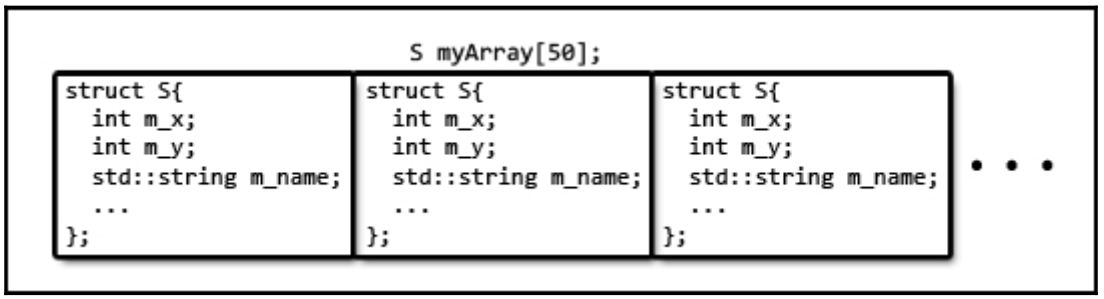
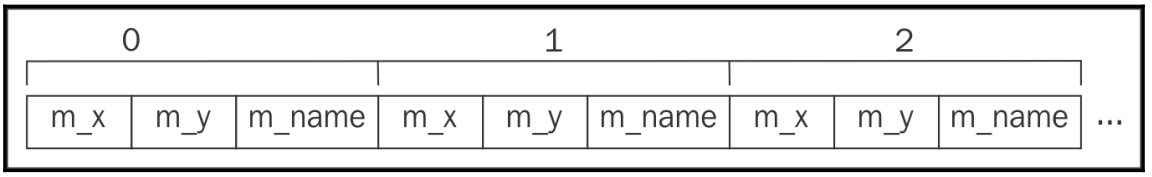
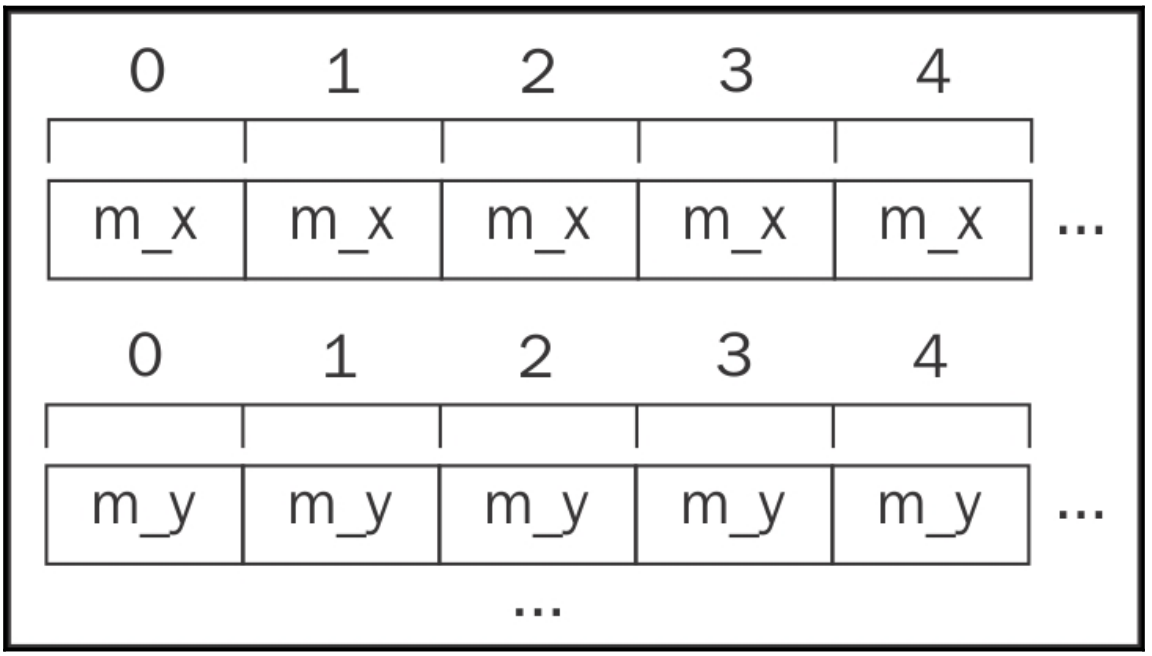
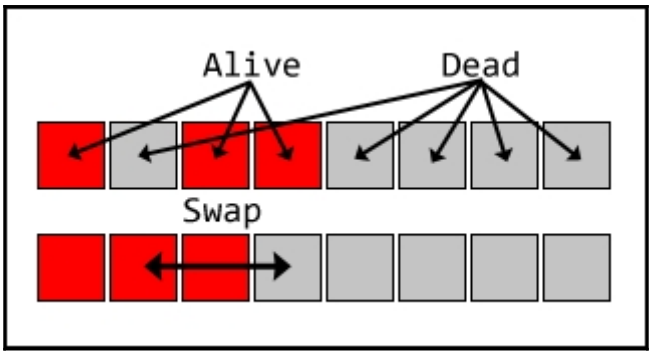
Chapter 2: Its Game Time! – Designing the Project

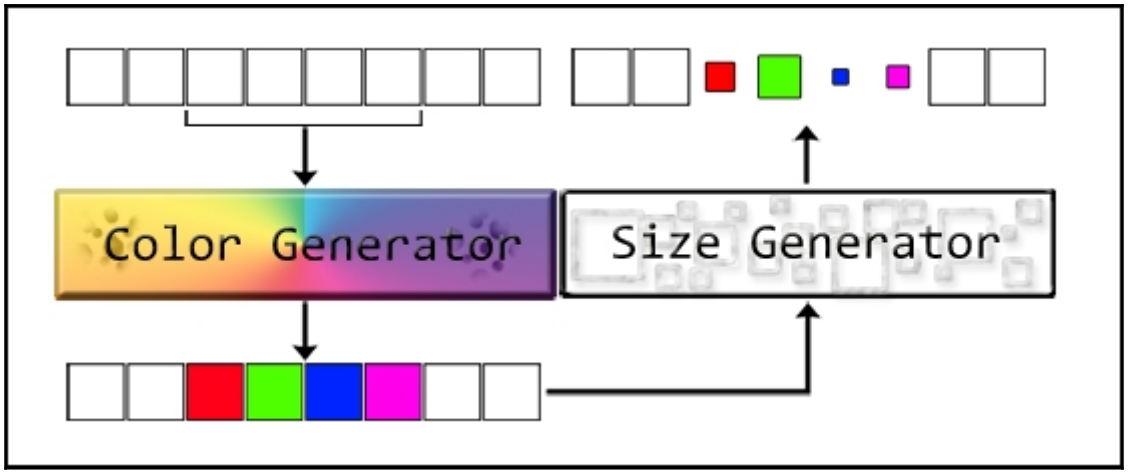
Chapter 3: Make It Rain! – Building a Particle System







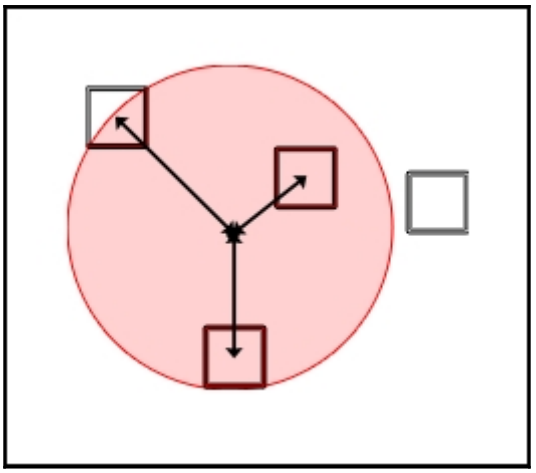




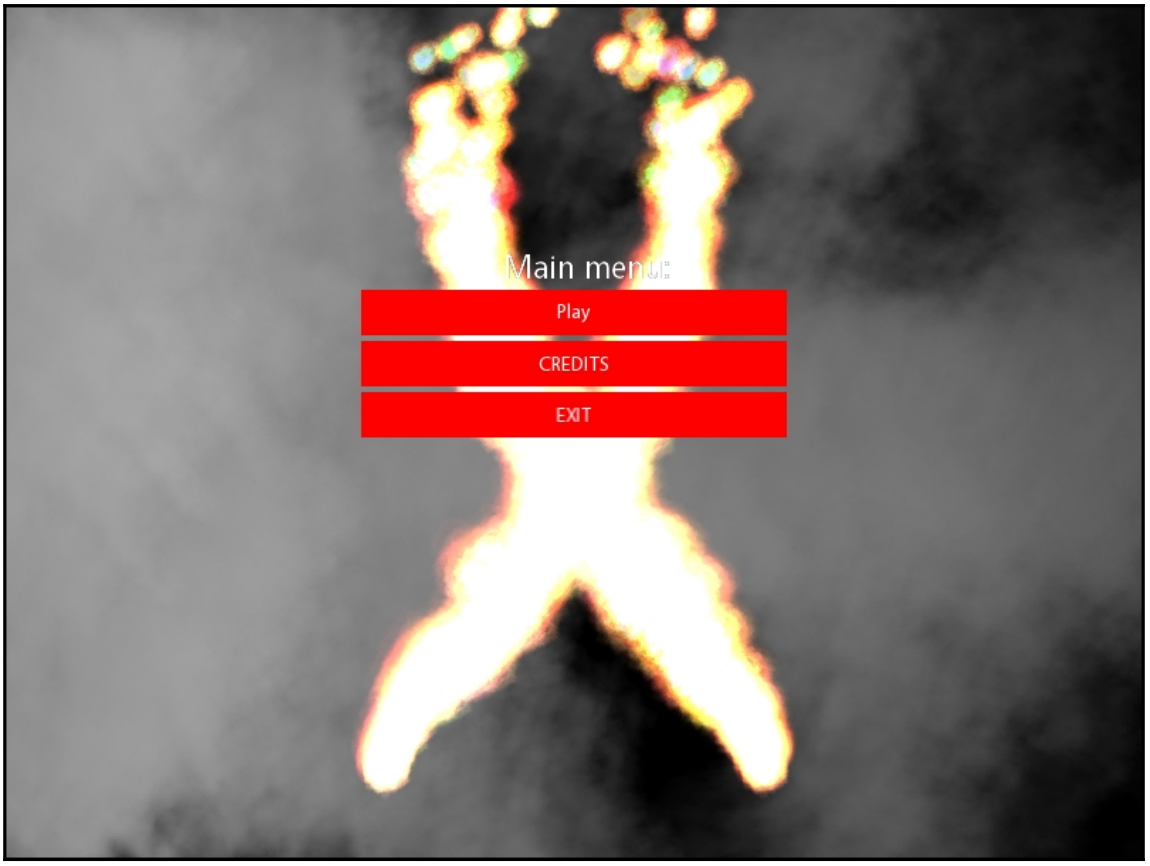

```
sf::Vector3f m_position[50];
sf::Vector3f m_velocity[50];
sf::RectangleShape m_RS[50];
```

