Portfolio

Finding Relevant Functions and their Usages

Collin McMillan[1]
Mark Grechanik[2]
Denys Poshyvanyk[1]
Qing Xie[2]
Chen Fu[2]

[1]College of William & Mary
[2]Accenture Technology Labs

Virtues of Programmers:

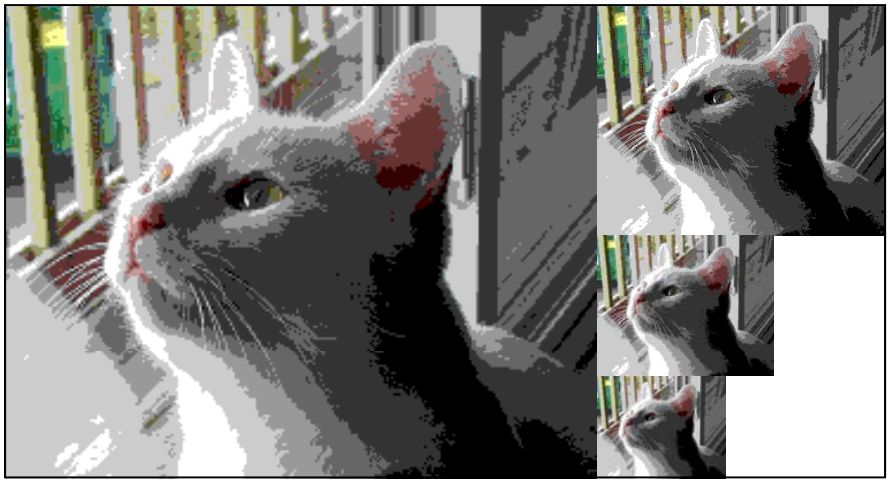
- **Laziness**
- Impatience
- Hubris

Larry Wall,

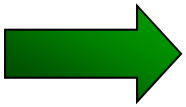
Inventor of Perl

Example Programming Task

Write a utility for dithering mip map images that are used for rendering texture.

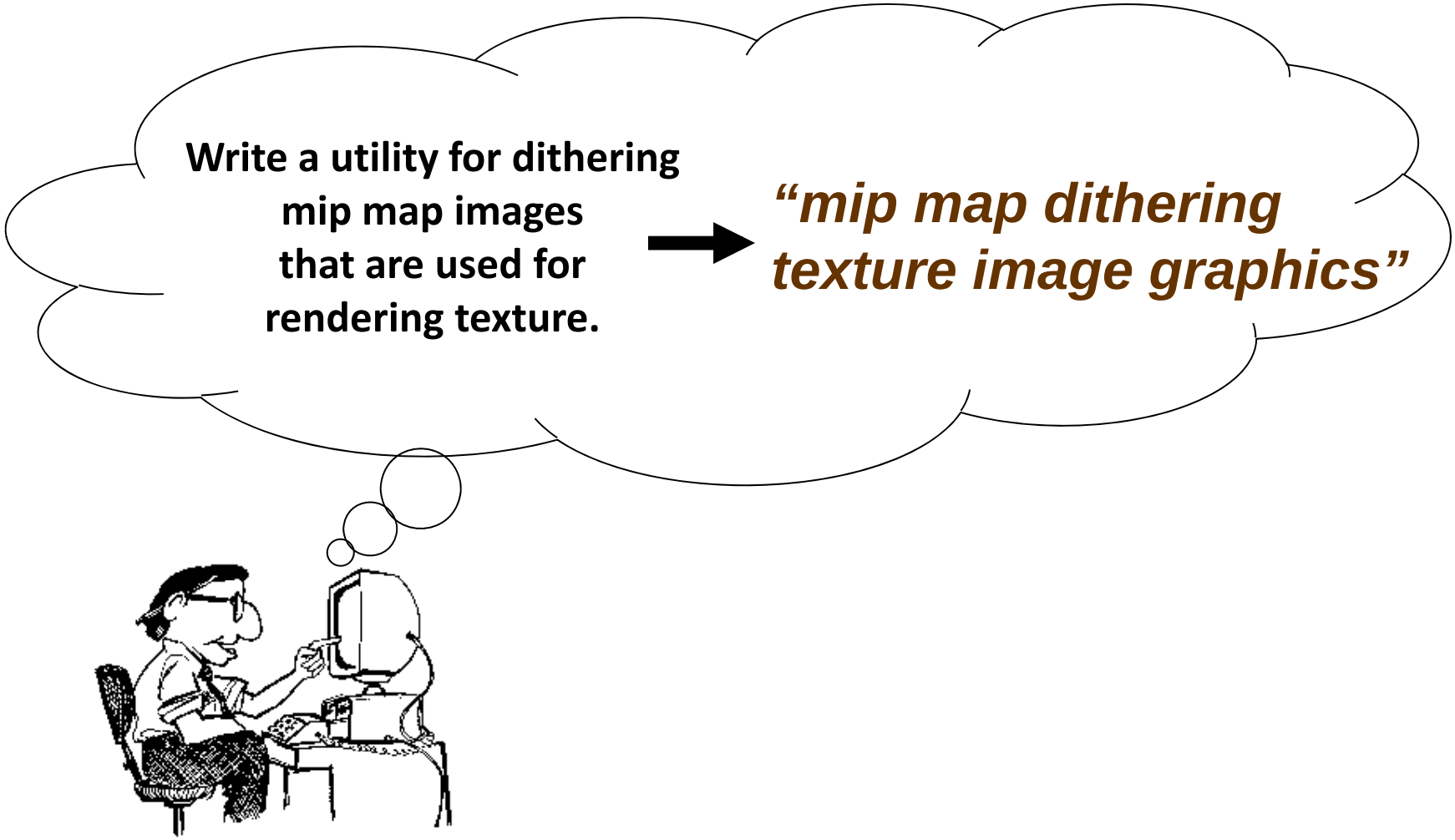


Mip Maps

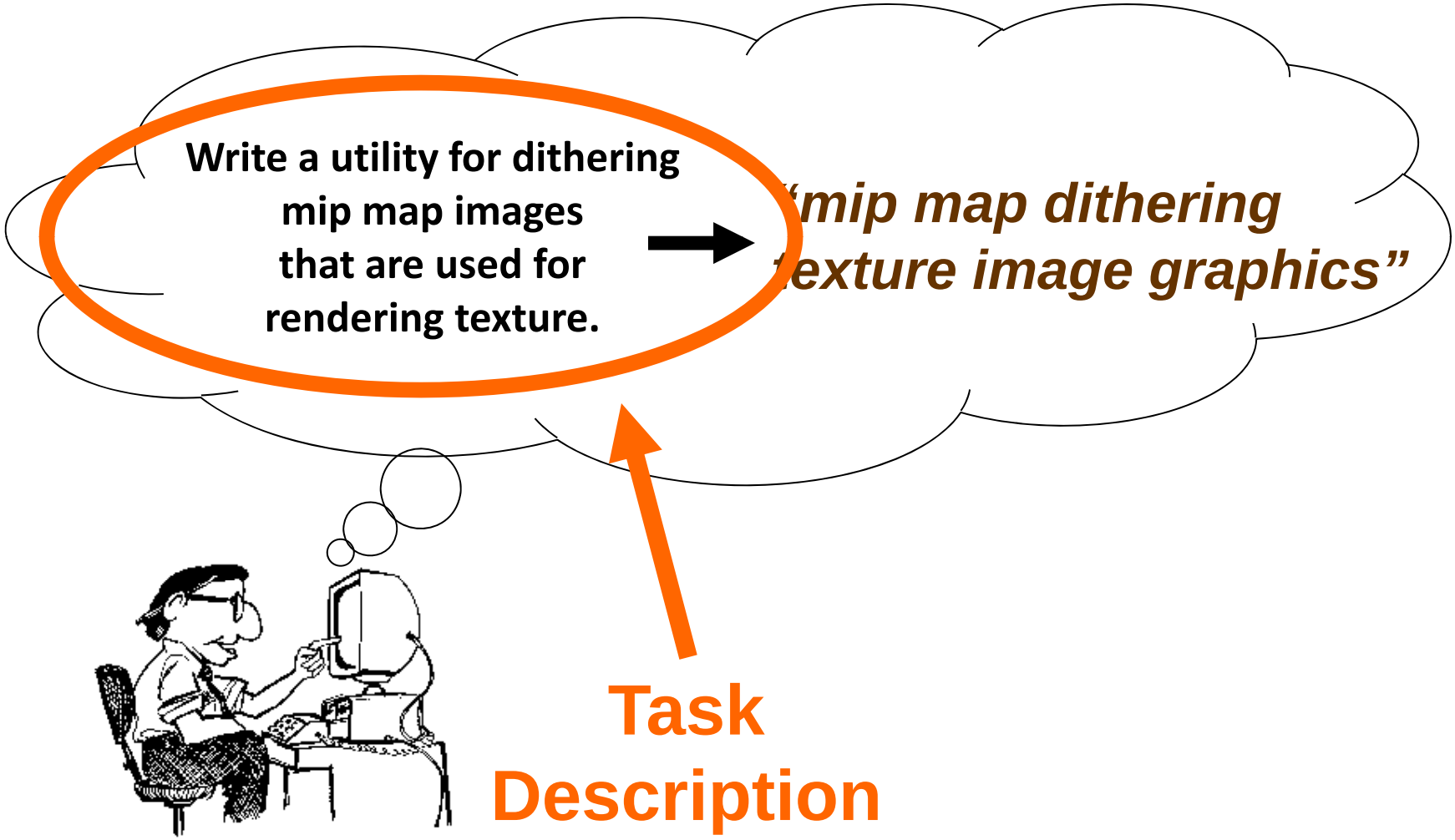


Dithering

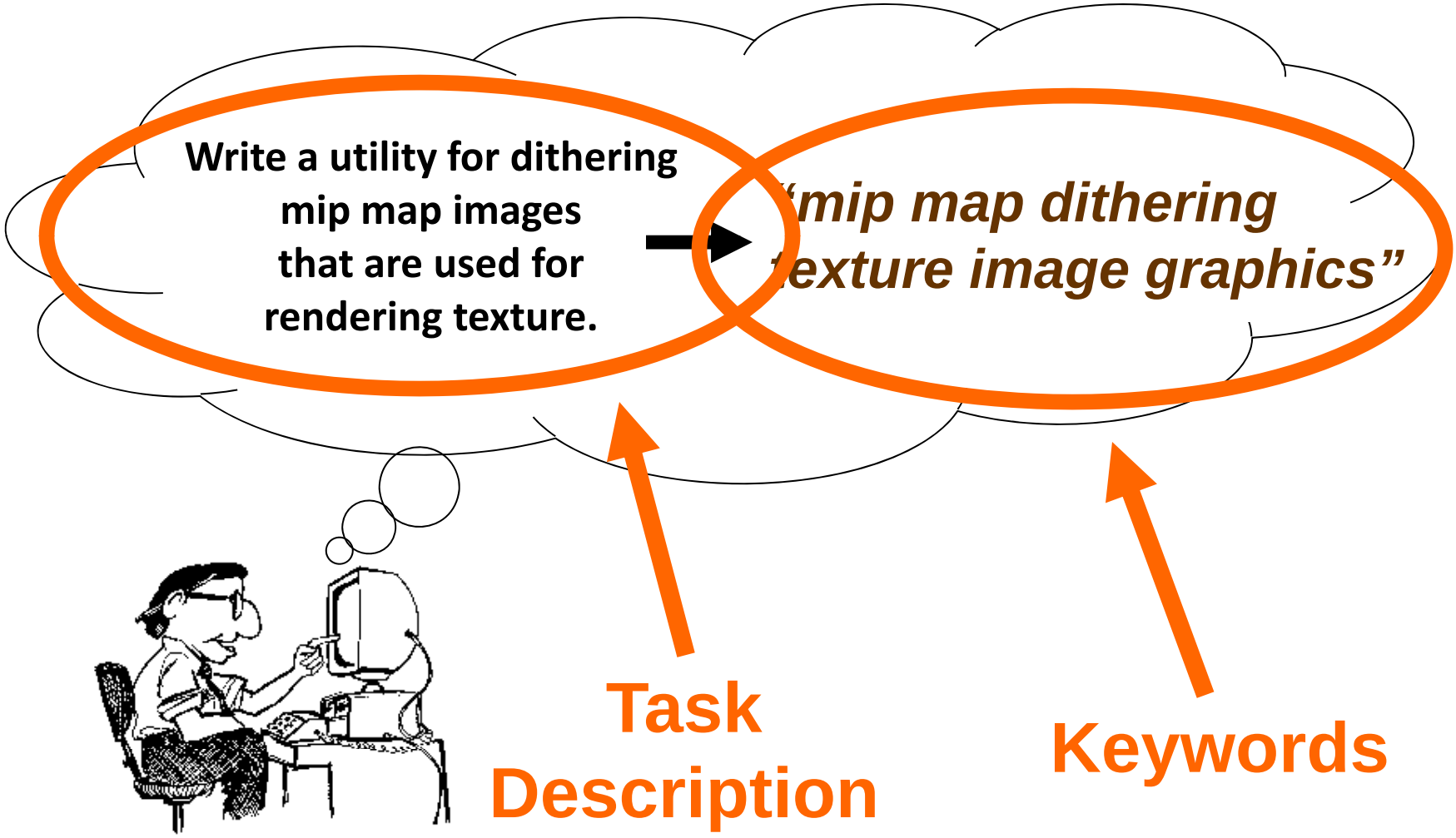
What Programmers Do: Use A Search Engine!



What Programmers Do: Use A Search Engine!



What Programmers Do: Use A Search Engine!

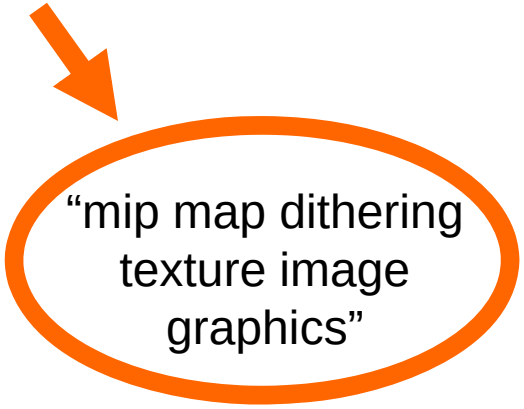


Example Source Code Search Engine

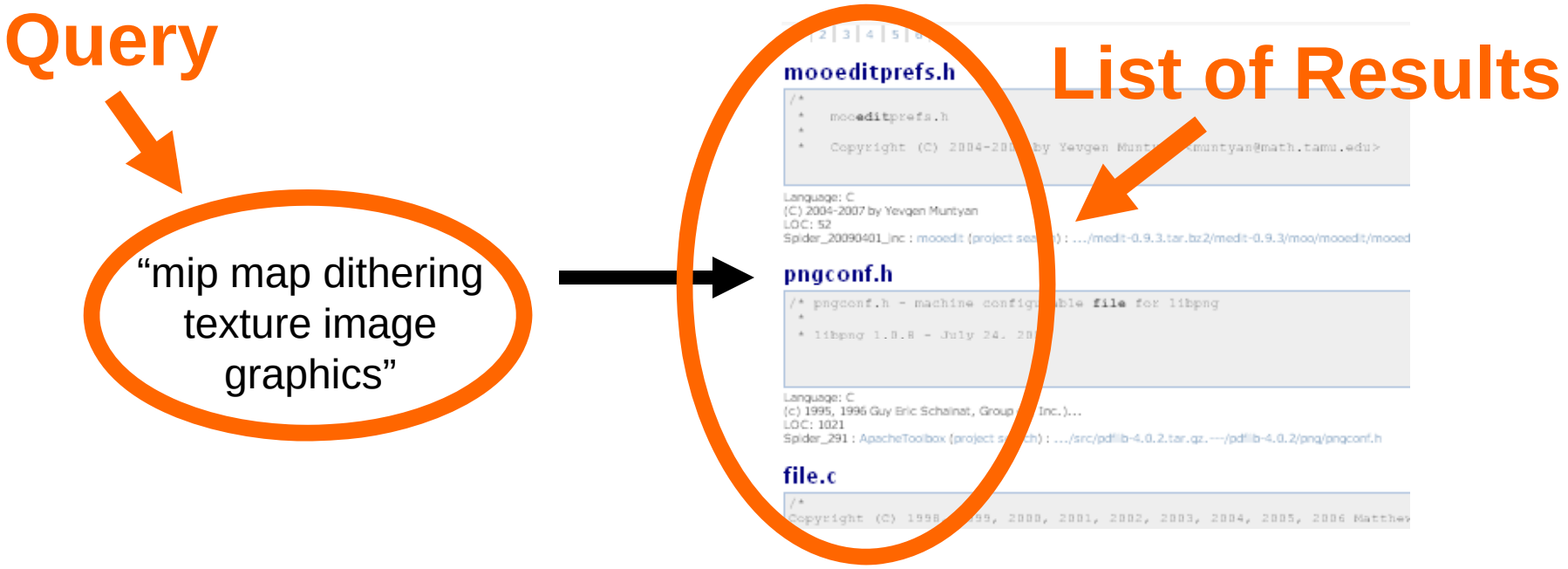
“mip map dithering texture image graphics”

Example Source Code Search Engine

Query



Example Source Code Search Engine



Users Prefer Web Search to Code Search

Of 35 Professional Programmers we surveyed:

- 12 did not search for code online
- 19 did not use code search engines
- 14 cited “irrelevant results” as the reason
- **Preferred Web Search to Code Search!**

Users Prefer Web Search to Code Search

Of 35 Professional Programmers we surveyed:

- 12 did not search for code online
- 19 did not use code search engines
- 14 cited “irrelevant results” as the reason
- **Preferred Web Search to Code Search!**

How Could This Happen?

Search Result - A Relevant Function

```
111 /*
112  * Perform variance-based color quantization on a 24-bit image.
113  */
114 int
115 txMipPal256(TxMip *pxMip, TxMip *txMip, int format, FxU32 dither, FxU32 compression)
116 {
117     int         w, h;
118     int         i;                          /* Counter */
119     int         OutColors;              /* # of entries computed */
120     int         Colormax;               /* quantized full-intensity */
121     float       Cfactor;                /* Conversion factor */
122 #if 0
123     uchar       *rgbmap;                /* how to map colors to palette indices */
124 #else
125     static uchar rgbmap[(1<<NBITS)*(1<<NBITS)*(1<<NBITS)]; /* how to map colors to pa
126 #endif
127     int         pixsize;
```

Search Result - A Relevant Function

```
111 /*
112  * Perform variance-based color quantization on a 24-bit image.
113  */
114 int
115 txMipPal256(TxMip *pxMip, TxMip *txMip, int format, FxU32 dither, FxU32 compression)
116 {
117     int         w, h;
118     int         i;                          /* Counter */
119     int         OutColors;                  /* # of entries computed */
120     int         Colormax;                   /* quantized full-intensity */
121     float       Cfactor;                    /* Conversion factor */
122 #if 0
123     uchar       *rgbmap;                    /* how to map colors to palette indice  */
124 #else
125     static uchar rgbmap[(1<<NBITS)*(1<<NBITS)*(1<<NBITS)]; /* how to map colors to pa
126 #endif
127     int         pixsize;
```

Keywords in Comments

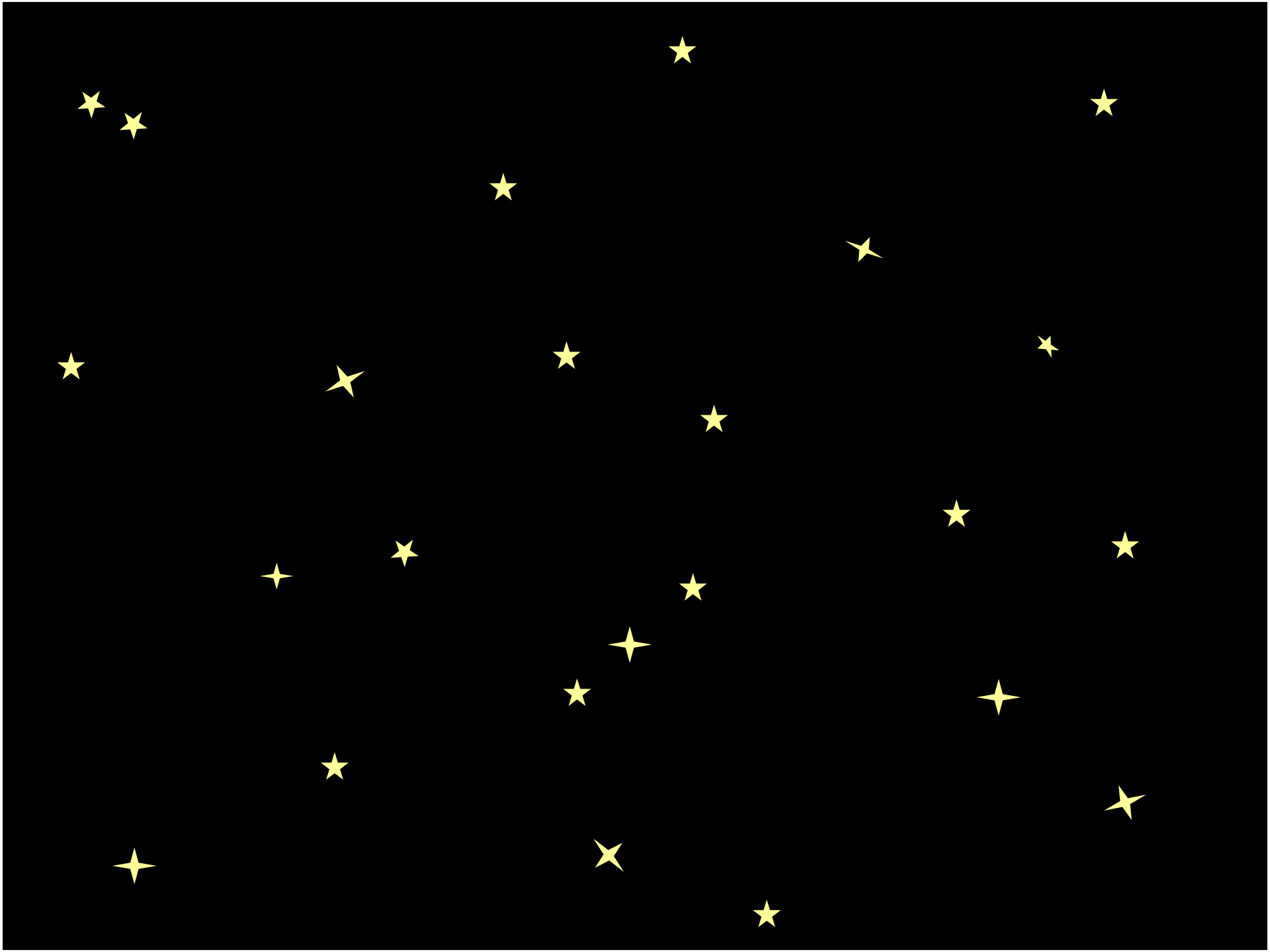
Find Usages of Relevant Functions

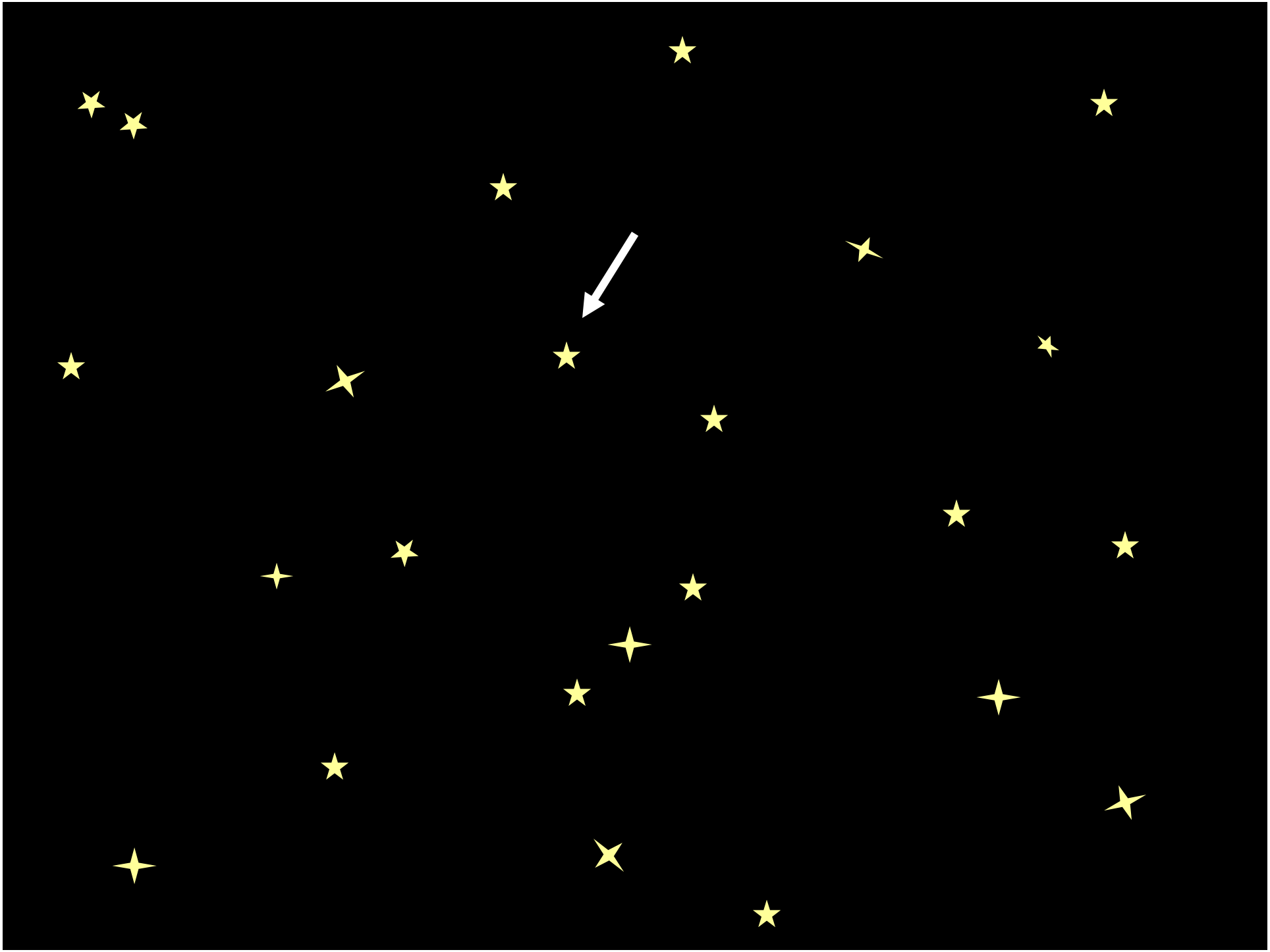
```
392       * Get a 256 color palette, to be used as samples
393       * Incidentally, convert src 32 bit image to dst 8 bit indexed image,
394       * with indices referring to the 256 color palette.
395       * Also incidentally, pack the alpha channel if necessary.
396       */
397      if( txVerbose )
398        {
399          printf("NCC Neural nets..."); fflush(stdout);
400        }
401      pxMip->format = (format == GR_TEXFMT_YIQ_422) ? GR_TEXFMT_P_8 :
402          GR_TEXFMT_AP_88;
403      ncolors = txMipPal256(pxMip, txMip, pxMip->format, 0, 0);
404      if( txVerbose )
405        {
406          printf("%d samples...", ncolors); fflush(stdout);
407        }
408      txMapPal256toYAB((FxU32 *)yabTable, (FxU8 *)map, ncolors, (FxU32 *)pxMip->pal);
409      if( txVerbose )
410        {
411          printf("eMax=(%3ld%3ld%3ld)...eAvg=(%3ld%3ld%3ld)\n",
412                 errG, errR, errB,
413                 totG/ncolors, totR/ncolors, totB/ncolors
```

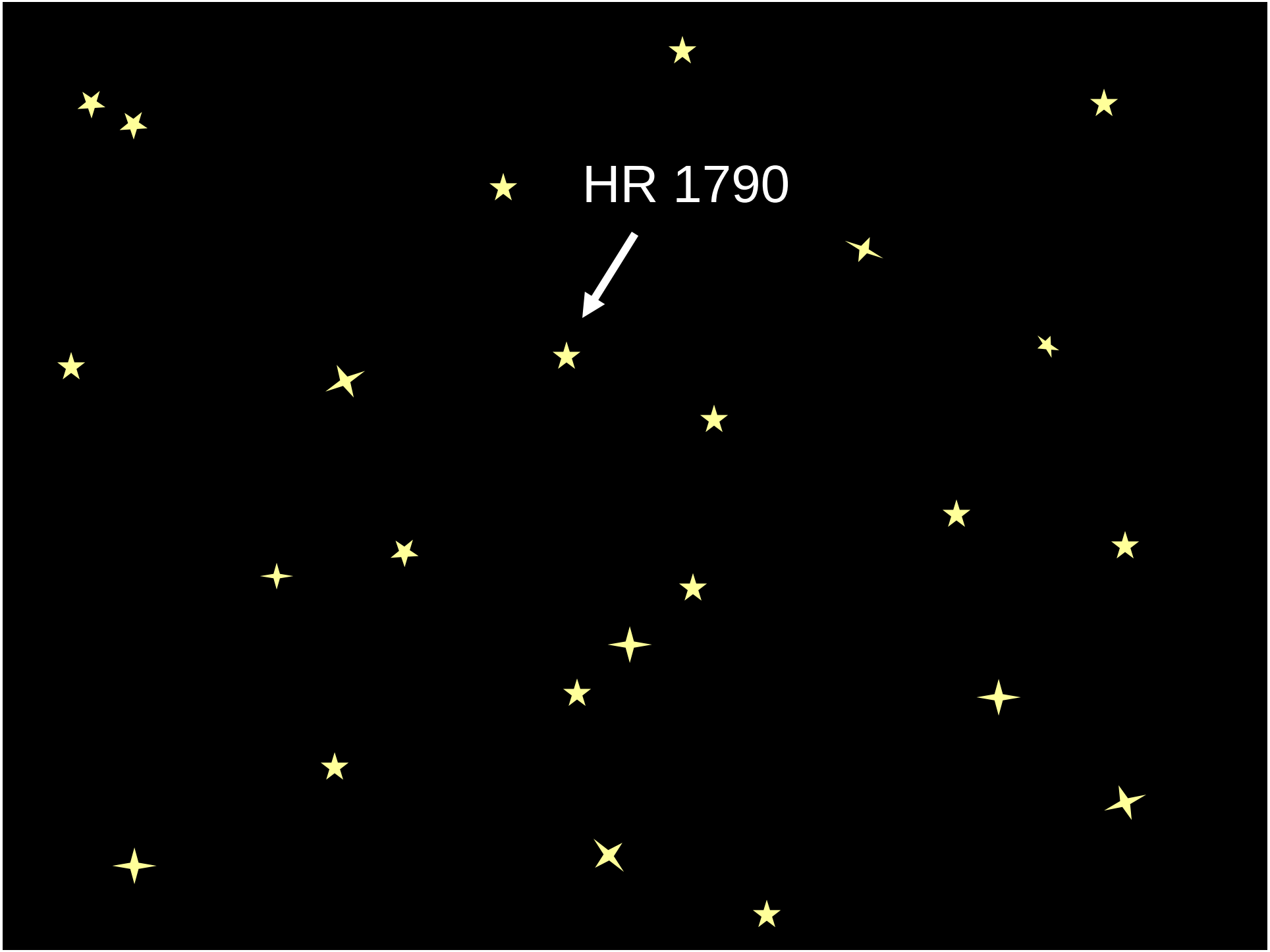
Find Usages of Relevant Functions

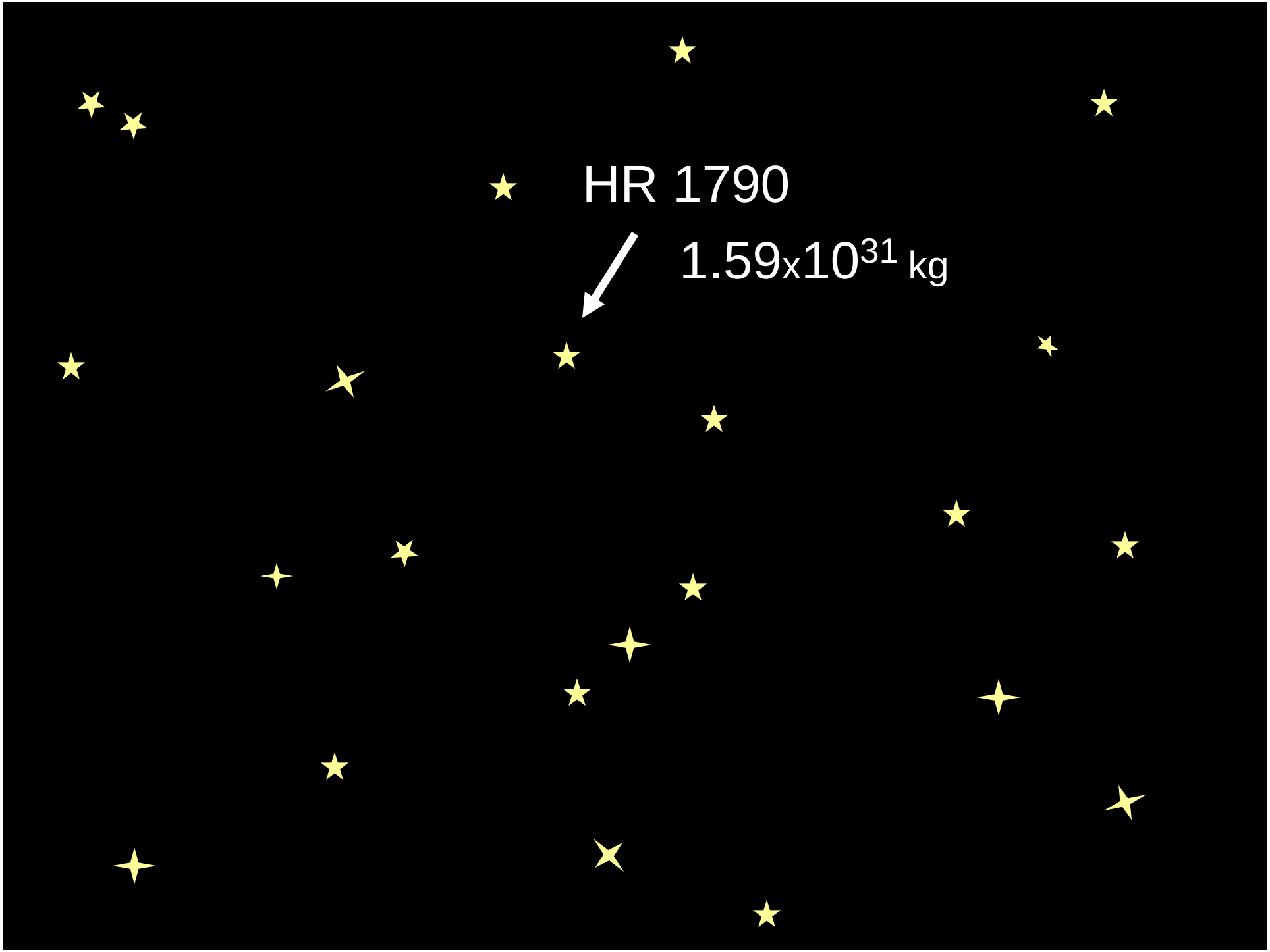
```
392       * Get a 256 color palette, to be used as samples
393       * Incidentally, convert src 32 bit image to dst 8 bit indexed image,
394       * with indices referring to the 256 color palette.
395       * Also incidentally, pack the alpha channel if necessary.
396       */
397      if( txVerbose )
398        {
399          printf("NCC Neural nets..."); fflush(stdout);
400        }
401      pxMip->format = (format == GR_TEXFMT_YIQ_422) ? GR_TEXFMT_P_8 :
402          GR_TEXFMT_AP_88;
403      ncolors = txMipPal256(pxMip, txMip, pxMip->format, 0, 0);
404      if( txVerbose )
405        {
406          printf("%d samples...", ncolors); fflush(stdout);
407        }
408      txMapPal256toYAB(FxU32 *)yabTable, (FxU8 *)map, ncolors, (FxU32 *)pxMip->pal);
409      if( txVerbose )
410        {
411          printf("eMax=(%3ld%3ld%3ld)...eAvg=(%3ld%3ld%3ld)\n",
412                 errG, errR, errB,
413                 totG/ncolors, totR/ncolors, totB/ncolors
```

