

Cave Exploration Starter Kit

1.3

Sun Jan 10 2016 16:02:50

Contents

1	Namespace Index	1
1.1	Packages	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	Namespace Documentation	9
4.1	CaveExploration Namespace Reference	9
4.1.1	Enumeration Type Documentation	12
4.1.1.1	AIState	12
4.1.1.2	NodeType	12
4.1.2	Function Documentation	12
4.1.2.1	InitWaypointDel(List< Node > path)	12
5	Class Documentation	13
5.1	CaveExploration.AudioPlayer Class Reference	13
5.1.1	Detailed Description	13
5.1.2	Member Function Documentation	13
5.1.2.1	IsPlaying()	13
5.1.2.2	PlaySound(AudioClip clip, float volume)	14
5.1.2.3	PlaySound(AudioClip clip, float volume, bool looped)	14
5.1.2.4	PlaySound(AudioClip clip, float volume, float pitch)	14
5.2	CaveExploration.BackgroundAudio Class Reference	14
5.2.1	Detailed Description	14
5.2.2	Member Data Documentation	15
5.2.2.1	BackgroundAudioTracks	15
5.2.2.2	Volume	15
5.3	CaveExploration.BobSprite Class Reference	15
5.3.1	Detailed Description	15
5.3.2	Member Data Documentation	15

5.3.2.1	MaxUpDown	15
5.3.2.2	Speed	15
5.3.3	Property Documentation	16
5.3.3.1	Enabled	16
5.4	CaveExploration.BottomCheck Class Reference	16
5.4.1	Detailed Description	16
5.4.2	Member Data Documentation	16
5.4.2.1	FallDamage	16
5.4.2.2	FallingHeightToDamage	17
5.4.2.3	GroundMask	17
5.4.2.4	particle	17
5.4.3	Property Documentation	17
5.4.3.1	IsGrounded	17
5.5	CaveExploration.Bounce Class Reference	17
5.5.1	Detailed Description	18
5.5.2	Member Function Documentation	18
5.5.2.1	PlayNearbyAudio()	18
5.5.2.2	Update()	18
5.5.3	Member Data Documentation	18
5.5.3.1	PlayerNearbyAudio	18
5.6	CaveExploration.CameraBackground Class Reference	18
5.6.1	Detailed Description	18
5.7	CaveExploration.CharacterSpeech Class Reference	19
5.7.1	Detailed Description	19
5.7.2	Member Function Documentation	19
5.7.2.1	Speak(string text)	19
5.7.3	Property Documentation	19
5.7.3.1	Speaking	19
5.8	CaveExploration.Collectible Class Reference	19
5.8.1	Detailed Description	20
5.8.2	Member Data Documentation	20
5.8.2.1	LightIntensityAdded	20
5.8.2.2	LightRadiusAdded	20
5.8.2.3	SeekDistance	20
5.8.2.4	SeekSpeed	20
5.9	CaveExploration.CollectiblePickedUpEvent Class Reference	20
5.10	CaveExploration.CollectibleSpawner Class Reference	21
5.10.1	Detailed Description	21
5.10.2	Member Function Documentation	22
5.10.2.1	OnLevelGenerated(GameEvent e)	22