Spatial Updater: `position += velocity;`

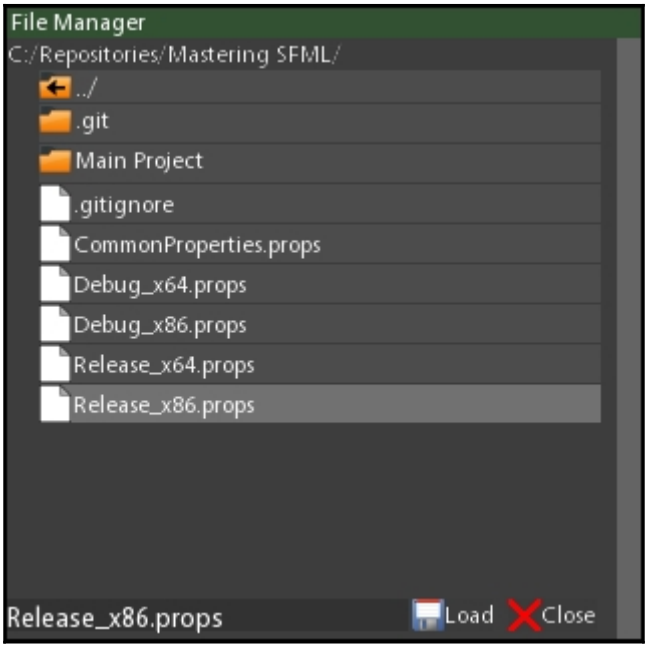
Drawable Updater: `drawable.setPosition(position);`

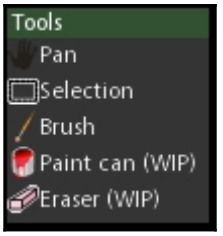



Additive
Alpha


Main menu
Play
CREDITS
EXIT

Chapter 4: Have Thy Gear Ready – Building Game Tools

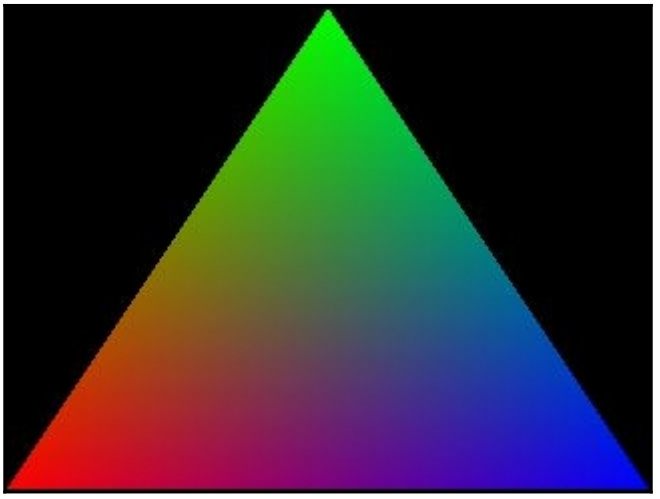




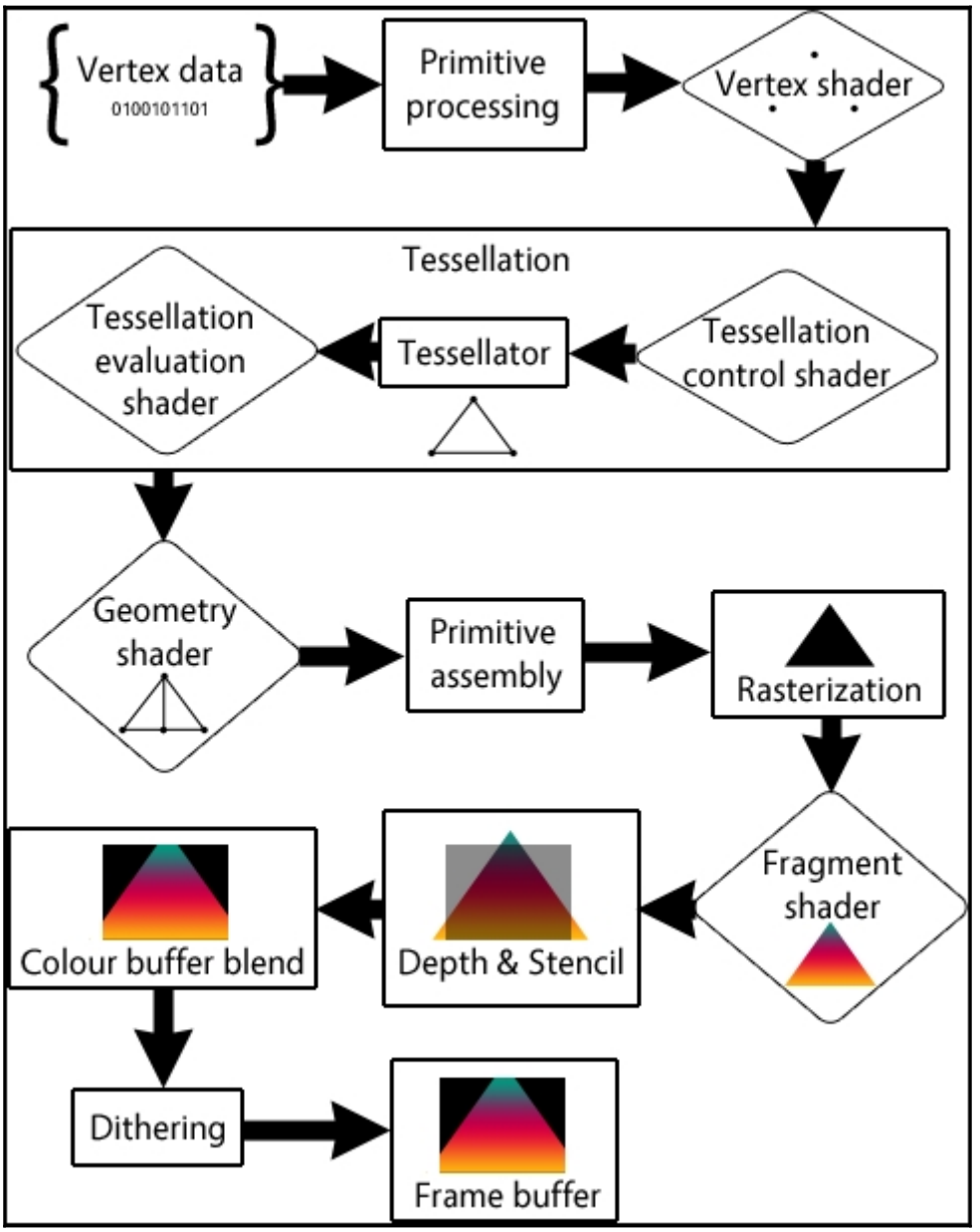
Chapter 5: Filling the Tool Belt – a few More Gadgets


Selection options
Mode: Tiles
Switch
Tile(s): 0
Select
Elevation:
0
Solid: False
Toggle
Copy
Remove
Show only selected layers
Tile selection
Close

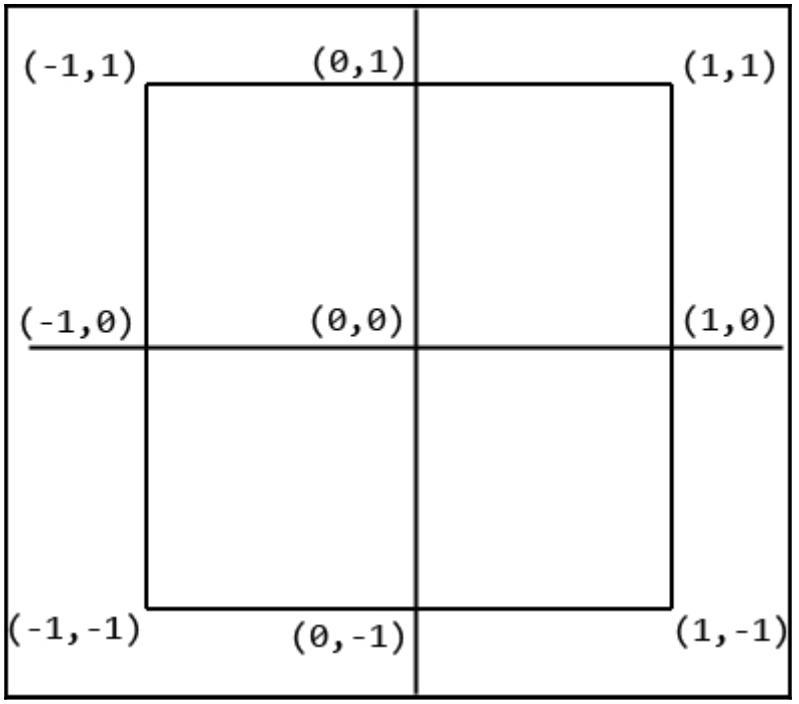
Chapter 6: Adding Some Finishing Touches – Using Shaders

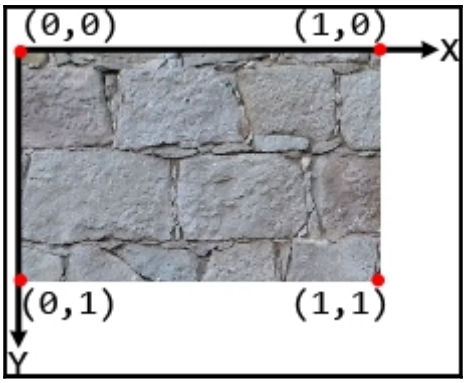


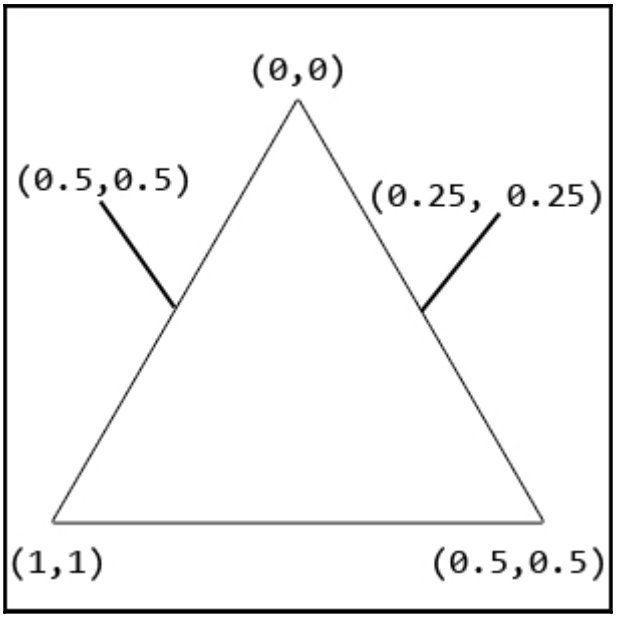
Chapter 7: One Step Forward, One Level Down – OpenGL Basics