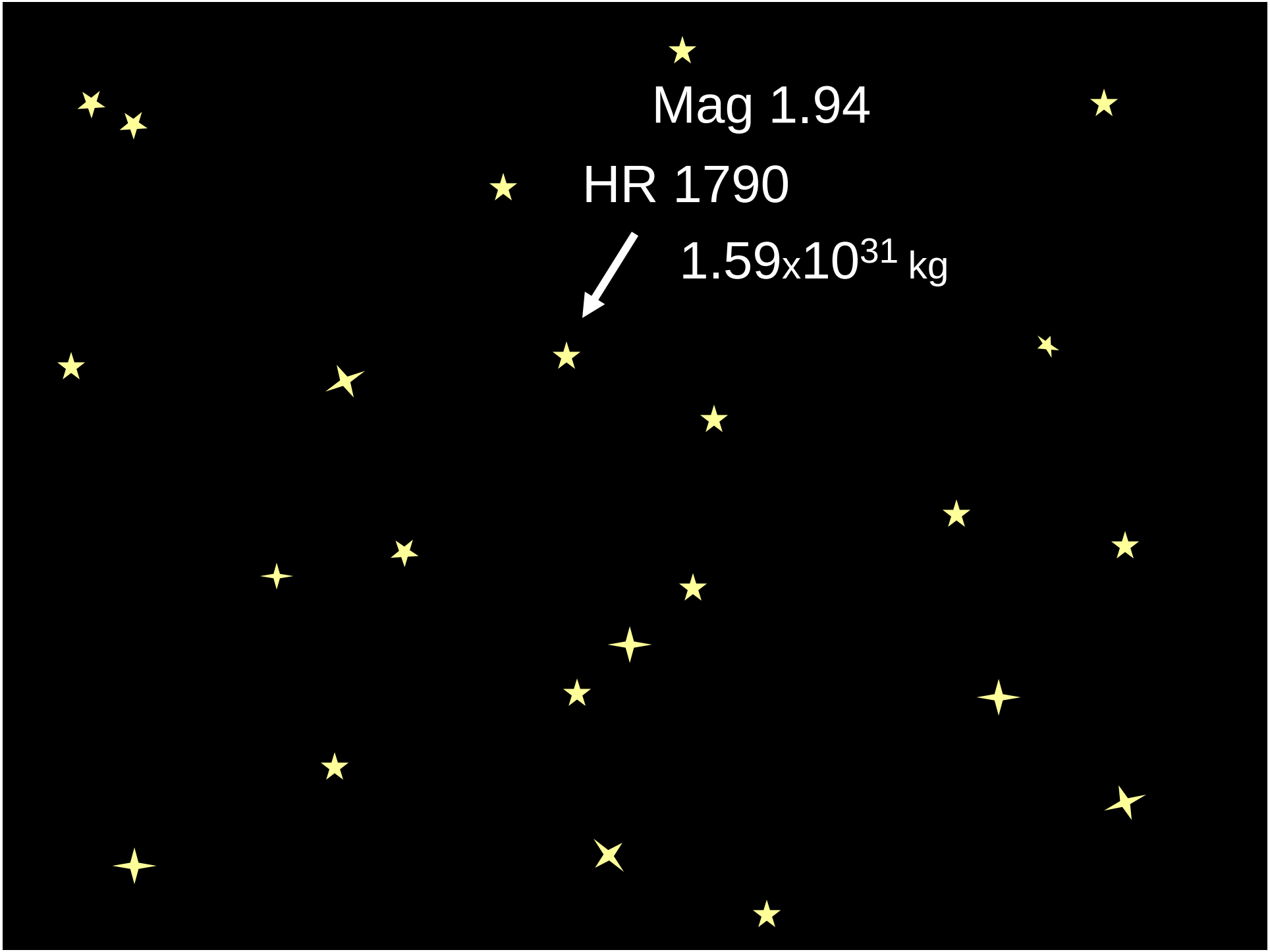
Function Called Here

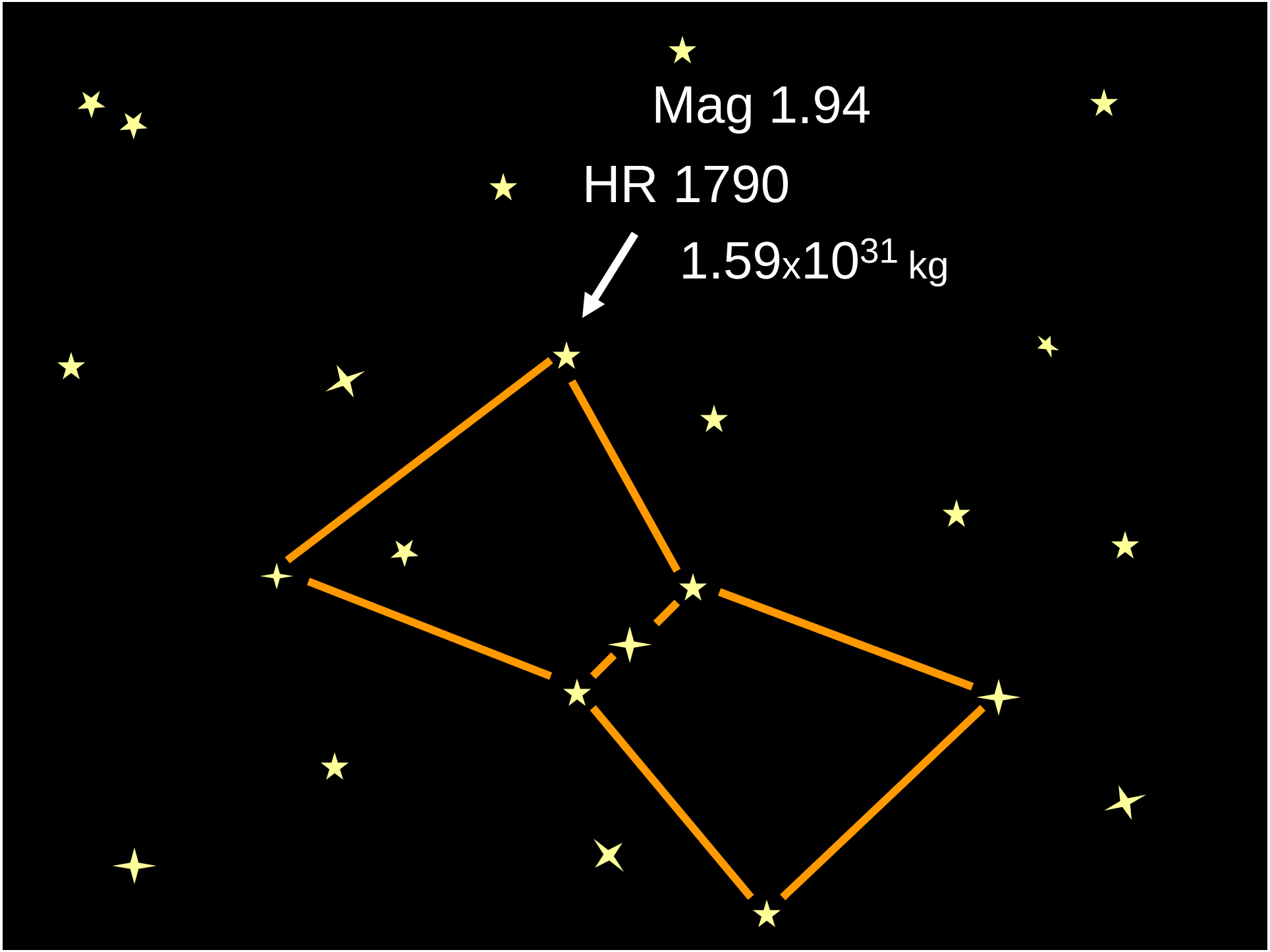


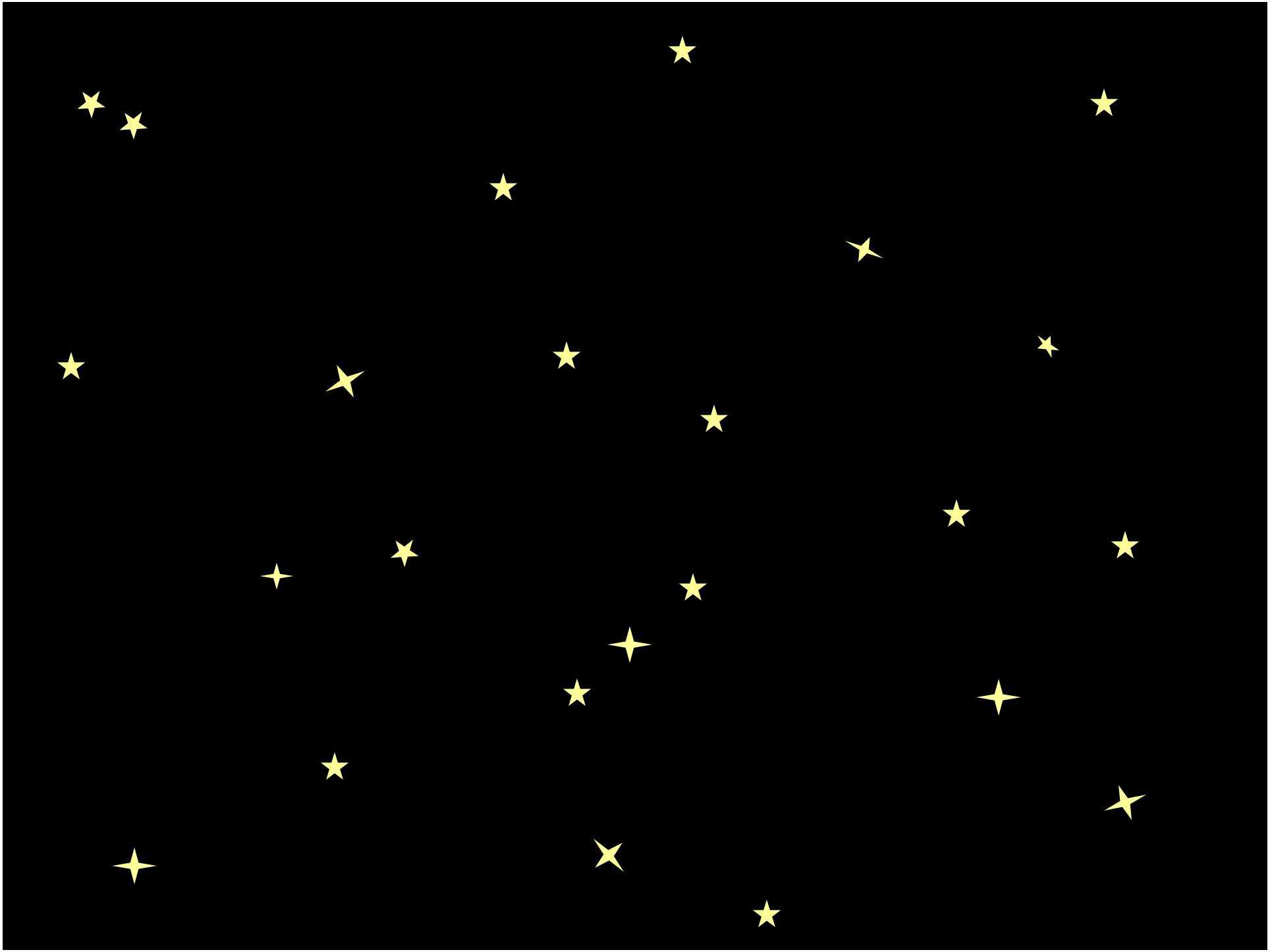


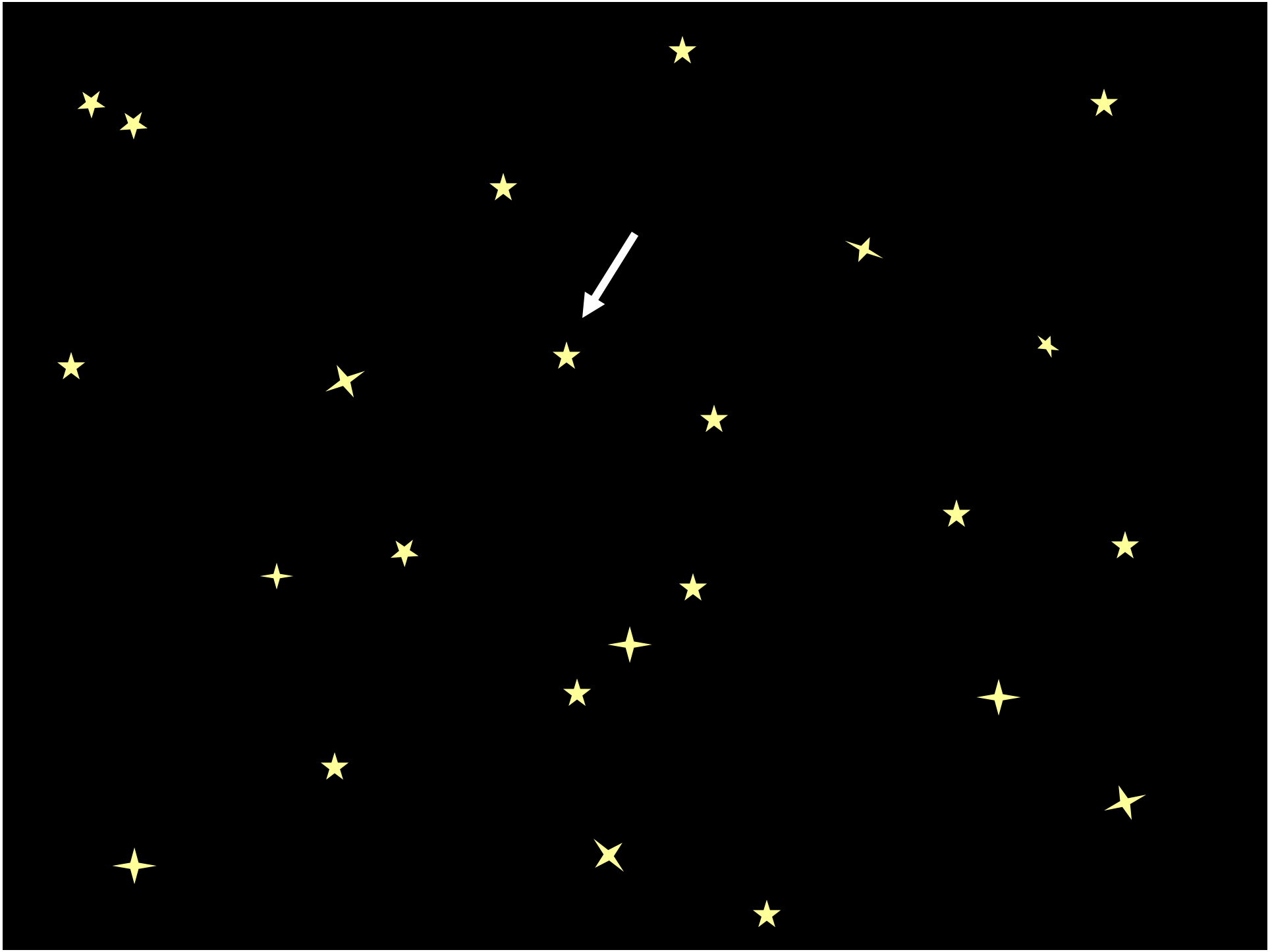

HR 1790

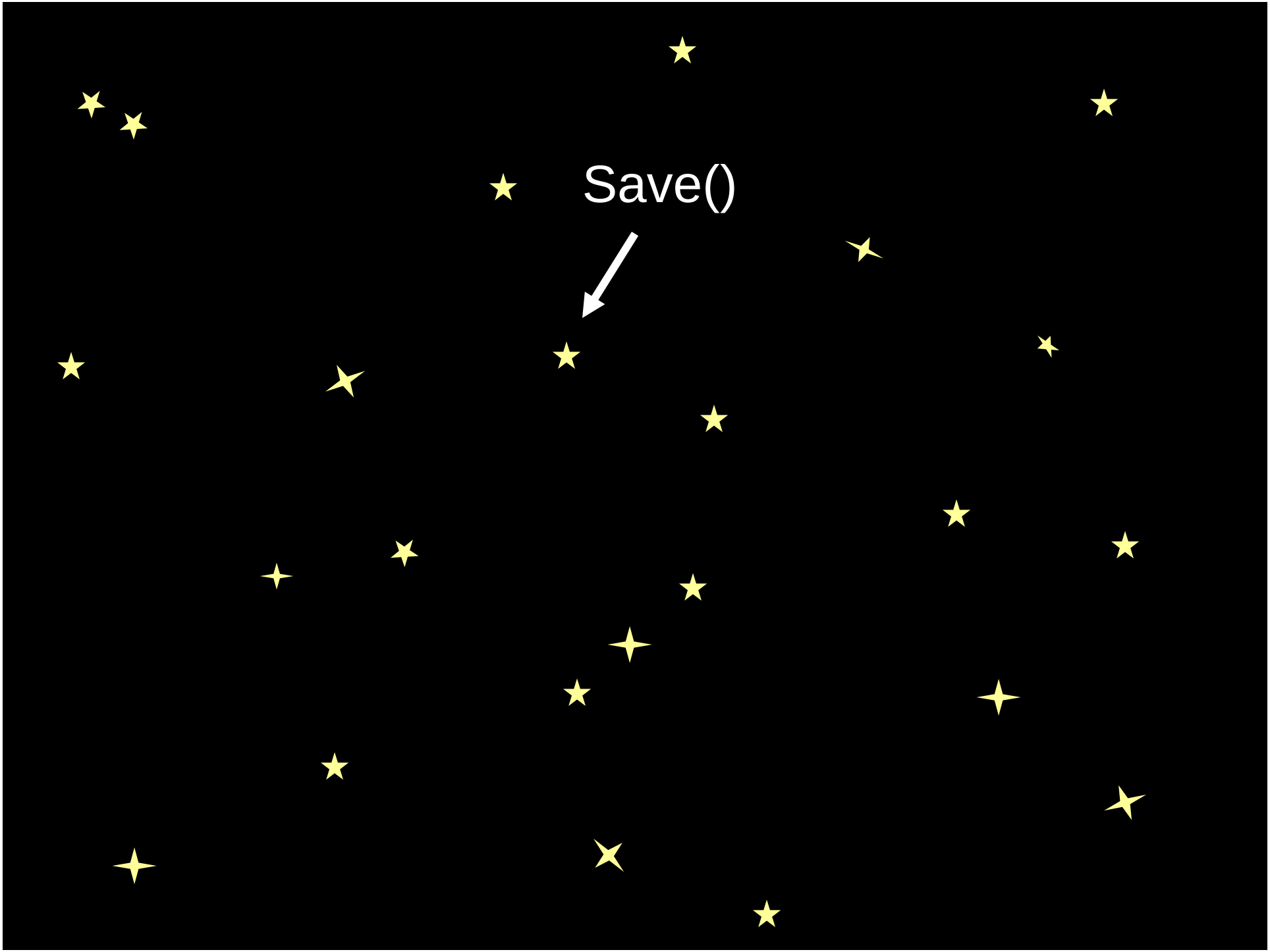

HR 1790
1.59x10^31 kg

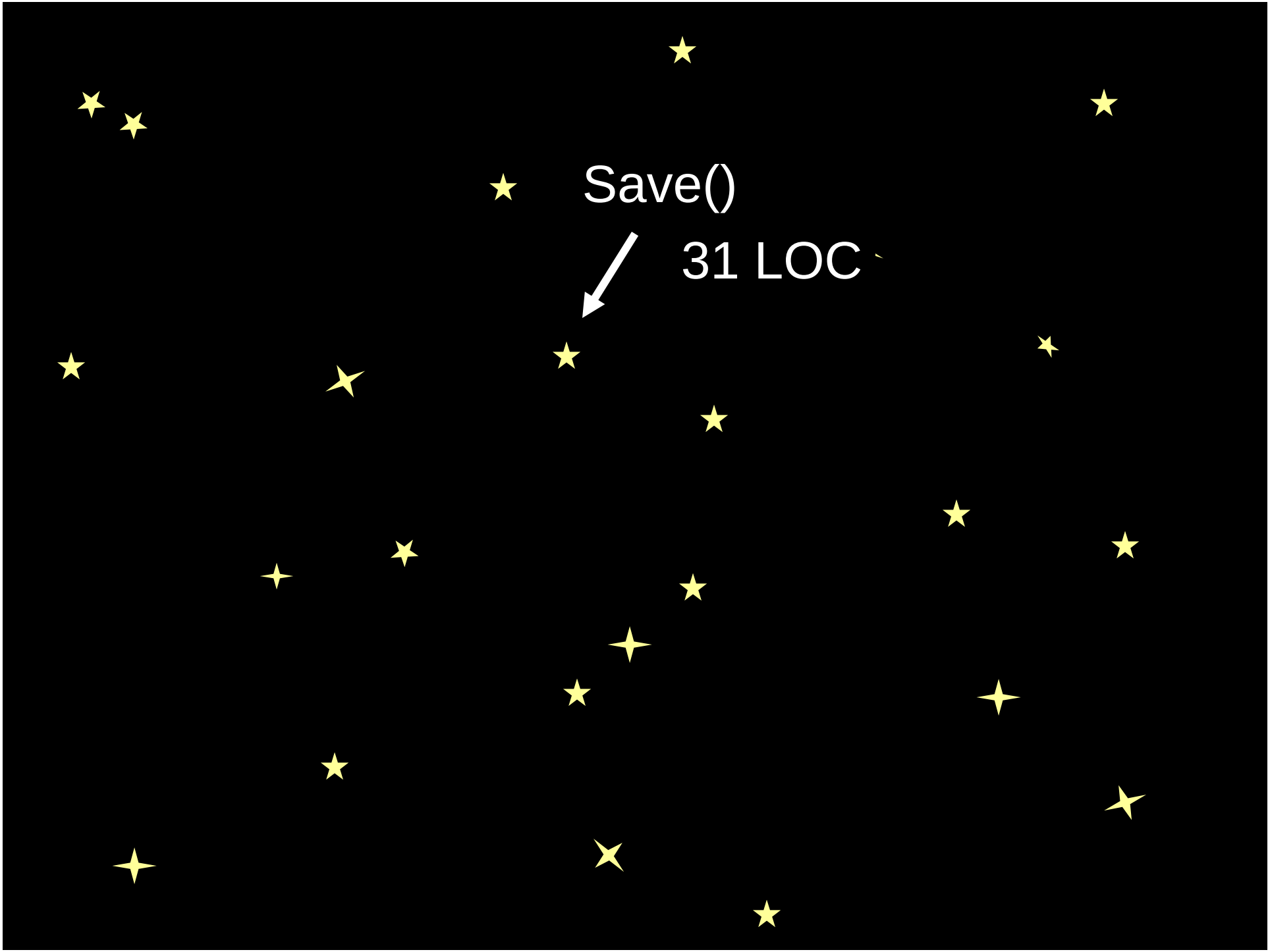

Mag 1.94
HR 1790
1.59x10^31 kg

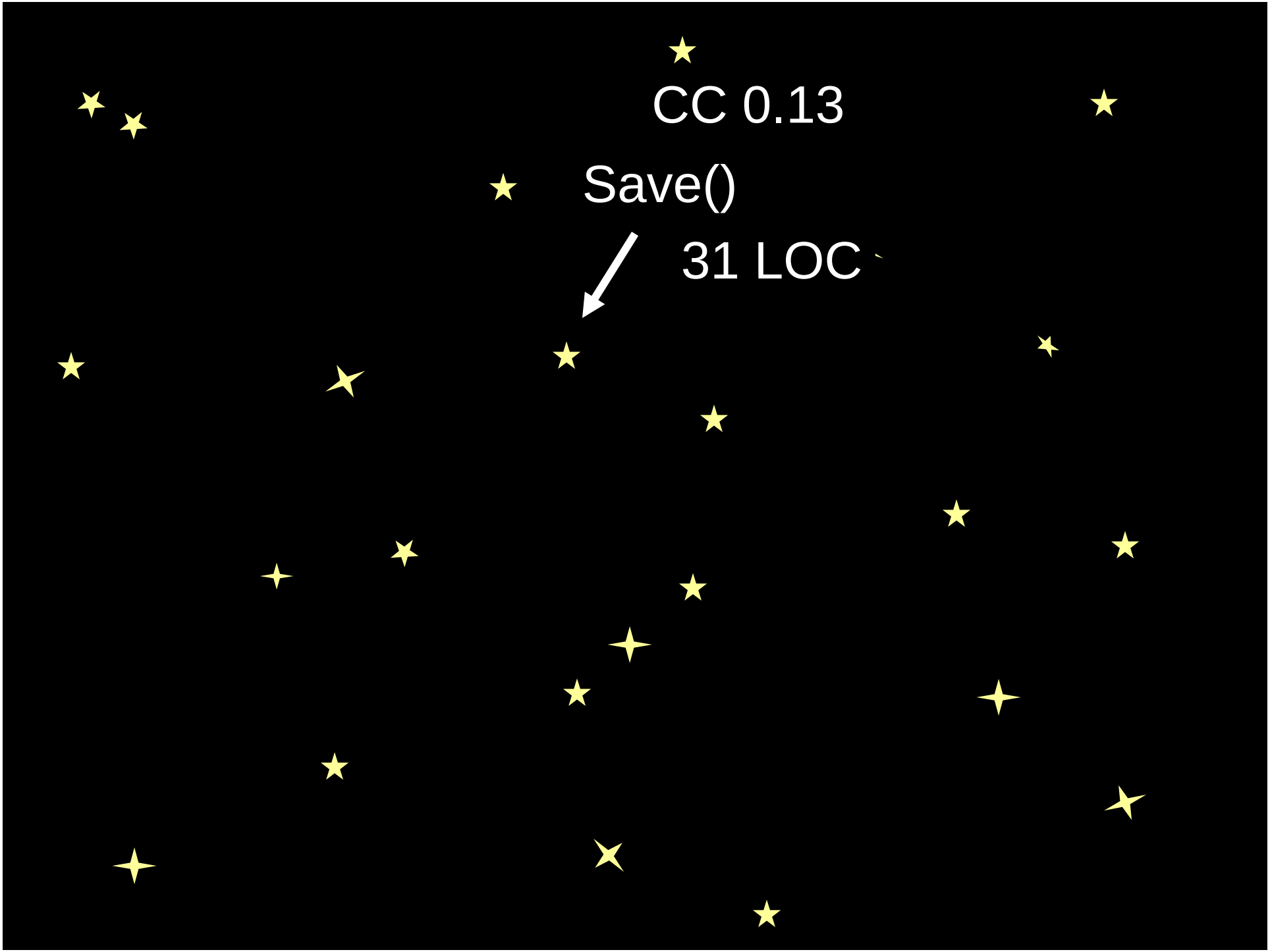

Mag 1.94
HR 1790
1.59x10^31 kg

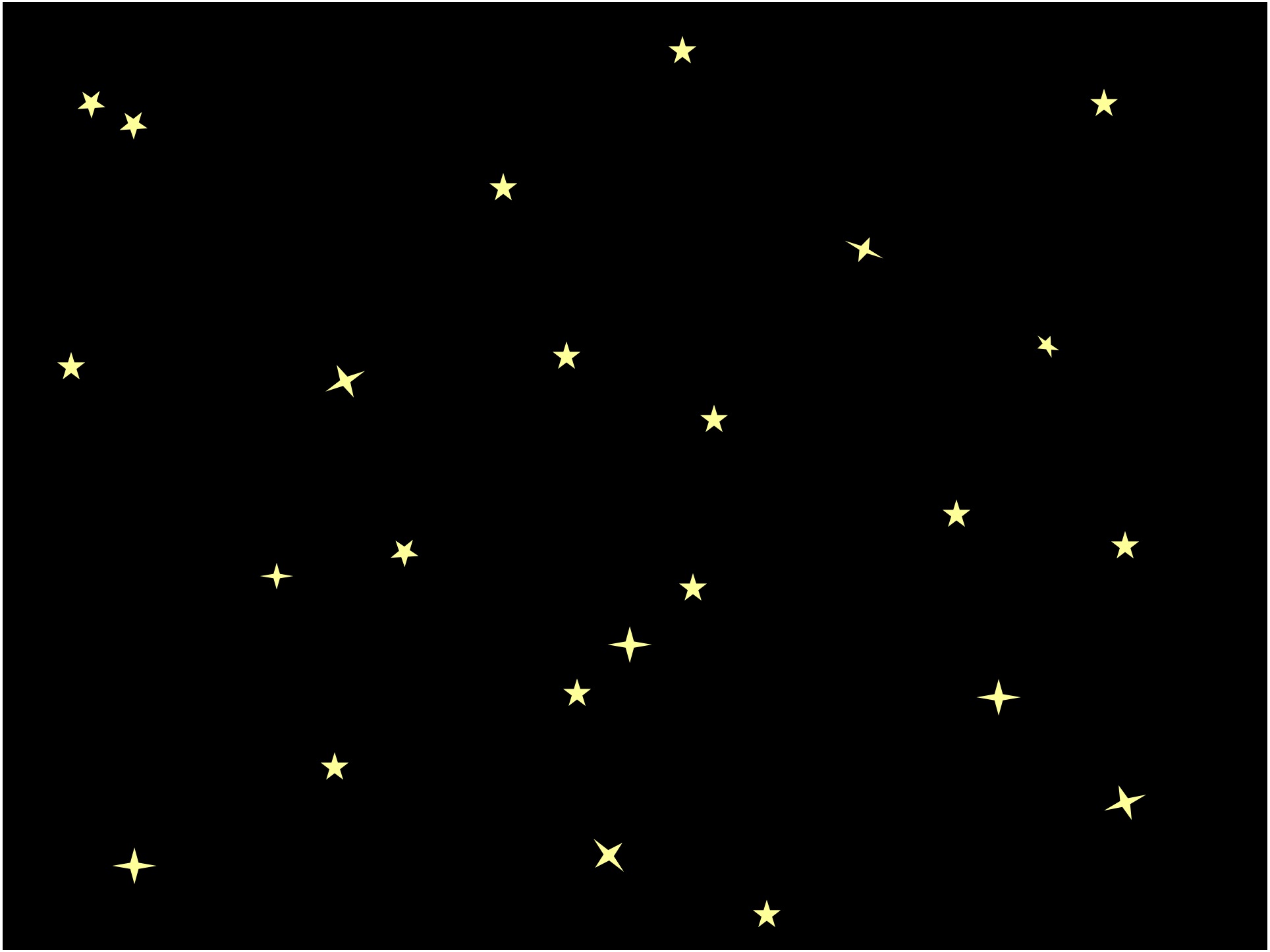


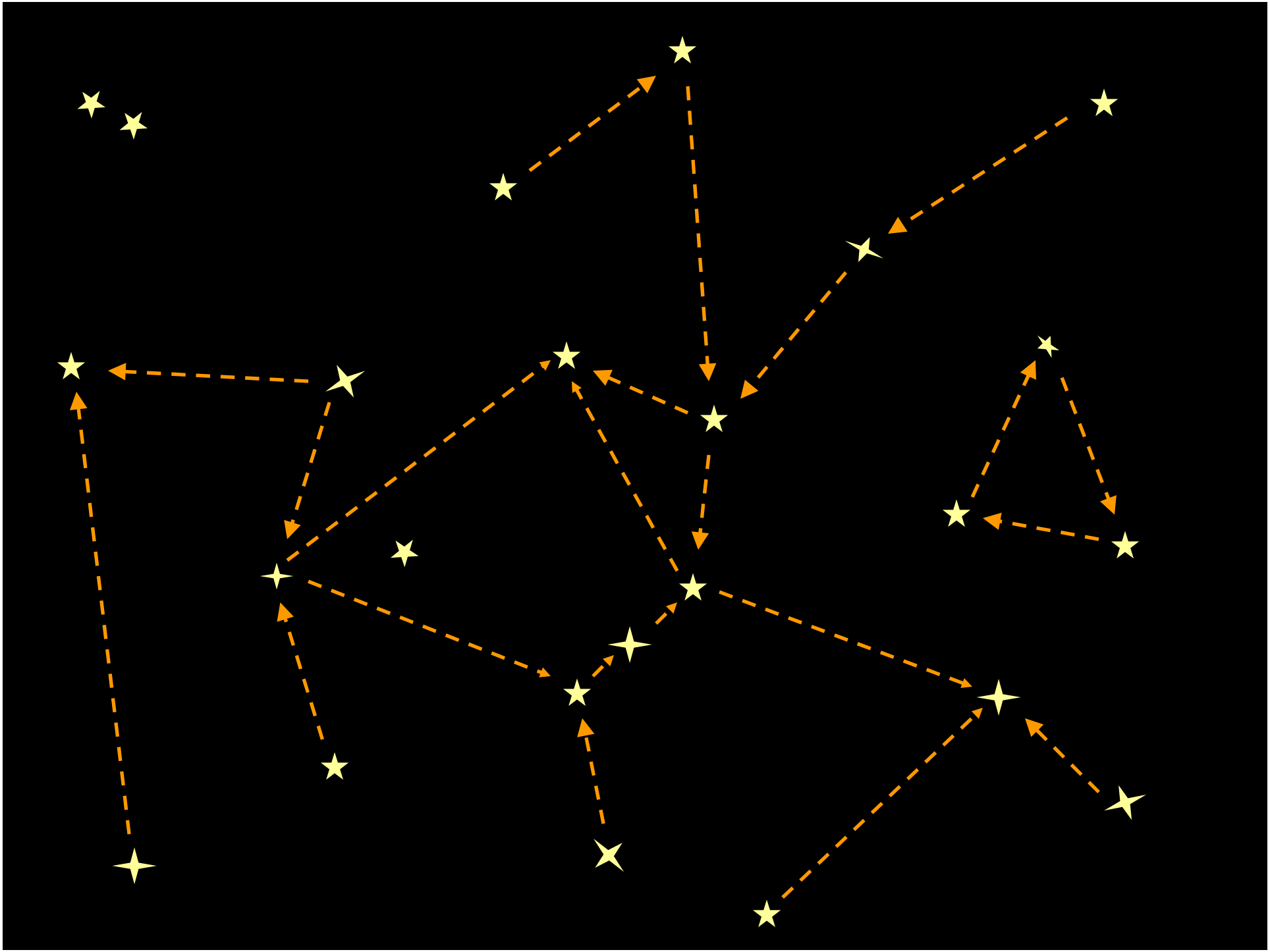


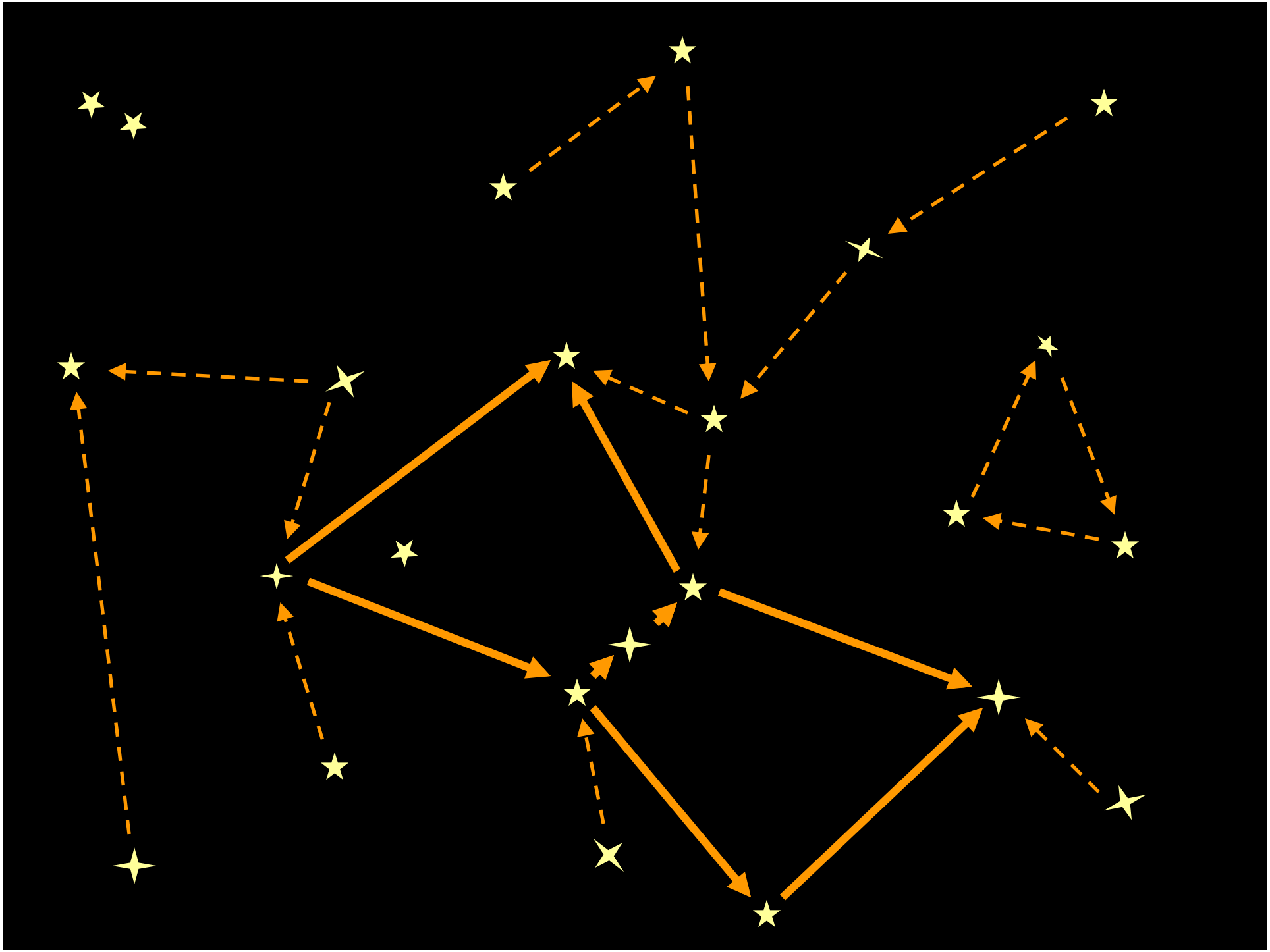

Save()


