ZARENIKIT

Voxel Framework

API Documentation

CHUNK CONTAINER

Name: ChunkContainer

Type: Public Sealed Class

Extends: MonoBehaviour

Description: This component is used to automatically generate chunks and allow chunks to reference each other's voxels easily. It is also useful for manipulating voxel grids in world coordinates.

Exposed Variables:

Access	Type	Name	Description
Public	ChunkObject	chunk	The Chunk prefab that will be used for auto-generation.
Public	bool	contiguous	If TRUE, voxel lookup is much faster, but all chunks must be generated side-by-side and never move.
Public	bool	generateOnStartup	Whether auto-generation is enabled.
Public	Vector3	size	The number of chunks that the container can hold in each direction.

Exposed Methods:

Access	Return Type	Name (Parameters)	Description
Public	int	GetVoxel(Vector3 worldpos)	Gets the voxel at the given world position.
Public	void	RemeshAround(Vector3 chunkPos)	Remeshes all chunks around a chunk at a given position.
Public	bool	SetVoxel(int voxel, Vector3 worldpos)	Sets the voxel ID at the given world position. Returns TRUE if successful and FALSE if not.

CHUNK OBJECT

Name: ChunkObject

Type: Public Class

Extends: MonoBehaviour

Description: This component serves as a base class for other chunks and provides the basic functionality needed to create and modify a voxel grid.

Exposed Variables:

Access	Type	Name	Description
Public	bool	immediatelyRemesh	If TRUE, the chunk will remesh exactly once every frame it is needed. If FALSE, the chunk will remesh in a coroutine.
Public	float	texturePadding	How much of the texture is assumed to be padding and cropped out in the mesh.

Exposed Methods:

Access	Return Type	Name (Parameters)	Description
Public	void	AbortReMesh()	Cancels any remesh operations that have been already scheduled to occur that frame.
Public	Vector3	ChunkToWorldCoords (Vector3 coords)	Returns the world position of a voxel's position in the grid.
Protected virtual	void	Feedback (ChunkStatus status)	Debugging method triggered when the chunk's status changes.
Public	void	Generate (IVoxelGenerator generator, bool remesh = true)	Populates the chunk with voxels from a given voxel generator.
Public	void	Generate (bool remesh = true)	Overload of above method, using the last used generator.
Public	Vector3	GetPhysicalSize()	Gets the size of the grid, accounting for the size of the voxels themselves.
Public	Vector3	GetSize()	Gets the size of the grid in voxels.
Public	int	GetSize()	Overload of the above method, returning the size of a specific dimension.
Public	int	GetVoxel(Vector3 localPos)	Gets the voxel ID of a voxel at the specified local position.

Public	bool	GetVoxelInstanceBool (int index, Vector3 localPos)	Returns the boolean voxel instance attribute of a voxel at the specified local position.
Public	float	GetVoxelInstanceFloat (int index, Vector3 localPos)	Returns the floating-point voxel instance attribute of a voxel at the specified local position.
Public	string	GetVoxelName (Vector3 localPos)	Returns the name of the voxel at the specified local position.
Public	bool	GetVoxelTypeBool (Vector3 localPos, string name)	Looks up the boolean voxel type attribute at the specified local position by name.
Public	bool	GetVoxelTypeBool (Vector3 localPos, int index)	Looks up the boolean voxel type attribute at the specified local position by index.
Public	float	GetVoxelTypeFloat (Vector3 localPos, string name)	Looks up the floating-point voxel type attribute at the specified local position by name.
Public	float	GetVoxelTypeBool (Vector3 localPos, int index)	Looks up the floating-point voxel type attribute at the specified local position by index.
Public	int	GetVoxelWithin (Vector3 localPos)	Gets the voxel ID of a voxel at the specified local position exclusively within the chunk.
Public	void	Initialize (VoxelLookup lut, int x = 16, int y = 16, int z = 16,float vSize = 1)	Initializes the voxel grid. This must be done at least once before voxel operations can occur.
Public	bool	IsSolid (Vector3 localPos)	Returns whether the voxel at the specified local position is solid and contributes to the physics mesh.
Public	bool	IsTransparent (Vector3 localPos)	Returns whether the voxel at the specified local position is marked as transparent.
Public	bool	IsVisible (Vector3 localPos)	Returns whether the voxel at the specified local position is visible and contributes to the visible mesh.
Public	VoxelLookup	Lookup()	Gets the lookup table currently in use.
Public	void	ReMesh()	Schedules the chunk to update its mesh on the next available frame.
Public	void	ReMeshImmediate()	Updates the chunk mesh immediately.

Public	bool	SetVoxel(int v, Vector3 localPos)	Attempts to set the voxel at the specified local position to the specified index. Returns TRUE if successful.
Public	bool	SetVoxelInstanceBool (Vector3 localPos, int index, bool value)	Attempts to set the boolean voxel instance attribute at the specified local position and index. Returns TRUE if successful.
Public	bool	SetVoxelInstanceFloat (Vector3 localPos, int index, float value)	Attempts to set the floating-point voxel instance attribute at the specified local position and index. Returns TRUE if successful.
Public	bool	SetVoxelWithin (int v, Vector3 localPos)	Attempts to set the voxel ID of a voxel at the specified local position exclusively within the chunk. Returns TRUE if successful.
Public	void	SwapTables (VoxelLookup table)	Changes the lookup table used by the chunk.
Public	float	VoxelSize()	Returns the scale of an individual voxel.
Public	Vector3	WorldToChunkCoords (Vector3 coords, bool round=false)	Returns the voxel's position in a grid at the given world position, optionally rounding it to the nearest voxel.