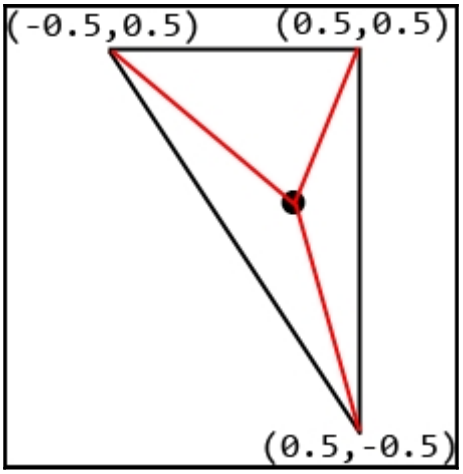
Position Color	Position Color	Position Color

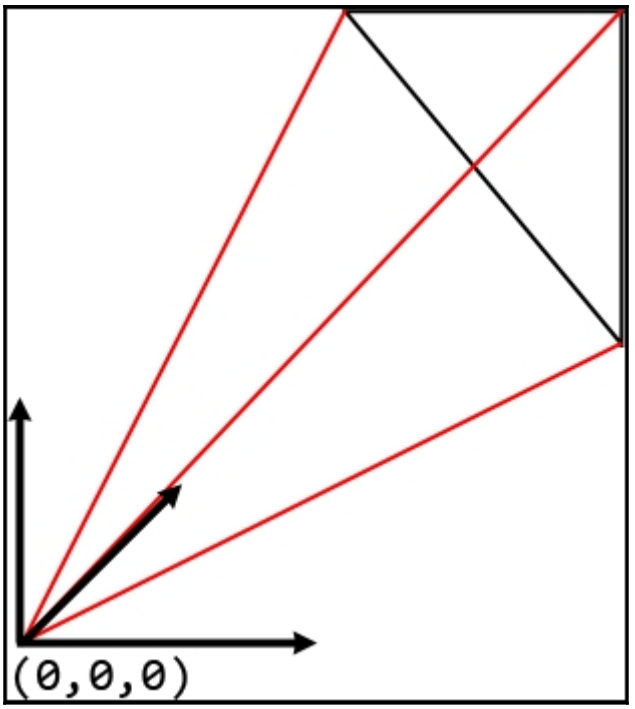



(0,0)
(1,0)
X
(0,1)
(1,1)
Y


(0,0)
(0.5,0.5)
(0.25, 0.25)
(1,1)
(0.5,0.5)

$$\begin{Bmatrix} 1, 0, 0, 0 \\ 0, 1, 0, 0 \\ 0, 0, 1, 0 \\ 0, 0, 0, 1 \end{Bmatrix}$$




(0,0,0)

Translation

$$\begin{Bmatrix} 1 & 0 & 0 & X \\ 0 & 1 & 0 & Y \\ 0 & 0 & 1 & Z \\ 0 & 0 & 0 & 1 \end{Bmatrix} * \begin{Bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\vartheta) & -\sin(\vartheta) & 0 \\ 0 & \sin(\vartheta) & \cos(\vartheta) & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix}$$

Rotation X

$$*$$

Rotation Y

$$\begin{Bmatrix} \cos(\vartheta) & 0 & \sin(\vartheta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\vartheta) & 0 & \cos(\vartheta) & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix}$$

$$*$$

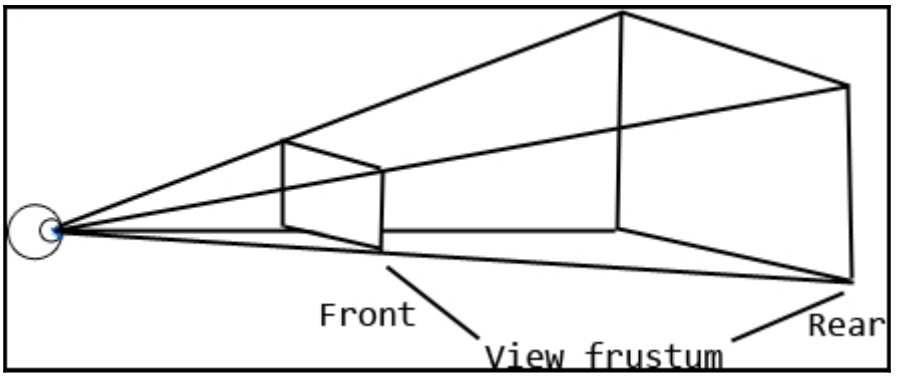
Rotation Z

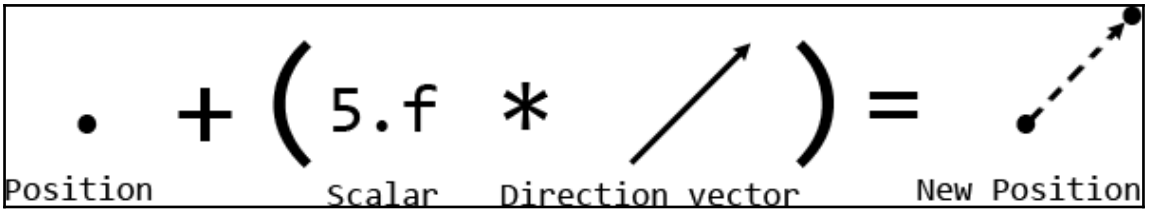
$$\begin{Bmatrix} \cos(\vartheta) & -\sin(\vartheta) & 0 & 0 \\ \sin(\vartheta) & \cos(\vartheta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix}$$

$$*$$

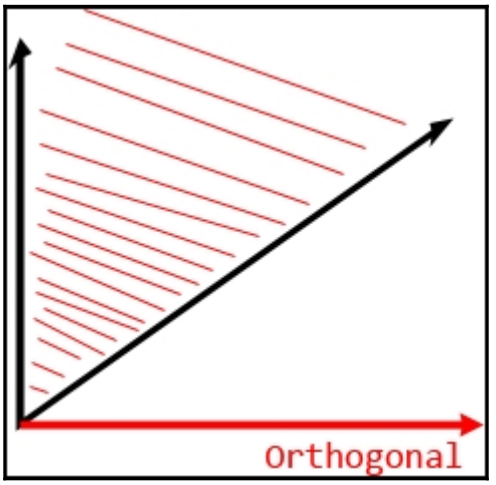
Scale

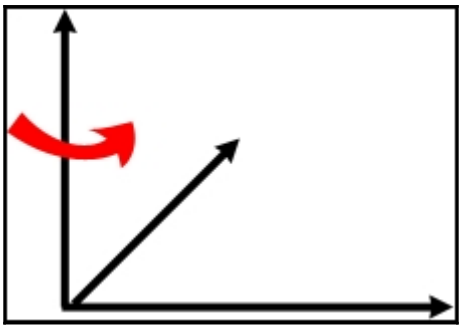
$$\begin{Bmatrix} sX & 0 & 0 & 0 \\ 0 & sY & 0 & 0 \\ 0 & 0 & sZ & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix}$$

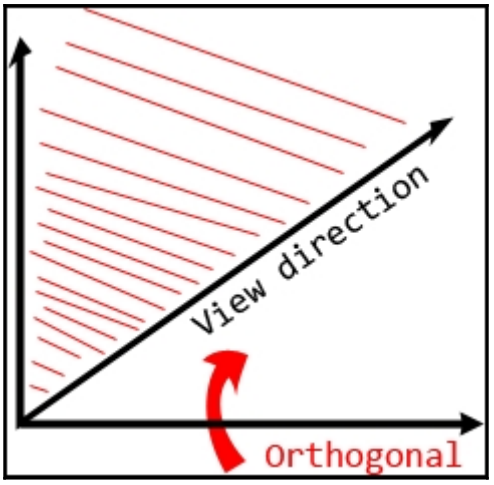


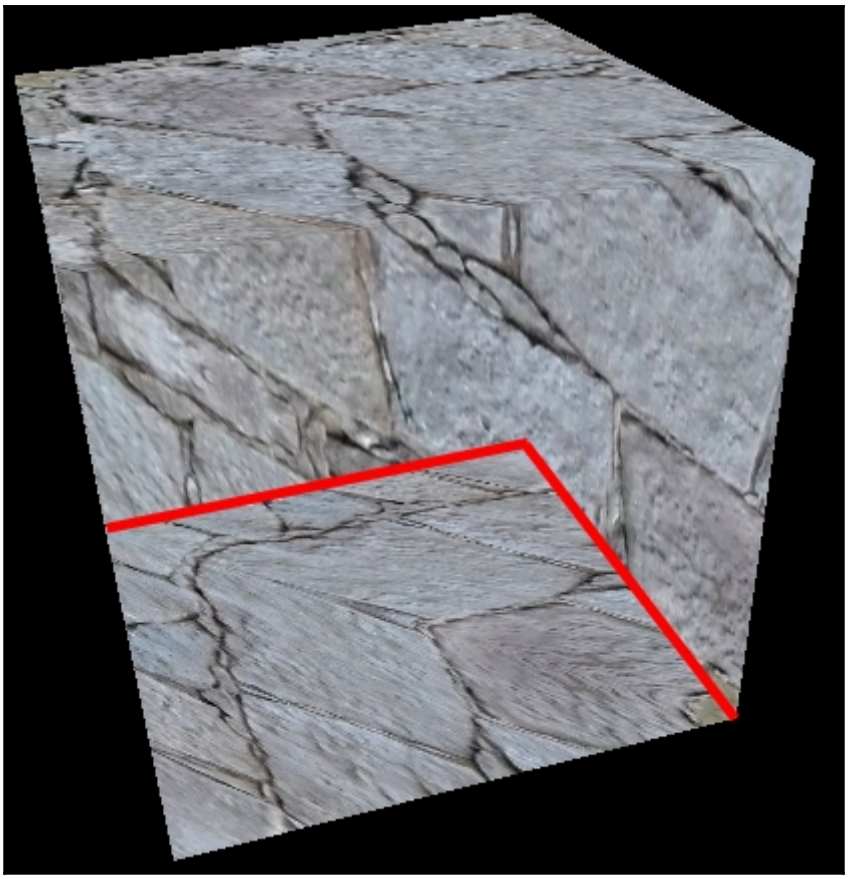


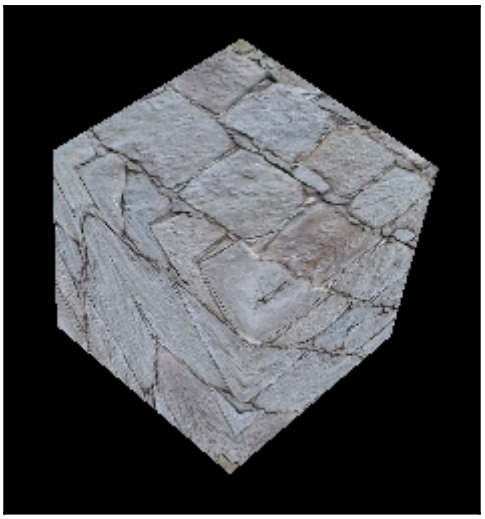
$$\vec{a} \times \vec{b} = \begin{bmatrix} a.y*b.z - a.z*b.y \\ a.z*b.x - a.x*b.z \\ a.x*b.y - a.y*b.x \end{bmatrix}$$