Save()
31 LOC


CC 0.13
Save()
31 LOC







Our Search Engine: Portfolio

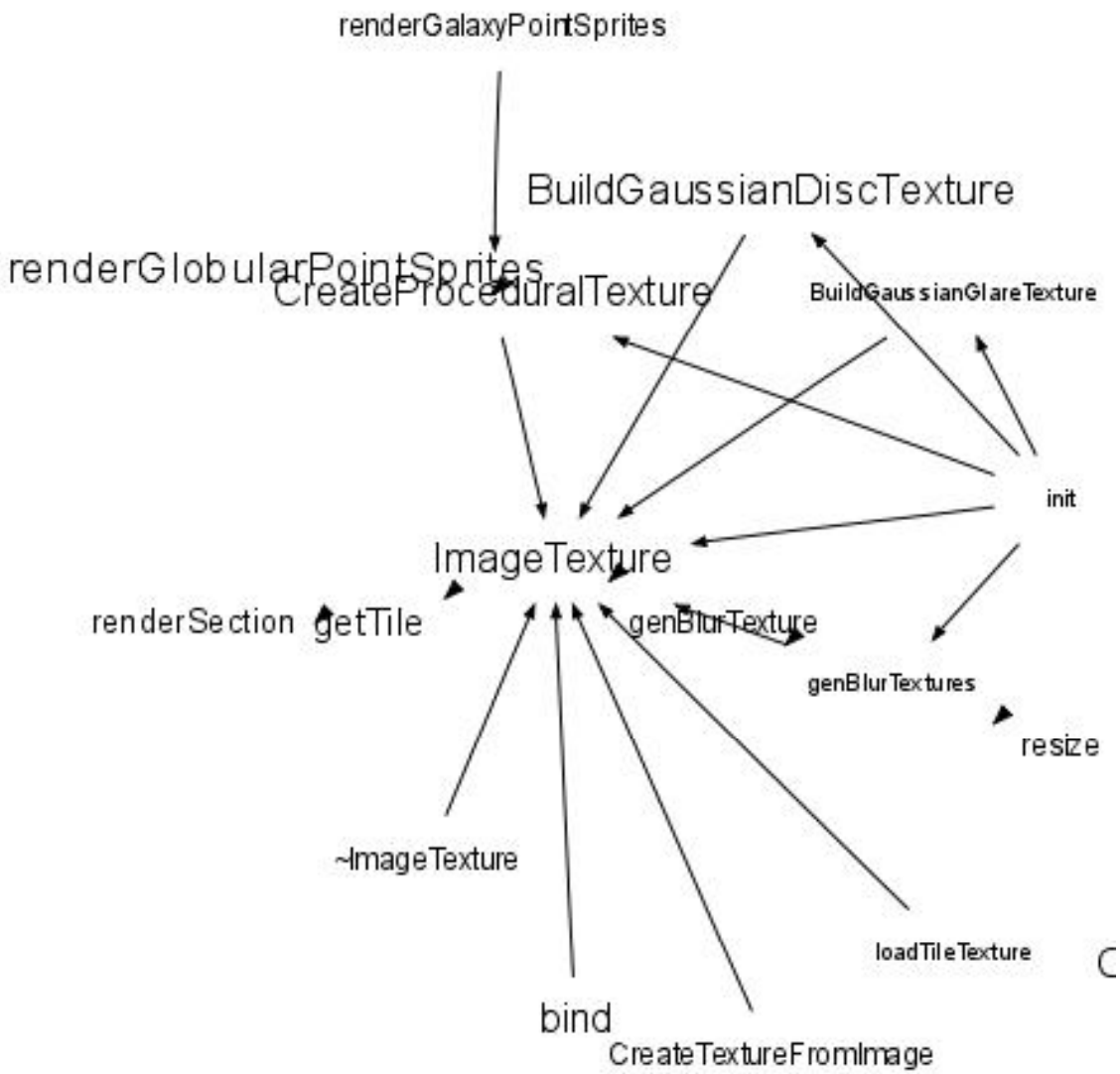
Our Contribution, Two Search Models:

1) Navigation Model for Software
2) Association Model for Software

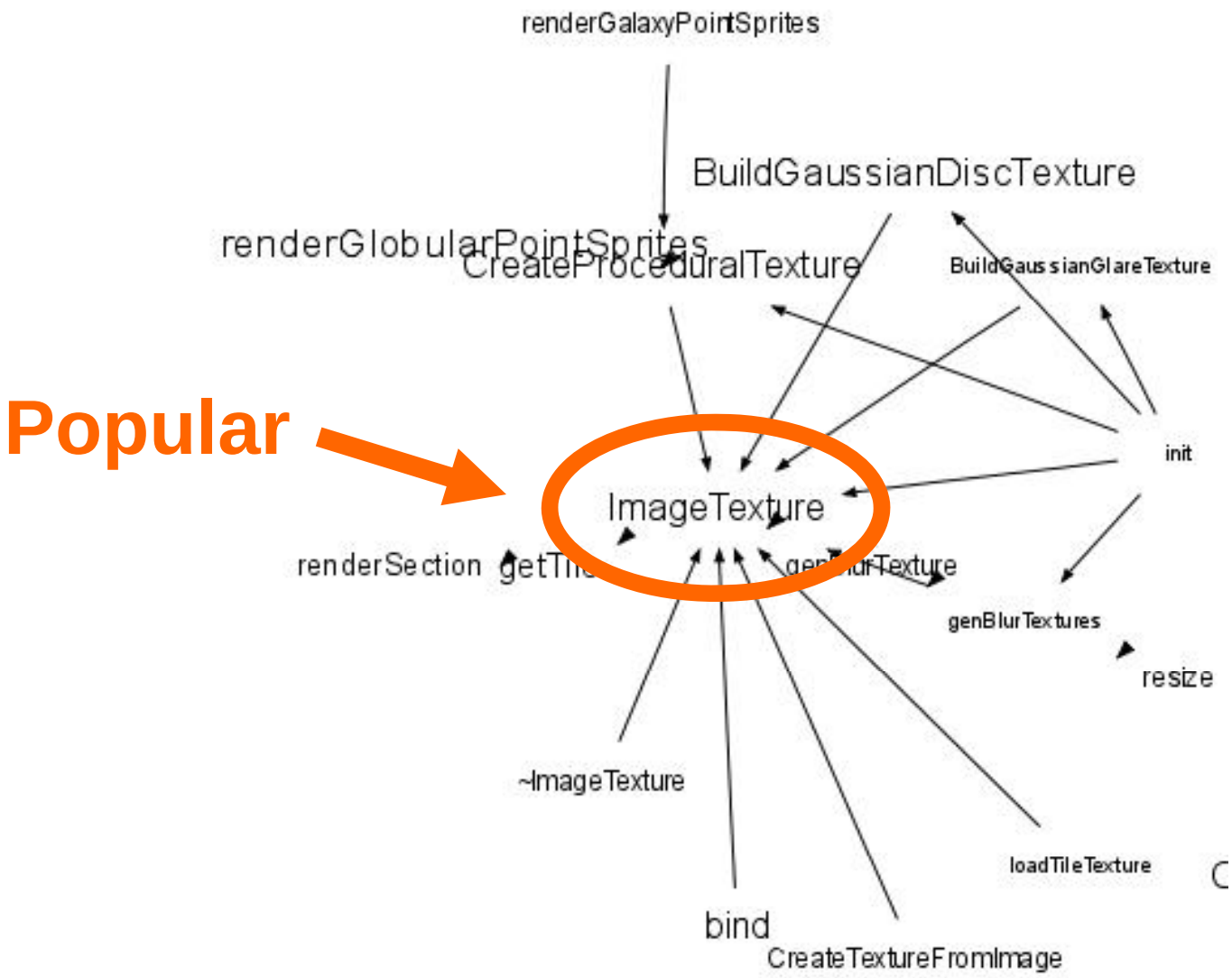
Navigation Model

Navigation Model - PageRank

Navigation Model - PageRank

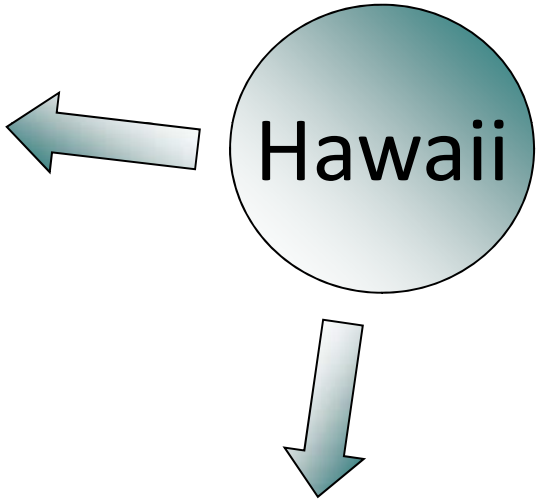


Navigation Model - PageRank

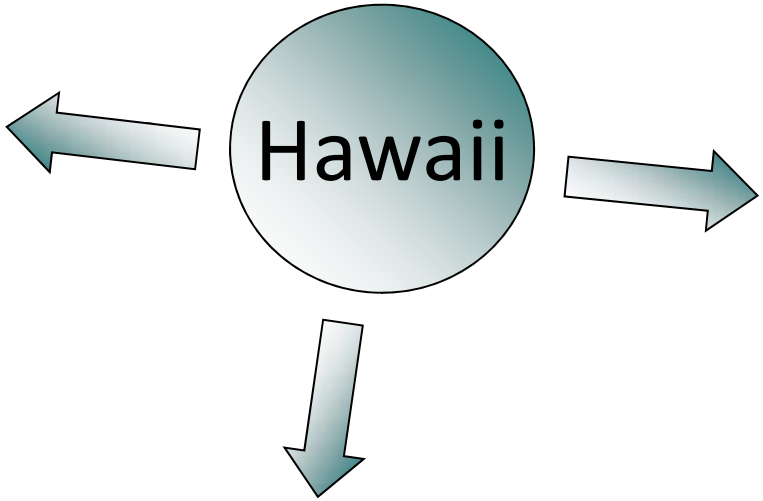


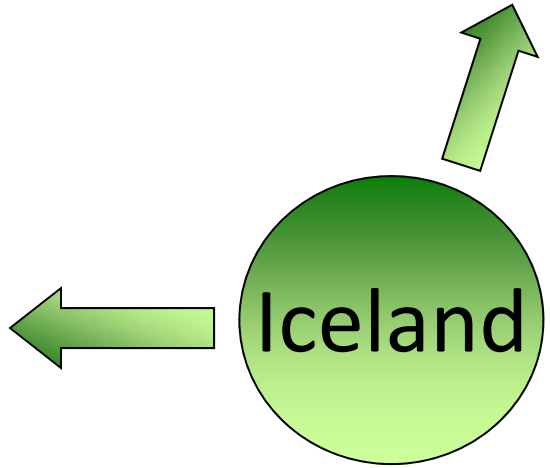
Spreading Activation in Associative Networks

Spreading Activation in Associative Networks



Spreading Activation in Associative Networks





In Portfolio: Function calls are associations

`initRenderer`

`LoadTextureFromFile`

`GlareTexture`

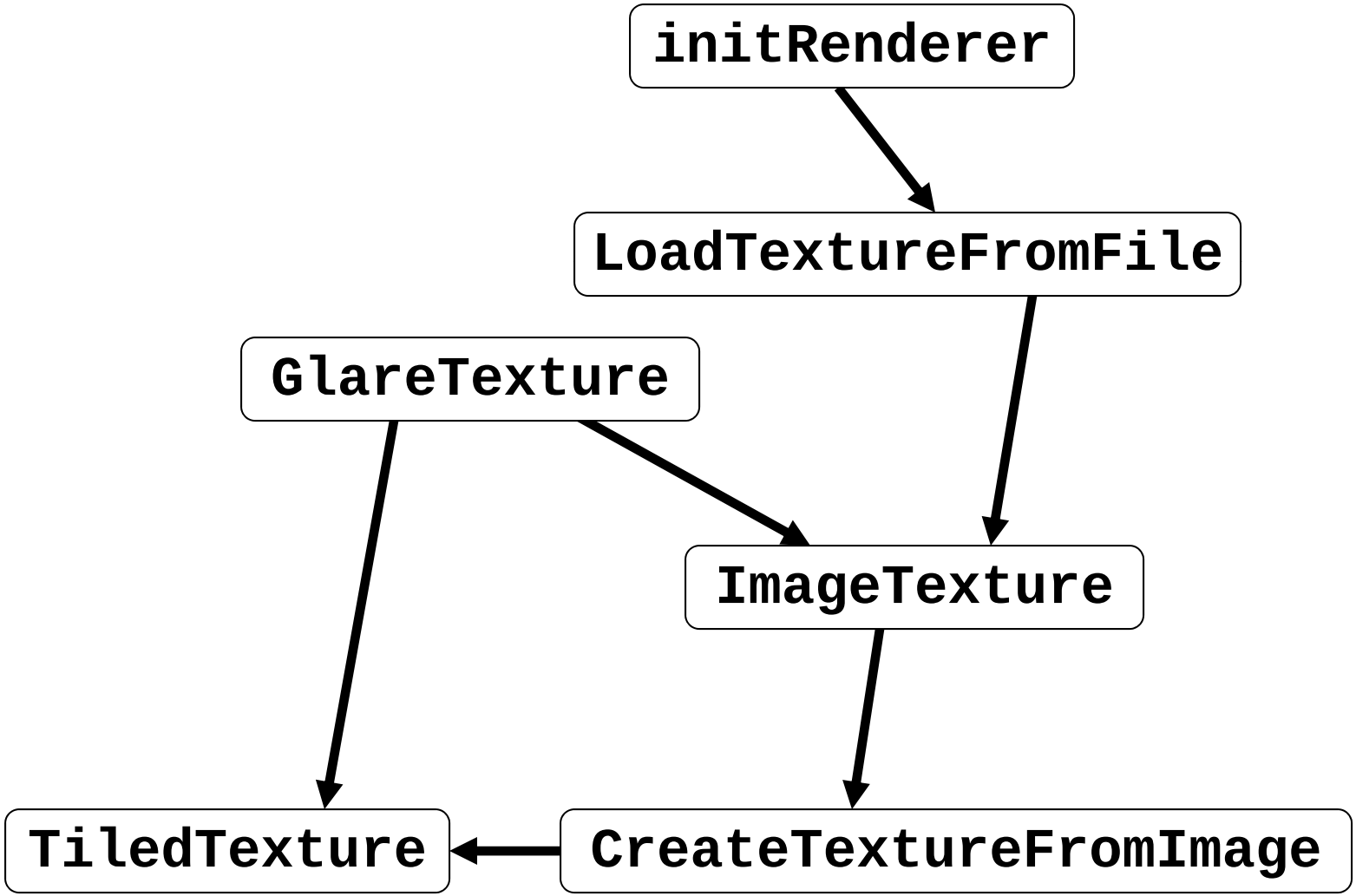
`ImageTexture`

`TiledTexture`

`CreateTextureFromImage`

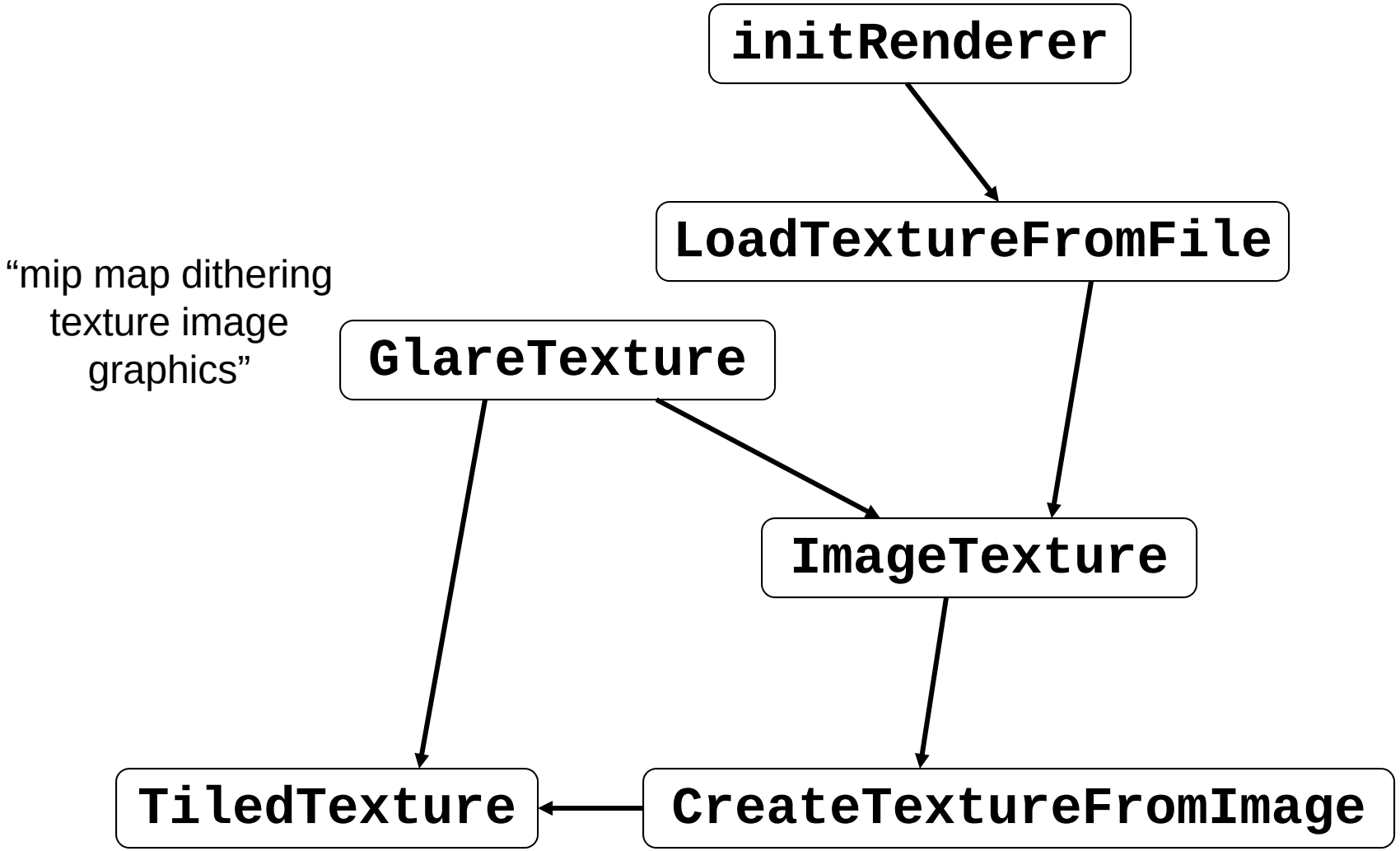
Functions from Celestia: http://www.shatters.net/celestia/

In Portfolio: Function calls are associations



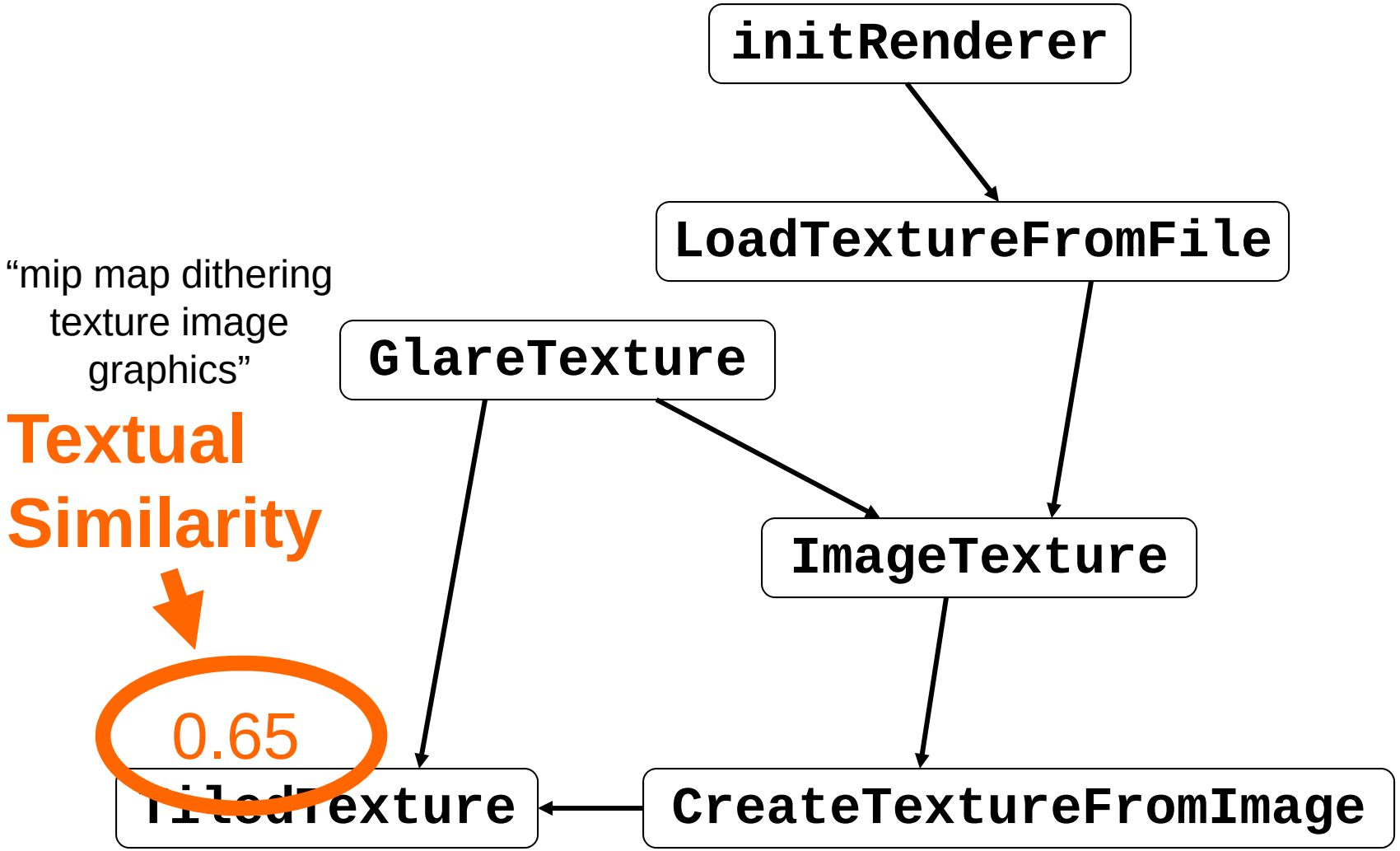
Functions from Celestia: http://www.shatters.net/celestia/

Association Model – Spreading Activation



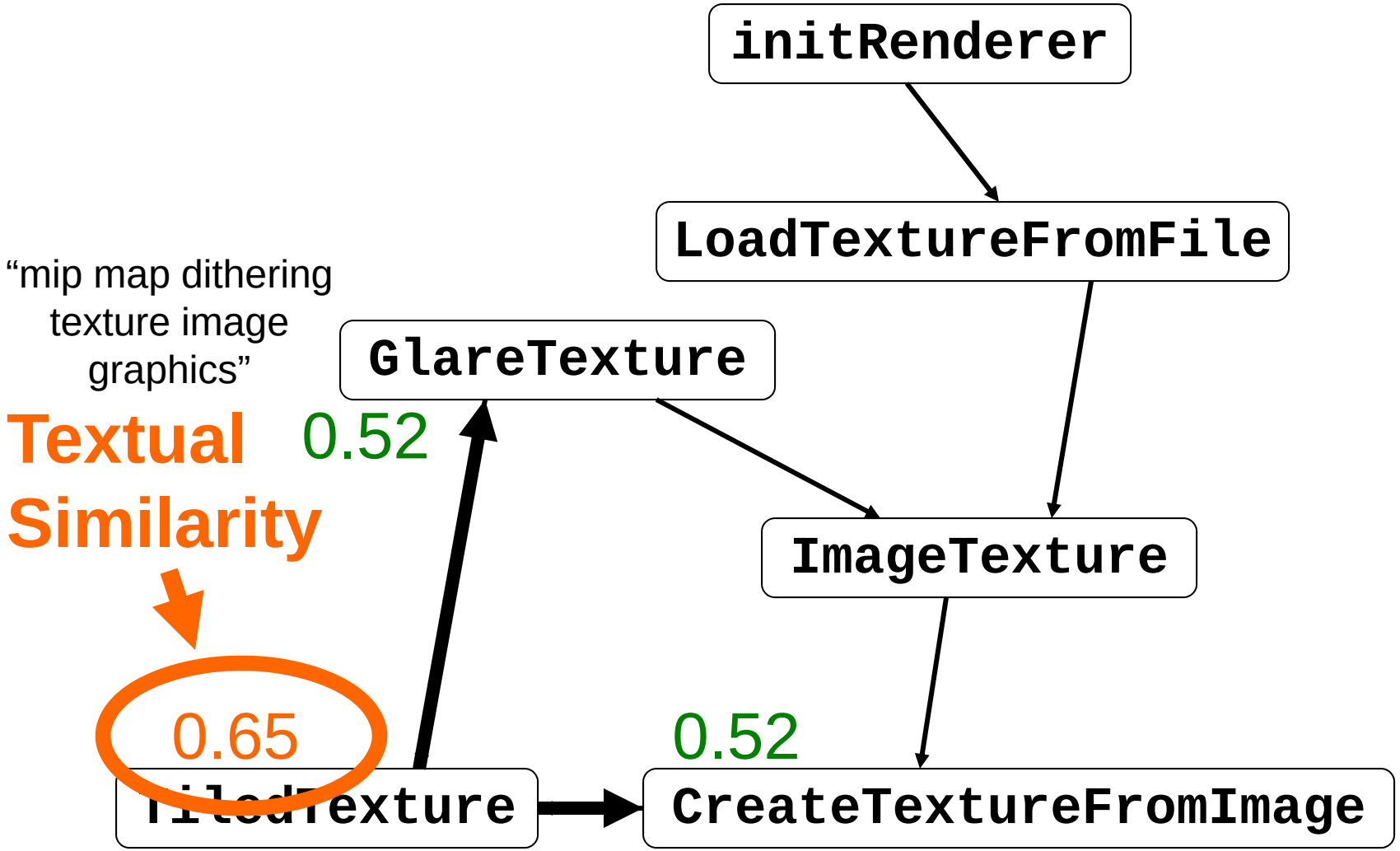
Functions from Celestia: http://www.shatters.net/celestia/

Association Model – Spreading Activation



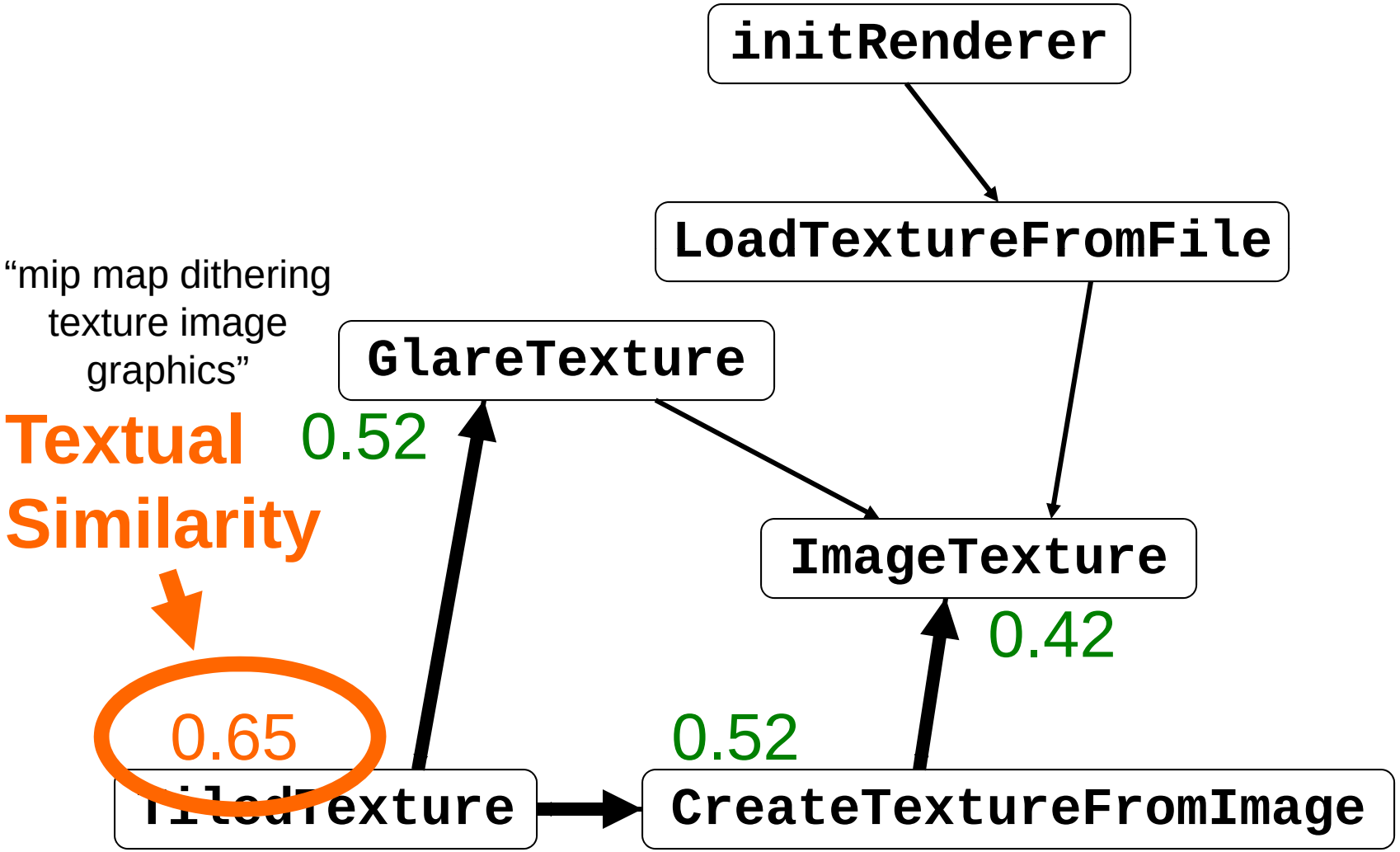
Functions from Celestia: http://www.shatters.net/celestia/

Association Model – Spreading Activation



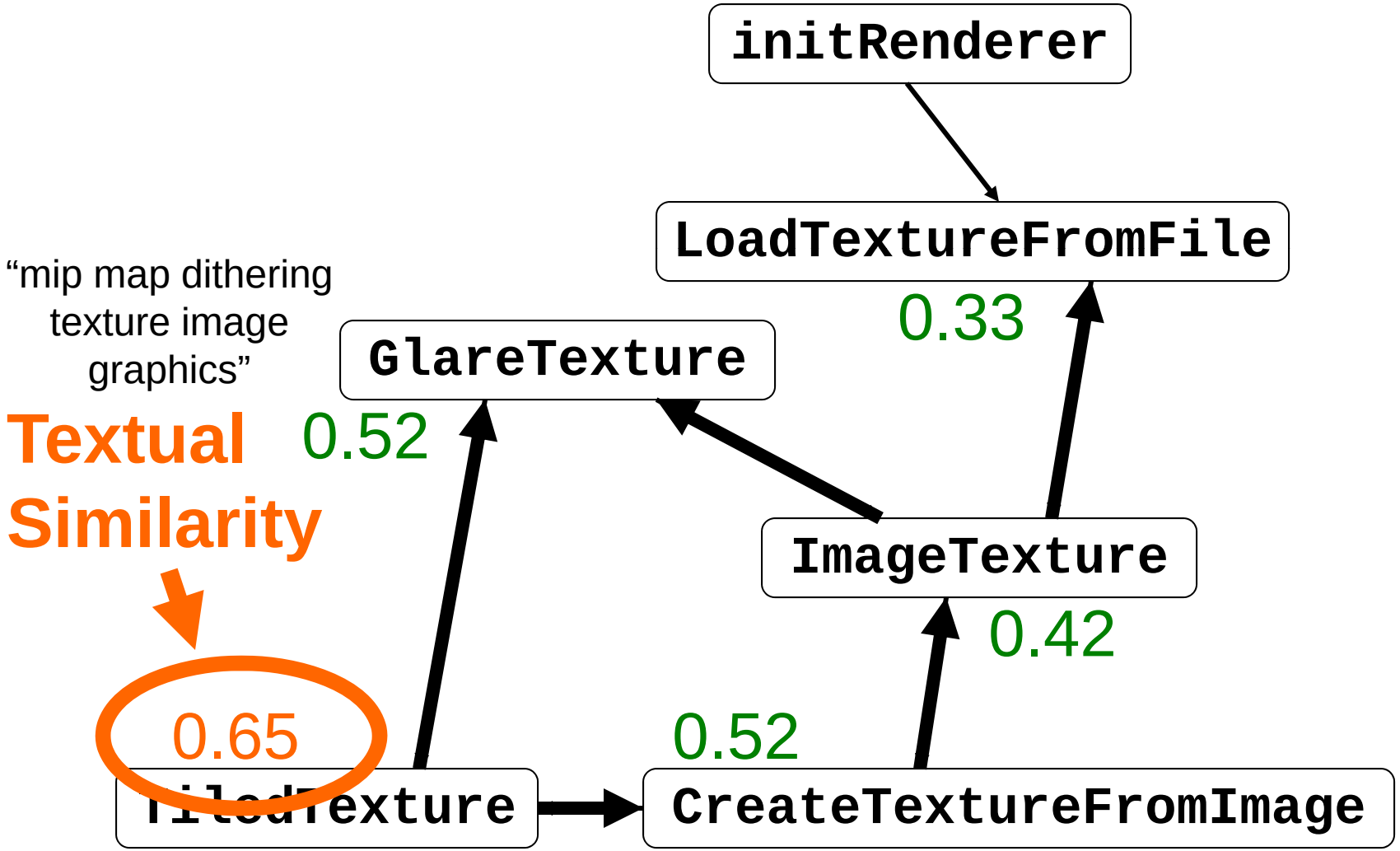
Functions from Celestia: http://www.shatters.net/celestia/

