

Szauer, Gabor. "Table of Contents." Hands-On C Game Animation Programming: Learn modern animation techniques from theory to implementation with C and OpenGL. Birmingham: Packt, 2020. i–vi. Bloomsbury Visual Arts. Web. 19 Dec. 2025. &lt;https://www.bloomsburyvisualarts.com/encyclopedia-chapter?docid=&amp;tocid=b-9781800207967-0000086&gt;;

Accessed from: [www.bloomsburyvisualarts.com](http://www.bloomsburyvisualarts.com)

Accessed on: Sat Dec 06 2025 16:00:00 Pacific Standard Time

Copyright © Gabor Szauer.

# Table of Contents

## Preface

---

### 1

#### Creating a Game Window

---

Technical requirements	12	Creating a window	17
Creating an empty project	12	Global variables	20
Creating the application class	14	Opening a window	20
Adding an OpenGL loader	15	Creating the event handler	26
Getting glad	16	Exploring the samples	28
Adding glad to the project	16	Summary	30

### 2

#### Implementing Vectors

---

Introducing vectors	32	component-wise operations	38
Creating a vector	32	Vector length	38
Epsilon	33	Normalizing vectors	39
Understanding component-wise operations	34	The angle between vectors	40
Vector addition	34	Vector projection and rejection	42
Vector subtraction	35	Vector reflection	43
Scaling vectors	36	Cross product	44
Multiplying vectors	36	Interpolating vectors	46
Dot product	37	Comparing vectors	48
Understanding non-		Exploring more vectors	49
		Summary	51

## 3

### Implementing Matrices

---

Technical requirements	54	Inverting a matrix	65
What is a matrix?	54	Transpose	66
Matrix storage	55	Determinant and minors of lower-order matrices	66
Creating a matrix	56		
Common matrix operations	59	Creating camera matrices	71
Comparing matrices	59	Frustum	71
Adding matrices	60	Perspective	72
Scaling a matrix	60	Orthographic	72
Matrix multiplication	61	Look at	73
Transforming vectors and points	63	Summary	74

## 4

### Implementing Quaternions

---

Creating quaternions	76	Transforming vectors	88
Angle axis	77	Interpolating quaternions	89
Creating rotations from one vector to another	78	Neighborhood	89
Retrieving quaternion data	80	Understanding the mix function	90
Common quaternion operations	80	Understanding the nlerp function	90
Comparison operations	81	Introduction to slerp	91
Dot product	82	Power	91
Length and squared length	82	Implementing slerp	92
Unit quaternions	83	Look rotation	92
Conjugate and inverse	84	Converting between quaternions and matrices	94
Multiplying quaternions	85	Summary	95

## 5

### Implementing Transforms

---

Creating the transform	98	Inverting transforms	100
Combining transforms	99	Mixing transforms	100

Converting transforms to matrices	101	Transforming points and vectors	104
Converting matrices into transforms	102	Summary	105

## 6

### Building an Abstract Renderer

---

Technical requirements	108	The IndexBuffer class declaration	125
<b>Working with shaders</b>	<b>108</b>	Implementing the IndexBuffer class	126
The Shader class declaration	108	<b>Rendering geometry</b>	127
Implementing the Shader class	110	<b>Working with textures</b>	129
<b>Working with buffers (attributes)</b>	<b>117</b>	Adding stb_image	130
The Attribute class declaration	118	The Texture class declaration	130
Implementing the Attribute class	119	Implementing the Texture class	131
<b>Working with uniforms</b>	<b>122</b>	<b>Simple shaders</b>	133
The Uniform class declaration	122	The vertex shader	133
Implementing the Uniform class	123	The fragment shader	134
<b>Working with index buffers</b>	<b>125</b>	<b>Summary</b>	135

## 7

### Exploring the glTF File Format

---

Technical requirements	138	Accessing data	141
<b>Exploring how glTF files are stored</b>	<b>138</b>	<b>Exploring cgltf</b>	143
glTF files store a scene, not a model	139	Integrating cgltf	144
<b>Exploring the glTF format</b>	<b>139</b>	<b>Exploring the sample assets</b>	146
The parts you need for animation	140	Exporting from Blender	146
The parts you don't need for animation	141	<b>Summary</b>	147