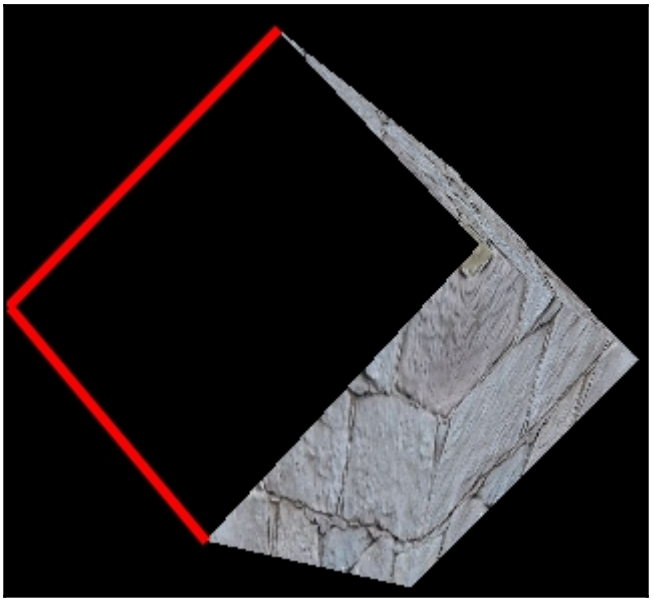


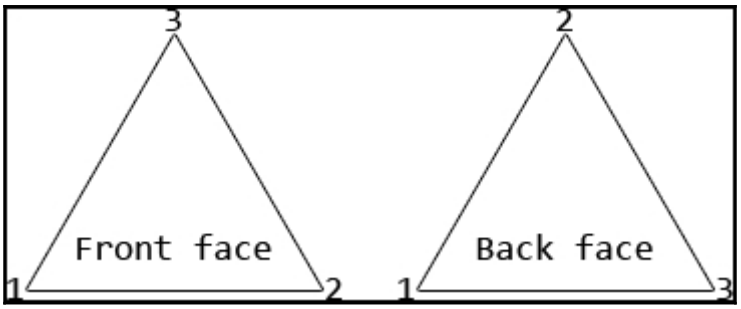



View direction
Orthogonal

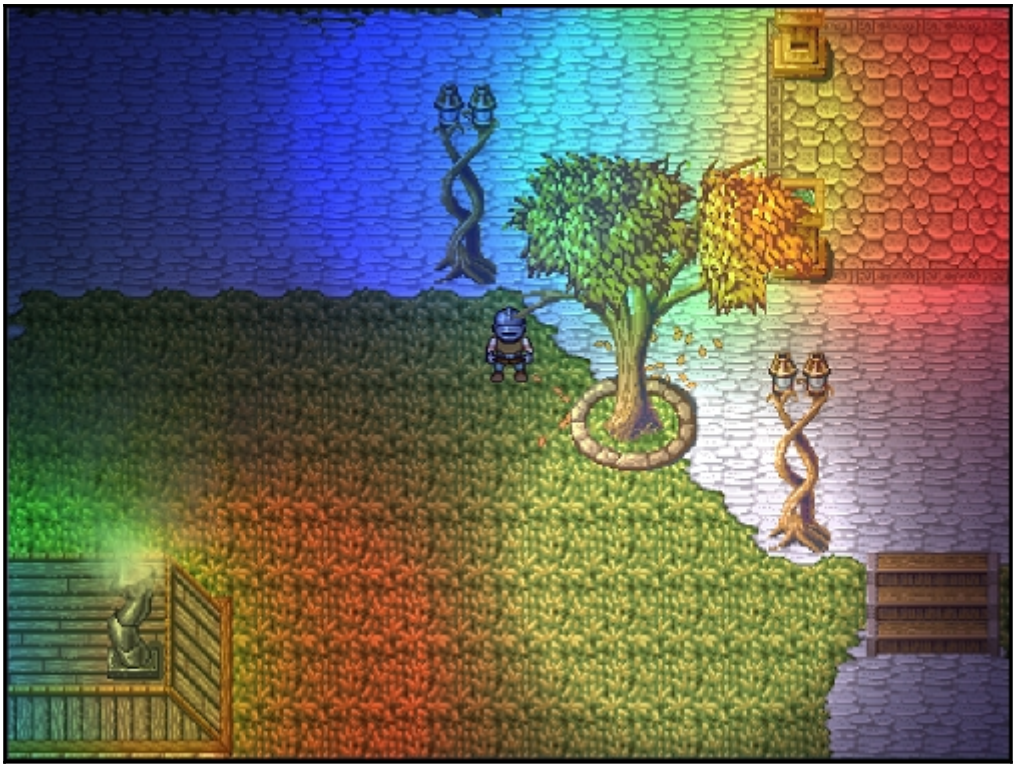


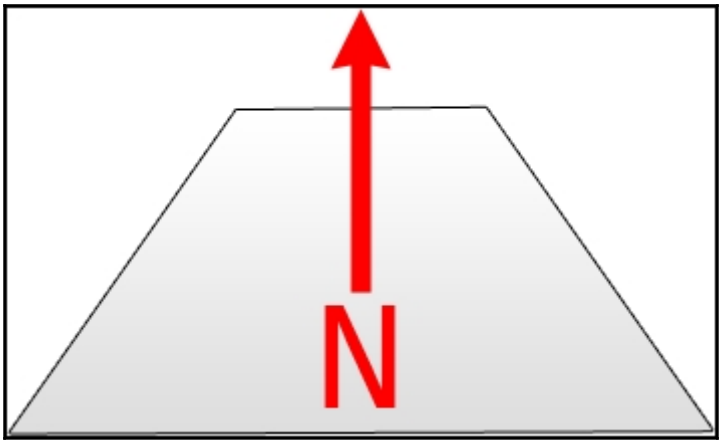


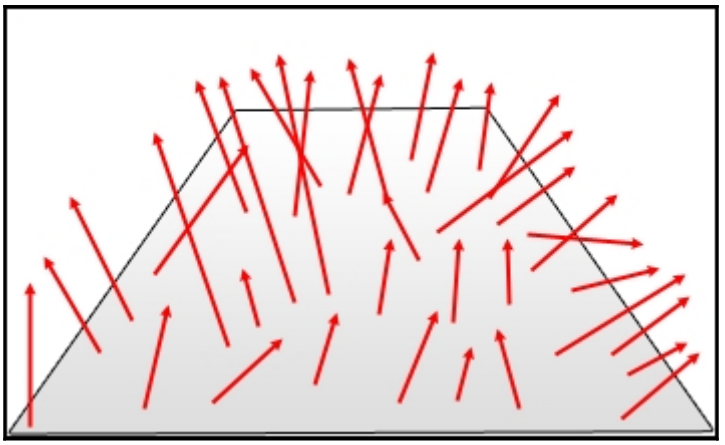


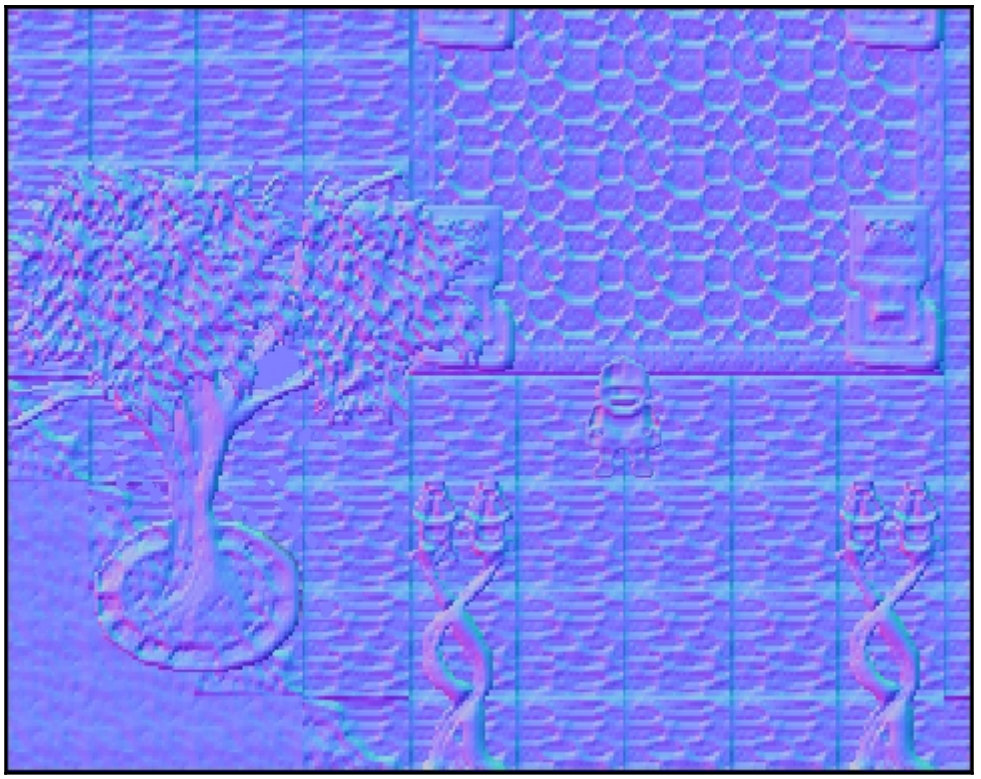

3
Front face
1
2
2
Back face
1
3

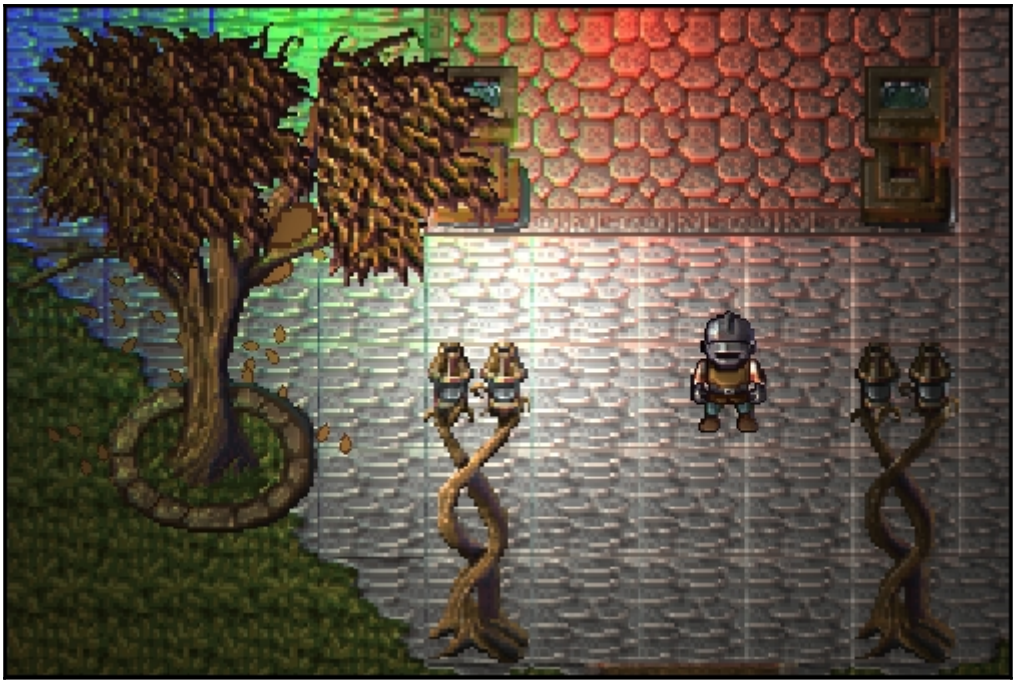
Chapter 8: Let There Be Light – An Introduction to Advanced Lighting