Association Model – Spreading Activation



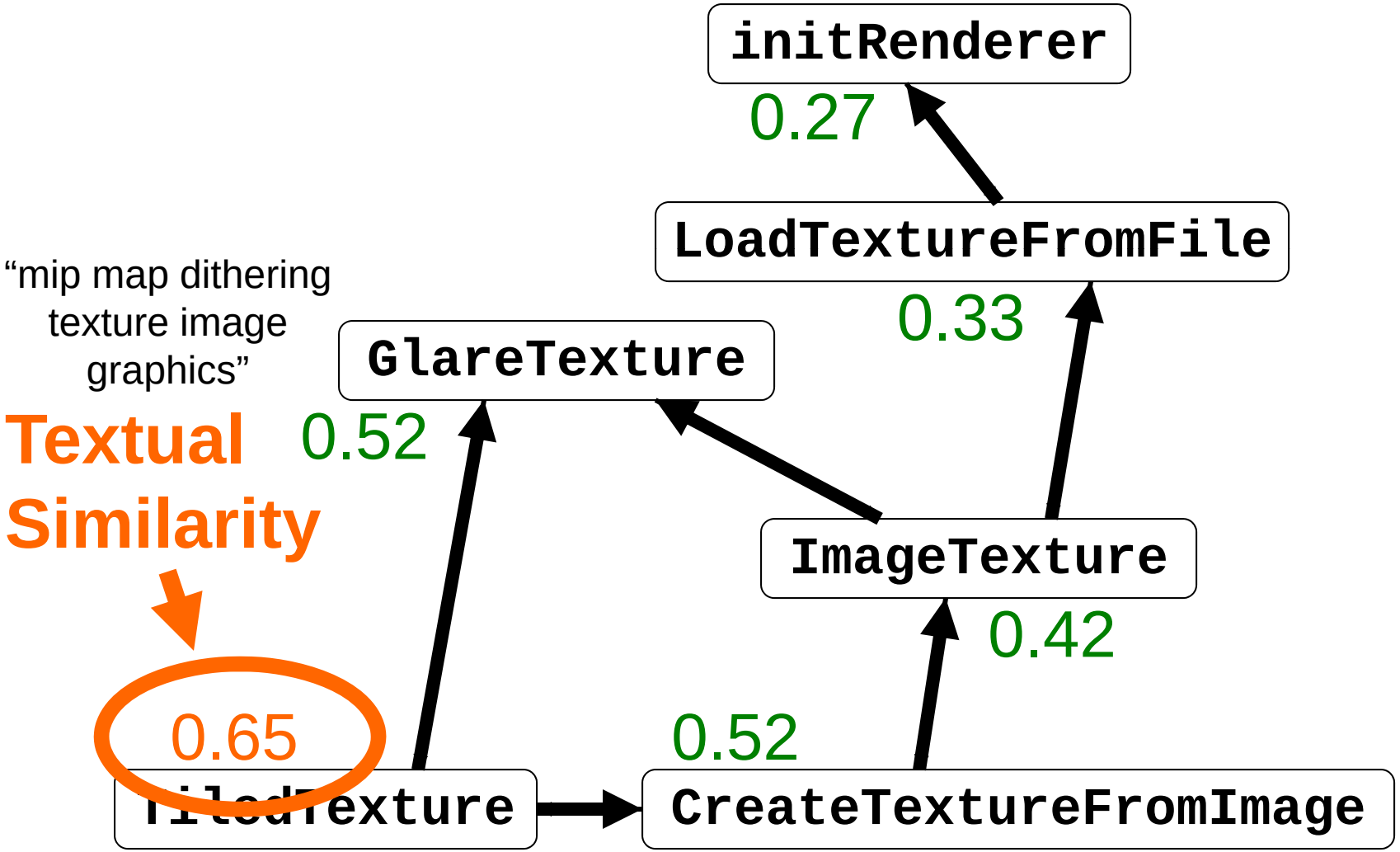
Functions from Celestia: http://www.shatters.net/celestia/

Association Model – Spreading Activation



Functions from Celestia: http://www.shatters.net/celestia/

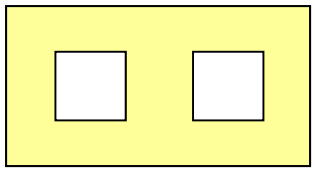
Association Model – Spreading Activation

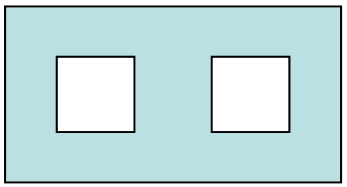


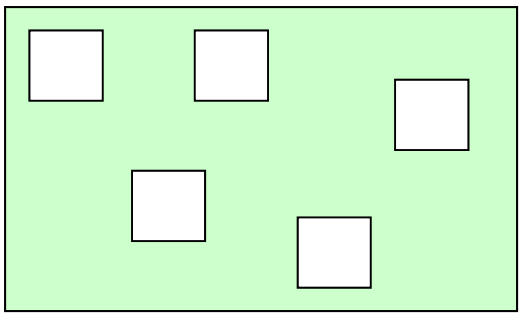
Functions from Celestia: http://www.shatters.net/celestia/