CHUNK STATUS

Name: ChunkStatus

Type: Public Enum

Description: Describes the chunk's current operation.

Values:

Name	Description
Generating	The chunk is currently using a generator to populate its grid with voxels
GridOp	The chunk's voxel grid is currently being modified.
Initialize	The chunk has just initialized and is ready for use.
Render	The chunk has just updated its mesh.

VOXEL GENERATOR

Name: `IVoxelGenerator`

Type: `Public Interface`

Description: `Used to populate a chunk with voxels.`

Required Methods:

Access	Return Type	Name (Parameters)	Description
`Public`	`int`	`Sample(Vector3 worldpos)`	`Returns a voxel ID. Use your generator to determine the ID based on the world position.`

THREADED AUTOMATOR [BETA]

Name: ThreadedAutomator

Type: Public Class

Extends: MonoBehaviour

Description: This component serves as a base class for custom multithreaded automators. *Note: This feature is currently under development. Use at your own risk.*

Exposed Variables:

Access	Type	Name	Description
Public	int	latency	Determines how many milliseconds the thread should sleep before continuing to work. It is recommended that this be greater than 0.

Exposed Methods:

Access	Return Type	Name (Parameters)	Description
Protected virtual	bool	Loop()	Called every time the thread loops. Return TRUE for the chunk to update its mesh.
Protected virtual	bool	Once()	Called once when the thread starts. Return TRUE for the chunk to update its mesh.
Protected virtual	bool	PerVoxel(Vector3 position)	Called once per voxel in the chunk. Return TRUE for the chunk to update its mesh.
Protected	void	WriteLine(string message, bool newline = true)	Used to print to the console for debugging. Note that using print() or Debug.Log() will not work.

VOXEL BEHAVIOUR

Name: VoxelBehaviour

Type: Public Class

Extends: MonoBehaviour

Description: This component serves as a base class for custom voxel behavior scripts. These scripts are similar to MonoBehaviour scripts, but deal with individual voxels.

Exposed Methods:

Access	Return Type	Name (Parameters)	Description
Protected	void	AbortReMesh()	Cancels any remesh operations that have been already scheduled to occur that frame.
Protected	int	GetVoxel(Vector3 localPos)	Gets the voxel ID of a voxel at the specified local position.
Protected	bool	GetVoxelInstanceBool (int index, Vector3 localPos)	Returns the boolean voxel instance attribute of a voxel at the specified local position.
Protected	float	GetVoxelInstanceFloat (int index, Vector3 localPos)	Returns the floating-point voxel instance attribute of a voxel at the specified local position.
Protected	string	GetVoxelName (Vector3 localPos)	Returns the name of the voxel at the specified local position.
Protected	bool	GetVoxelTypeBool (Vector3 localPos, string name)	Looks up the boolean voxel type attribute at the specified local position by name.
Protected	bool	GetVoxelTypeBool (Vector3 localPos, int index)	Looks up the boolean voxel type attribute at the specified local position by index.
Protected	float	GetVoxelTypeFloat (Vector3 localPos, string name)	Looks up the floating-point voxel type attribute at the specified local position by name.
Protected	float	GetVoxelTypeBool (Vector3 localPos, int index)	Looks up the floating-point voxel type attribute at the specified local position by index.
Public	void	ReMesh()	Schedules the chunk to update its mesh on the next available frame.
Protected	void	ReMeshImmediate()	Updates the chunk mesh immediately.
Protected	bool	SetVoxel(int v, Vector3 localPos)	Attempts to set the voxel at the specified local position to the specified index. Returns TRUE if successful.
Protected	bool	SetVoxelInstanceBool (Vector3 localPos, int index, bool value)	Attempts to set the boolean voxel instance attribute at the specified local position and index. Returns TRUE if successful.

Public	bool	SetVoxelInstanceFloat (Vector3 localPos, int index, float value)	Attempts to set the floating-point voxel instance attribute at the specified local position and index. Returns TRUE if successful.

VOXEL DATA

Name: VoxelData

Type: Public Struct

Description: Contains data about a specific voxel, and optionally a second 'back' voxel that may pertain to the function of that voxel. This 'back' voxel refers to the voxel adjacent to the face being clicked or collided with in a VoxelBehaviour.

Exposed Variables:

Access	Type	Name	Description
Public	int	ID	The ID of the voxel.
Public	Vector3	worldPos	The world position of the voxel.
Public	Vector3	coords	The chunk coordinates of the voxel.
Public	int	backID	The ID of the 'back' voxel.
Public	Vector3	backWorldPos	The world position of the 'back' voxel.
Public	Vector3	backCoords	The chunk coordinates of the 'back' voxel.

VOXEL LOOKUP

Name: ChunkContainer

Type: Public Sealed Class

Extends: ScriptableObject

Description: This class is used to find information about voxels by ID.

Exposed Variables:

Access	Type	Name	Description
Public	Texture2D	atlas	The texture atlas used by the voxels
Public	Texture2D[]	sources	The textures used by each individual voxel type.
Public	String[]	FloatAttribNames	The names of the floating-point voxel type attributes in this table.
Public	String[]	BoolAttribNames	The names of the boolean voxel type attributes in this table.

VOXEL RANDOM GENERATOR

Name: VoxelRandomGenerator

Type: Public Class

Extends: IVoxelGenerator

Description: This class is a placeholder that is used to generate random voxels in a chunk. This class produces completely random results, and for most applications it is best to use a custom generator. This class is purely intended for debugging and testing purposes.

Exposed Methods:

Access	Return Type	Name (Parameters)	Description
Public	int	Sample(Vector3 worldPos)	Gives a random result, ignoring the position entirely.