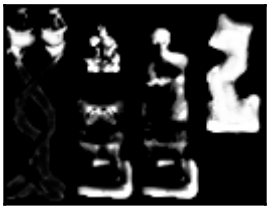


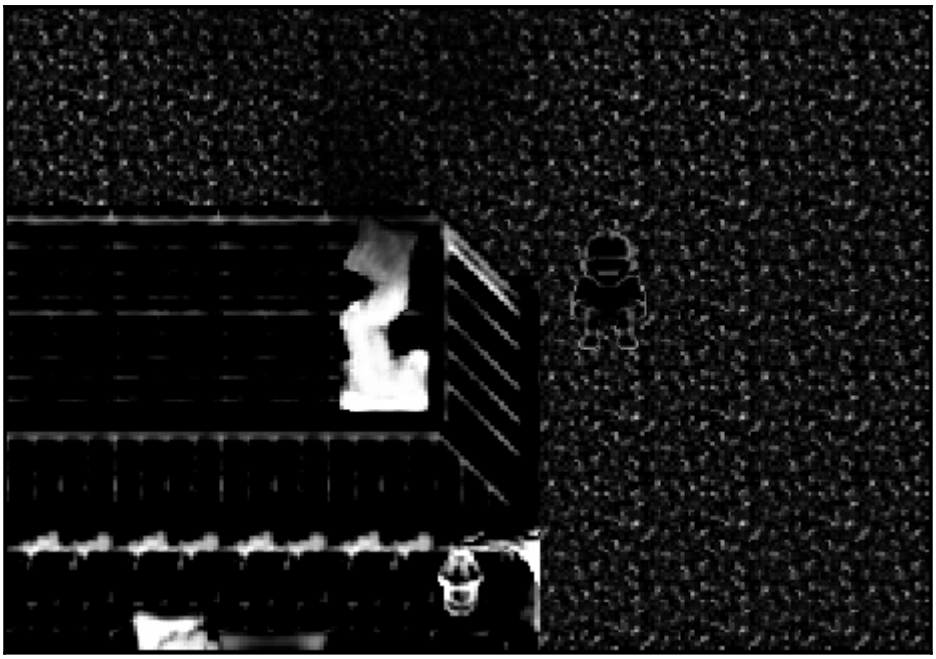

N

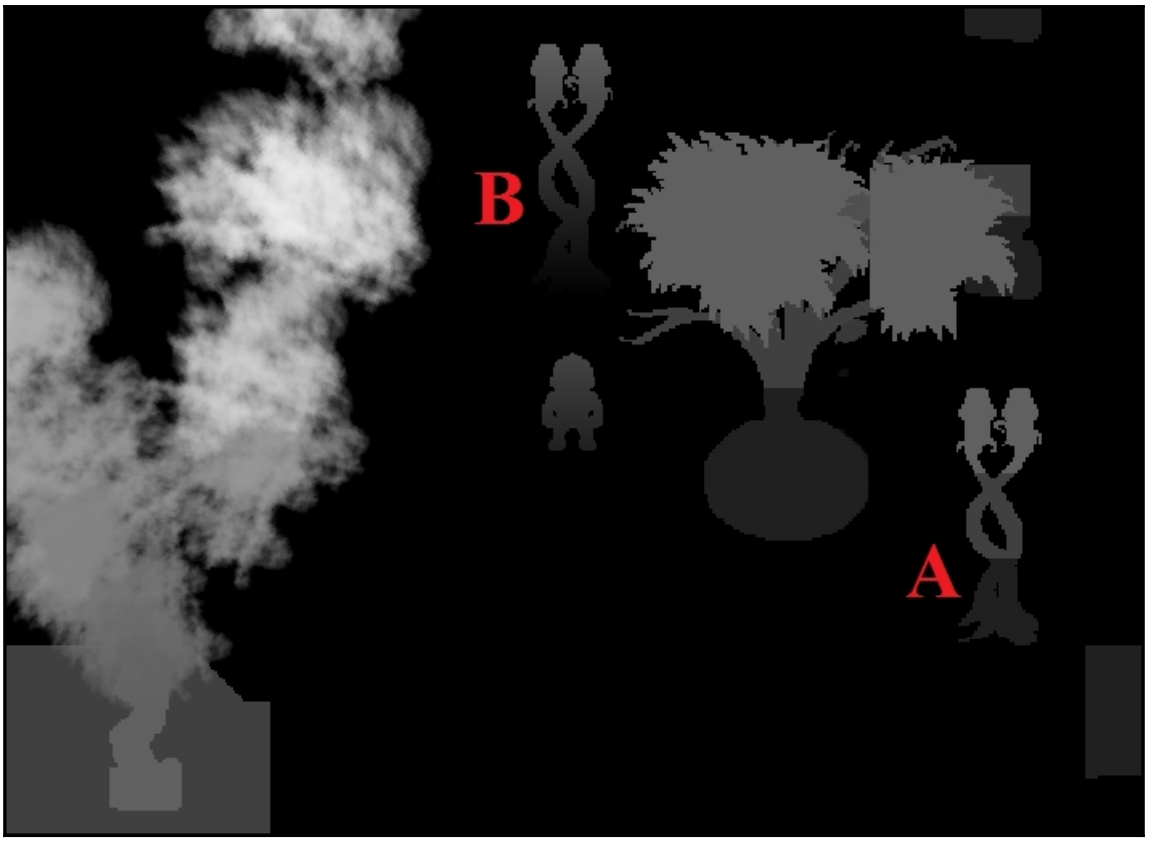


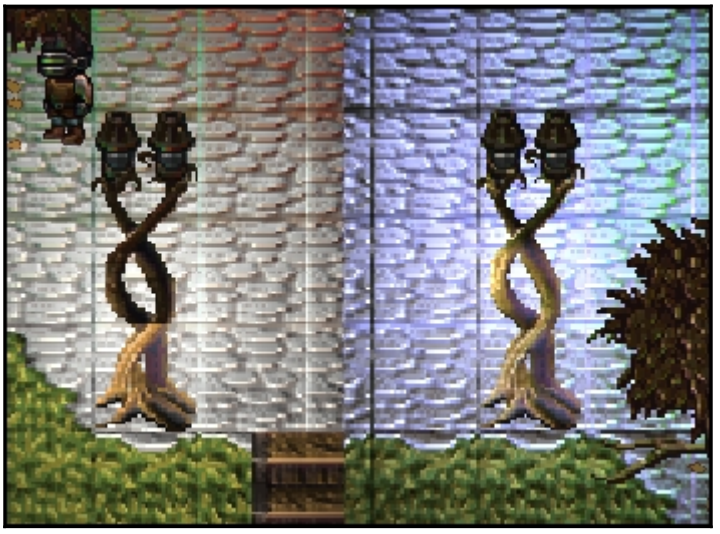




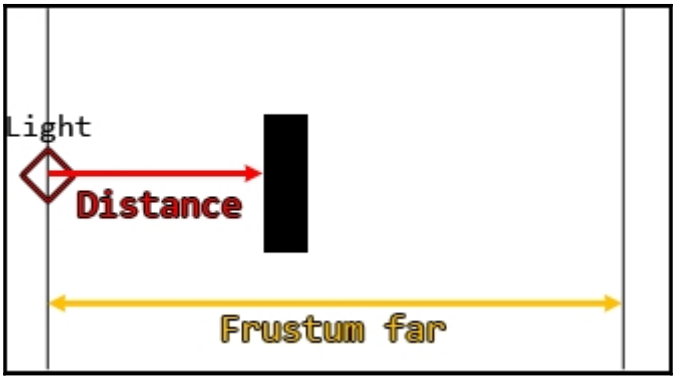


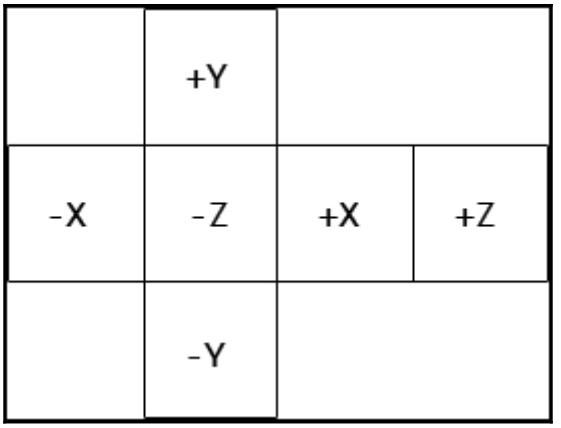


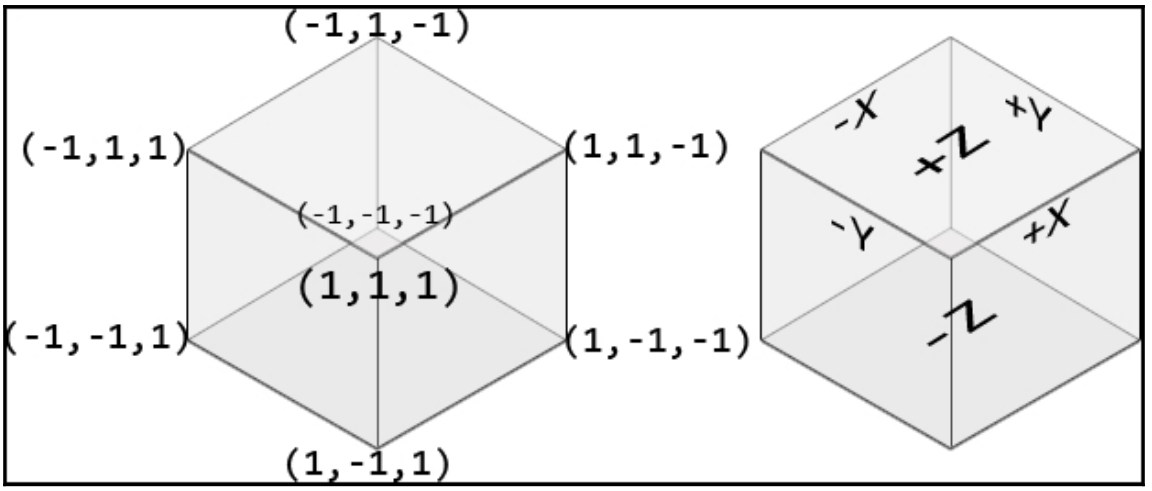

B
A

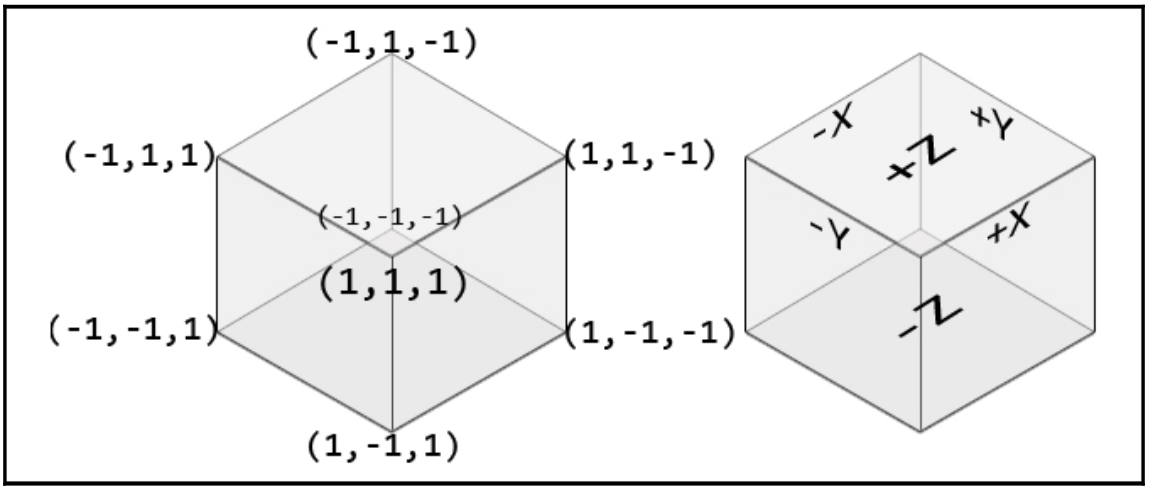


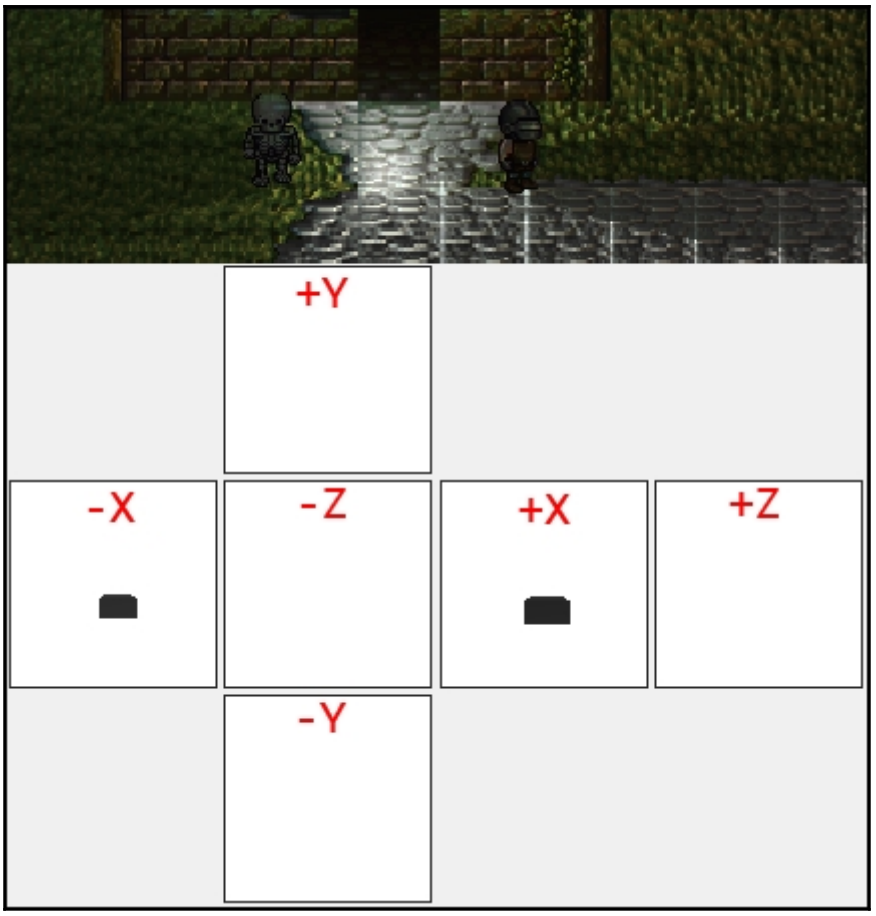
Chapter 9: The Speed of Dark – Lighting and Shadows