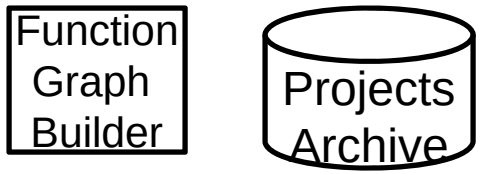

Projects
Archive

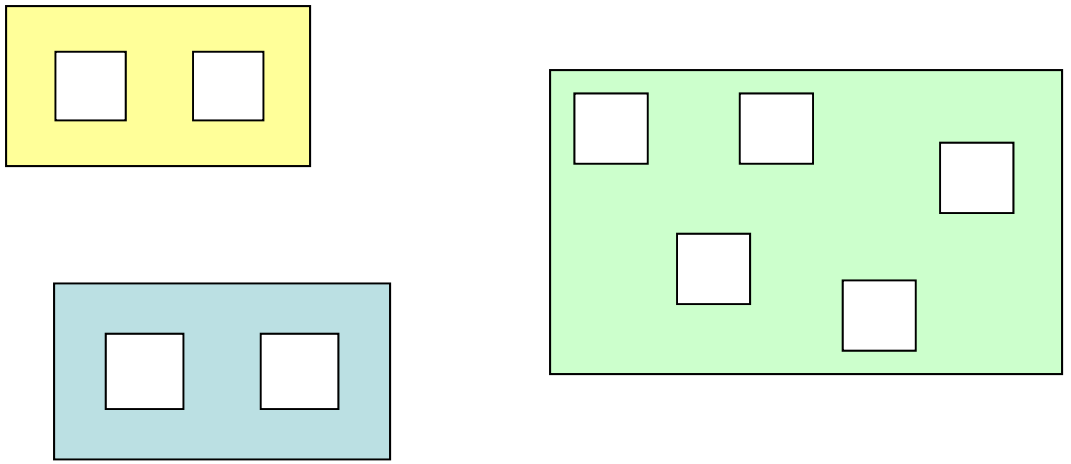

Projects
Archive

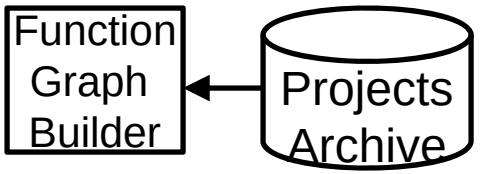


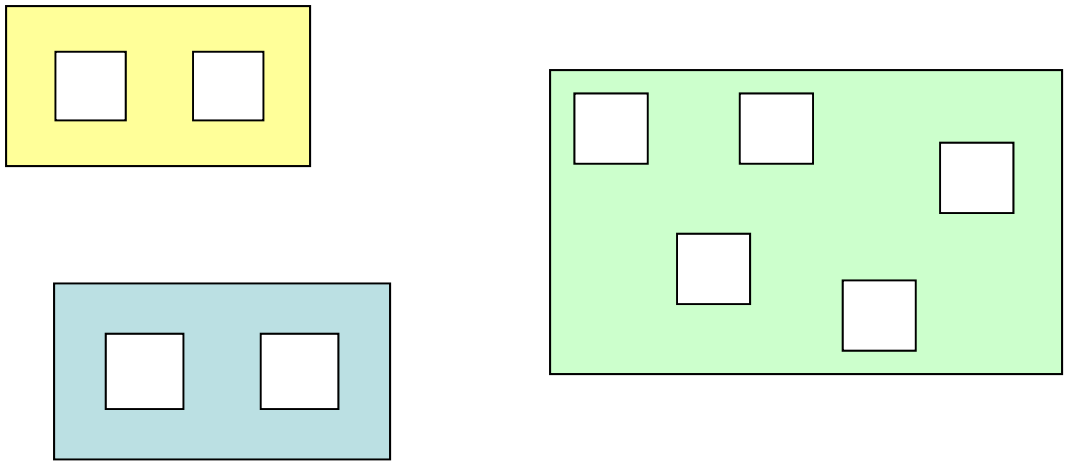


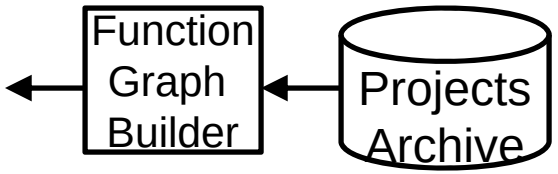


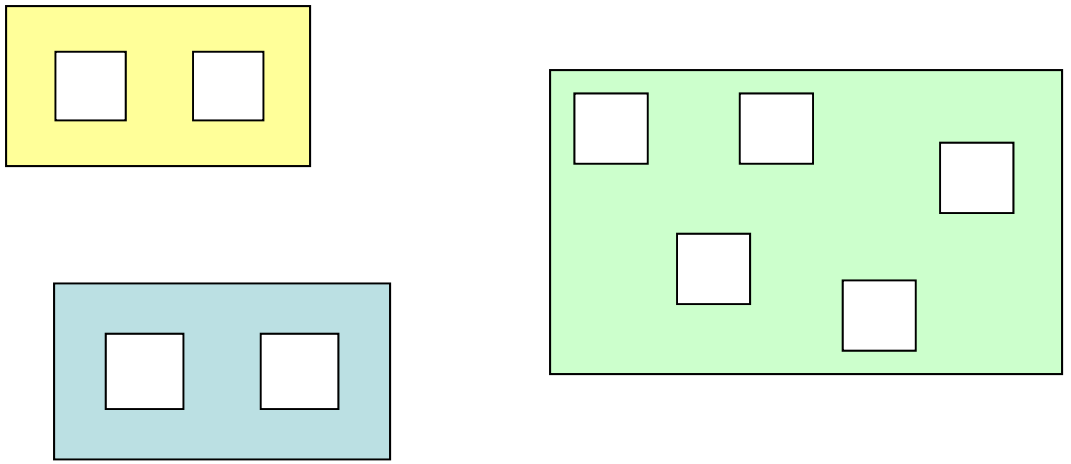

Function
Graph
Builder
Projects
Archive

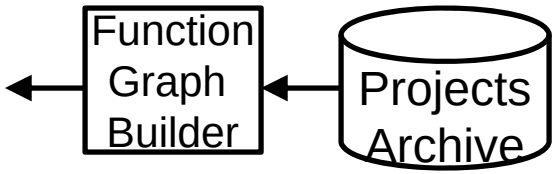


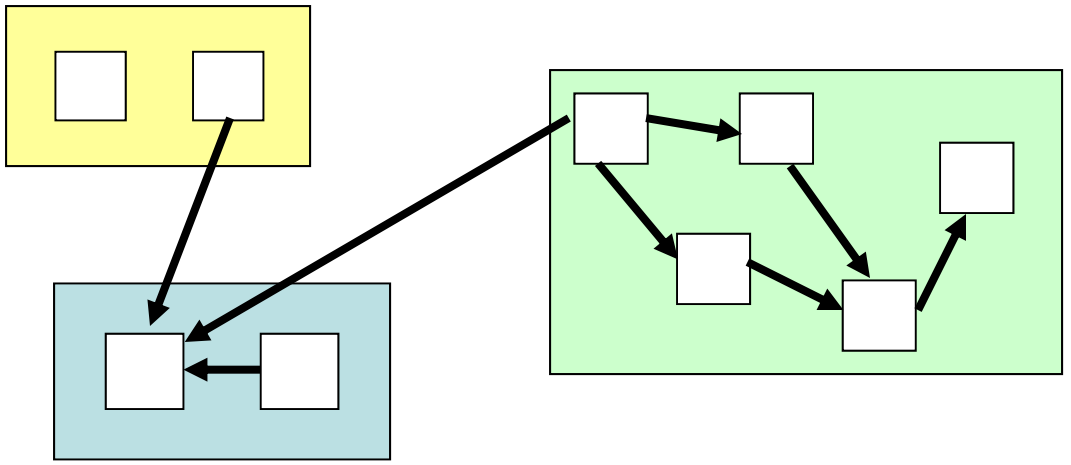

Function
Graph
Builder
Projects
Archive

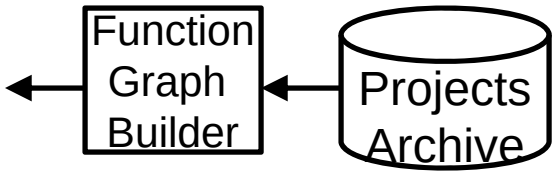


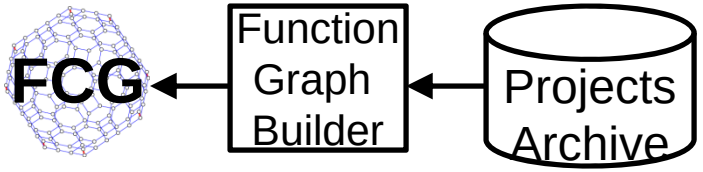

Function
Graph
Builder
Projects
Archive

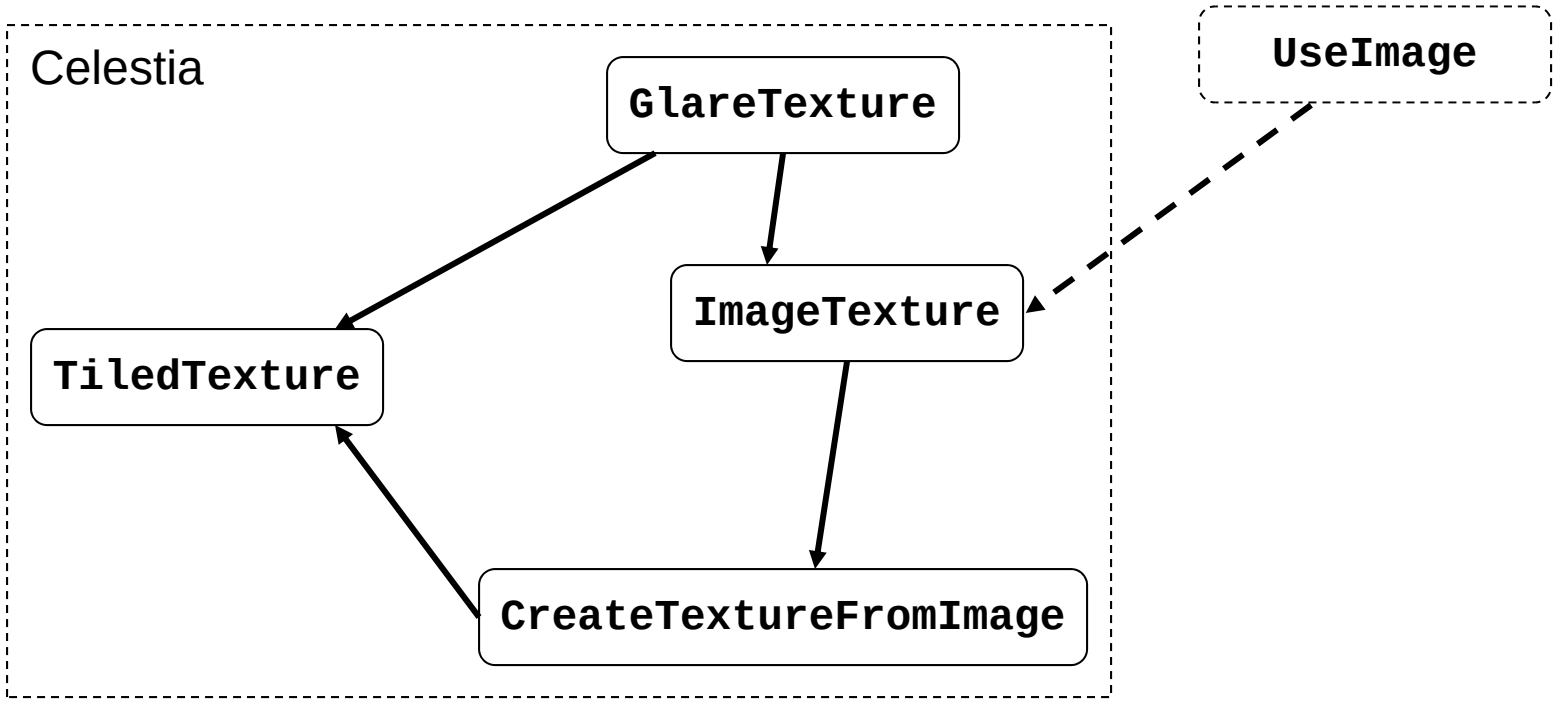


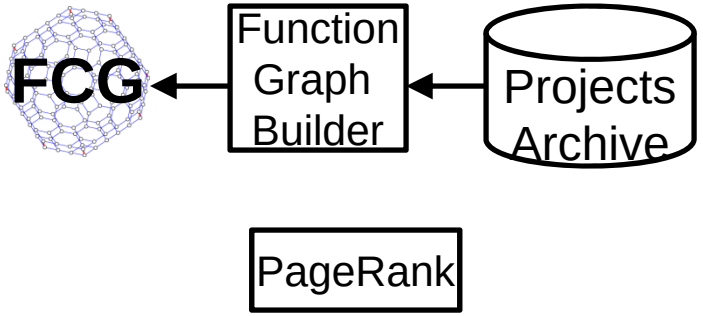

Function
Graph
Builder
Projects
Archive

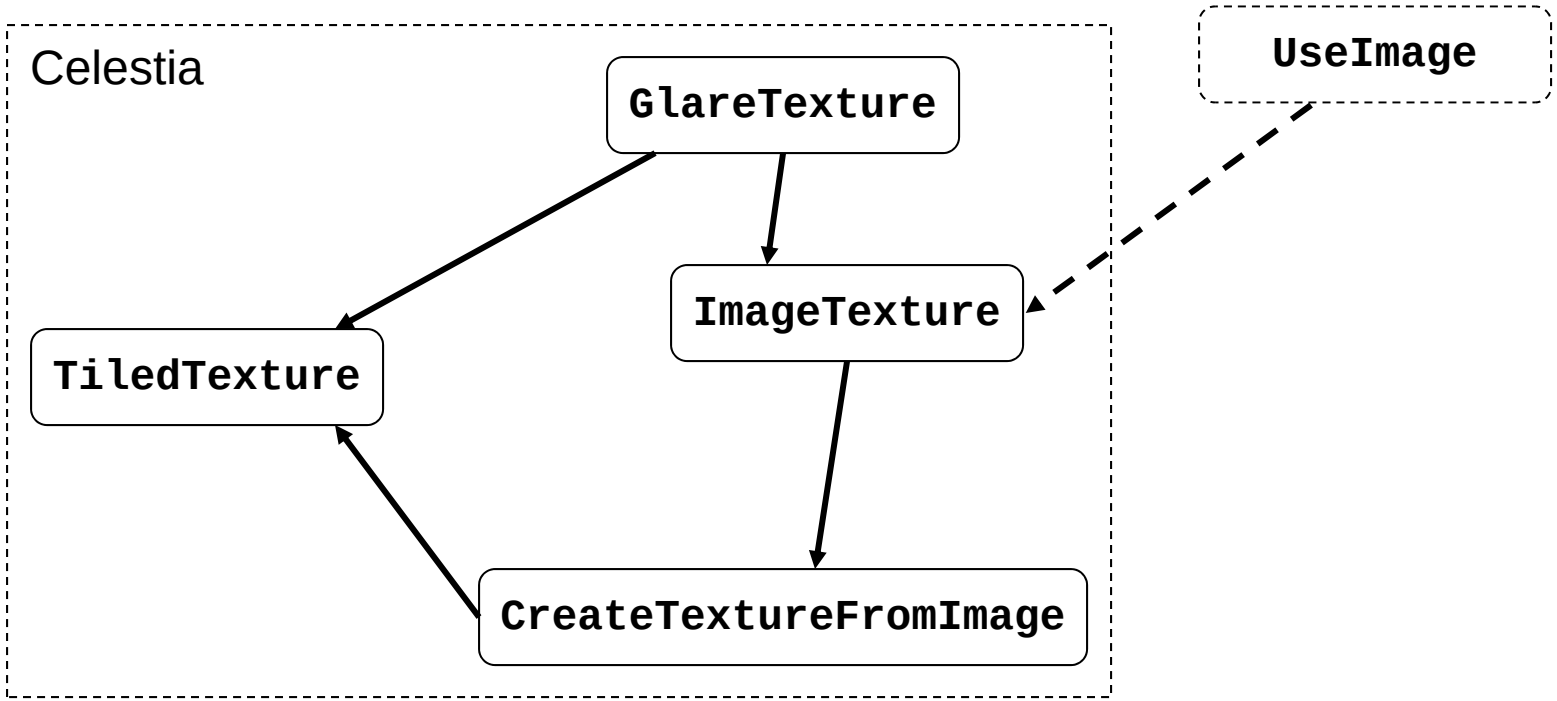


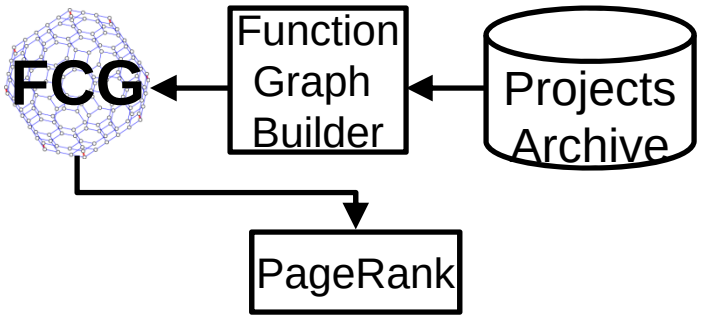

Function
Graph
Builder
Projects
Archive

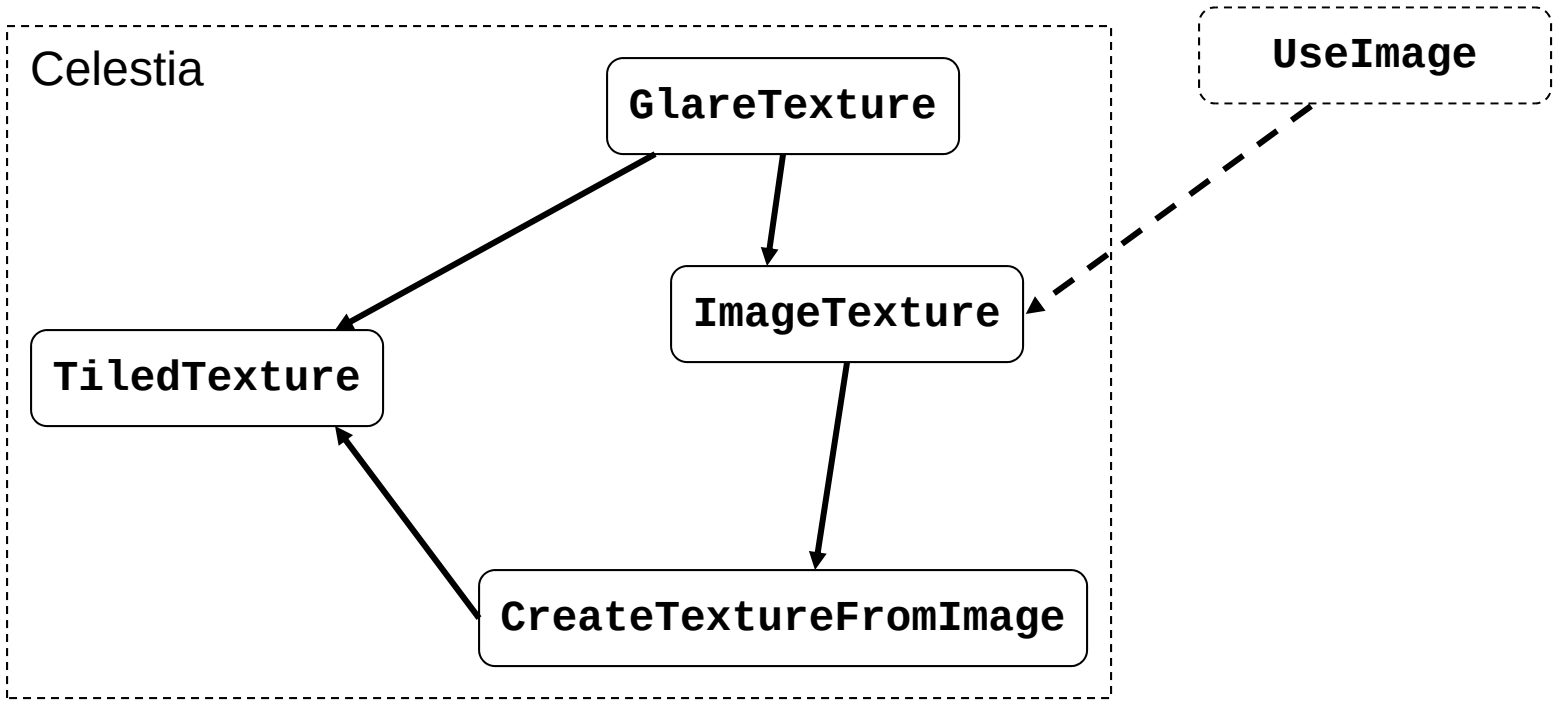

FCG
Function
Graph
Builder
Projects
Archive

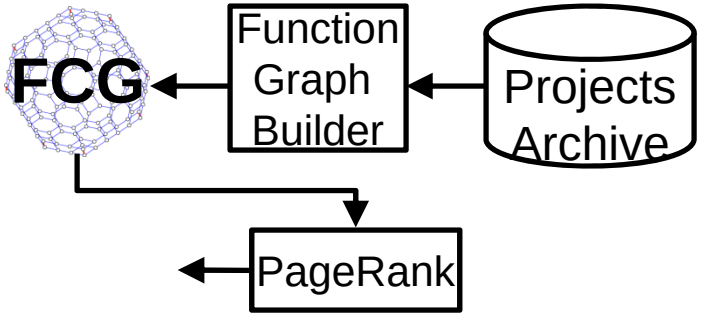

Celestia
GlareTexture
UseImage
ImageTexture
TiledTexture
CreateTextureFromImage

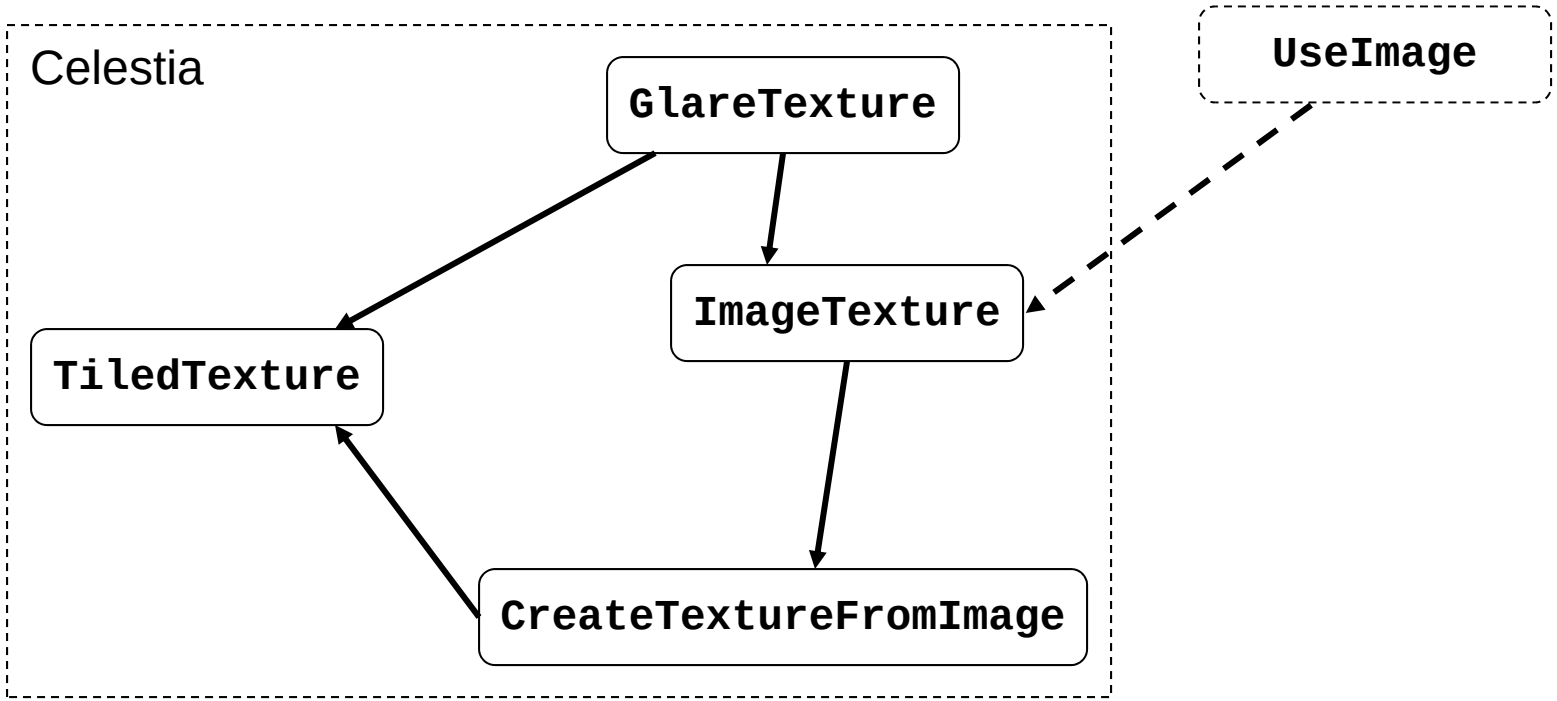

FCG
Function
Graph
Builder
Projects
Archive
PageRank

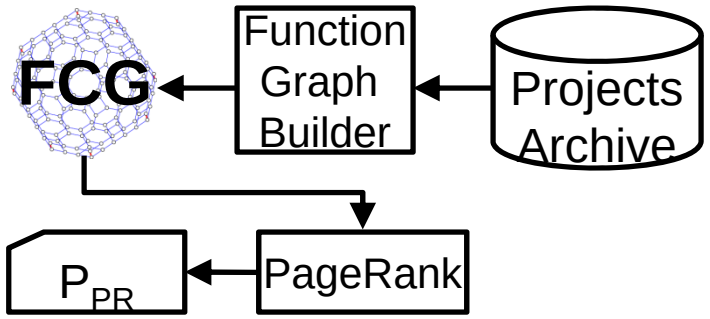

Celestia
GlareTexture
UseImage
ImageTexture
TiledTexture
CreateTextureFromImage

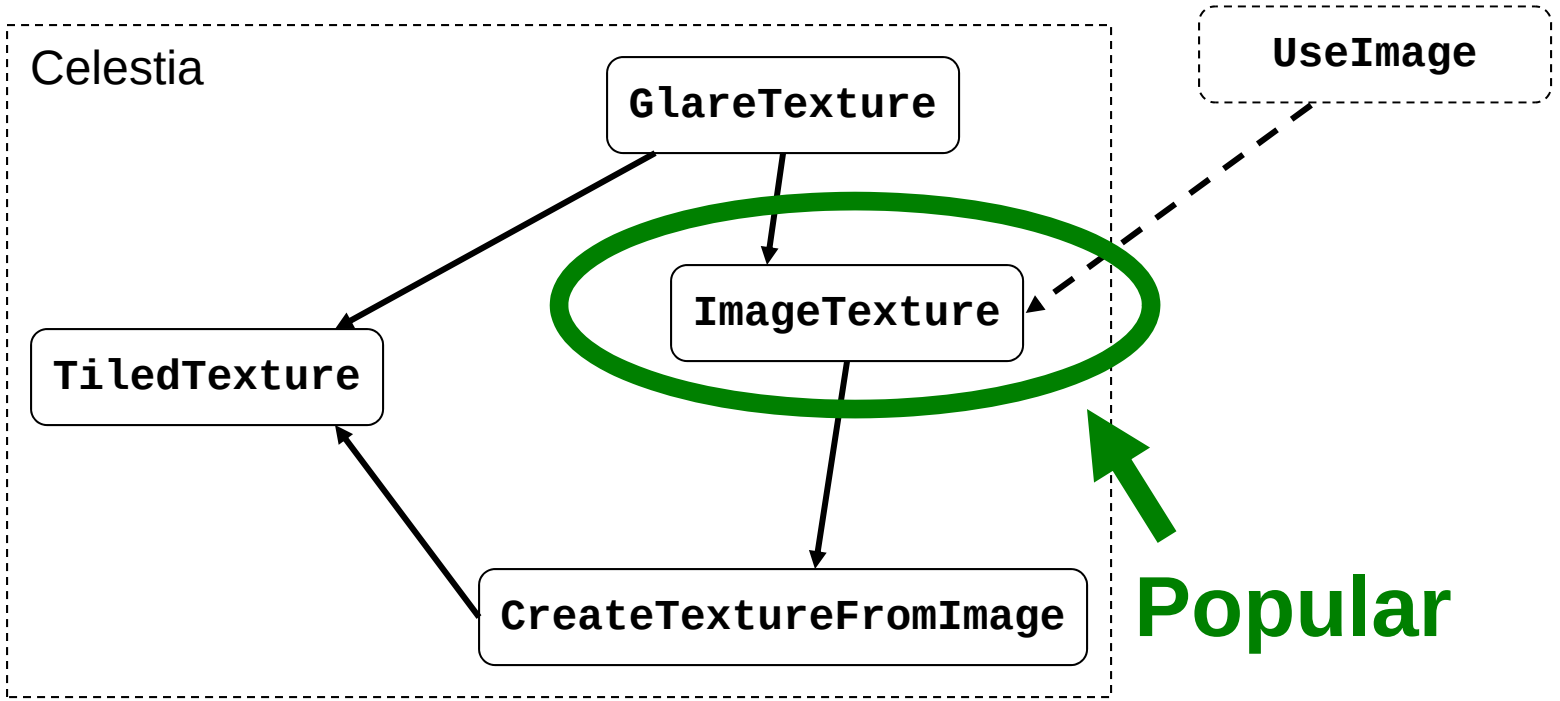

FCG
Function Graph Builder
Projects Archive
PageRank

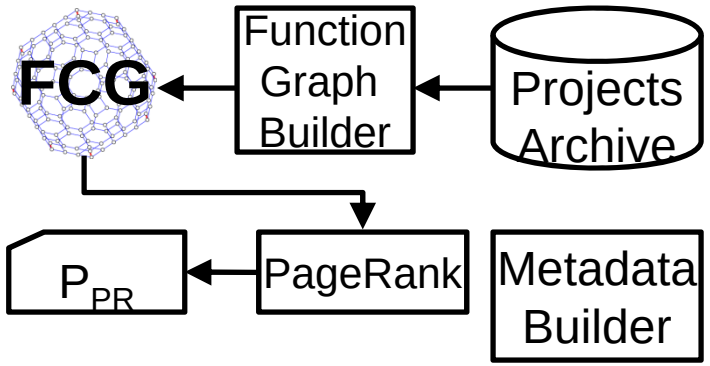

Celestia
GlareTexture
UseImage
ImageTexture
TiledTexture
CreateTextureFromImage

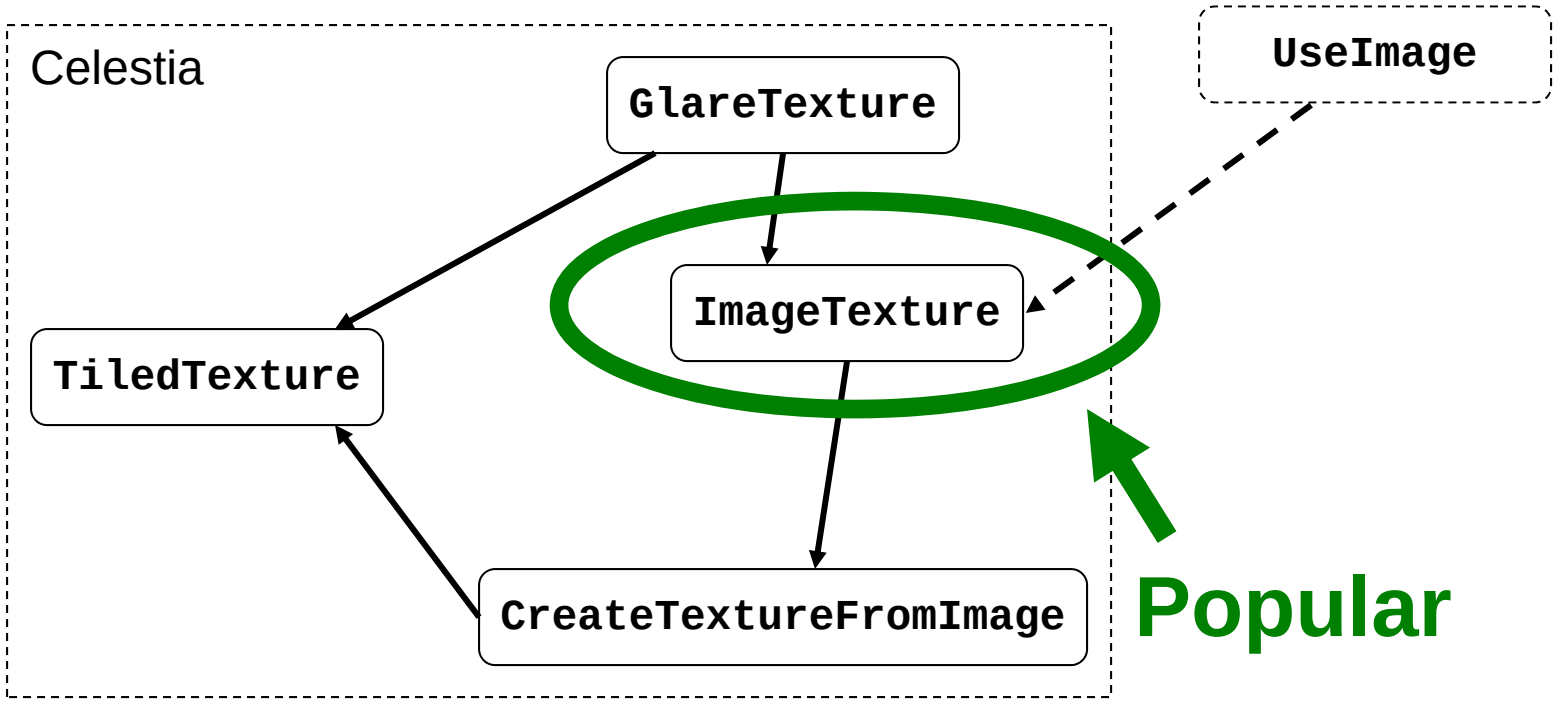

FCG
Function
Graph
Builder
Projects
Archive
PageRank

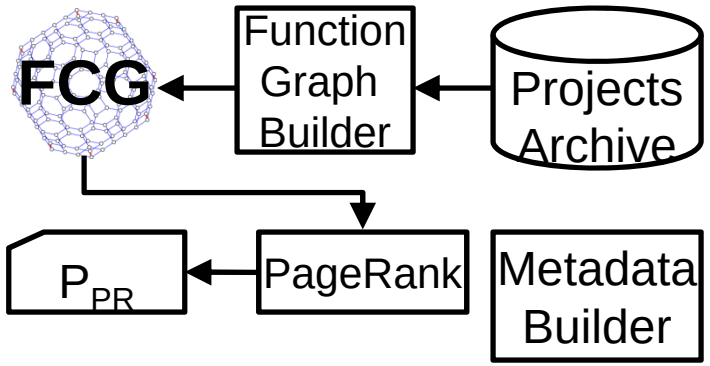

Celestia
GlareTexture
UseImage
ImageTexture
TiledTexture
CreateTextureFromImage

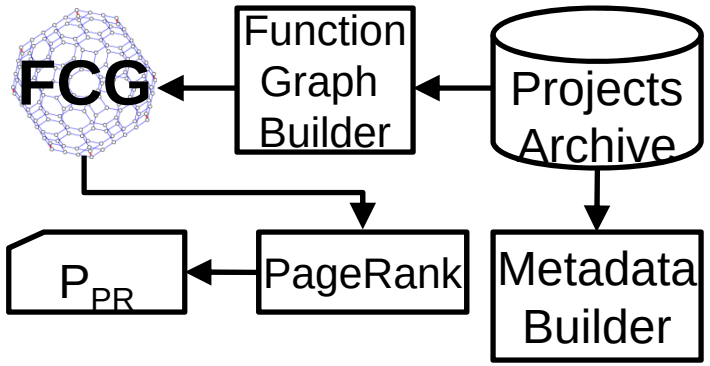

FCG
Function Graph Builder
Projects Archive
P_{PR}
PageRank

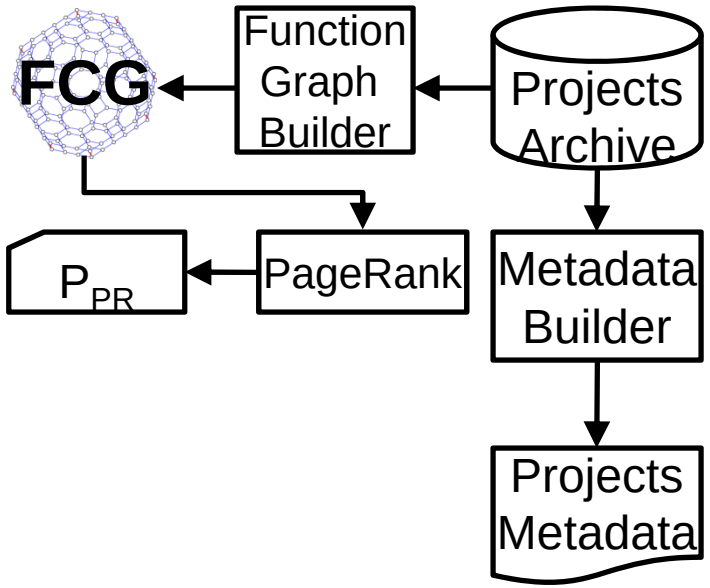

Celestia
GlareTexture
UseImage
ImageTexture
TiledTexture
CreateTextureFromImage
Popular

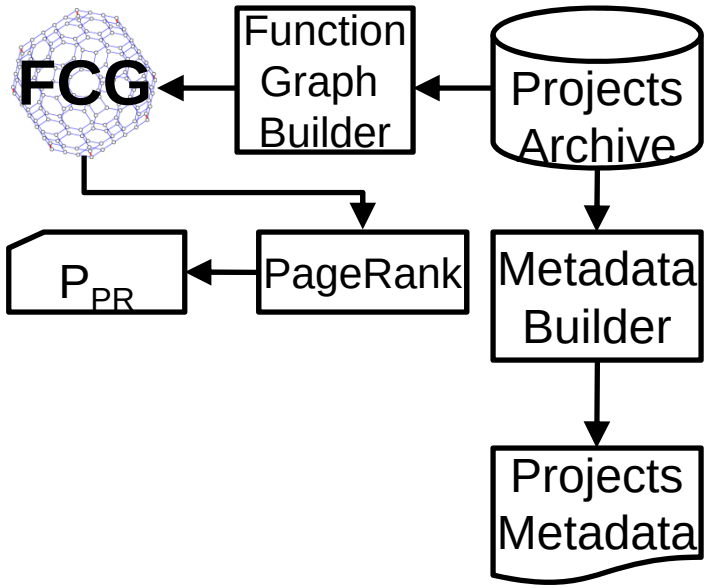

FCG
Function Graph Builder
Projects Archive
P_{PR}
PageRank
Metadata Builder


Celestia
GlareTexture
UseImage
ImageTexture
TiledTexture
CreateTextureFromImage
Popular

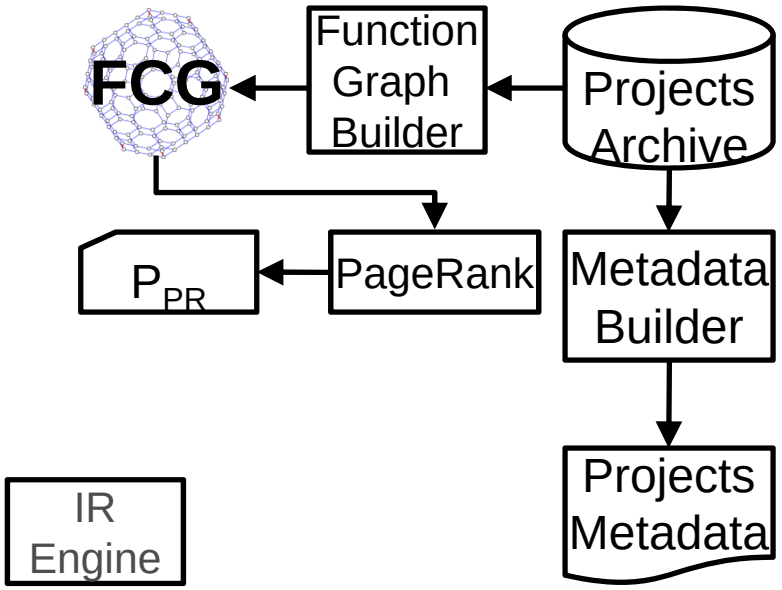

FCG
Function Graph Builder
Projects Archive
P_{PR}
PageRank
Metadata Builder


FCG
Function Graph Builder
Projects Archive
P_{PR}
PageRank
Metadata Builder

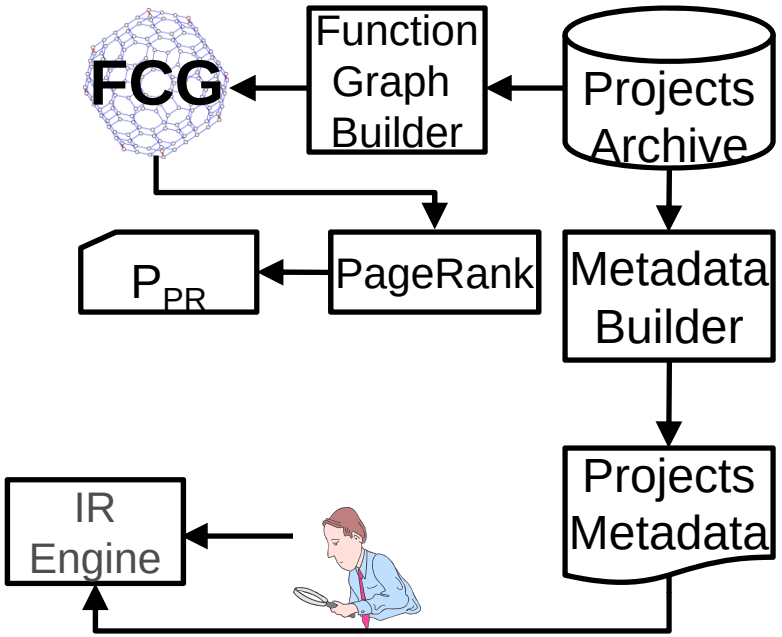

FCG
Function Graph Builder
Projects Archive
P_{PR}
PageRank
Metadata Builder
Projects Metadata




```
/* Tiling Method for Mip Maps */
static Texture* TiledTexture(Image& img)
{
  if (GetTextureCaps().nonPow2Supported)
  {
    /* prepares mip maps for dithering */
    if (mipMode == Texture::DefaultMipMaps)
      mipMode = Texture::AutoMipMaps;
  }
  …
}
```

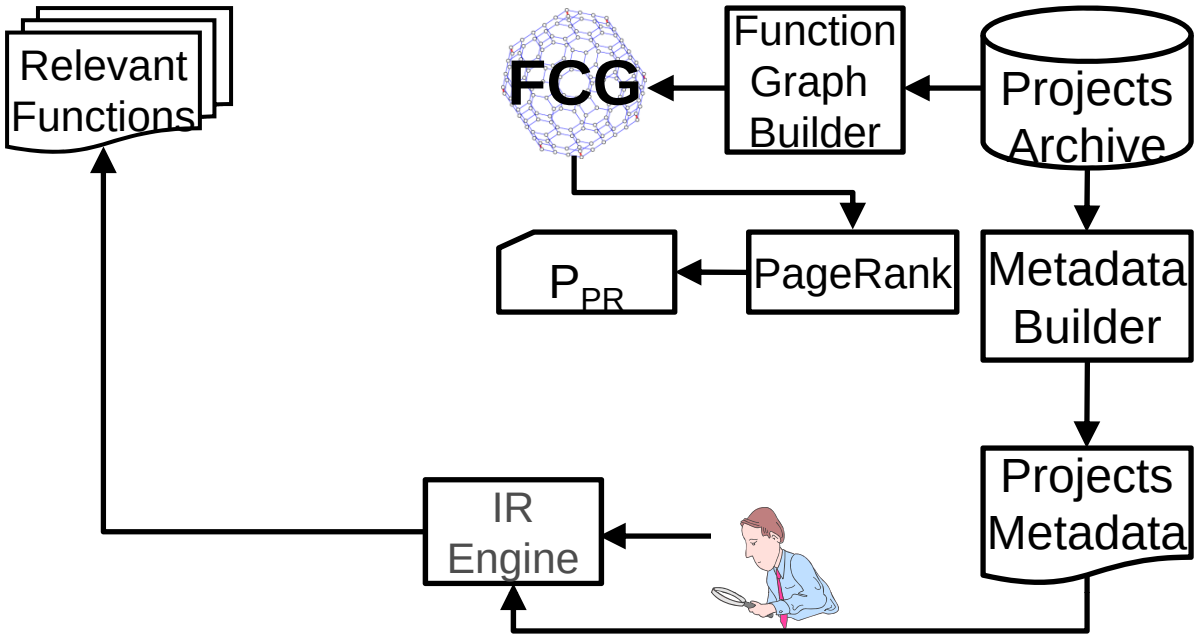



```
/* Tiling Method for Mip Maps */
static Texture* TiledTexture(Image& img)
{
  if (GetTextureCaps().nonPow2Supported)
  {
    /* prepares mip maps for dithering */
    if (mipMode == Texture::DefaultMipMaps)
      mipMode = Texture::AutoMipMaps;
  }
  …
}
```



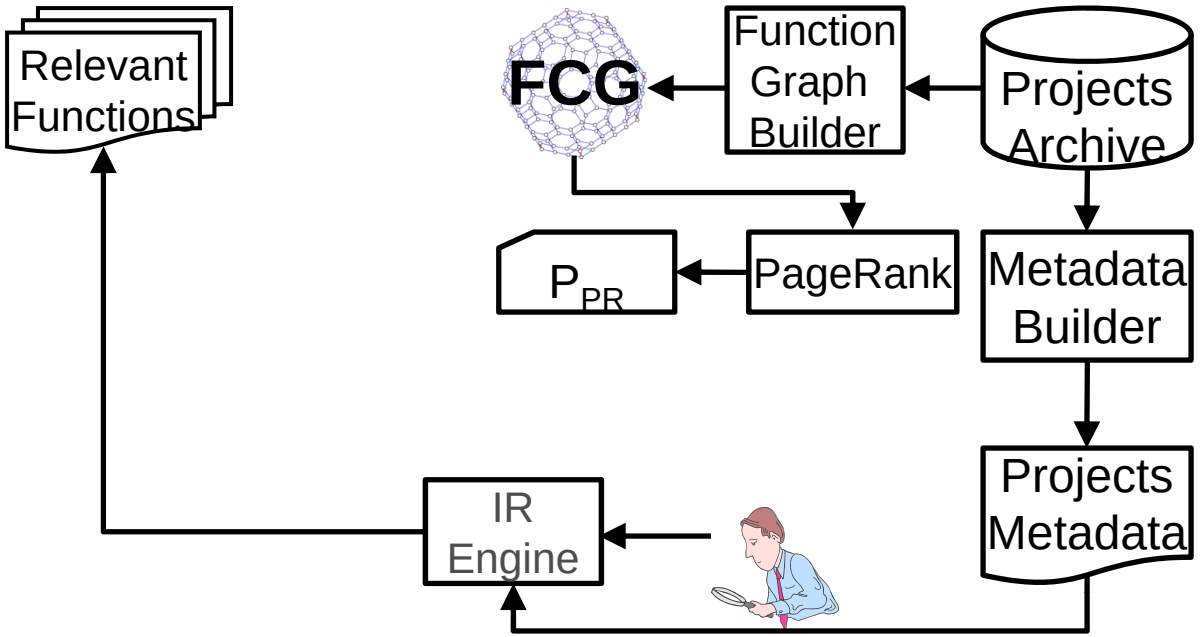
“mip map dithering” →

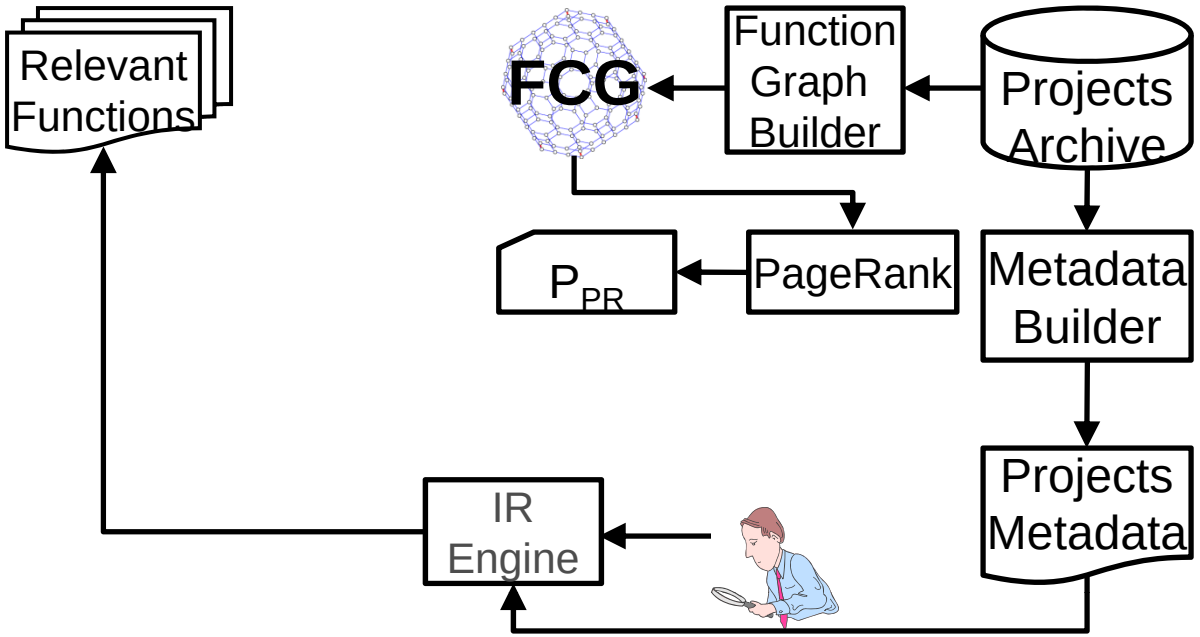
```
/* Tiling Method for Mip Maps */
static Texture* TiledTexture(Image& img)
{
  if (GetTextureCaps().nonPow2Supported)
  {
    /* prepares mip maps for dithering */
    if (mipMode == Texture::DefaultMipMaps)
      mipMode = Texture::AutoMipMaps;
  }
  …
}
```



“mip map dithering” →

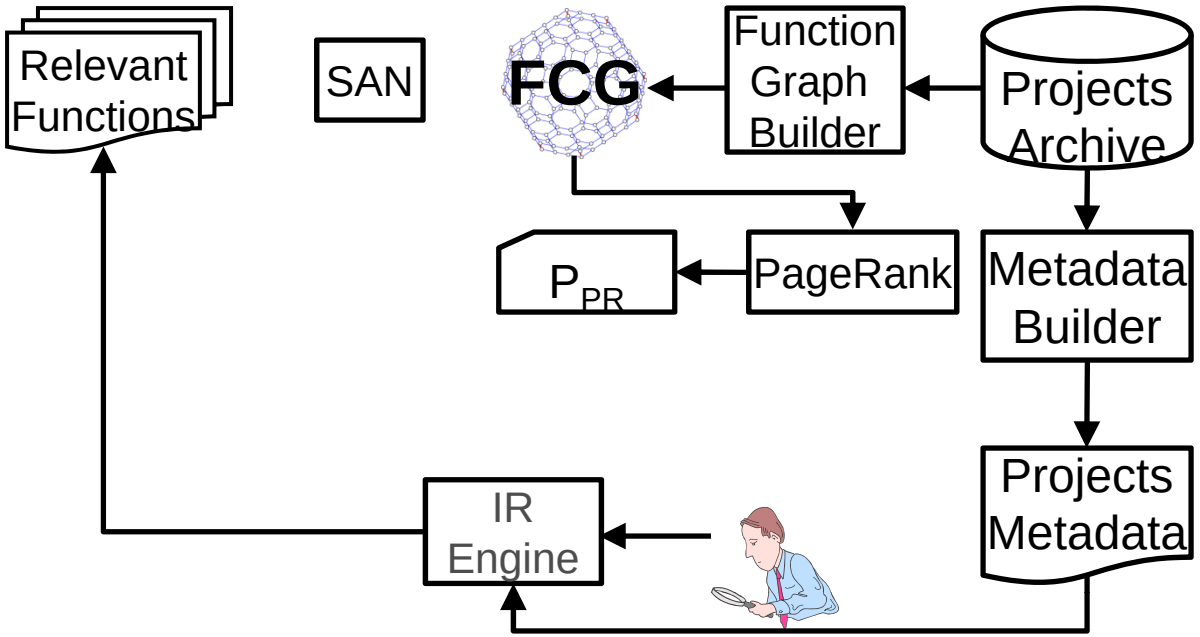
```
/* Tiling Method for Mip Maps */
static Texture* TiledTexture(Image& img)
{
  if (GetTextureCaps().nonPow2Supported)
  {
    /* prepares mip maps for dithering */
    if (mipMode == Texture::DefaultMipMaps)
      mipMode = Texture::AutoMipMaps;
  }
  …
}
```


Relevant Functions
FCG
Function Graph Builder
Projects Archive
P_{PR}
PageRank
Metadata Builder
Projects Metadata
IR Engine



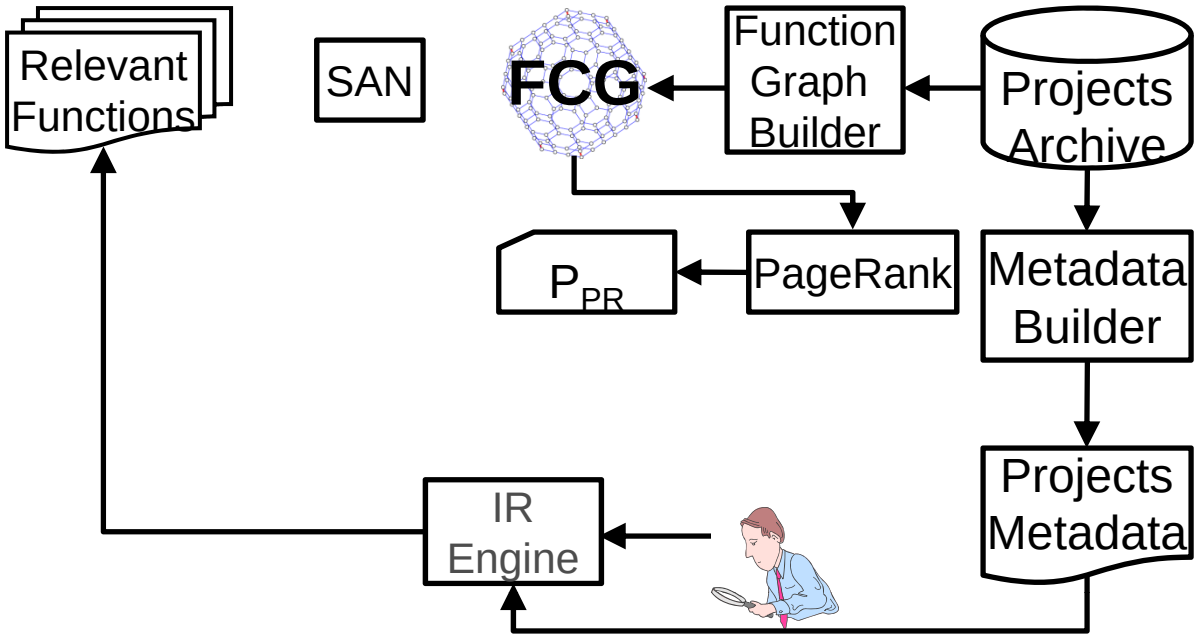
$SAN_{0.65}$

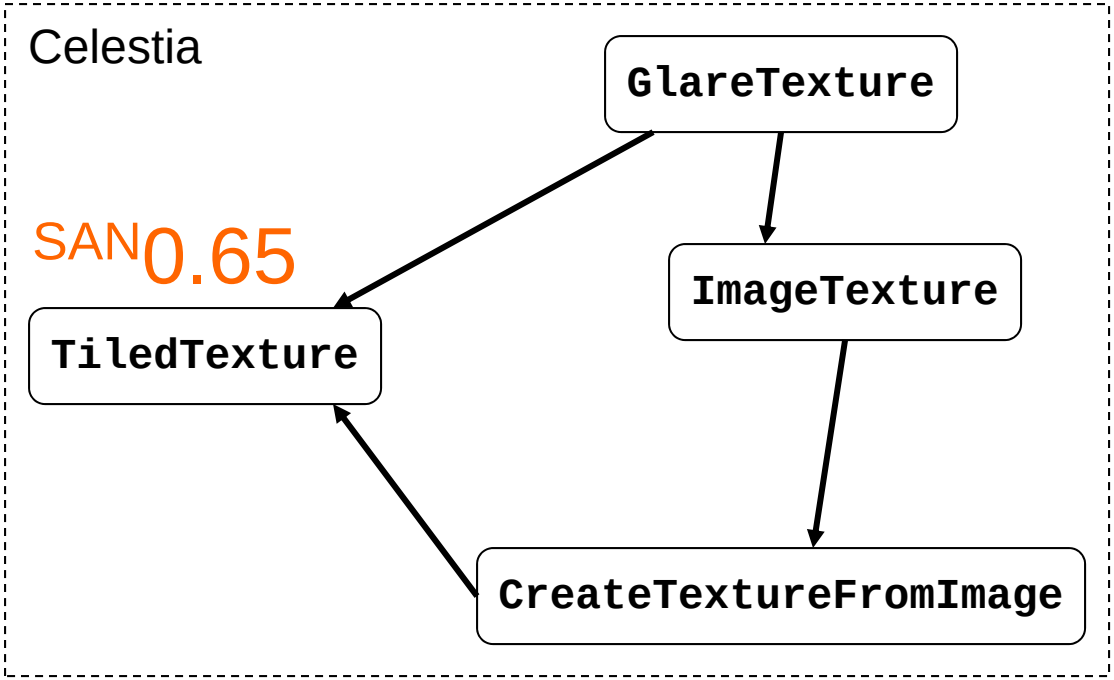
`TiledTexture`

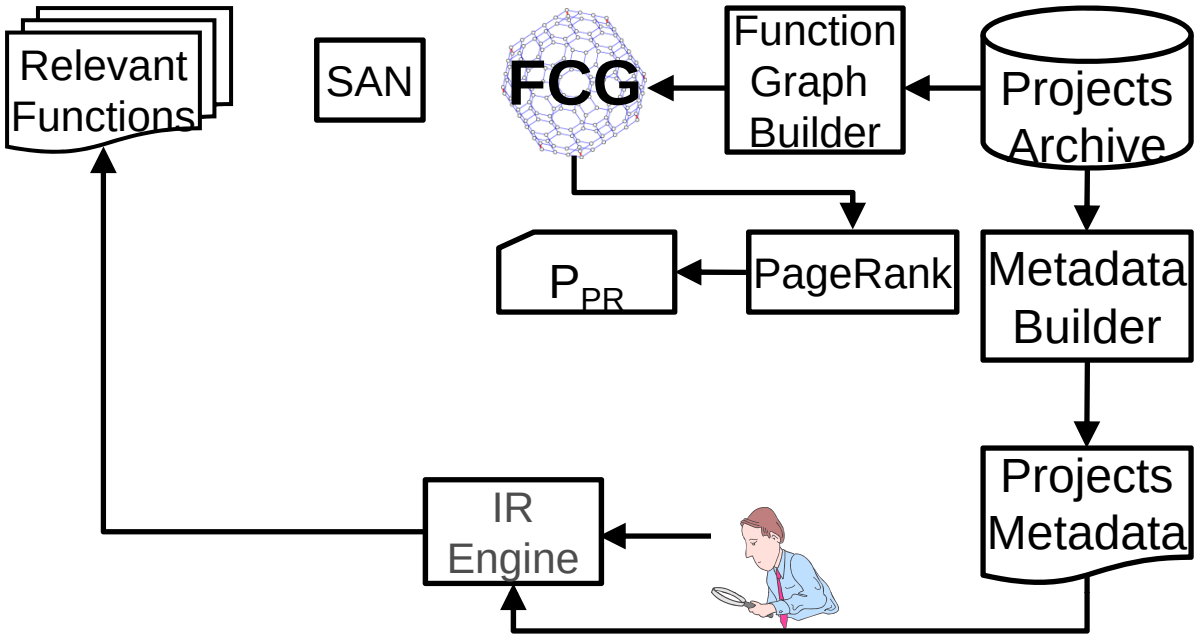


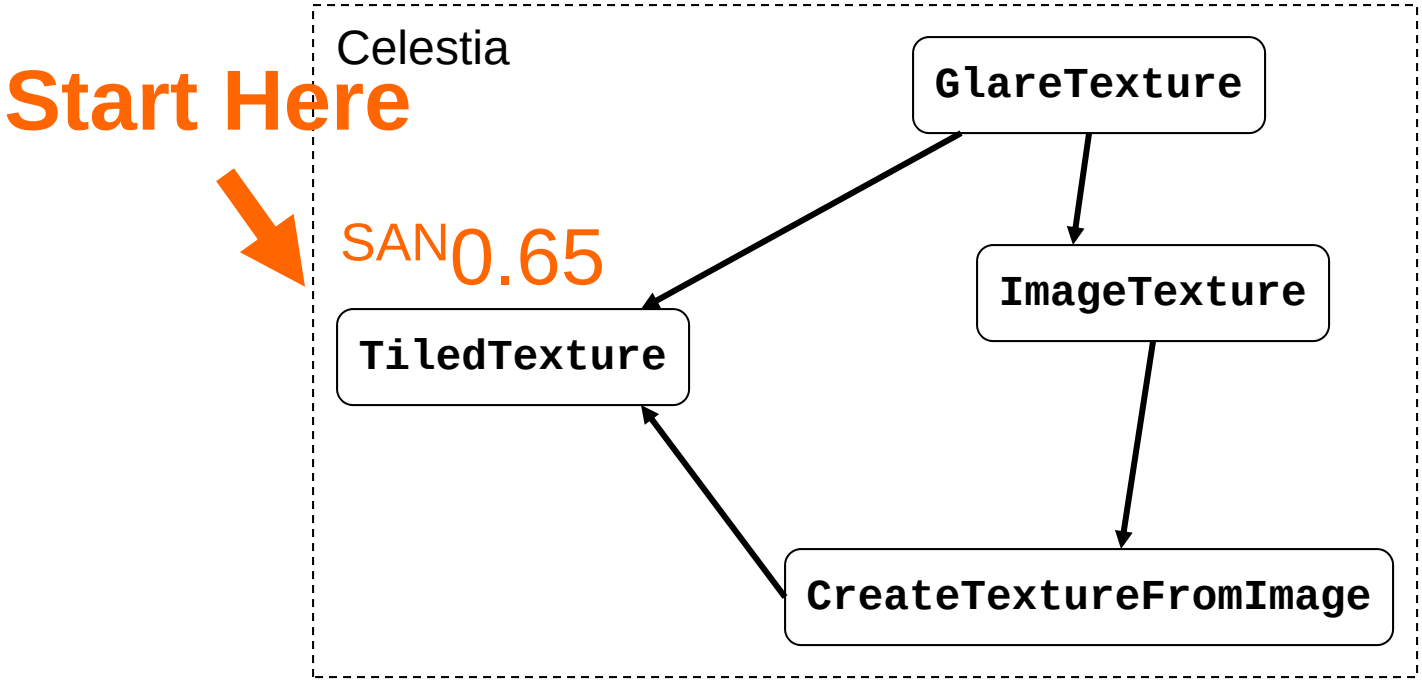
$SAN_{0.65}$

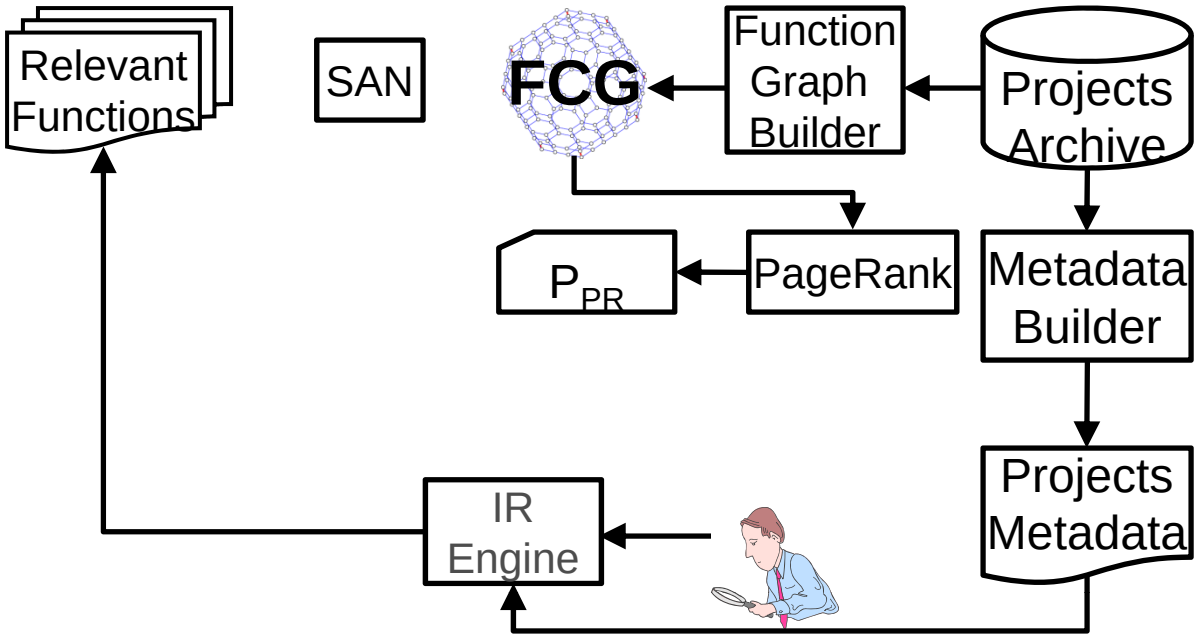
`TiledTexture`

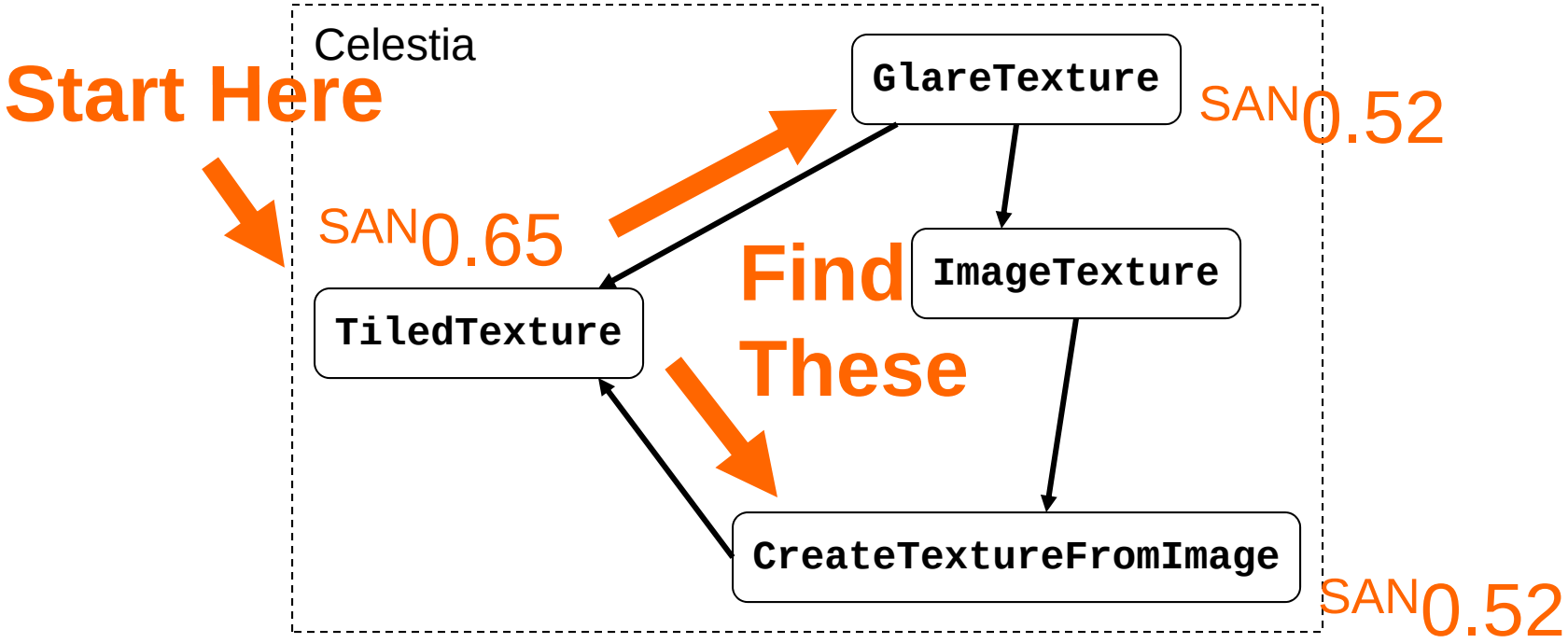

Relevant Functions
SAN
FCG
Function Graph Builder
Projects Archive
P_{PR}
PageRank
Metadata Builder
Projects Metadata
IR Engine