5.10.3	Member Data Documentation	22
5.10.3.1	Collectible	22
5.10.3.2	CollectibleWallLimit	22
5.10.3.3	MaxCollectibles	22
5.10.3.4	SpawnChance	22
5.11	CaveExploration.DamagePlayerOnCollision Class Reference	22
5.11.1	Detailed Description	23
5.11.2	Member Data Documentation	23
5.11.2.1	DamageAmount	23
5.11.2.2	DamageForce	23
5.12	CaveExploration.DirectorFullGame Class Reference	23
5.12.1	Detailed Description	23
5.12.2	Member Data Documentation	23
5.12.2.1	EndCollectiblePrefab	23
5.12.2.2	PlayerPrefab	24
5.12.2.3	Seed	24
5.13	CaveExploration.DirectorGridGenerator Class Reference	24
5.13.1	Detailed Description	24
5.13.2	Member Data Documentation	24
5.13.2.1	Seed	24
5.14	CaveExploration.EnableColliderOnTriggerExit Class Reference	24
5.14.1	Detailed Description	25
5.14.2	Member Data Documentation	25
5.14.2.1	MultipleColliders	25
5.14.2.2	Tag	25
5.15	CaveExploration.EnemiesPlaced Class Reference	25
5.16	CaveExploration.EnemyAI Class Reference	26
5.16.1	Detailed Description	26
5.16.2	Member Function Documentation	26
5.16.2.1	Update()	26
5.17	CaveExploration.EnemyHealth Class Reference	27
5.17.1	Detailed Description	27
5.17.2	Member Function Documentation	27
5.17.2.1	Kill(Vector3 projectilePosition, float force)	27
5.17.3	Member Data Documentation	27
5.17.3.1	CountTowardsEnemyCount	27
5.17.3.2	DeathDuration	27
5.17.3.3	SoundOnDeath	28
5.18	CaveExploration.EnemyKillCountSpeech Class Reference	28
5.18.1	Detailed Description	28

5.18.2	Member Data Documentation	28
5.18.2.1	EnemyKilledSpeech	28
5.19	CaveExploration.EnemyKilled Class Reference	28
5.20	CaveExploration.EnemyProjectile Class Reference	29
5.20.1	Detailed Description	29
5.20.2	Member Function Documentation	29
5.20.2.1	ReturnToPool()	29
5.20.3	Member Data Documentation	29
5.20.3.1	DamageAmount	29
5.20.3.2	DamageForce	29
5.20.3.3	MaxTimeAlive	29
5.21	CaveExploration.EnemySpawner Class Reference	30
5.21.1	Detailed Description	30
5.21.2	Member Function Documentation	30
5.21.2.1	OnLevelGenerated(GameEvent e)	30
5.21.3	Member Data Documentation	31
5.21.3.1	Enemy	31
5.21.3.2	MaxEnemies	31
5.21.3.3	PlacementType	31
5.21.3.4	RandomRotation	31
5.22	CaveExploration.Events Class Reference	31
5.23	CaveExploration.Explode Class Reference	31
5.23.1	Detailed Description	32
5.24	CaveExploration.Explosion Class Reference	32
5.24.1	Detailed Description	32
5.24.2	Member Function Documentation	33
5.24.2.1	PlayExplosionSound()	33
5.24.2.2	ReturnToPool()	33
5.24.3	Member Data Documentation	33
5.24.3.1	ExplosionClip	33
5.24.3.2	ExplosionForce	33
5.25	CaveExploration.FollowPlayer Class Reference	33
5.25.1	Detailed Description	33
5.25.2	Member Data Documentation	33
5.25.2.1	Displacement	33
5.26	CaveExploration.GameEvent Class Reference	34
5.26.1	Detailed Description	34
5.27	CaveExploration.GridManager Class Reference	34
5.27.1	Detailed Description	35
5.27.2	Member Function Documentation	36