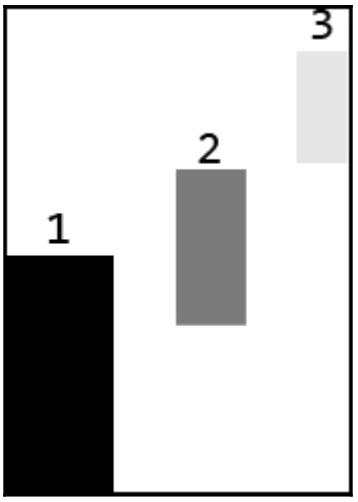


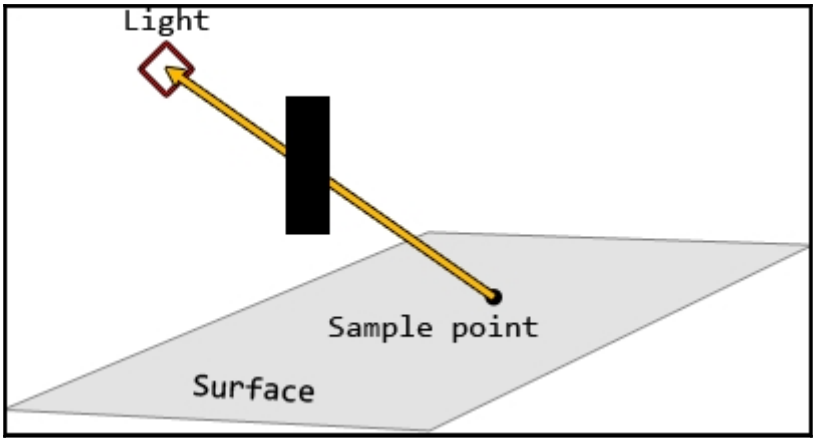


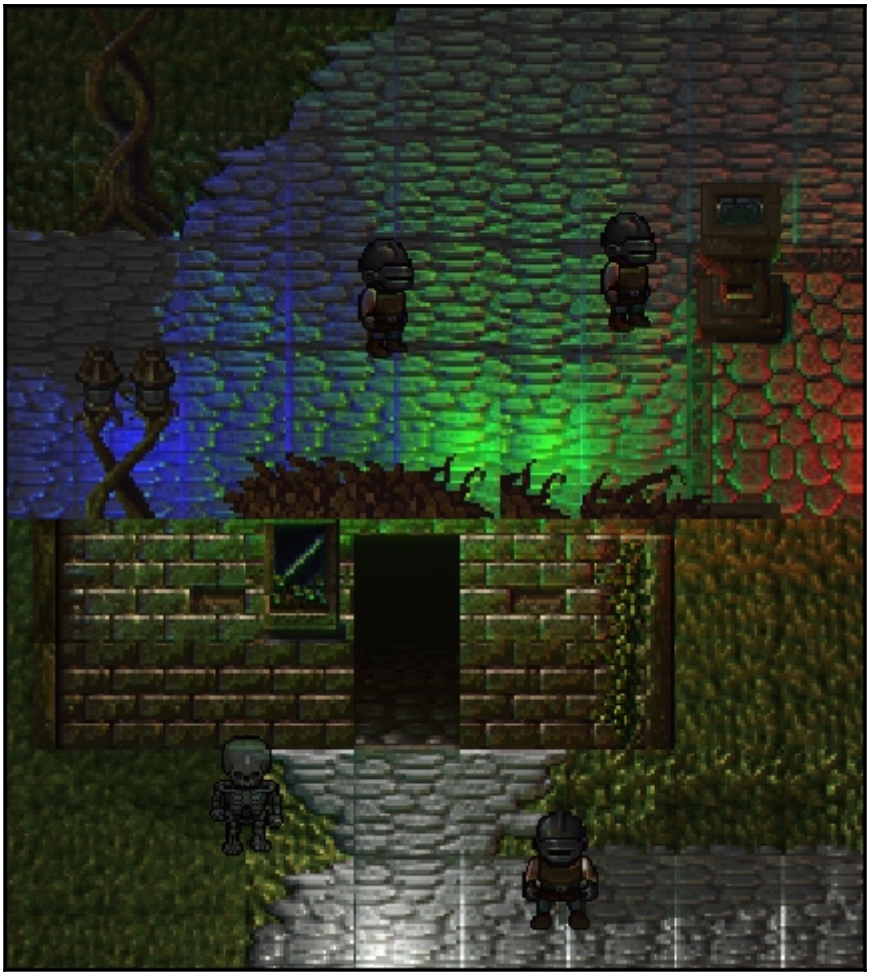

(-1,1,-1)
(-1,1,1)
(1,1,-1)
(-1,-1,-1)
(1,1,1)
(-1,-1,1)
(1,-1,-1)
(1,-1,1)
-X
+Y
+Z
-Y
+X
-Z


(-1,1,-1)
(-1,1,1)
(1,1,-1)
(-1,-1,-1)
(1,1,1)
(-1,-1,1)
(1,-1,-1)
(1,-1,1)
-X
+Y
+Z
-Y
+X
-Z

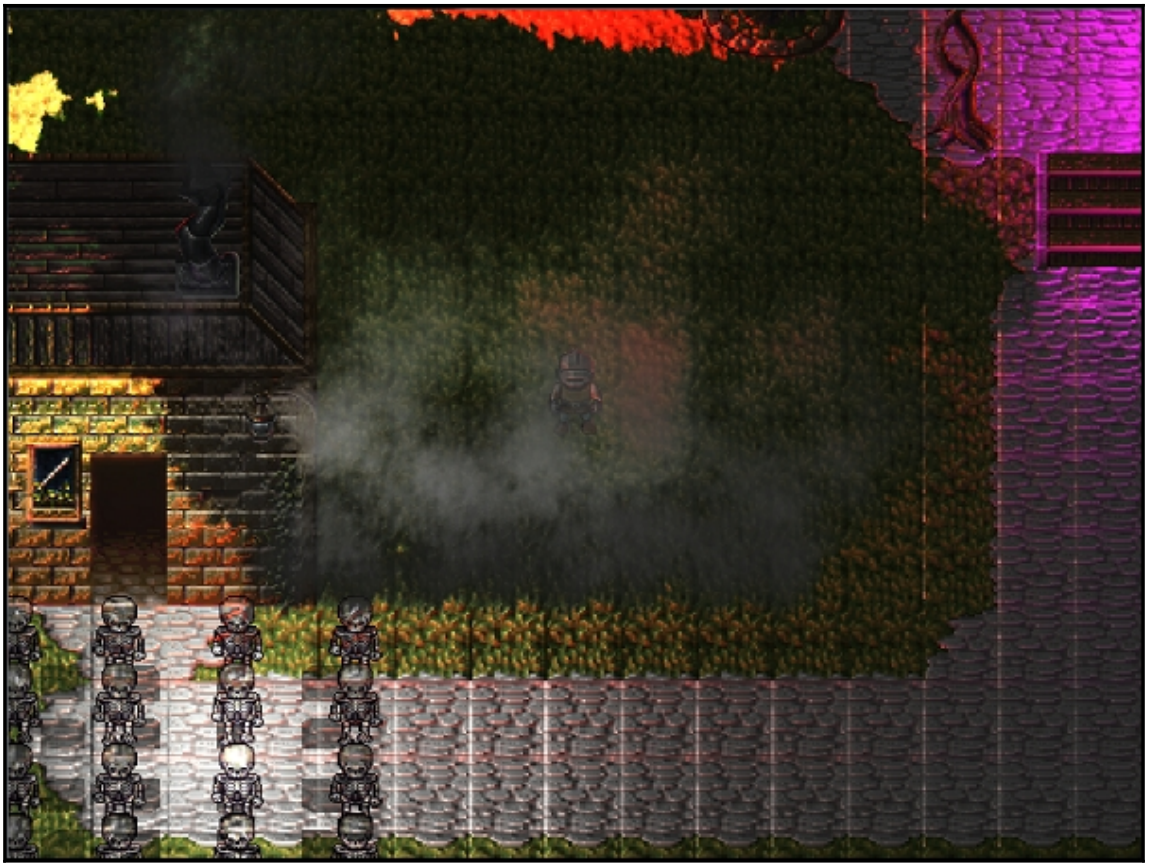

+Y
-X
-Z
+X
+Z
-Y

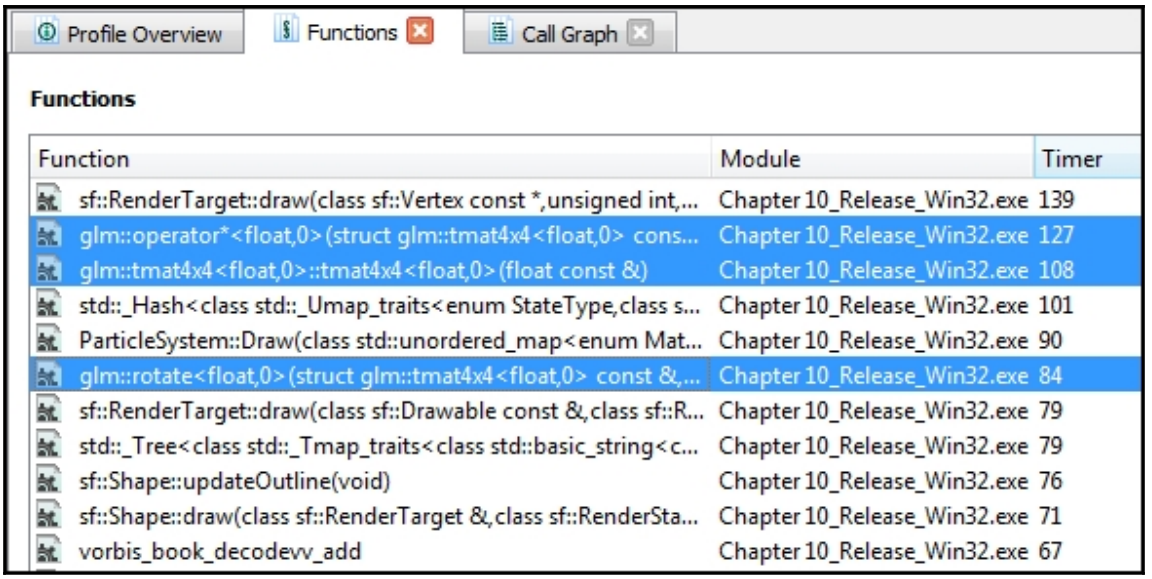

3
2
1

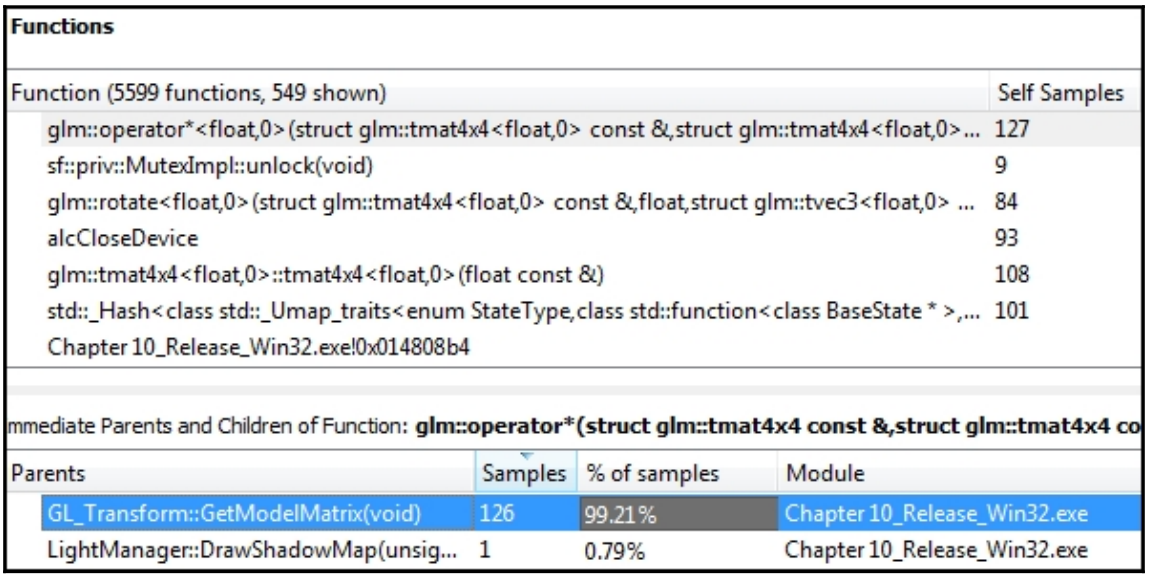

Light
Sample point
Surface



Chapter 10: A Chapter You Shouldn't Skip – Final Optimizations




Profile Overview
Functions
Call Graph
Functions
Function
Module
Timer
sf::RenderTarget::draw(class sf::Vertex const *,unsigned int,...
Chapter 10_Release_Win32.exe
139
glm::operator*<float,0>(struct glm::tmat4x4<float,0> cons...
Chapter 10_Release_Win32.exe
127
glm::tmat4x4<float,0>::tmat4x4<float,0>(float const &)
Chapter 10_Release_Win32.exe
108
std::_Hash<class std::_Umap_traits<enum StateType,class s...
Chapter 10_Release_Win32.exe
101
ParticleSystem::Draw(class std::unordered_map<enum Mat...
Chapter 10_Release_Win32.exe
90
glm::rotate<float,0>(struct glm::tmat4x4<float,0> const &,...
Chapter 10_Release_Win32.exe
84
sf::RenderTarget::draw(class sf::Drawable const &,class sf::R...
Chapter 10_Release_Win32.exe
79
std::_Tree<class std::_Tmap_traits<class std::basic_string<c...
Chapter 10_Release_Win32.exe
79
sf::Shape::updateOutline(void)
Chapter 10_Release_Win32.exe
76
sf::Shape::draw(class sf::RenderTarget &,class sf::RenderSta...
Chapter 10_Release_Win32.exe
71
vorbis_book_decodevv_add
Chapter 10_Release_Win32.exe
67


Functions
Function (5599 functions, 549 shown)
Self Samples
glm::operator*<float,0>(struct glm::tmat4x4<float,0> const &,struct glm::tmat4x4<float,0>...