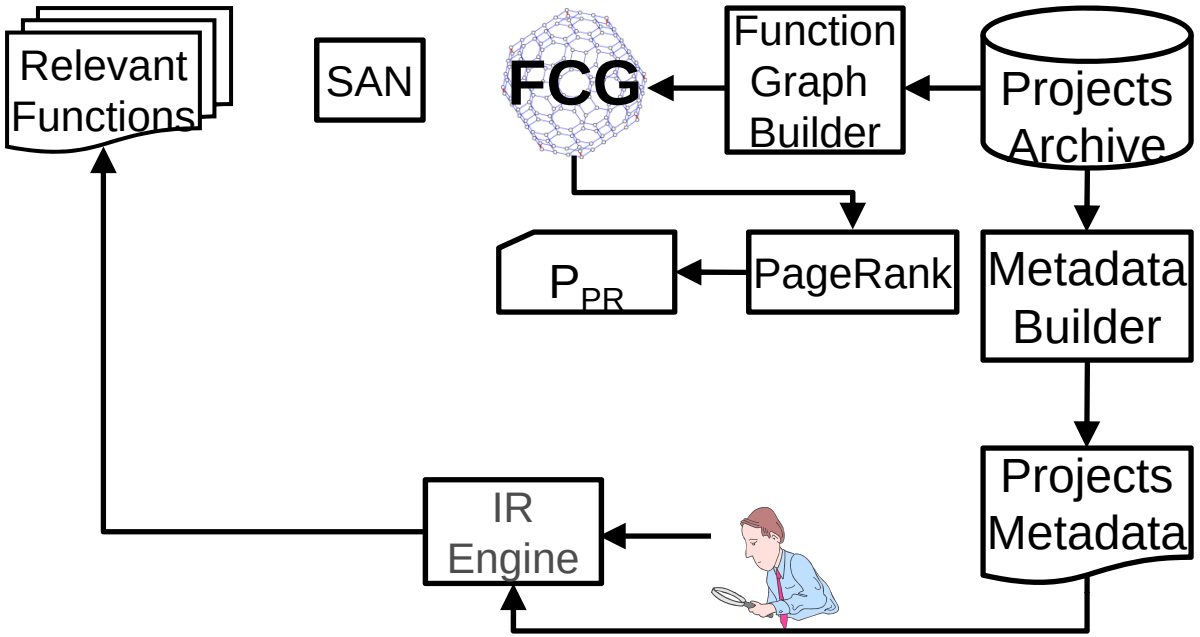

Celestia
GlareTexture
SAN 0.65
ImageTexture
TiledTexture
CreateTextureFromImage

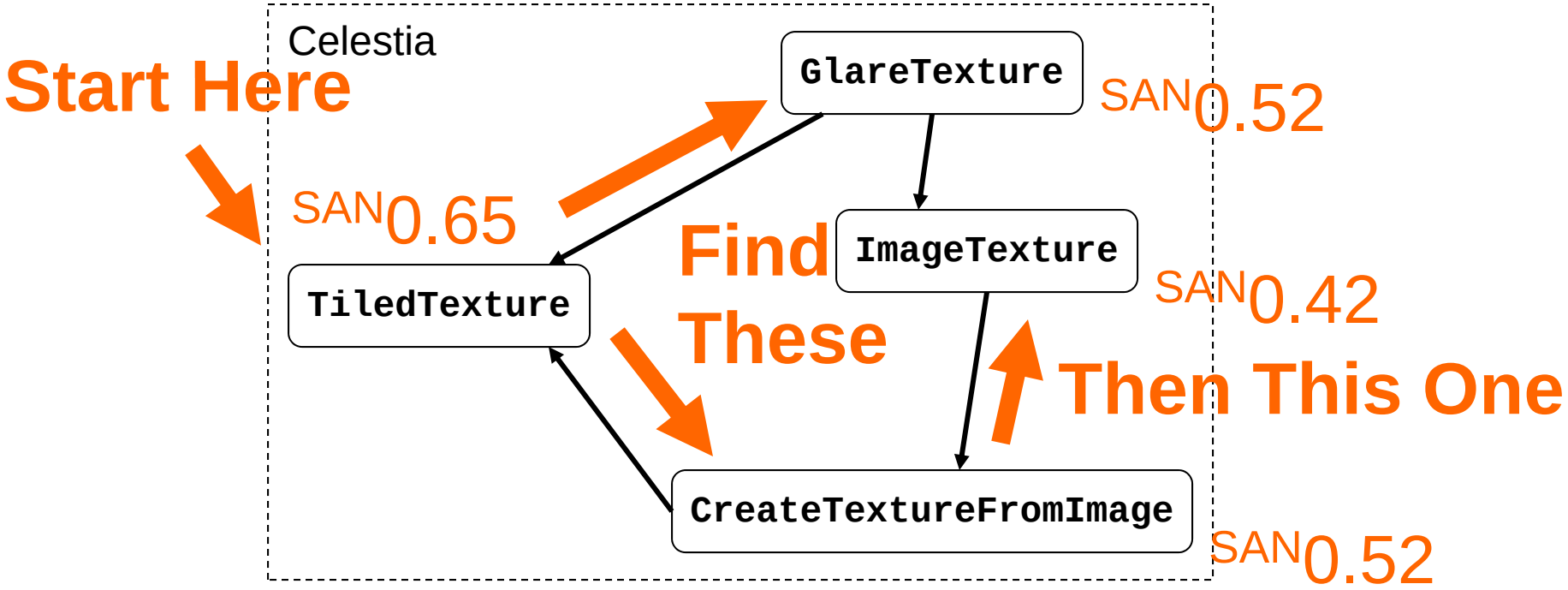

Relevant Functions
SAN
FCG
Function Graph Builder
Projects Archive
P_{PR}
PageRank
Metadata Builder
Projects Metadata
IR Engine

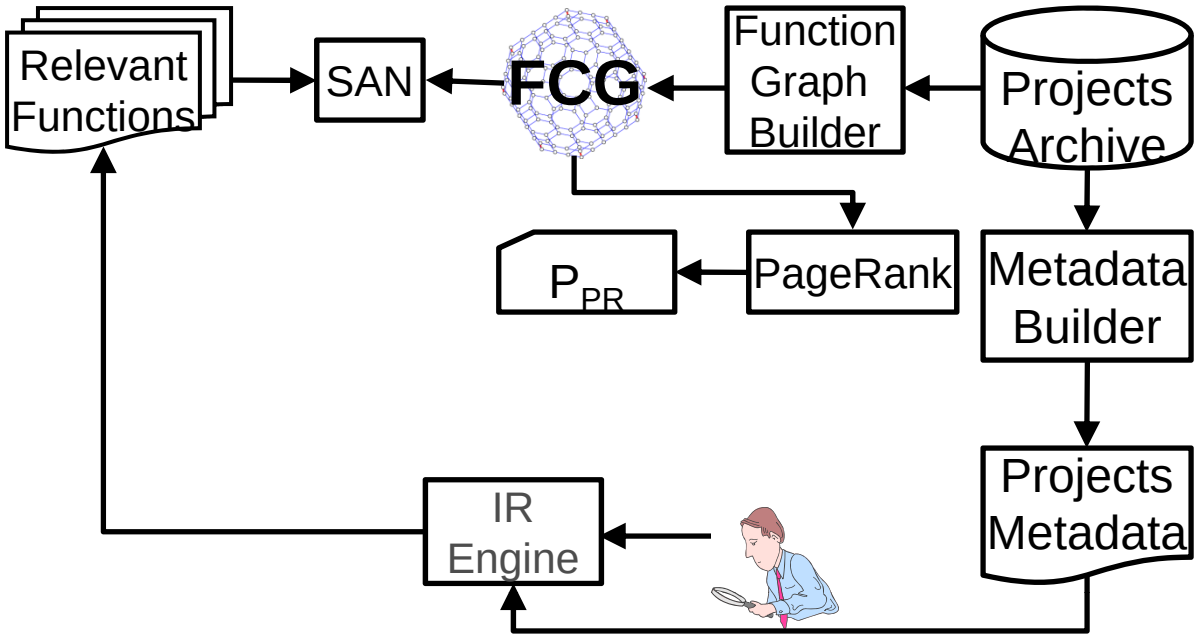

Start Here
SAN 0.65
Celestia
GlareTexture
ImageTexture
TiledTexture
CreateTextureFromImage

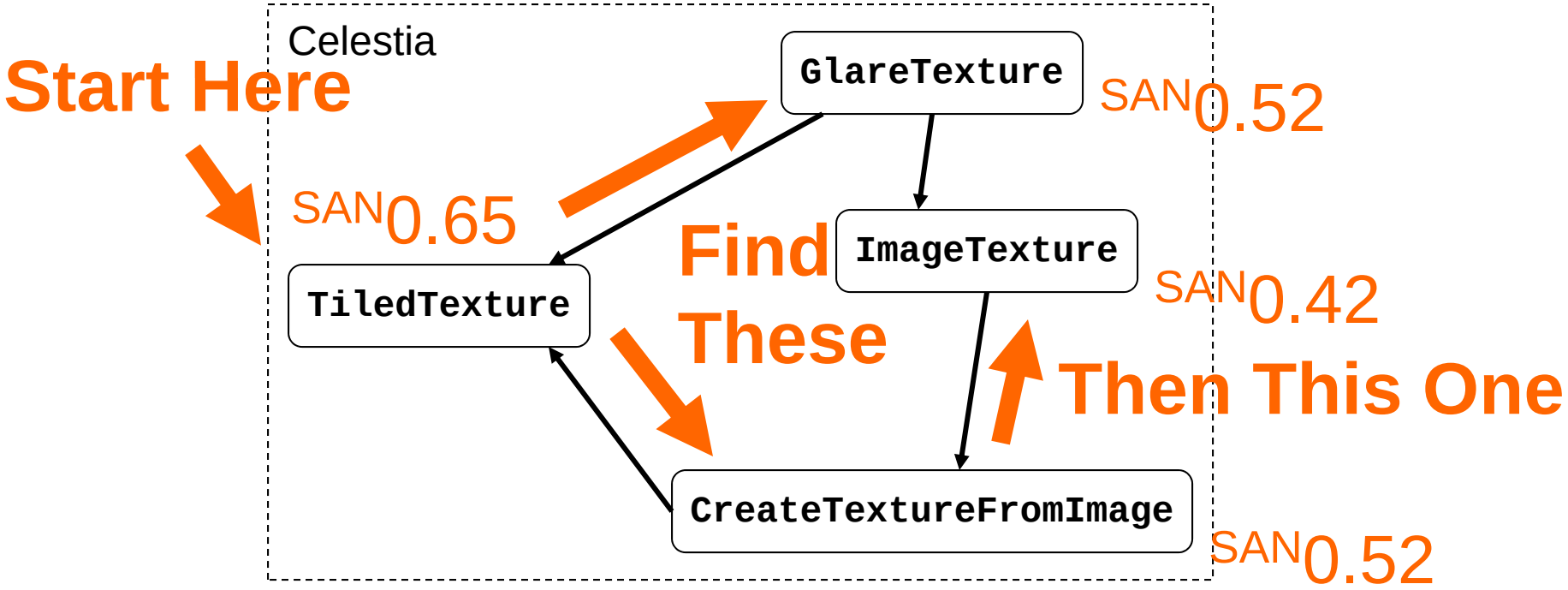

Relevant Functions
SAN
FCG
Function Graph Builder
Projects Archive
P_{PR}
PageRank
Metadata Builder
Projects Metadata
IR Engine

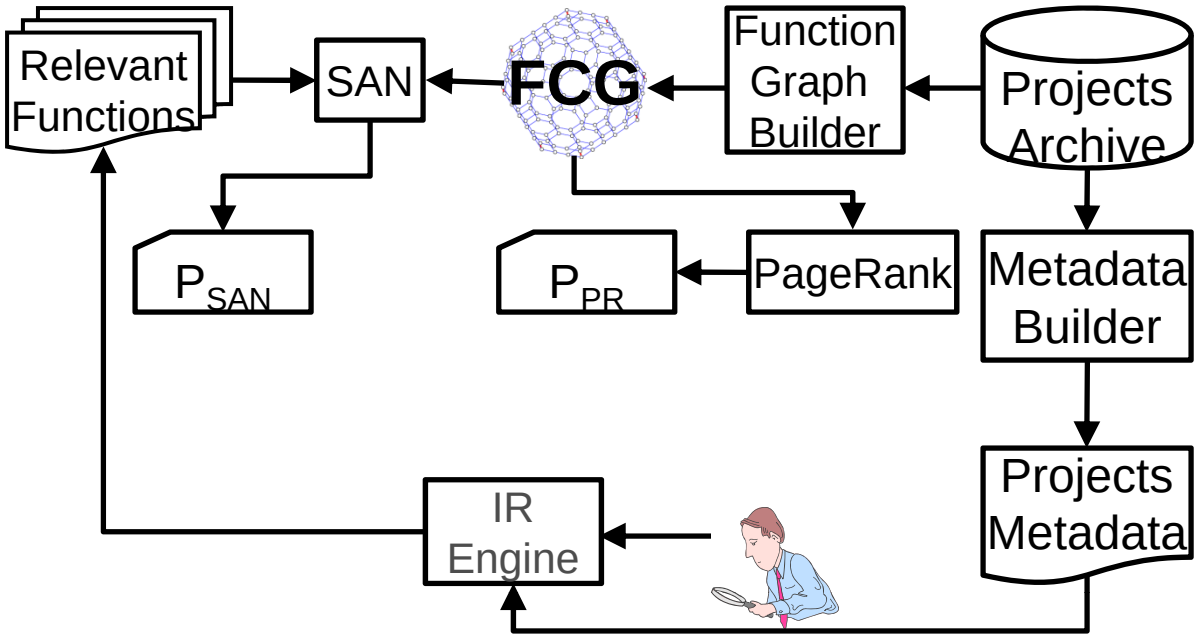

Celestia
Start Here
GlareTexture
SAN 0.52
SAN 0.65
Find These
ImageTexture
TiledTexture
CreateTextureFromImage
SAN 0.52

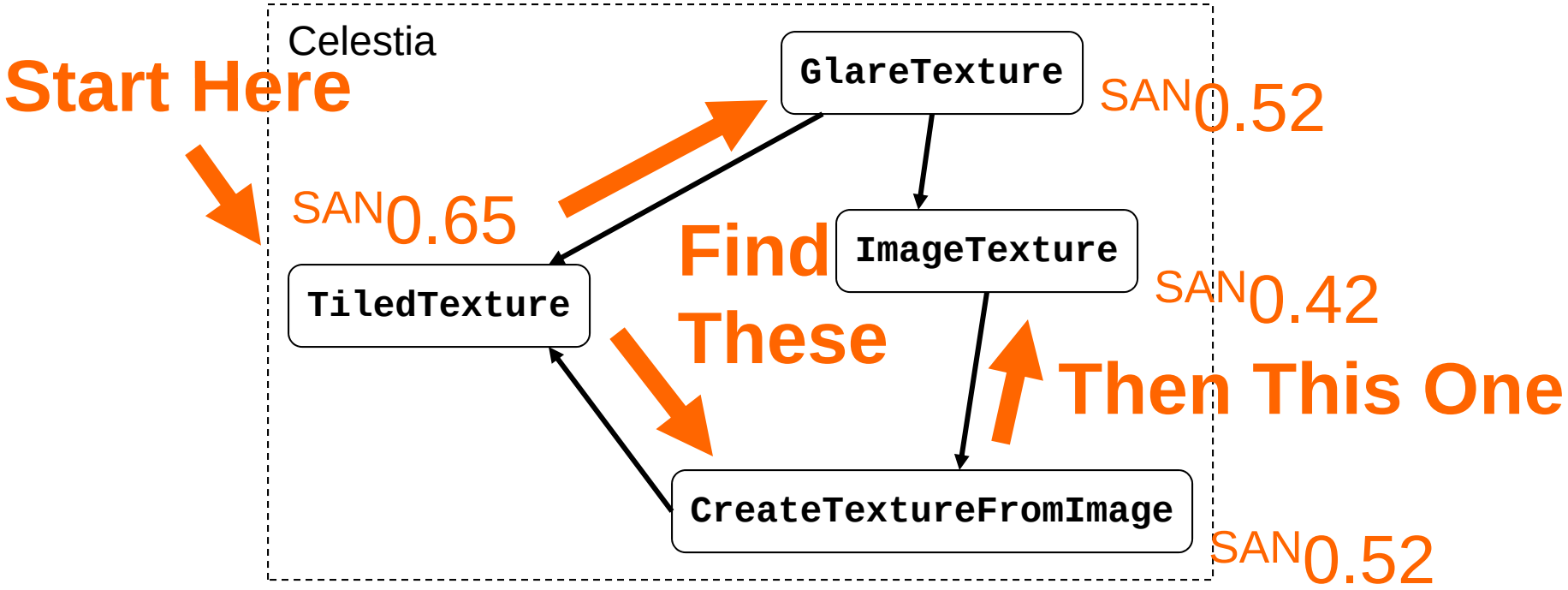

Relevant Functions
SAN
FCG
Function Graph Builder
Projects Archive
P_{PR}
PageRank
Metadata Builder
Projects Metadata
IR Engine

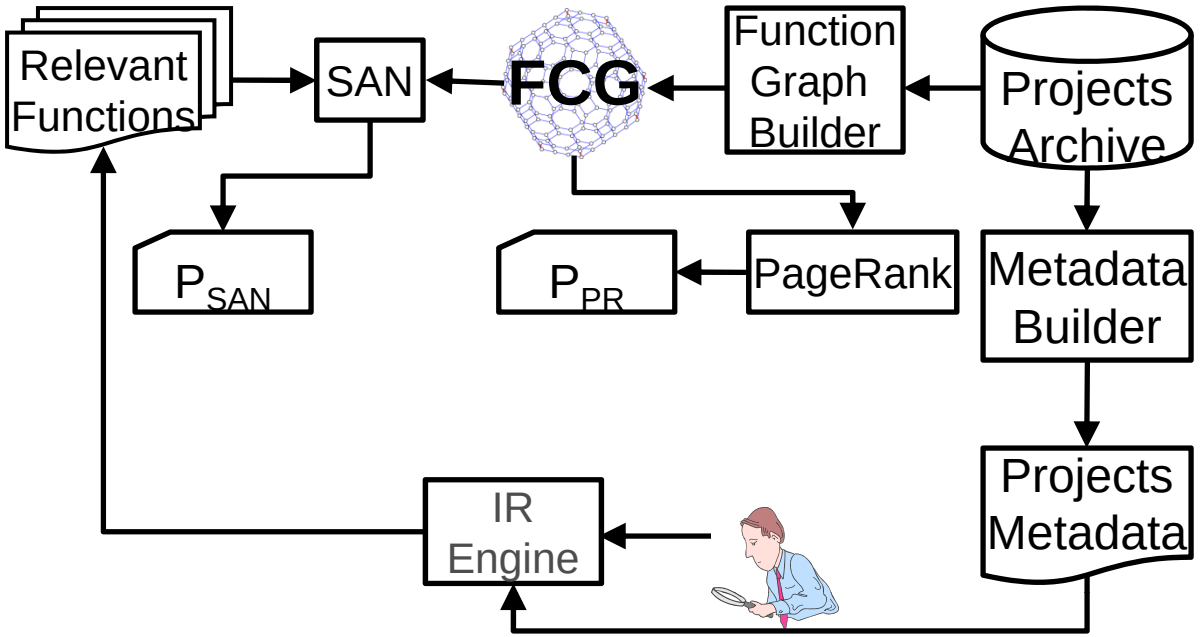

Celestia
Start Here
GlareTexture
SAN 0.52
SAN 0.65
Find These
ImageTexture
SAN 0.42
TiledTexture
Then This One
CreateTextureFromImage
SAN 0.52

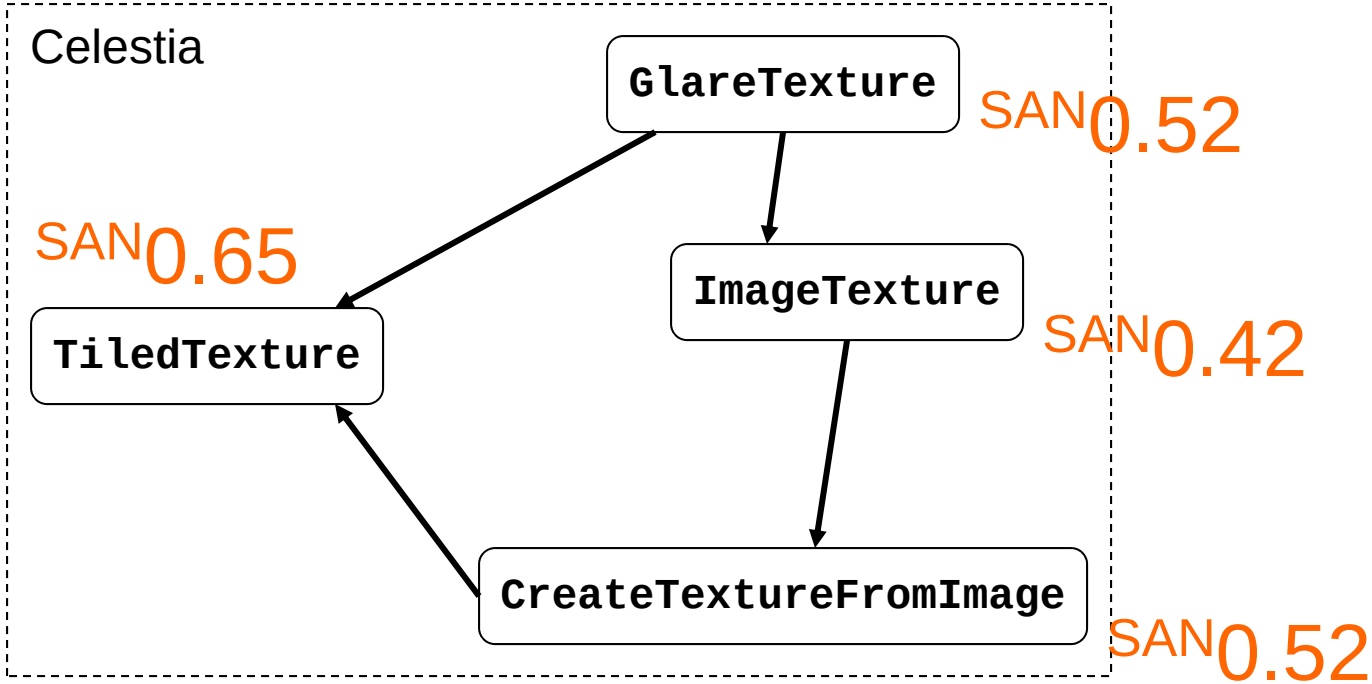

Relevant Functions
SAN
FCG
Function Graph Builder
Projects Archive
P_{PR}
PageRank
Metadata Builder
Projects Metadata
IR Engine

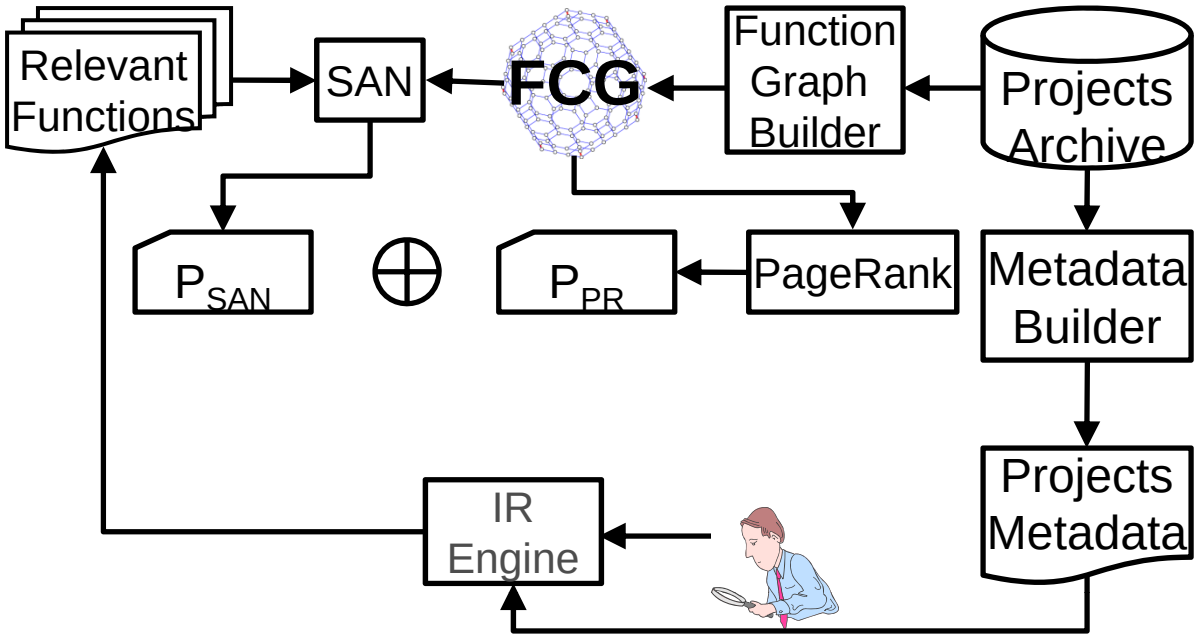

Celestia
Start Here
GlareTexture
SAN 0.52
SAN 0.65
Find These
ImageTexture
SAN 0.42
TiledTexture
Then This One
CreateTextureFromImage
SAN 0.52

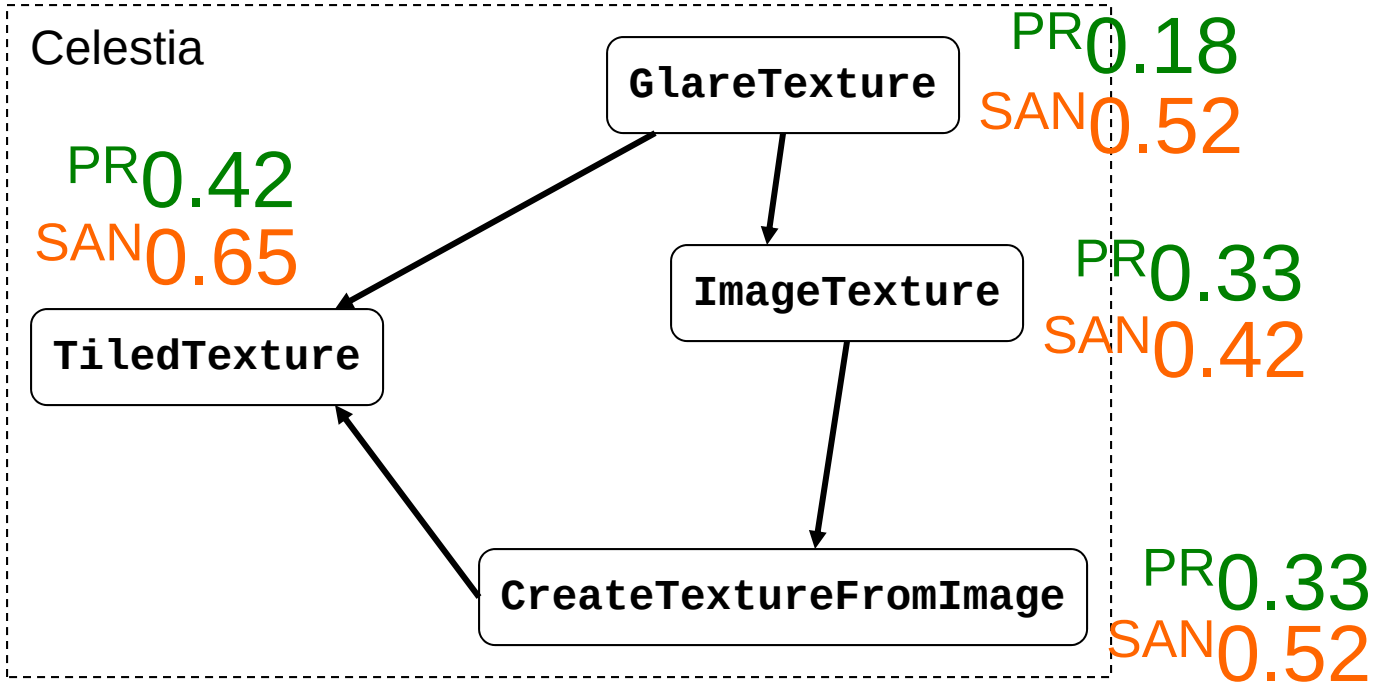

Relevant Functions
SAN
FCG
Function Graph Builder
Projects Archive
P_{SAN}
P_{PR}
PageRank
Metadata Builder
Projects Metadata
IR Engine

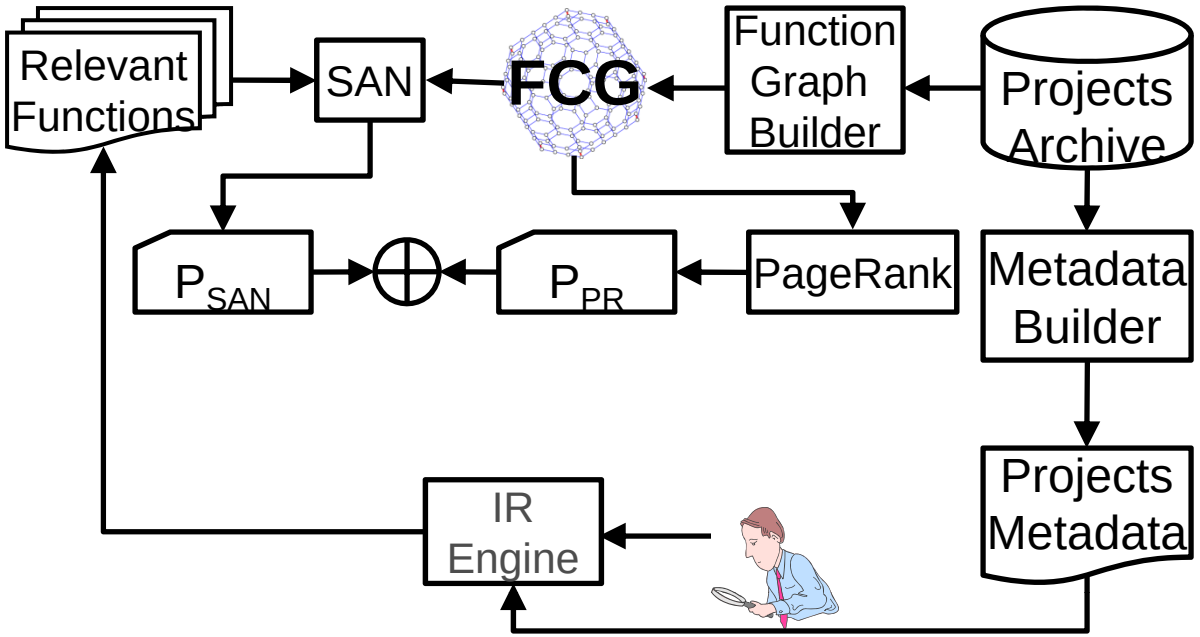

Celestia
Start Here
GlareTexture
SAN 0.52
SAN 0.65
Find These
ImageTexture
SAN 0.42
TiledTexture
Then This One
CreateTextureFromImage
SAN 0.52

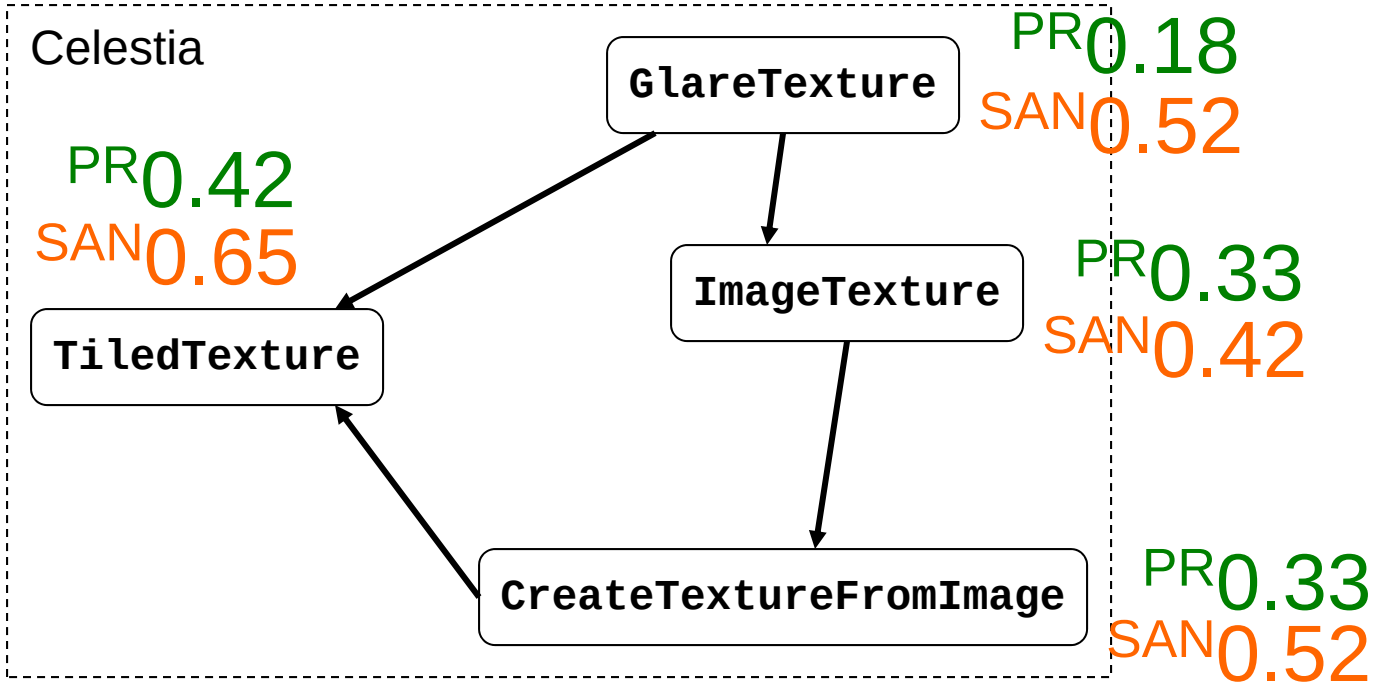

Relevant Functions
SAN
FCG
Function Graph Builder
Projects Archive
P_{SAN}
P_{PR}
PageRank
Metadata Builder
Projects Metadata
IR Engine