5.27.2.1	CountWallMooreNeighbours(Vector2 coord)	36
5.27.2.2	GenerateWithSeed(int seed, bool firstLevel)	36
5.27.2.3	GetBackgroundNodes()	36
5.27.2.4	GetFloorNodeMaxDistanceFromStartNode(float maxDistance, Node startNode)	36
5.27.2.5	GetFloorNodeMinDistanceFromStartNode(float minDistance, Node startNode)	36
5.27.2.6	GetRandomBackgroundNode()	37
5.27.2.7	GetRandomFloorNode()	37
5.27.3	Member Data Documentation	37
5.27.3.1	Cell	37
5.27.3.2	ChanceToBecomeWall	37
5.27.3.3	FloorsToWallConversion	37
5.27.3.4	GridSize	37
5.27.3.5	NumberOfTransistionSteps	37
5.27.3.6	WallsToFloorConversion	37
5.27.4	Property Documentation	38
5.27.4.1	EndNode	38
5.27.4.2	Grid	38
5.27.4.3	instance	38
5.27.4.4	StartNode	38
5.27.4.5	TexturePack	38
5.28	CaveExploration.Health Class Reference	38
5.28.1	Detailed Description	39
5.28.2	Member Function Documentation	39
5.28.2.1	OnDamage(PlayerDamagedEvent e)	39
5.28.3	Member Data Documentation	39
5.28.3.1	DamageAnimationTime	39
5.28.3.2	HurtColour	39
5.28.3.3	HurtSpeech	39
5.28.3.4	LowHealthSpeech	39
5.28.3.5	LowHealthThreshold	39
5.28.3.6	MaxHealth	40
5.29	CaveExploration.IntroductorySpeech Class Reference	40
5.29.1	Detailed Description	40
5.29.2	Member Data Documentation	40
5.29.2.1	Speech	40
5.29.3	Property Documentation	40
5.29.3.1	InProgress	40
5.29.3.2	instance	41
5.30	CaveExploration.IntroductorySpeechFinishedEvent Class Reference	41
5.30.1	Detailed Description	41

5.31	CaveExploration.JetAudio Class Reference	41
5.31.1	Detailed Description	41
5.31.2	Member Data Documentation	42
5.31.2.1	JetAudioClip	42
5.31.2.2	Volume	42
5.32	CaveExploration.Jetpack Class Reference	42
5.32.1	Detailed Description	42
5.32.2	Member Function Documentation	43
5.32.2.1	FuelRecharging()	43
5.32.3	Member Data Documentation	43
5.32.3.1	JetForce	43
5.32.3.2	JetFuel	43
5.32.3.3	JetFuelWaste	43
5.32.3.4	MaxForce	43
5.32.4	Property Documentation	43
5.32.4.1	CanJet	43
5.32.4.2	UsingJet	43
5.33	CaveExploration.JetpackBar Class Reference	43
5.33.1	Detailed Description	44
5.33.2	Member Function Documentation	44
5.33.2.1	Disable()	44
5.33.2.2	Enable()	44
5.33.2.3	UpdateColour(float currentFuel)	44
5.33.2.4	UpdateLocalScaleY(float scaleY)	44
5.33.3	Member Data Documentation	45
5.33.3.1	ColourWhenEmptyFuel	45
5.33.3.2	ColourWhenFullFuel	45
5.34	CaveExploration.LevelCompleteEvent Class Reference	45
5.34.1	Detailed Description	45
5.34.2	Constructor & Destructor Documentation	45
5.34.2.1	LevelCompleteEvent(int nextSeed)	45
5.34.3	Property Documentation	46
5.34.3.1	NextSeed	46
5.35	CaveExploration.LevelEnd Class Reference	46
5.35.1	Detailed Description	46
5.36	CaveExploration.LevelGeneratedEvent Class Reference	46
5.36.1	Detailed Description	46
5.37	CaveExploration.LevelGeneratedSpeechRequired Class Reference	47
5.37.1	Detailed Description	47
5.38	CaveExploration.LightDecreaseEvent Class Reference	47

5.38.1 Detailed Description	47
5.38.2 Constructor & Destructor Documentation	48
5.38.2.1 LightDecreaseEvent(float?lightAmount=null, float?radiusAmount=null)	48
5.38.3 Property Documentation	48
5.38.3.1 LightAmount	48
5.38.3.2 RadiusAmount	48
5.39 CaveExploration.LightIntensityPulse Class Reference	48
5.39.1 Detailed Description	48
5.40 CaveExploration.LightPulse Class Reference	49
5.40.1 Detailed Description	49
5.40.2 Member Data Documentation	49
5.40.2.1 Maximum	49
5.40.2.2 Minimum	49
5.40.2.3 Speed	49
5.41 CaveExploration.LightRangePulse Class Reference	50
5.41.1 Detailed Description	50
5.42 CaveExploration.Node Class Reference	50
5.42.1 Detailed Description	51
5.42.2 Constructor & Destructor Documentation	51
5.42.2.1 Node()	51
5.42.2.2 Node(Vector2 coordinates)	51
5.42.2.3 Node(Vector2 coordinates, NodeType state)	51
5.42.2.4 Node(Vector2 coordinates, NodeType state, bool isOccupied)	51
5.42.3 Member Function Documentation	52
5.42.3.1 CompareTo(object obj)	52
5.42.3.2 GetFScore()	52
5.42.4 Property Documentation	52
5.42.4.1 Coordinates	52
5.42.4.2 GScore	52
5.42.4.3 HScore	52
5.42.4.4 IsObstacle	52
5.42.4.5 IsOccupied	52
5.42.4.6 NodeState	53
5.42.4.7 Parent	53
5.42.4.8 Position	53
5.43 CaveExploration.NodeCluster Class Reference	53
5.43.1 Detailed Description	53
5.43.2 Constructor & Destructor Documentation	53
5.43.2.1 NodeCluster()	53
5.43.3 Property Documentation	53