127
sf::priv::MutexImpl::unlock(void)
9
glm::rotate<float,0>(struct glm::tmat4x4<float,0> const &,float,struct glm::tvec3<float,0> ...
84
alcCloseDevice
93
glm::tmat4x4<float,0>::tmat4x4<float,0>(float const &)
108
std::_Hash<class std::_Umap_traits<enum StateType,class std::function<class BaseState * >,...
101
Chapter 10_Release_Win32.exe!0x014808b4
mmediate Parents and Children of Function: glm::operator*(struct glm::tmat4x4 const &,struct glm::tmat4x4 co
Parents
Samples
% of samples
Module
GL_Transform::GetModelMatrix(void)
126
99.21%
Chapter 10_Release_Win32.exe
LightManager::DrawShadowMap(unsig...
1
0.79%
Chapter 10_Release_Win32.exe

Line	Address	Source Code	Hotspot Samples	% of Hotspot Samples	Timer
14	0x13aec20	glm::mat4 GL_Transform::GetModelMatrix() {			
15	0x13aec38	glm::mat4 matrix_pos = glm::translate(m_position);	2	16.67%	2
16	0x13aec6b	glm::mat4 matrix_scale = glm::scale(m_scale);	3	25.00%	3
17		// Represent each stored rotation as a different matrix, because we store angles.			
18		// Directional vector x, y, z			
19	0x13aec99	glm::mat4 matrix_rotX = glm::rotate(m_rotation.x, glm::vec3(1, 0, 0));	2	16.67%	2
20	0x13aecf0	glm::mat4 matrix_rotY = glm::rotate(m_rotation.y, glm::vec3(0, 1, 0));	2	16.67%	2
21	0x13aed47	glm::mat4 matrix_rotZ = glm::rotate(m_rotation.z, glm::vec3(0, 0, 1));	2	16.67%	2
22		// Create a rotation matrix. Multiply in reverse order it needs to be applied.			
23	0x13aeda1	glm::mat4 matrix_rotation = matrix_rotZ * matrix_rotY * matrix_rotX;			
24		// Apply transforms in reverse order they need to be applied in.			
25	0x13aedce	return matrix_pos * matrix_rotation * matrix_scale;	1	8.33%	1
26	0x13aedf6	}			

