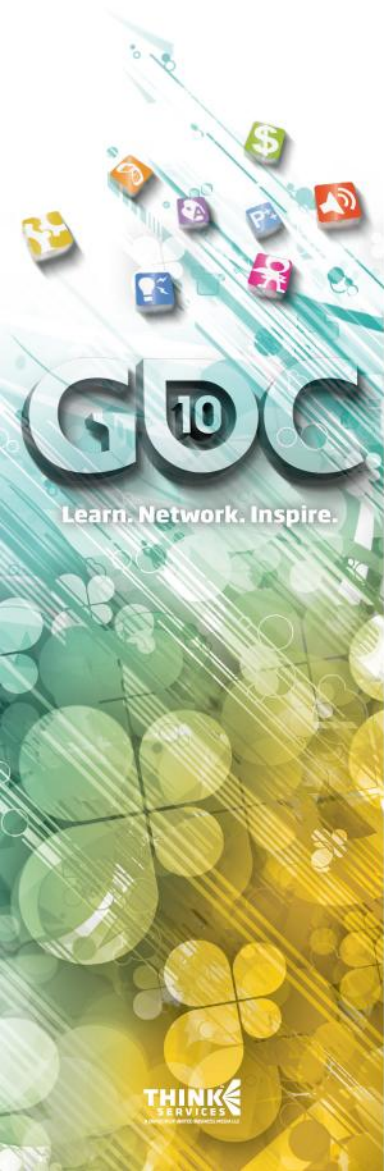


GK_Performance_Tier3
SYMBOLS = 366
PROJECTIONS = 31
BASE = Bishada GX8800



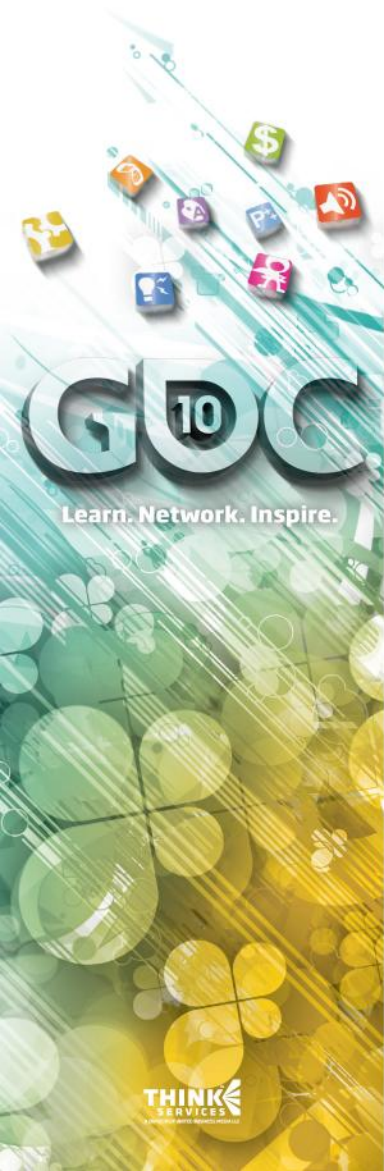
Future Work

- ⌚ Blend-weight morphing
- ⌚ Physics simulation on hair and clothes
- ⌚ Improved hair rendering
- ⌚ DXT Compression
- ⌚ Better descriptor compression
- ⌚ Persistent disk cache



Conclusions

- ④ Customisation needs time and resources
 - Roughly 50% of graphics team workload
 - Significant proportion of memory and CPU/GPU resources dedicated to it
- ④ Get art pipeline sorted first
- ④ Flexibility is key
 - For artists and players
- ④ Distance fields are fantastic



Questions?

maurizio.sciglio@realtimeworlds.com

simon.taylor@realtimeworlds.com

www.realtimeworlds.com

www.apb.com