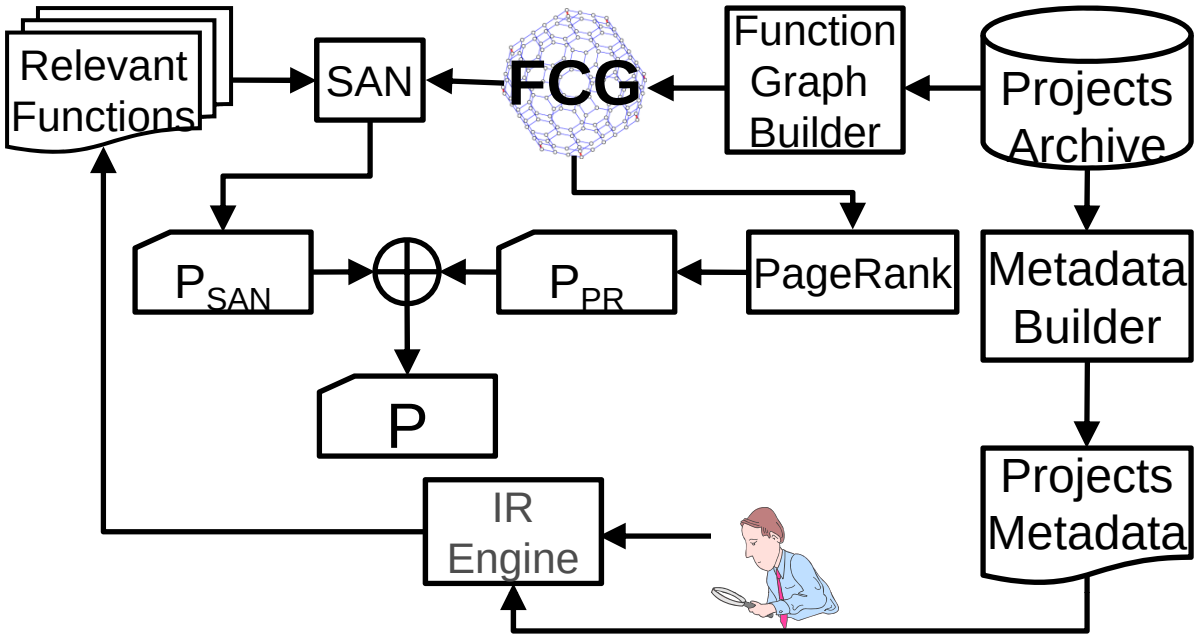

Celestia
GlareTexture
SAN 0.52
SAN 0.65
ImageTexture
TiledTexture
SAN 0.42
CreateTextureFromImage
SAN 0.52

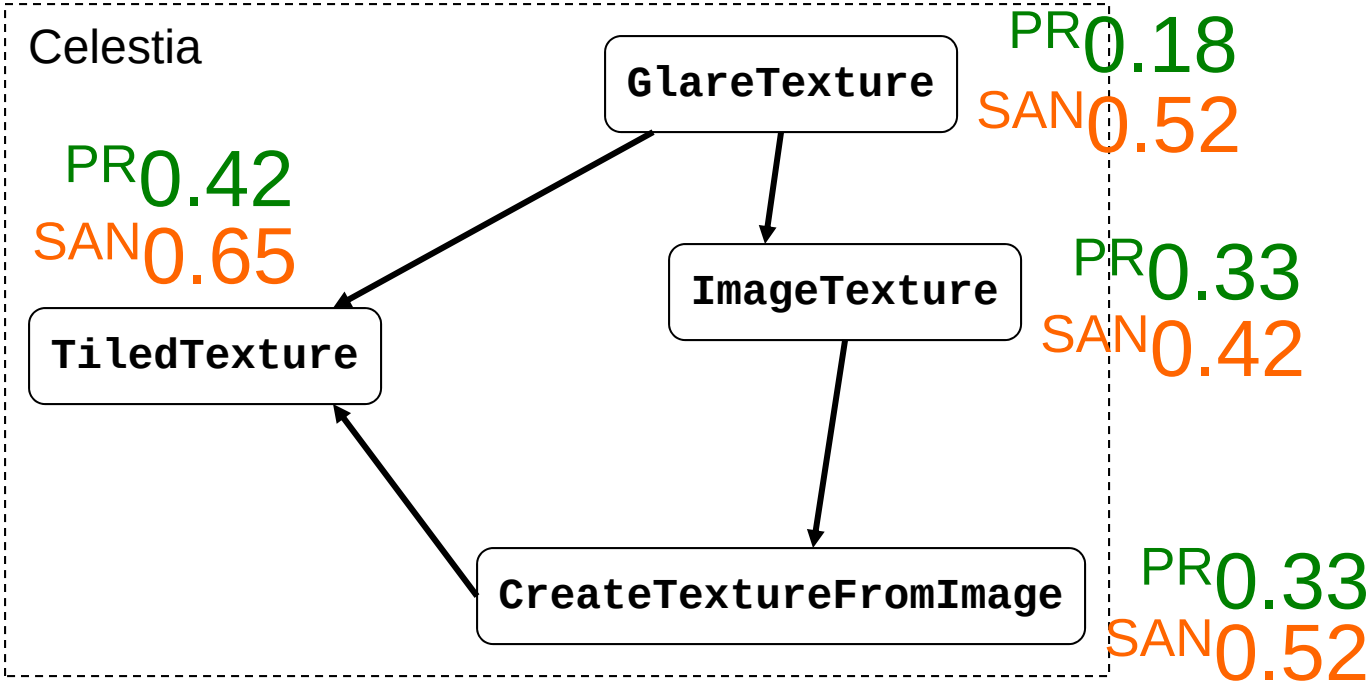

Relevant Functions
SAN
FCG
Function Graph Builder
Projects Archive
P_{SAN}
P_{PR}
PageRank
Metadata Builder
Projects Metadata
IR Engine

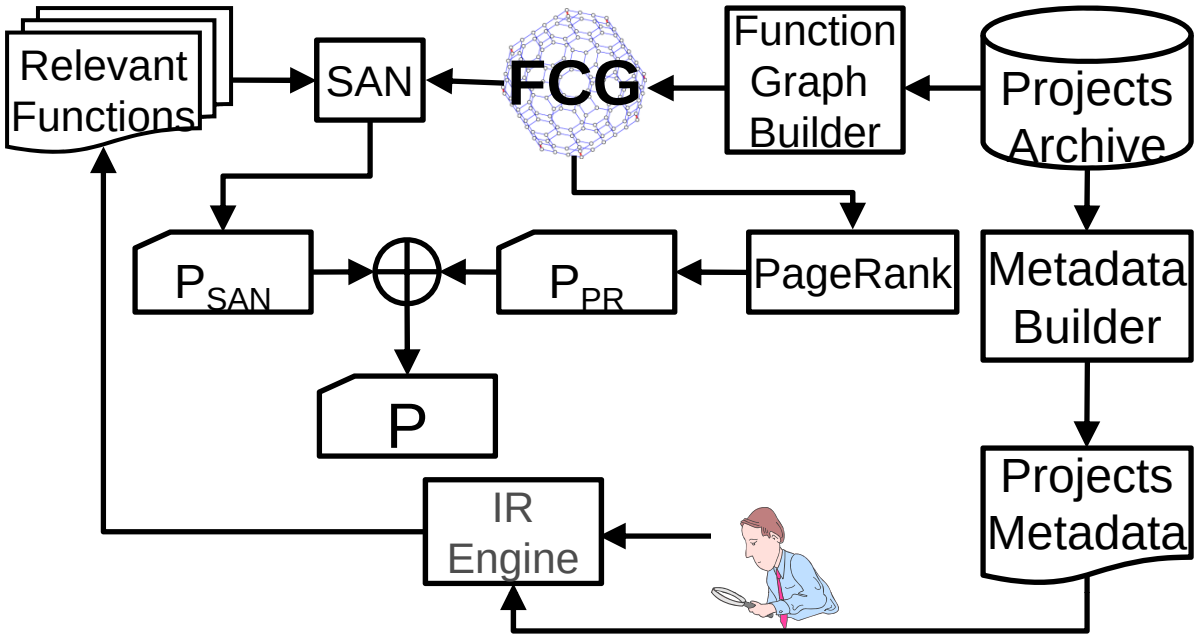

Celestia
GlareTexture
PR0.18
SAN0.52
PR0.42
SAN0.65
ImageTexture
PR0.33
SAN0.42
TiledTexture
CreateTextureFromImage
PR0.33
SAN0.52

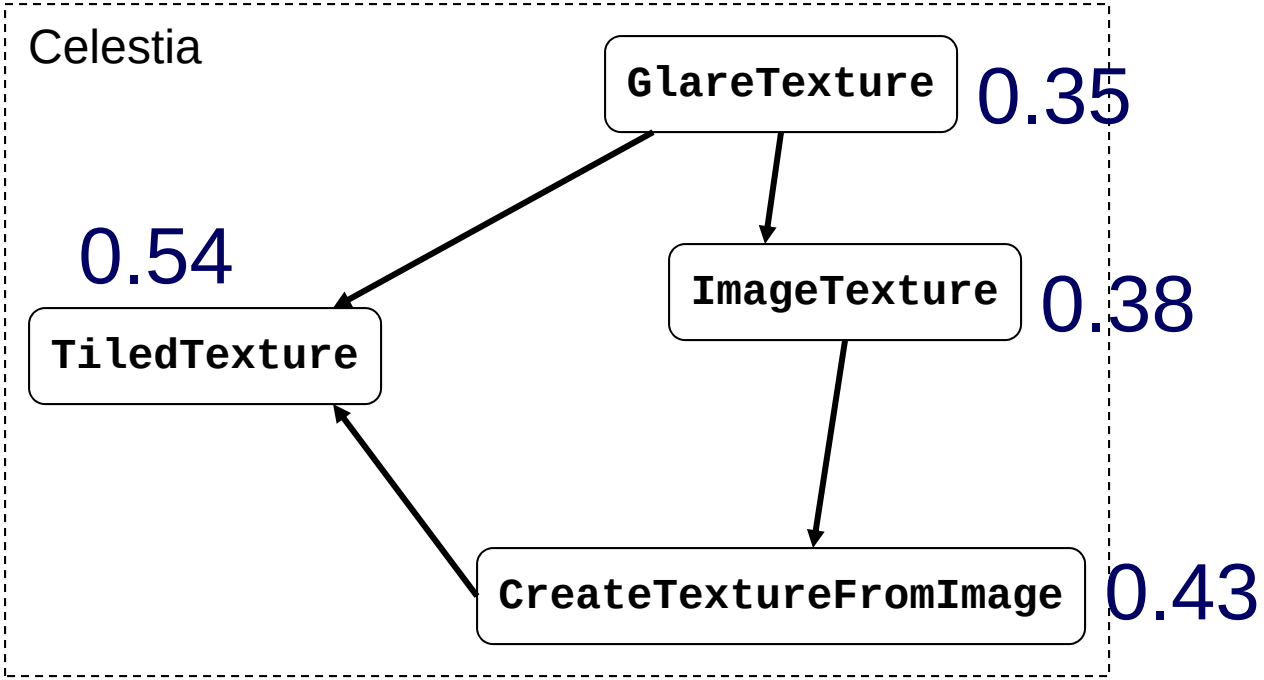

Relevant Functions
SAN
FCG
Function Graph Builder
Projects Archive
P_{SAN}
P_{PR}
PageRank
Metadata Builder
Projects Metadata
IR Engine

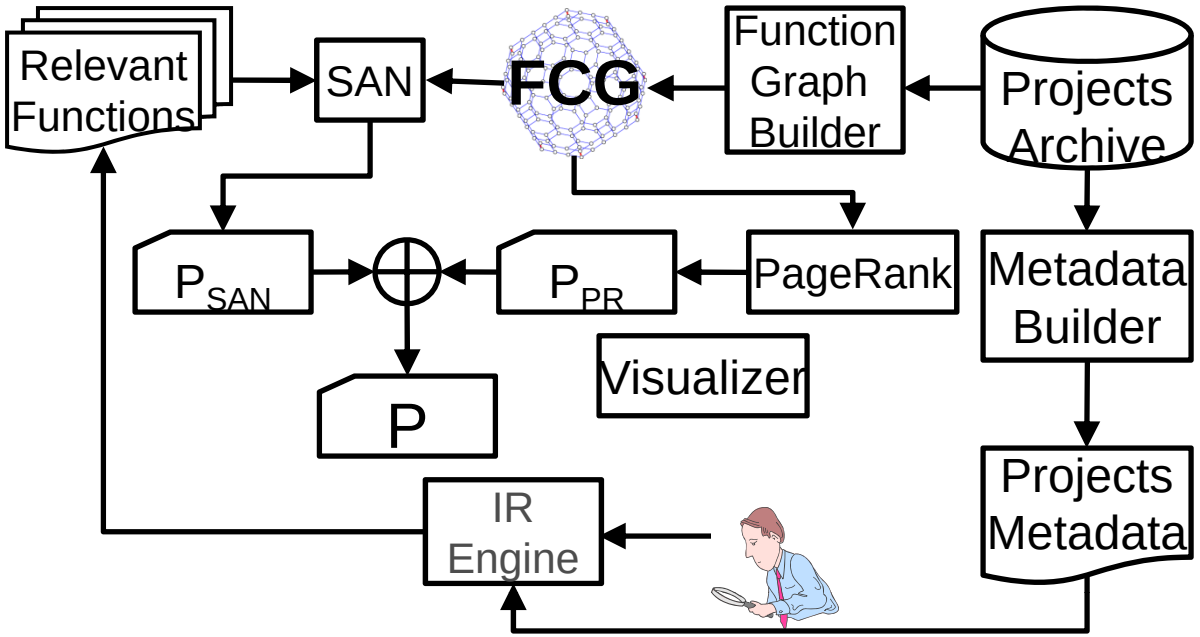

Celestia
GlareTexture
PR0.18
SAN0.52
PR0.42
SAN0.65
ImageTexture
PR0.33
SAN0.42
TiledTexture
CreateTextureFromImage
PR0.33
SAN0.52

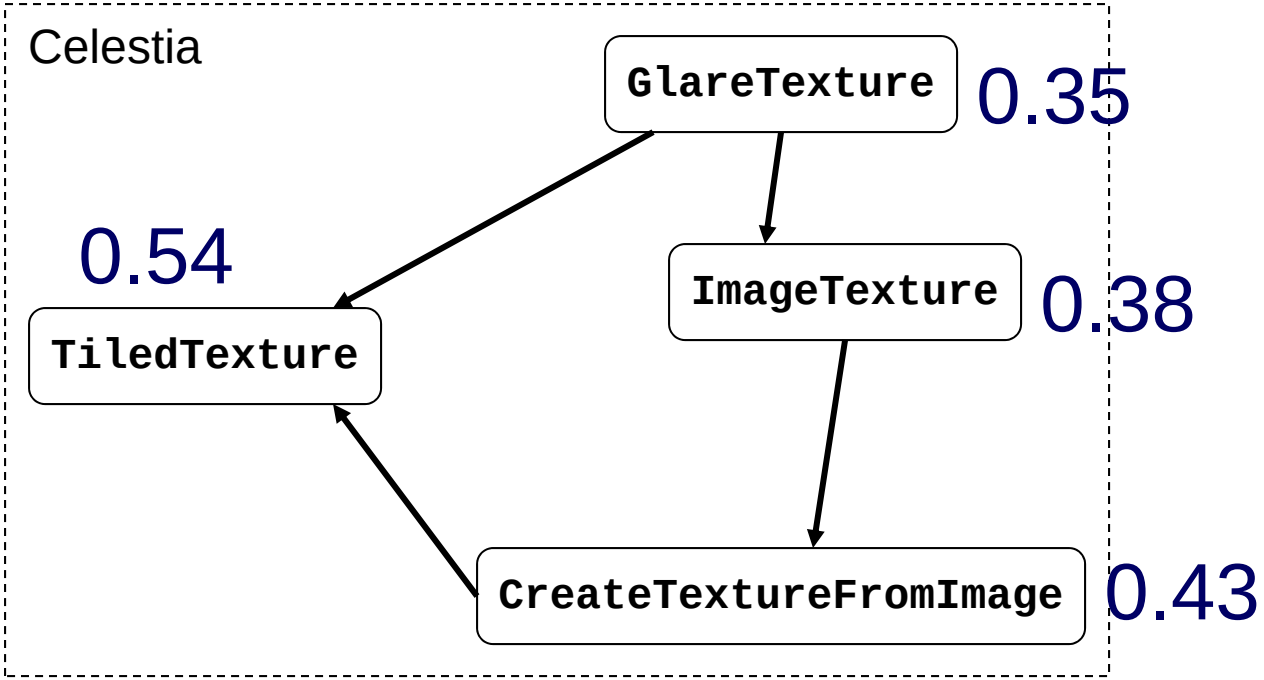

Relevant Functions
SAN
FCG
Function Graph Builder
Projects Archive
P_{SAN}
P_{PR}
PageRank
Metadata Builder
P
IR Engine
Projects Metadata

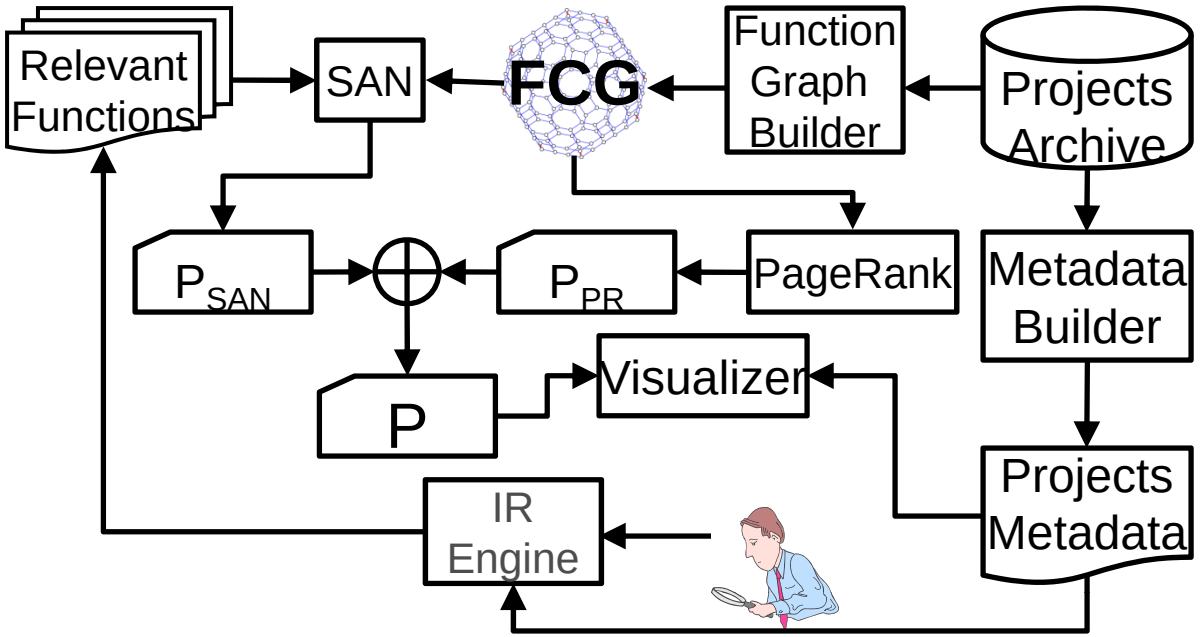

Celestia
GlareTexture
PR 0.18
SAN 0.52
PR 0.42
SAN 0.65
ImageTexture
PR 0.33
SAN 0.42
TiledTexture
CreateTextureFromImage
PR 0.33
SAN 0.52

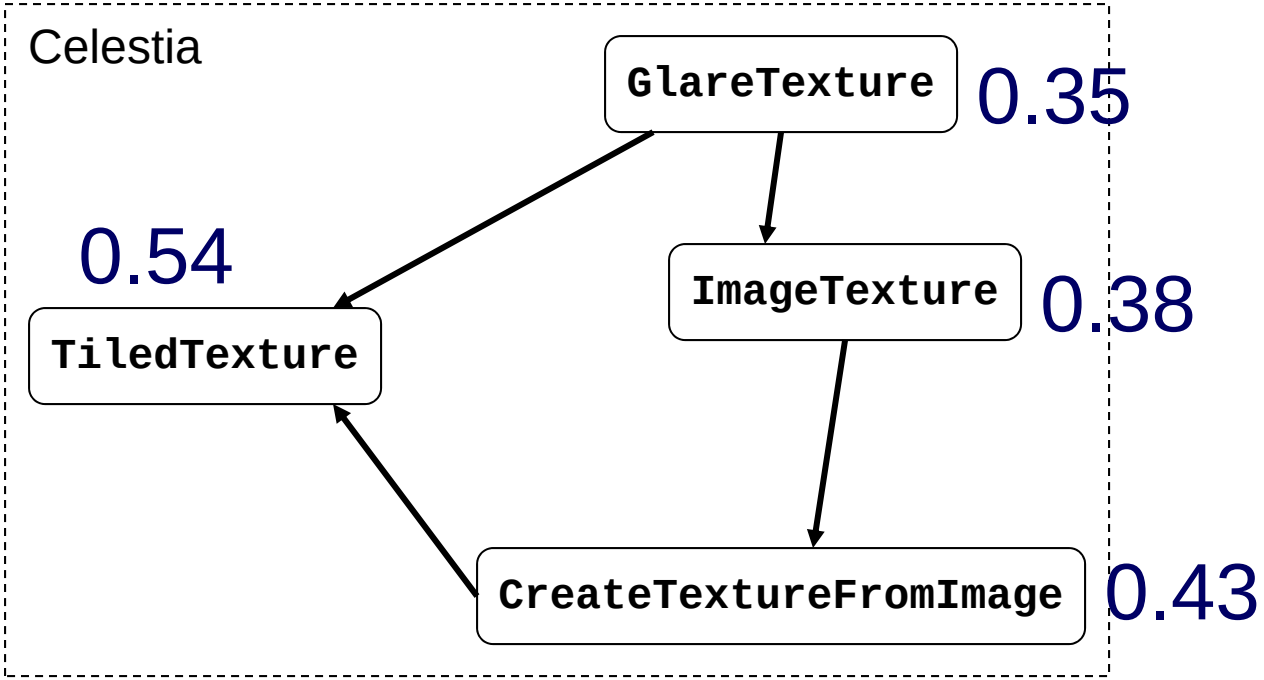

Relevant Functions
SAN
FCG
Function Graph Builder
Projects Archive
P_{SAN}
P_{PR}
PageRank
Metadata Builder
P
IR Engine
Projects Metadata

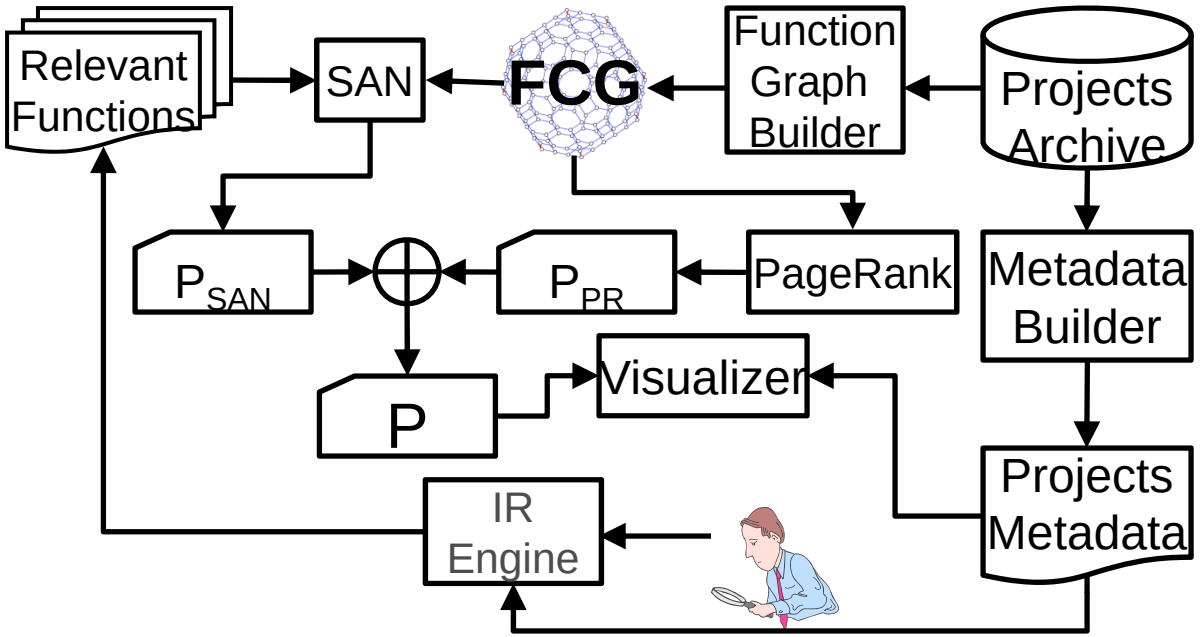

Celestia
GlareTexture
0.35
0.54
ImageTexture
0.38
TiledTexture
CreateTextureFromImage
0.43

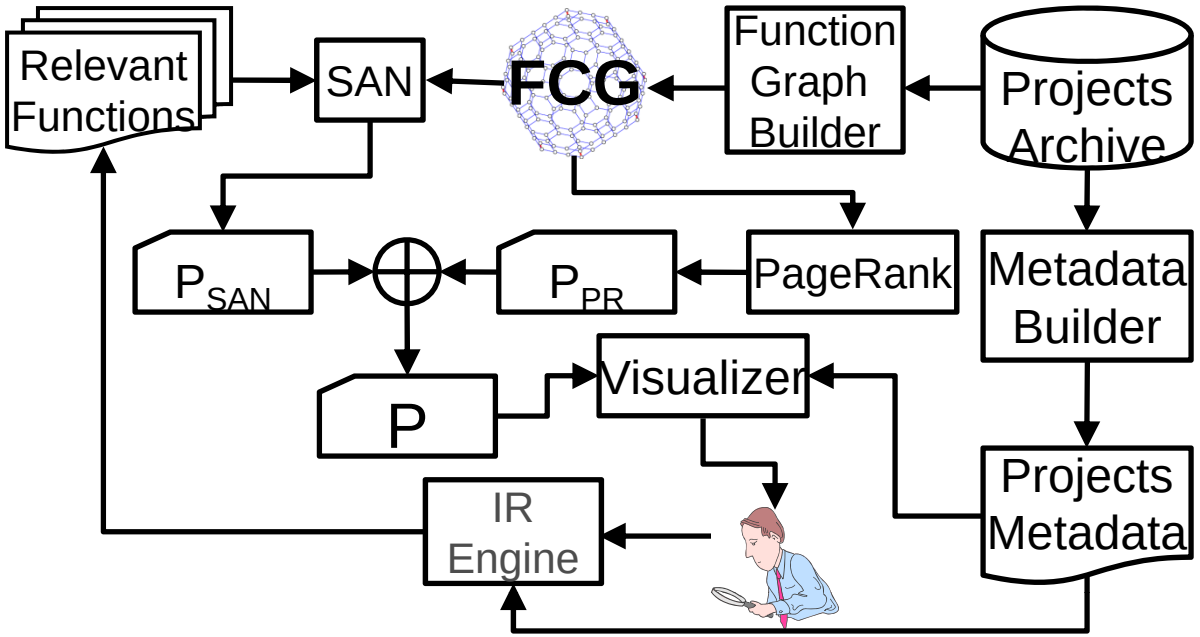

Relevant Functions
SAN
FCG
Function Graph Builder
Projects Archive
P_{SAN}
P_{PR}
PageRank
Metadata Builder
P
Visualizer
IR Engine
Projects Metadata

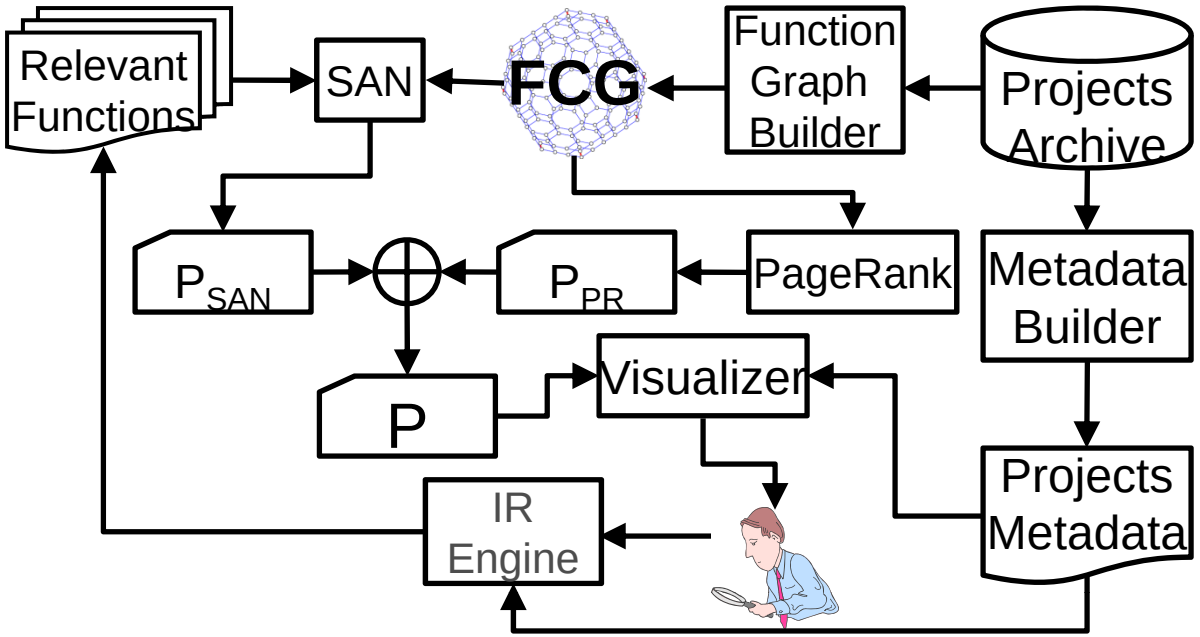

Celestia
GlareTexture
0.35
0.54
ImageTexture
0.38
TiledTexture
CreateTextureFromImage
0.43


Relevant Functions
SAN
FCG
Function Graph Builder
Projects Archive
P_{SAN}
P_{PR}
PageRank
Metadata Builder
P
Visualizer
IR Engine
Projects Metadata

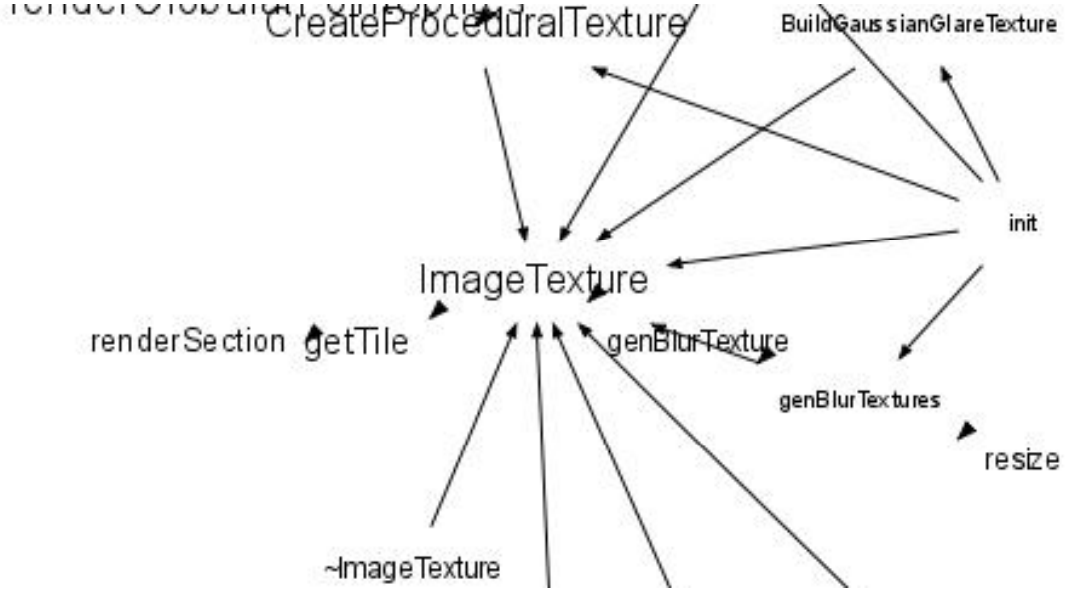

Celestia
GlareTexture
0.35
0.54
ImageTexture
0.38
TiledTexture
CreateTextureFromImage
0.43


Relevant Functions
SAN
FCG
Function Graph Builder
Projects Archive
P_{SAN}
P_{PR}
PageRank
Metadata Builder
P
Visualizer
IR Engine
Projects Metadata


Relevant Functions
SAN
FCG
Function Graph Builder
Projects Archive
P_{SAN}
P_{PR}
PageRank
Metadata Builder
P
Visualizer
IR Engine
Projects Metadata



reloadConfiguration	ultimate
CreateProceduralTexture	celestia
genBlurTextures	celestia
CreateTextureFromImage	celestia
viewOptionsMenu	ultimate
CreateMipMaps	cs-pseud
genBlurTexture	celestia
BuildGaussianGlareTexture	celestia
BuildGaussianDiscTexture	celestia
loadTileTexture	celestia
getTile	celestia
renderGalaxyPointSprites	celestia



Software Archive

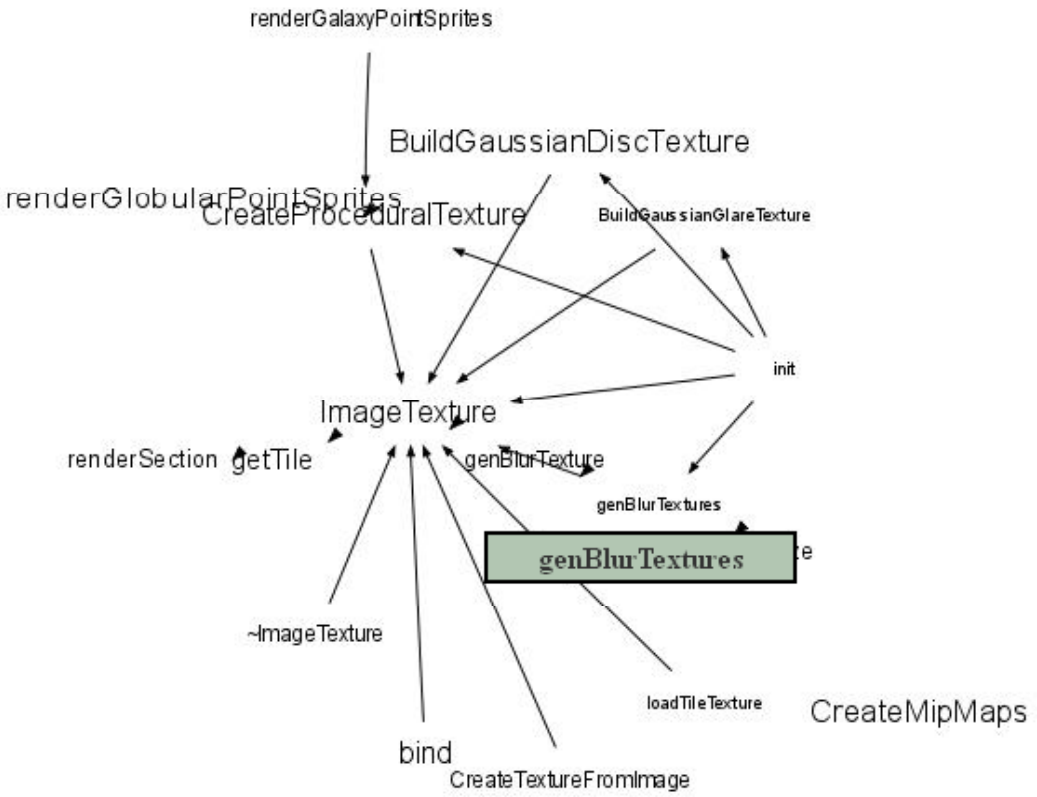
FreeBSD Ports

- 18,203 C/C++ Projects
- 2.4 Million Files
- 8.5 Million Functions
- 32 Million Function Calls
- 270 Million Total Lines of Code

Portfolio Interface

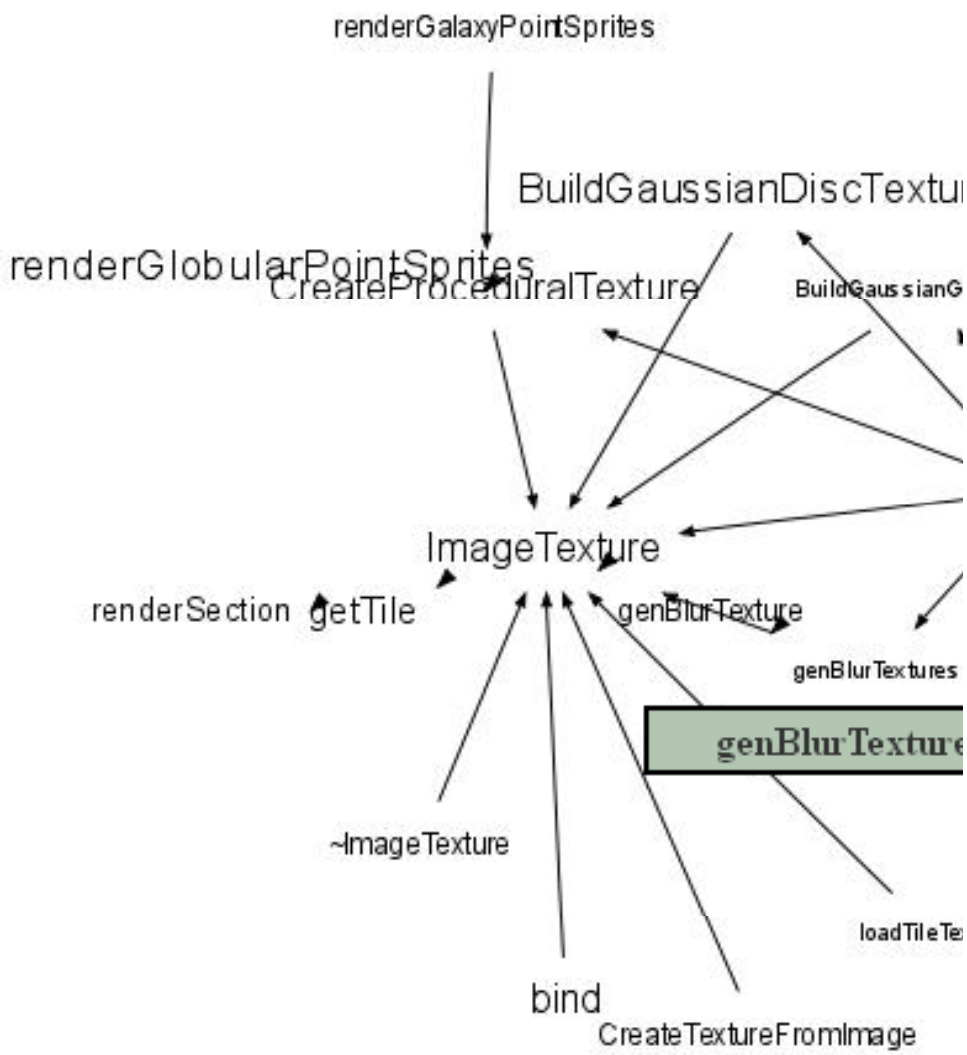
"mip map dithering texture image graphics"

Function Name	Project
CGameWinSystem	ultimate
CGraphicWorld	ultimate
CWinSystem	ultimate
CRenderer	ultimate
ImageTexture	celestia
CSound	ultimate
reloadConfiguration	ultimate
CreateProceduralTexture	celestia
genBlurTextures	celestia
CreateTextureFromImage	celestia
viewOptionsMenu	ultimate
CreateMipMaps	cs-pseud
genBlurTexture	celestia
BuildGaussianGlareTexture	celestia
BuildGaussianDiscTexture	celestia
loadTileTexture	celestia
getTile	celestia
renderGalaxyPointSprites	celestia
renderSection	celestia
renderGlobularPointSprites	celestia
Init	vdrift-2
resize	celestia
init	celestia
~ImageTexture	celestia
bind	celestia



Function **genBlurTextures** is in file **ports_20k/celestia/celestia-1.6.0/src/celengine/render.cpp**.