OpenGL API Call Summary

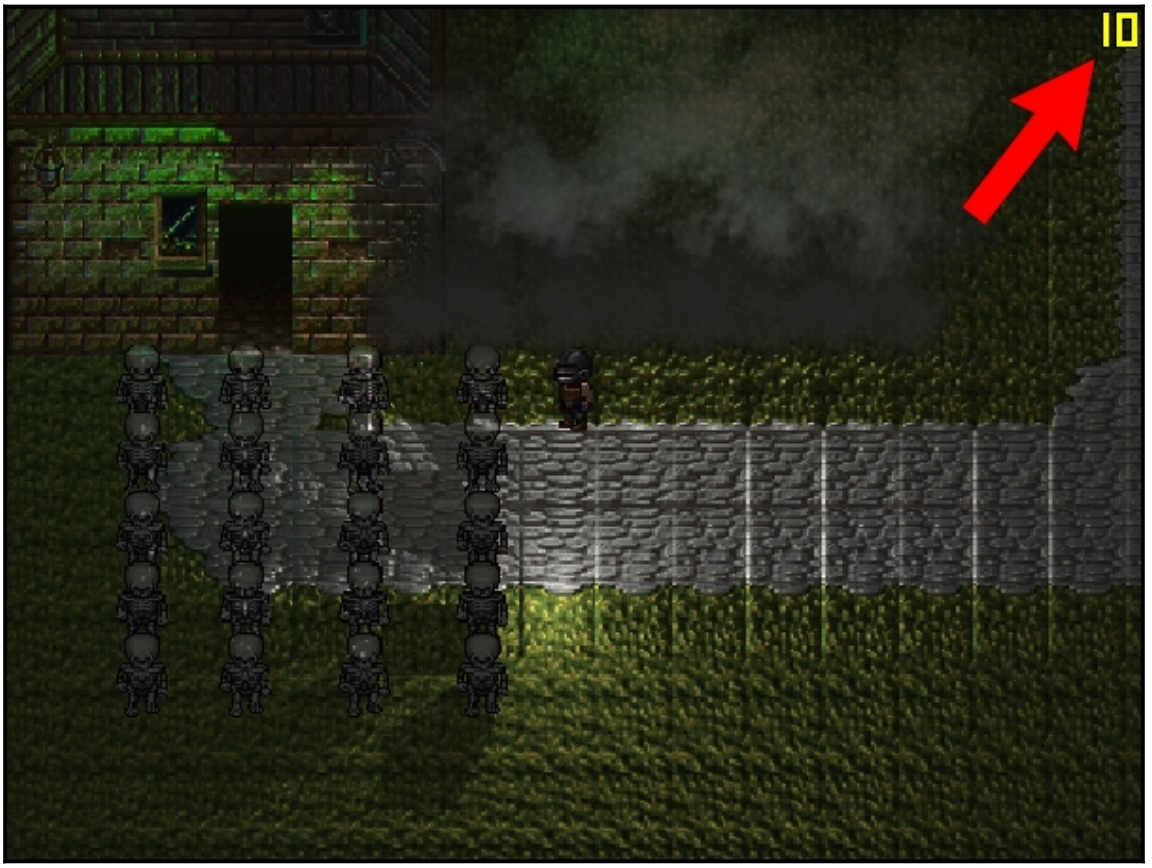
Filter

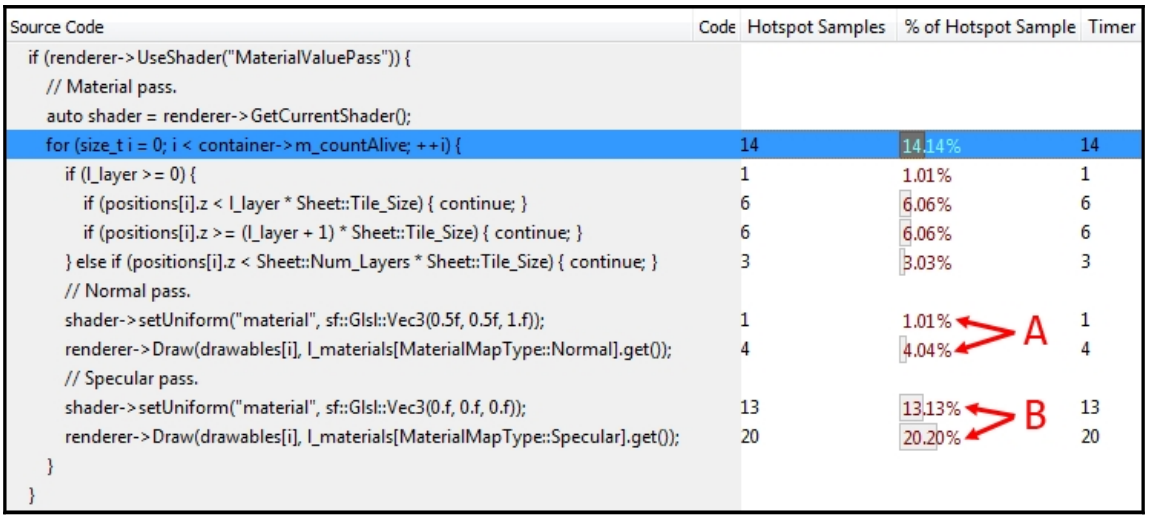
Drag a column header and drop it here to group by that column

	Name	Count	Capture Time %	Total Time (µs)	Min (µs)	Avg (µs)	Max (µs)
1	glBindBuffer	213126	0.12	82,579.257	0.256	0.387	661.154
2	glUseProgramObjectARB	178639	0.13	87,052.339	0.253	0.487	536.910
3	glBindVertexArray	106563	0.07	49,376.405	0.257	0.463	872.059
4	glFlush	76501	0.11	75,080.383	0.615	0.981	81.768
5	glActiveTextureARB	65913	0.04	24,828.946	0.256	0.376	37.048
6	glDrawArrays	58759	37.92	25,378,947.979	395.801	431.915	1,962.933
7	glUniformMatrix4fv	58233	0.04	28,550.424	0.277	0.490	5.756
8	glDrawElements	53279	33.98	22,745,252.819	394.393	426.908	1,439.381
9	glLoadMatrixf	40421	0.03	18,190.259	0.265	0.450	399.597
10	glVertexPointer	37413	0.02	14,960.612	0.283	0.399	20.320

Functions

Function	Module	Timer
ForceUpdater::Update(float,class ParticleContainer *)	Chapter 10_Release_Win32.exe	2
GL_Transform::GetModelMatrix(void)	Chapter 10_Release_Win32.exe	3
glBindTexture	Chapter 10_Release_Win32.exe	1
glColorPointer	Chapter 10_Release_Win32.exe	8
glLoadMatrixf	Chapter 10_Release_Win32.exe	2
glm::tmat4x4<float,0>::tmat4x4<float,0>(float const &)	Chapter 10_Release_Win32.exe	2
glVertexPointer	Chapter 10_Release_Win32.exe	4





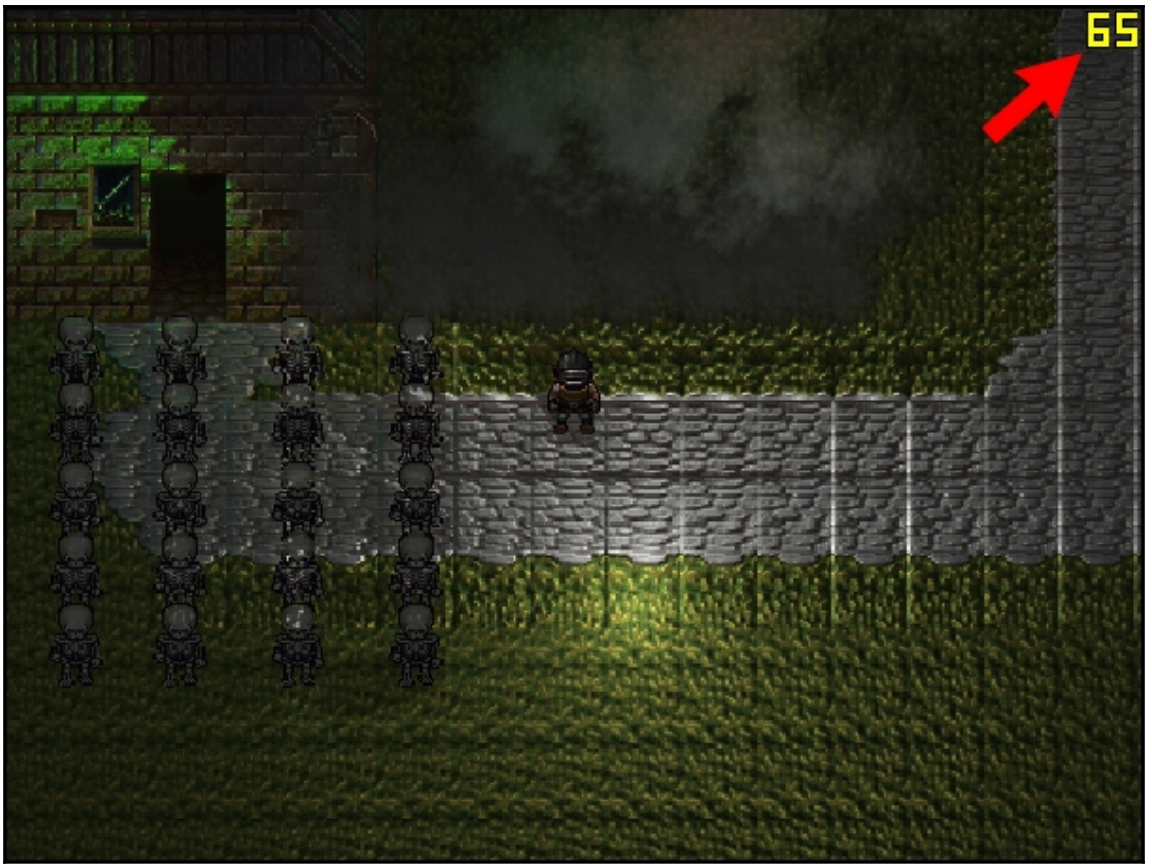
Source Code	Code	Hotspot Samples	% of Hotspot Sample	Timer
if (renderer->UseShader("MaterialValuePass")) {				
// Material pass.				
auto shader = renderer->GetCurrentShader();				
for (size_t i = 0; i < container->m_countAlive; ++i) {		14	14.14%	14
if (l_layer >= 0) {		1	1.01%	1
if (positions[i].z < l_layer * Sheet::Tile_Size) { continue; }		6	6.06%	6
if (positions[i].z >= (l_layer + 1) * Sheet::Tile_Size) { continue; }		6	6.06%	6
} else if (positions[i].z < Sheet::Num_Layers * Sheet::Tile_Size) { continue; }		3	3.03%	3
// Normal pass.				
shader->setUniform("material", sf::Glsl::Vec3(0.5f, 0.5f, 1.f));		1	1.01%	1
renderer->Draw(drawables[i], l_materials[MaterialMapType::Normal].get());		4	4.04%	4
// Specular pass.				
shader->setUniform("material", sf::Glsl::Vec3(0.f, 0.f, 0.f));		13	13.13%	13
renderer->Draw(drawables[i], l_materials[MaterialMapType::Specular].get());		20	20.20%	20
}				
}				

A

B

Functions

Function	Module	Timer
ParticleSystem::Draw(class std::unordered_map<enum Mat...	Chapter 10_Release_Win32.exe	99
• • •		
sf::RenderTexture::activate(bool)	Chapter 10_Release_Win32.exe	47
sf::ThreadLocal::setValue(void *)	Chapter 10_Release_Win32.exe	47
sf::Lock::~Lock(void)	Chapter 10_Release_Win32.exe	40
Renderer::Draw(class sf::Shape const &,class sf::RenderTarg...	Chapter 10_Release_Win32.exe	36
sf::Context::setActive(bool)	Chapter 10_Release_Win32.exe	32
sf::GlResource::TransientContextLock::~TransientContextLo...	Chapter 10_Release_Win32.exe	32
sf::Shader::bind(class sf::Shader const *)	Chapter 10_Release_Win32.exe	32
sf::Transformable::getTransform(void)	Chapter 10_Release_Win32.exe	32
sf::priv::GlContext::setActive(bool)	Chapter 10_Release_Win32.exe	31



Functions

Function	Module	Timer
ParticleSystem::Draw(class std::unordered_map< enum Mat...	Chapter 10_Release_Win32.exe	331
sf::Shape::updateOutline(void)	Chapter 10_Release_Win32.exe	281
sf::RenderTarget::draw(class sf::Vertex const *,unsigned int,...	Chapter 10_Release_Win32.exe	255
std::_Hash< class std::_Umap_traits< enum StateType,class s...	Chapter 10_Release_Win32.exe	225
sf::Transform::combine(class sf::Transform const &)	Chapter 10_Release_Win32.exe	217