# 8

## Creating Curves, Frames, and Tracks

---

Understanding cubic Bézier splines	150	Declaring the Track class	161
		Implementing the Track class	163
Understanding cubic Hermite splines	156	Creating the TransformTrack class	174
Interpolation types	157	Declaring the TransformTrack class	174
Creating the Frame struct	159	Implementing the TransformTrack class	176
Creating the Track class	160	Summary	179

# 9

## Implementing Animation Clips

---

Implementing poses	182	glTF - loading the rest pose	194
Declaring the Pose class	183	glTF - loading joint names	196
Implementing the Pose class	184	glTF - loading animation clips	197
Implementing clips	188	Summary	201
Declaring the Clip class	188		
Implementing the Clip class	190		

# 10

## Mesh Skinning

---

Exploring meshes	204	glTF - loading the bind pose	216
Understanding skinning	205	glTF - loading a skeleton	218
Exploring rigid skinning	207	Implementing meshes	218
The rigid skinning pipeline	209	The Mesh class declaration	218
Exploring smooth skinning	210	The Mesh class implementation	221
Implementing skeletons	212	glTF - loading meshes	229
The Skeleton class declaration	212	Implementing GPU skinning	235
The Skeleton class implementation	214	Summary	237

# 11

## Optimizing the Animation Pipeline

---

Pre-generating the skin matrix	240	Creating FastClip	253
Generating the skin matrix	240	<b>The Pose palette generation</b>	254
CPU skinning	241	Changing the GetMatrixPalette function	255
GPU skinning	243	Reordering joints	256
<b>Storing the skin palette in a texture</b>	244	Reordering clips	259
<b>Faster sampling</b>	245	Reordering meshes	260
Optimizing the Track class	245	<b>Exploring Pose::GetGlobalTransform</b>	261
Converting tracks	250	<b>Summary</b>	261
Creating FastTransformTrack	251		

# 12

## Blending between Animations

---

<b>Pose blending</b>	264	Implementing the cross-fade controller	268
Declaring the Blend function	264	<b>Additive blending</b>	271
Implementing the Blend function	265	Declaring additive animations	272
<b>Crossfading animations</b>	266	Implementing additive animations	272
Creating helper classes	266	<b>Summary</b>	274
Declaring the cross-fade controller	267		

# 13

## Implementing Inverse Kinematics

---

<b>Creating a CCD solver</b>	276	<b>Implementing constraints</b>	290
Declaring the CCD solver	277	Ball-and-socket constraint	292
Implementing the CCD solver	278	Hinge constraint	293
<b>Creating a FABRIK solver</b>	282	<b>Using IK to align a character's feet to the ground</b>	294
Declaring the FABRIK solver	283	Finding the foot goals	295
Implementing the FABRIK solver	285		

Interpolating the foot goals	296	Foot alignment	298
Vertical character placement	297	<b>Summary</b>	<b>299</b>
IK pass	297	Further reading	299

## 14

### Using Dual Quaternions for Skinning

---

Introducing dual quaternions	302	<b>Skinning with dual quaternions</b>	<b>310</b>
<b>Implementing dual quaternions</b>	<b>303</b>	Modifying the pose class	311
Implementing dual quaternion operations	304	Modifying the skeleton class	312
Measuring, normalizing, and inverting dual quaternions	305	Creating new uniform types	313
Converting transforms and dual quaternions	307	Creating a dual quaternion shader	314
Transforming vectors and points	309	<b>Understanding how to use dual quaternion skinning</b>	<b>318</b>
		<b>Summary</b>	<b>320</b>

## 15

### Rendering Instanced Crowds

---

Storing data in textures	322	Implementing the Crowd class	339
Reading data from textures	323	Using the Crowd class	344
Encoding animation data	323	<b>Blending animations</b>	<b>345</b>
Exploring per-instance data	324	<b>Exploring texture formats</b>	<b>346</b>
<b>Creating an animation texture</b>	<b>324</b>	<b>Combining animation textures</b>	<b>346</b>
Declaring the AnimTexture class	325	<b>Optimizing texel fetches</b>	<b>347</b>
Implementing the AnimTexture class	326	Limiting influences	347
Animation baker	331	Limiting animated components	348
Creating a crowd shader	333	Not interpolating	348
Creating the Crowd utility class	337	Combining these optimizations	349
		<b>Summary</b>	<b>349</b>

---

### Other Books You May Enjoy

---

Leave a review - let other readers know what you think    353