5.43.3.1	Nodes	53
5.44	CaveExploration.NodeClusterManager Class Reference	54
5.44.1	Detailed Description	54
5.44.2	Member Function Documentation	54
5.44.2.1	CalculateMainCluster()	54
5.44.2.2	ConnectClusters()	55
5.44.2.3	ConvertDisconnectedClustersToNodeType(NodeType nodeType)	55
5.44.2.4	IdentifyClusters(NodeList nodes, Rect size)	55
5.44.3	Property Documentation	55
5.44.3.1	Clusters	55
5.44.3.2	MainCluster	55
5.45	CaveExploration.NodeList Class Reference	55
5.45.1	Detailed Description	56
5.45.2	Member Function Documentation	56
5.45.2.1	Add(Node node)	56
5.45.2.2	Contains(Node node)	56
5.45.2.3	ContainsNodeTypeAtPosition(Vector2 coord, NodeType nodeType)	56
5.45.2.4	First()	57
5.45.2.5	GetAdjacentNodes(Vector2 cellCoordinate, bool includeObstacles)	57
5.45.2.6	GetNodeFromGridCoordinate(Vector2 coord)	57
5.45.2.7	GetNodeFromPosition(Vector2 position)	57
5.45.2.8	IsValidCoordinate(Vector2 coord)	57
5.45.2.9	Replace(Node origNode, Node newNode)	58
5.45.2.10	Replace(Vector2 origNodeCoord, Node newNode)	58
5.45.3	Property Documentation	58
5.45.3.1	Count	58
5.46	CaveExploration.ObjectManager Class Reference	58
5.46.1	Detailed Description	59
5.46.2	Member Function Documentation	59
5.46.2.1	GetObject(string prefabName, Vector2 position, Quaternion rotation)	59
5.46.2.2	GetObject(string prefabName, Vector2 position)	59
5.46.2.3	GetObject(string prefabName, Vector2 position, bool onlyPooledObjects)	59
5.46.2.4	RemoveObject(GameObject obj)	60
5.46.2.5	RemoveObjects()	60
5.46.3	Property Documentation	60
5.46.3.1	instance	60
5.47	CaveExploration.ObjectPool Class Reference	60
5.47.1	Detailed Description	61
5.47.2	Member Function Documentation	61
5.47.2.1	GetObjectForType(string objectType, bool onlyPooled)	61