Function Name	Project
CGameWinSystem	ultimate
CGraphicWorld	ultimate
CWinSystem	ultimate
CRenderer	ultimate
ImageTexture	celestia
CSound	ultimate
reloadConfiguration	ultimate
CreateProceduralTexture	celestia
genBlurTextures	celestia
CreateTextureFromImage	celestia
viewOptionsMenu	ultimate
CreateMipMaps	cs-pseud
genBlurTexture	celestia
BuildGaussianGlareTexture	celestia
BuildGaussianDiscTexture	celestia
loadTileTexture	celestia
getTile	celestia
renderGalaxyPointSprites	celestia
renderSection	celestia
renderGlobularPointSprites	celestia
Init	vdrift-2
resize	celestia
init	celestia
~ImageTexture	celestia
bind	celestia

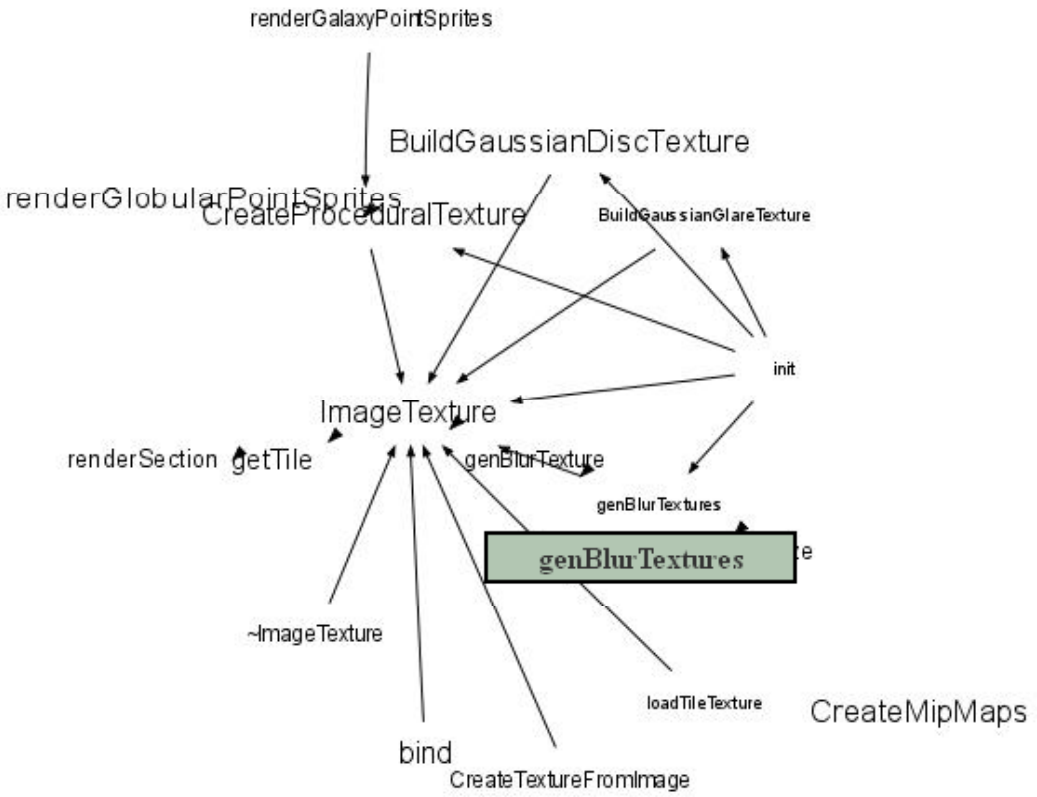


Function **genBlurTextures** is in file **ports_20k/celestia/celestia-1.6.0**

Portfolio Interface

"mip map dithering texture image graphics"

Function Name	Project
CGameWinSystem	ultimate
CGraphicWorld	ultimate
CWinSystem	ultimate
CRenderer	ultimate
ImageTexture	celestia
CSound	ultimate
reloadConfiguration	ultimate
CreateProceduralTexture	celestia
genBlurTextures	celestia
CreateTextureFromImage	celestia
viewOptionsMenu	ultimate
CreateMipMaps	cs-pseud
genBlurTexture	celestia
BuildGaussianGlareTexture	celestia
BuildGaussianDiscTexture	celestia
loadTileTexture	celestia
getTile	celestia
renderGalaxyPointSprites	celestia
renderSection	celestia
renderGlobularPointSprites	celestia
Init	vdrift-2
resize	celestia
init	celestia
~ImageTexture	celestia
bind	celestia

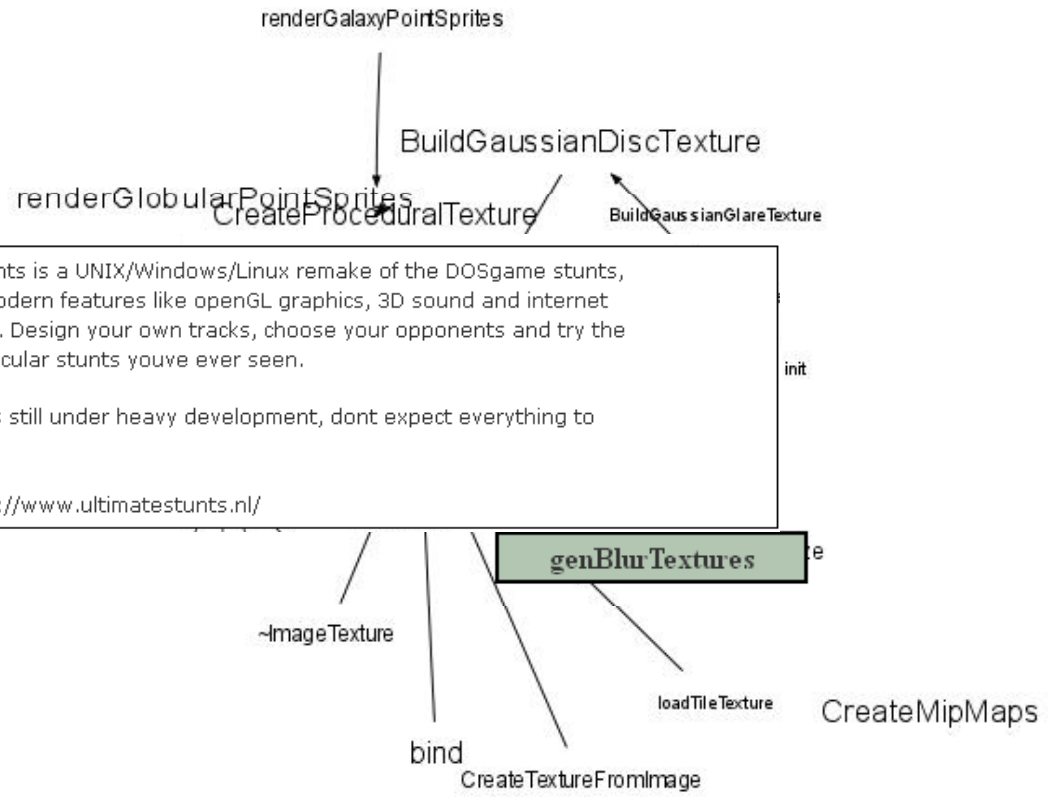


Function **genBlurTextures** is in file **ports_20k/celestia/celestia-1.6.0/src/celengine/render.cpp**.

Portfolio Interface

"mip map dithering texture image graphics"

Function Name	Project
CGameWinSystem	ultimate
CGraphicWorld	ultimate
CWinSystem	ultimate
CRenderer	ultimate
ImageTexture	celestia
CSound	ultimate
reloadConfiguration	ultimate
CreateProceduralTexture	
genBlurTextures	
CreateTextureFromImage	
viewOptionsMenu	
CreateMipMaps	
genBlurTexture	
BuildGaussianGlareTexture	
BuildGaussianDiscTexture	
loadTileTexture	celestia
getTile	celestia
renderGalaxyPointSprites	celestia
renderSection	celestia
renderGlobularPointSprites	celestia
Init	vdrift-2
resize	celestia
init	celestia
~ImageTexture	celestia
bind	celestia

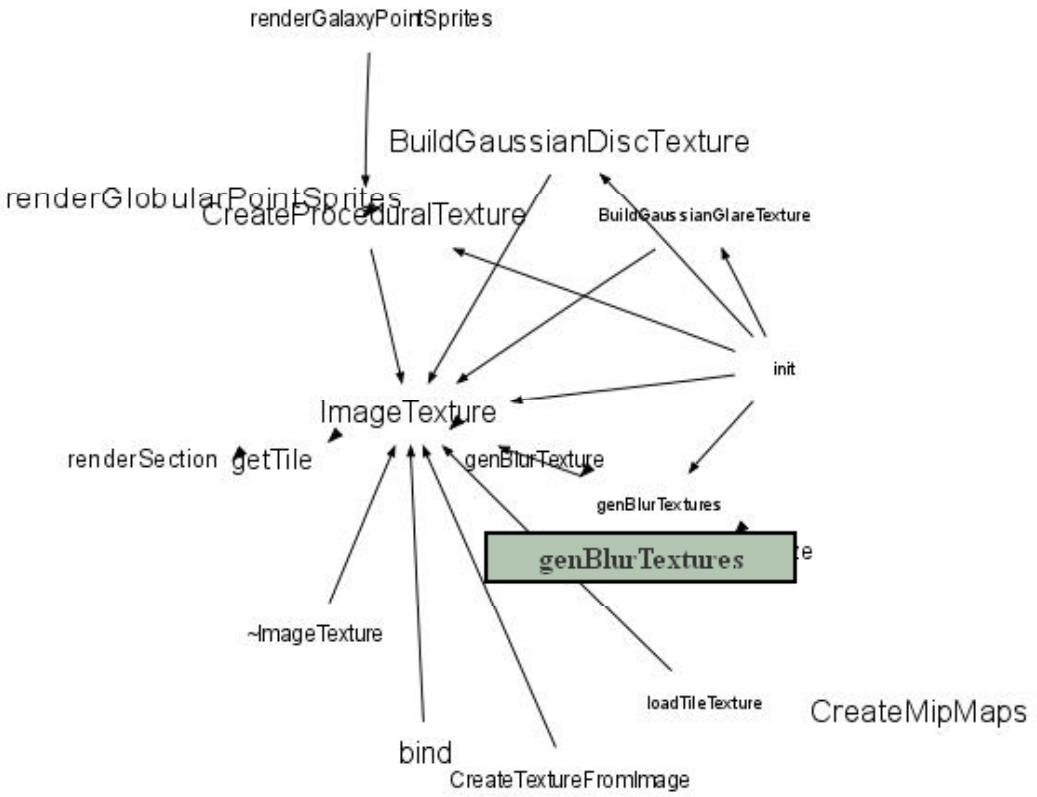


Function **genBlurTextures** is in file **ports_20k/celestia/celestia-1.6.0/src/celengine/render.cpp**.

Portfolio Interface

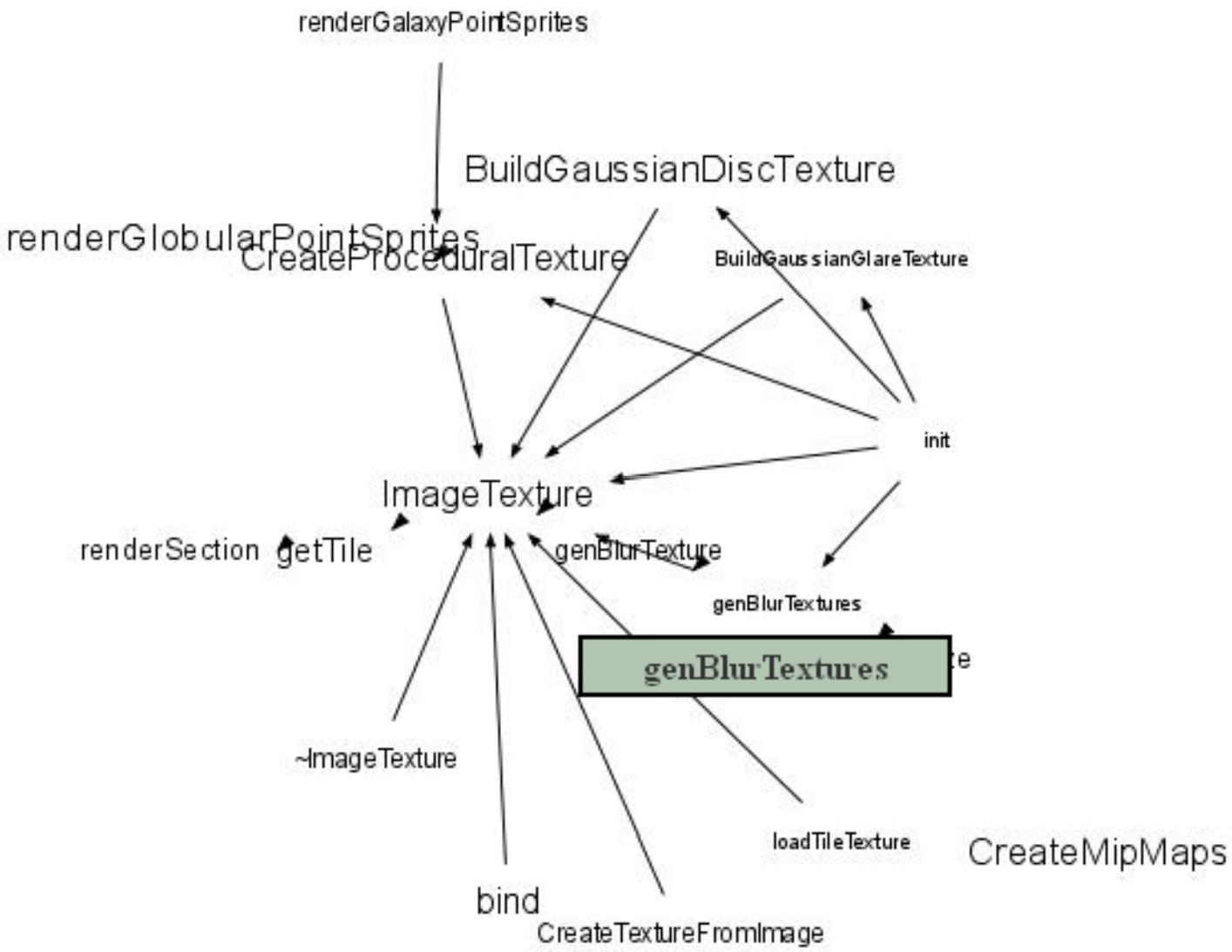
"mip map dithering texture image graphics"

Function Name	Project
CGameWinSystem	ultimate
CGraphicWorld	ultimate
CWinSystem	ultimate
CRenderer	ultimate
ImageTexture	celestia
CSound	ultimate
reloadConfiguration	ultimate
CreateProceduralTexture	celestia
genBlurTextures	celestia
CreateTextureFromImage	celestia
viewOptionsMenu	ultimate
CreateMipMaps	cs-pseud
genBlurTexture	celestia
BuildGaussianGlareTexture	celestia
BuildGaussianDiscTexture	celestia
loadTileTexture	celestia
getTile	celestia
renderGalaxyPointSprites	celestia
renderSection	celestia
renderGlobularPointSprites	celestia
Init	vdrift-2
resize	celestia
init	celestia
~ImageTexture	celestia
bind	celestia



Function **genBlurTextures** is in file **ports_20k/celestia/celestia-1.6.0/src/celengine/render.cpp**.

ultimate
ultimate
ultimate
ultimate
celestia
ultimate
ultimate
celestia
celestia
celestia
ultimate
cs-pseud
celestia
celestia
celestia
celestia
celestia
celestia
celestia
celestia
celestia
vdrift-2
celestia
celestia
celestia
celestia



Function **genBlurTextures** is in file **ports_20k/celestia/celestia-1.6.0/src/celengine/render.cpp**.

Experiment

To compare **Portfolio** with **Google Code Search** and **Koders**

49 C/C++ Programmers participated

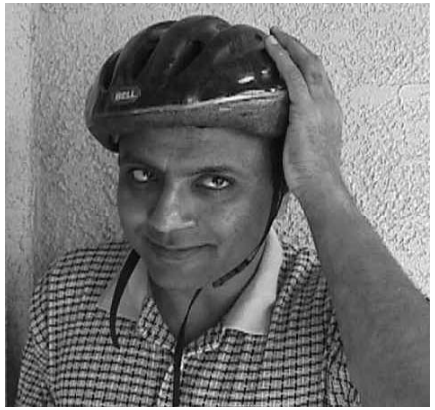
- 44 Professionals from Accenture
- 5 Students from the University of Illinois at Chicago

Large Case Studies are Rare

"First, it is very difficult to scale human experiments to get quantitative, significant measures of usefulness; this type of large-scale human study is very rare.

Second, comparing different recommenders using human evaluators would involve carefully designed, time-consuming experiments; this is also extremely rare."

Saul, Filkov, Devanbu, Bird
Recommending Random Walks, ESEC/FSE'07



Participants' Role

1) Receive Task and Search Engine

Write a utility for dithering mip map images that are used for rendering texture.

2) Translate Task to Query, enter into Engine

Likert Scale - Confidence

 Completely irrelevant – there is absolutely nothing that the participant can use from this retrieved code fragments, nothing in it is related to keywords that the participant chose based on the descriptions of the tasks.

 Mostly irrelevant – a retrieved code fragment is only remotely relevant to a given task; it is unclear how to reuse it.

 Mostly relevant – a retrieved code fragment is relevant to a given task and participant can understand with some modest effort how to reuse it to solve a given task.

 Highly relevant – The participant is highly confident that code fragment can be reused and s/he clearly see how to use it.

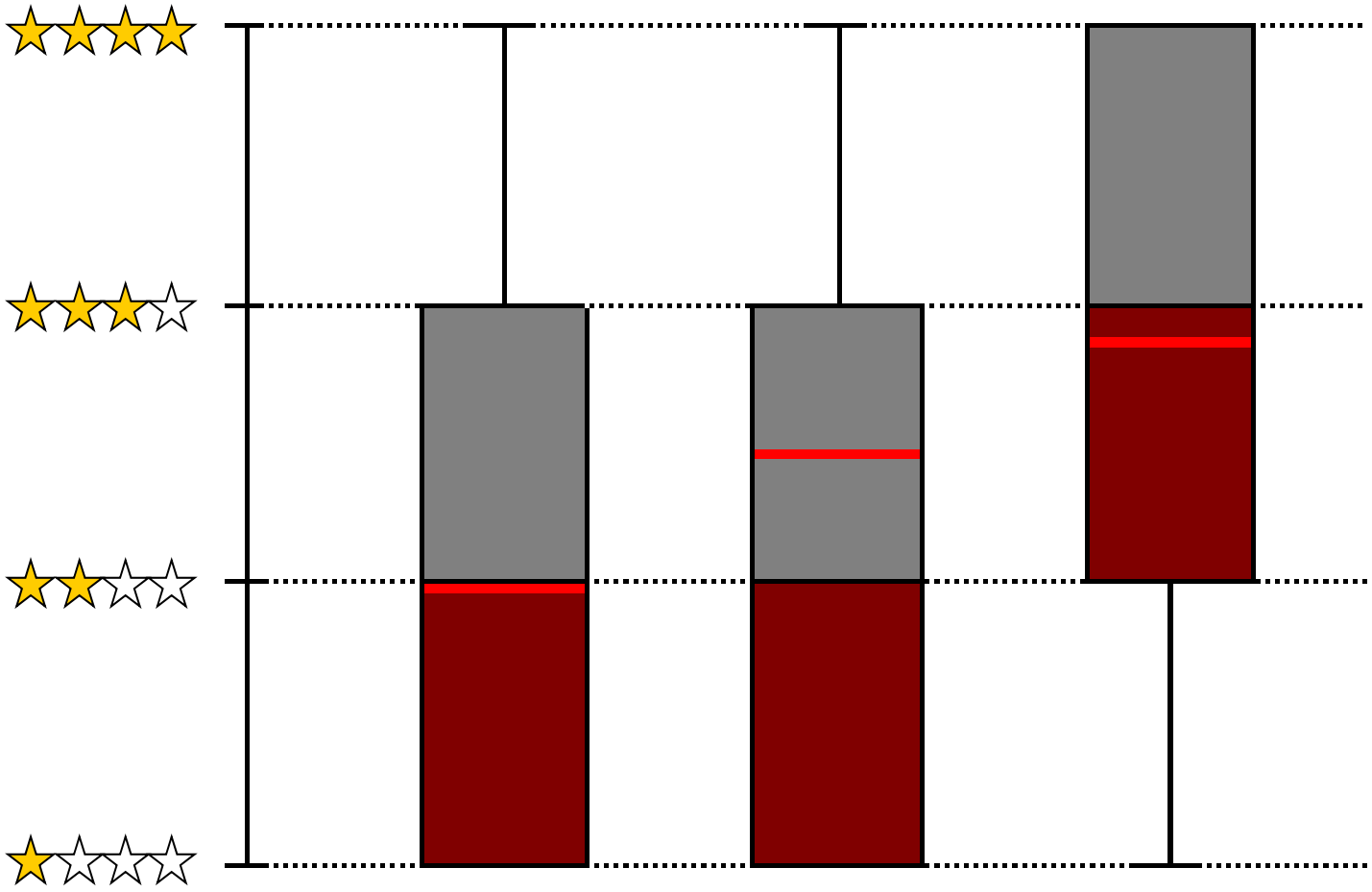
Analysis of the Results

Metrics:

- Confidence (C)
- Precision (P)
- Normalized Discounted Cumulative Gain (NG)

Search Engine	Queries Entered	Responses Rated
Portfolio	184	1276
Google Code Search	198	1373
Koders	208	1486

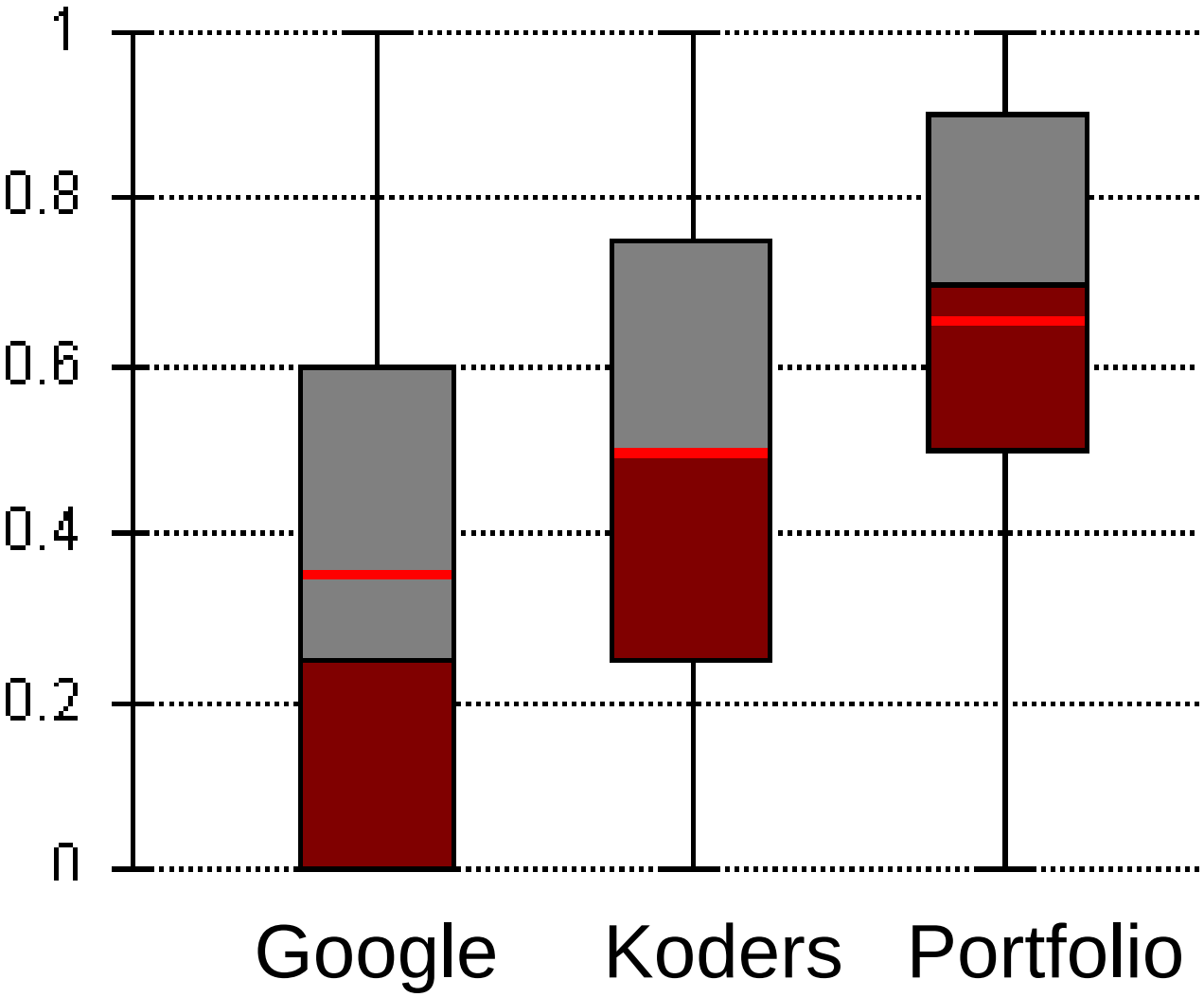
Results – Confidence



Google Koders Portfolio

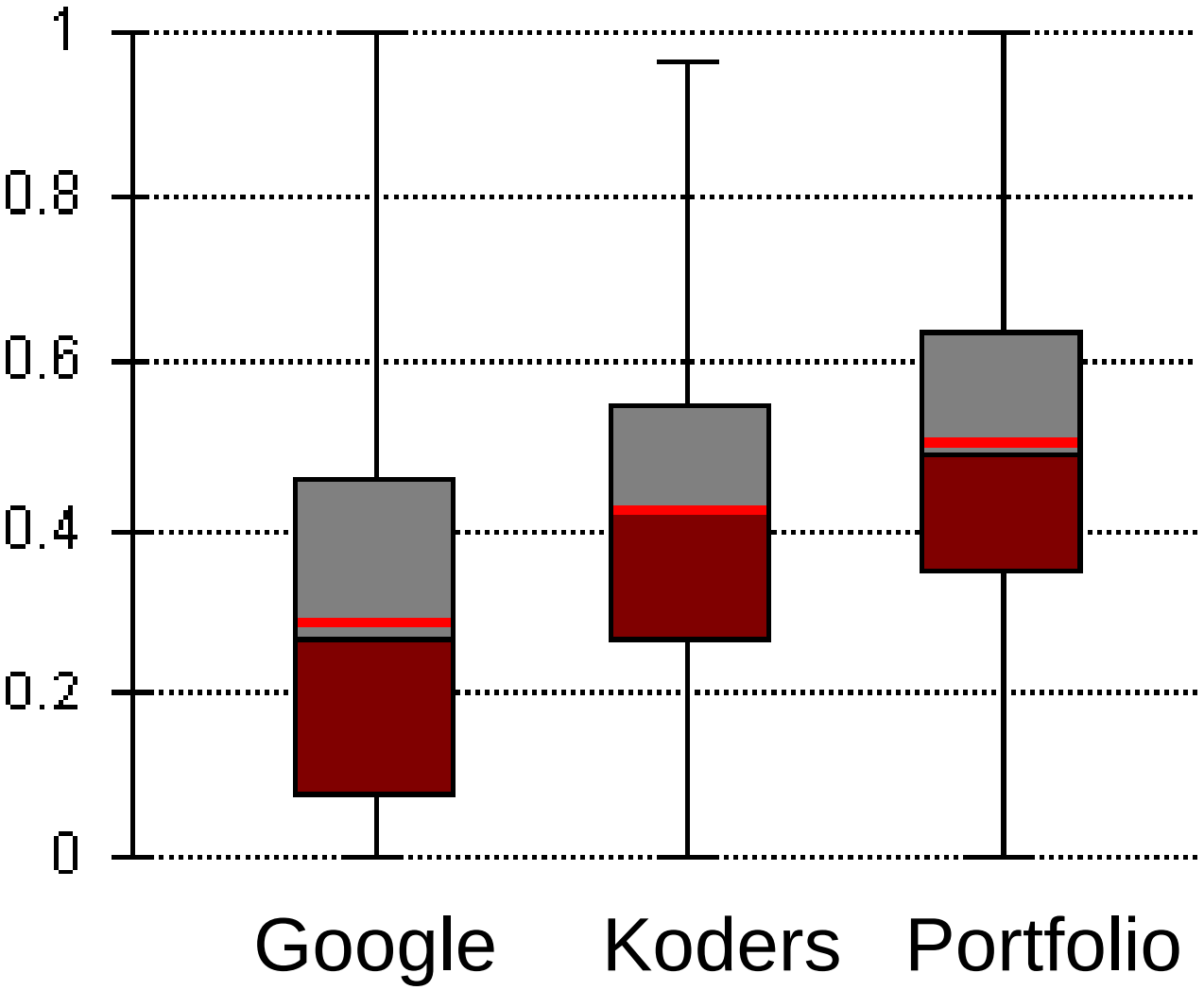
p	$<5.0 \cdot 10^{-108}$
F	261.3
F_{crit}	3.01

Results – Precision



p	$<8.6 \cdot 10^{-22}$
F	52.5
F_{crit}	3.01

Results – Normalized Discounted Gain



p	$<2.5 \cdot 10^{-18}$
F	43.8
F_{crit}	3.01

Statistical Analysis – ANOVA

Null Hypothesis rejected in all cases:

$\mathbf{H_0}$ – There is no difference in the C, P, or NG mean values among users of Portfolio, Google Code Search, and Koders.

$\mathbf{H_1}$ – There is statistically-significant difference in the numbers of C, P, and NG mean values among users of Portfolio, Google Code Search, and Koders.

Metric	p	F	$F_{critical}$
Confidence	$< 5.0 \cdot 10^{-108}$	261.3	3.01
Precision	$< 8.6 \cdot 10^{-22}$	52.5	3.01
Discounted Gain	$< 2.5 \cdot 10^{-18}$	43.8	3.01

Statistical Analysis – ANOVA

Null Hypothesis rejected in all cases:

~~H_0 – There is no difference in the C, P, or NG mean values among users of Portfolio, Google Code Search, and Koders.~~

H_1 – There is statistically-significant difference in the numbers of C, P, and NG mean values among users of Portfolio, Google Code Search, and Koders.

Metric	p	F	$F_{critical}$
Confidence	$< 5.0 \cdot 10^{-108}$	261.3	3.01
Precision	$< 8.6 \cdot 10^{-22}$	52.5	3.01
Discounted Gain	$< 2.5 \cdot 10^{-18}$	43.8	3.01

Programmer Experience Relations

Do more-experienced programmers report different results than less-experienced programmers?

Null Hypothesis **not** rejected:

$\mathbf{H_0}$ – There is no difference in the C, P, or NG mean values among experienced and less-experienced users of Portfolio, Google Koder Search, and Koders.

$\mathbf{H_1}$ – There is statistically-significant difference in the numbers of C, P, and NG mean values among experienced and less-experienced users of Portfolio, Google Code Search, and Koders.

Programmer Experience Relations

Do more-experienced programmers report different results than less-experienced programmers?

Null Hypothesis **not** rejected:

H_0 – There is no difference in the C, P, or NG mean values among experienced and less-experienced users of Portfolio, Google Koder Search, and Koders.

~~**H_1** – There is statistically-significant difference in the numbers of C, P, and NG mean values among experienced and less-experienced users of Portfolio, Google Code Search, and Koders.~~

Responses from Programmers

"The search engine Portfolio is a good search tool ... developers won't waste time exploring different projects or functions."

"Portfolio looks into functional not based exactly on the wording like Google, so when it's found the right function, the search is really on target."

"The 'code web' of search results was very helpful for finding out which things to analyze."

Suggestions from Programmers

"The best addition to Portfolio would be the ability to navigate though functions much like an IDE can."

"I would like to see more feedback from the search and more options on how to search."

"If query is misspelled, [Portfolio] does not return suggestions."

Ongoing Improvements

Data Available

- All Source Code in Our Repository
- Function Dependency Extractor: FUNDEX
- Case Study Tasks and Responses

Programmer Access

- SOAP web service
- Java search now available

See **http://www.searchportfolio.net/**

Visit us online and at Facebook!

http://www.searchportfolio.net/