5.47.2.2	PoolObject(GameObject obj)	61
5.47.3	Member Data Documentation	61
5.47.3.1	amountToBuffer	61
5.47.3.2	containerObject	61
5.47.3.3	objectPrefabs	62
5.47.3.4	pooledObjects	62
5.48	CaveExploration.PathManager Class Reference	62
5.48.1	Detailed Description	62
5.48.2	Member Function Documentation	62
5.48.2.1	GetShortestPath(Node orig, Node dest, float wallMovementCost, bool include↔ Obstacles)	62
5.49	CaveExploration.Player Class Reference	63
5.49.1	Detailed Description	63
5.49.2	Member Data Documentation	63
5.49.2.1	IsFacingRight	63
5.49.2.2	JumpForce	64
5.49.2.3	MaxRunSpeed	64
5.49.2.4	MaxWalkSpeed	64
5.49.3	Property Documentation	64
5.49.3.1	HasJumped	64
5.49.3.2	IsDead	64
5.49.3.3	MoveSpeed	64
5.50	CaveExploration.PlayerAnimation Class Reference	64
5.50.1	Detailed Description	65
5.50.2	Member Function Documentation	65
5.50.2.1	FinishedSpawning()	65
5.50.3	Property Documentation	65
5.50.3.1	IsSpawning	65
5.51	CaveExploration.PlayerAudio Class Reference	65
5.51.1	Member Function Documentation	66
5.51.1.1	OnPlayerDamaged()	66
5.51.1.2	PlaySpawnSound()	66
5.51.1.3	PlayStepAudio()	66
5.51.2	Member Data Documentation	66
5.51.2.1	FallingAudio	66
5.51.2.2	HurtAudio	66
5.51.2.3	JumpAudio	66
5.51.2.4	SpawnSound	66
5.51.2.5	StepClips	66
5.52	CaveExploration.PlayerDamagedEvent Class Reference	67

5.52.1 Detailed Description	67
5.52.2 Constructor & Destructor Documentation	67
5.52.2.1 PlayerDamagedEvent(int damageAmount)	67
5.52.2.2 PlayerDamagedEvent(int damageAmount, Vector2?forceDirection, float force)	67
5.52.3 Member Function Documentation	68
5.52.3.1 DamageForce()	68
5.52.4 Property Documentation	68
5.52.4.1 DamageAmount	68
5.53 CaveExploration.PlayerKilledEvent Class Reference	68
5.53.1 Detailed Description	68
5.54 CaveExploration.PlayerLight Class Reference	68
5.54.1 Detailed Description	69
5.54.2 Member Data Documentation	69
5.54.2.1 DamageAnimationTime	69
5.54.2.2 LightDecreaseColour	69
5.54.2.3 LightDecreaseOverTime	69
5.54.2.4 LightDecreaseSpeech	69
5.54.2.5 LightLowSpeech	70
5.54.2.6 LowLightThreshold	70
5.54.2.7 PickUpAudio	70
5.54.2.8 RadiusDecreaseOverTime	70
5.54.2.9 RestartLevelOnEmptyLight	70
5.55 CaveExploration.PlayerSpawnedEvent Class Reference	70
5.55.1 Detailed Description	70
5.55.2 Constructor & Destructor Documentation	71
5.55.2.1 PlayerSpawnedEvent(Transform player)	71
5.55.3 Property Documentation	71
5.55.3.1 Player	71
5.56 CaveExploration.Projectile Class Reference	71
5.56.1 Detailed Description	71
5.56.2 Member Data Documentation	71
5.56.2.1 MaxTimeAlive	71
5.56.2.2 RotationSpeed	72
5.56.2.3 SeekEnemies	72
5.56.2.4 SeekSpeed	72
5.57 CaveExploration.SeekPlayer Class Reference	72
5.57.1 Detailed Description	73
5.57.2 Member Function Documentation	73
5.57.2.1 BurstMoveTowardsPlayer()	73
5.57.2.2 Update()	73

5.57.3	Member Data Documentation	73
5.57.3.1	MaxForce	73
5.57.3.2	MoveAudio	73
5.57.3.3	MoveBurstForce	73
5.57.3.4	RotationSpeed	73
5.57.3.5	TimeBetweenMovementBursts	73
5.58	CaveExploration.ShootPlayer Class Reference	74
5.58.1	Detailed Description	74
5.58.2	Member Function Documentation	75
5.58.2.1	ShootProjectile()	75
5.58.2.2	Update()	75
5.58.3	Member Data Documentation	75
5.58.3.1	IsFacingRight	75
5.58.3.2	MoveSpeed	75
5.58.3.3	Projectile	75
5.58.3.4	ProjectileForce	75
5.58.3.5	ProjectileSpawnLocation	75
5.58.3.6	ShootAudioClip	75
5.59	CaveExploration.SideCheck Class Reference	75
5.59.1	Detailed Description	76
5.59.2	Member Data Documentation	76
5.59.2.1	GroundMask	76
5.59.2.2	KnockbackForce	76
5.60	SortingLayer Class Reference	76
5.60.1	Detailed Description	76
5.60.2	Member Data Documentation	77
5.60.2.1	SortingOrder	77
5.61	CaveExploration.TexturePack Class Reference	77
5.61.1	Detailed Description	78
5.61.2	Member Function Documentation	78
5.61.2.1	GetSpriteFromCellType(NodeType cellType)	78
5.61.2.2	GetSpriteSize(NodeType cellType, Vector2 localScale)	78
5.61.3	Member Data Documentation	78
5.61.3.1	Background	78
5.61.3.2	Enabled	78
5.61.3.3	WallBottomLeft	78
5.61.3.4	WallBottomMiddle	78
5.61.3.5	WallBottomRight	78
5.61.3.6	WallMiddle	79
5.61.3.7	WallMiddleLeft	79

5.61.3.8	WallMiddleRight	79
5.61.3.9	WallTopLeft	79
5.61.3.10	WallTopMiddle	79
5.61.3.11	WallTopRight	79
5.62	CaveExploration.ThrowLight Class Reference	79
5.62.1	Detailed Description	80
5.62.2	Member Data Documentation	80
5.62.2.1	Capacity	80
5.62.2.2	Force	80
5.62.2.3	SpeechOnEmpty	80
5.62.2.4	Throwable	80
5.63	CaveExploration.TopCheck Class Reference	80
5.63.1	Detailed Description	80
5.63.2	Member Data Documentation	81
5.63.2.1	GroundMask	81
5.63.2.2	KnockbackForce	81
5.63.2.3	particle	81
5.64	CaveExploration.Utilities Class Reference	81
5.64.1	Detailed Description	82
5.64.2	Member Function Documentation	82
5.64.2.1	CreateRays(Vector2 position, Vector2 heading, float angle, int number)	82
5.64.2.2	CreateRays(Vector2 position, Vector2 heading, float angle)	82
5.64.2.3	FindObjectOfType< T >(string scriptName)	83
5.64.2.4	FindObjectWithTag(string scriptName, string tag)	83
5.64.2.5	GetComponent< T >(string scriptName, Transform owner)	83
5.64.2.6	GetComponent< T >(string scriptName)	83
5.64.2.7	GetGridCoordinateForPosition(Vector2 position)	83
5.64.2.8	GetNodePosition(Node node)	83
5.64.2.9	GetPrefab(string scriptName, string prefabName)	83
5.64.2.10	IncrementTowards(float n, float target, float acc)	83
5.64.2.11	InstantiatePrefab(string scriptName, GameObject prefab)	84
5.64.2.12	InstantiatePrefabAtPosition(string scriptName, GameObject prefab, Vector2 position)	84
5.64.3	Member Data Documentation	84
5.64.3.1	IsDebug	84
5.64.4	Property Documentation	84
5.64.4.1	instance	84
5.65	CaveExploration.WaypointManager Class Reference	84
5.65.1	Detailed Description	85
5.65.2	Member Function Documentation	85

5.65.2.1	GetCurrentWaypoint()	85
5.65.2.2	GetPosition(Vector2 currentPosition)	85
5.65.2.3	GetNextReactivePosition(Vector2 currentPosition)	85
5.65.2.4	HasReachedCurrentWaypoint(Vector2 characterPosition)	86
5.65.2.5	Initialised()	86
5.65.2.6	InitialiseWaypointsFromNodes(List< Node > path)	86
5.65.3	Member Data Documentation	86
5.65.3.1	IsLooped	86
5.65.3.2	WaypointProximity	86
5.65.3.3	Waypoints	86
5.65.4	Property Documentation	86
5.65.4.1	IsComplete	86
Index		87

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

CaveExploration	9
---	---

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

CaveExploration.Events	31
CaveExploration.GameEvent	34
CaveExploration.CollectiblePickedUpEvent	20
CaveExploration.EnemiesPlaced	25
CaveExploration.EnemyKilled	28
CaveExploration.IntroductorySpeechFinishedEvent	41
CaveExploration.LevelCompleteEvent	45
CaveExploration.LevelGeneratedEvent	46
CaveExploration.LevelGeneratedSpeechRequired	47
CaveExploration.LightDecreaseEvent	47
CaveExploration.PlayerDamagedEvent	67
CaveExploration.PlayerKilledEvent	68
CaveExploration.PlayerSpawnedEvent	70
IComparable	
CaveExploration.Node	50
MonoBehaviour	
CaveExploration.AudioPlayer	13
CaveExploration.BackgroundAudio	14
CaveExploration.BobSprite	15
CaveExploration.BottomCheck	16
CaveExploration.CameraBackground	18
CaveExploration.CharacterSpeech	19
CaveExploration.Collectible	19
CaveExploration.CollectibleSpawner	21
CaveExploration.DamagePlayerOnCollision	22
CaveExploration.DirectorFullGame	23
CaveExploration.DirectorGridGenerator	24
CaveExploration.EnableColliderOnTriggerExit	24
CaveExploration.EnemyAI	26
CaveExploration.Bounce	17
CaveExploration.SeekPlayer	72
CaveExploration.ShootPlayer	74
CaveExploration.EnemyHealth	27
CaveExploration.EnemyKillCountSpeech	28
CaveExploration.EnemyProjectile	29
CaveExploration.EnemySpawner	30
CaveExploration.Explode	31

CaveExploration.Explosion	32
CaveExploration.FollowPlayer	33
CaveExploration.GridManager	34
CaveExploration.Health	38
CaveExploration.IntroductorySpeech	40
CaveExploration.JetAudio	41
CaveExploration.Jetpack	42
CaveExploration.JetpackBar	43
CaveExploration.LevelEnd	46
CaveExploration.LightPulse	49
CaveExploration.LightIntensityPulse	48
CaveExploration.LightRangePulse	50
CaveExploration.NodeClusterManager	54
CaveExploration.ObjectManager	58
CaveExploration.ObjectPool	60
CaveExploration.PathManager	62
CaveExploration.Player	63
CaveExploration.PlayerAnimation	64
CaveExploration.PlayerAudio	65
CaveExploration.PlayerLight	68
CaveExploration.Projectile	71
CaveExploration.SideCheck	75
CaveExploration.TexturePack	77
CaveExploration.ThrowLight	79
CaveExploration.TopCheck	80
CaveExploration.Utilities	81
CaveExploration.WaypointManager	84
SortingLayer	76
CaveExploration.NodeCluster	53
CaveExploration.NodeList	55

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

CaveExploration.AudioPlayer	Handles playing audio clips, includes methods to handle volume, pitch, looped status, and playing songs forwards/backwards.	13
CaveExploration.BackgroundAudio	Plays a background audio clip on start.	14
CaveExploration.BobSprite	"Bobs" sprite up and down.	15
CaveExploration.BottomCheck	Used to determine if the player is grounded and apply fall damage.	16
CaveExploration.Bounce	AI for the "bounce" enemy. Handles animation, audio and AI states.	17
CaveExploration.CameraBackground	Handles changing of background image based on currently used texture pack.	18
CaveExploration.CharacterSpeech	Responsible for showing character speech (text) on screen.	19
CaveExploration.Collectible	Attach to collectible in game. Handles seeking player, and giving player a light boost on collection.	19
CaveExploration.CollectiblePickedUpEvent		20
CaveExploration.CollectibleSpawner	Spawns collectibles on round start.	21
CaveExploration.DamagePlayerOnCollision	Attach to any object to damage the player when a collision with this object occurs.	22
CaveExploration.DirectorFullGame	Manages the game scene. Starts level, player, and collectible placement/creation. Handles player dead and level complete events.	23
CaveExploration.DirectorGridGenerator	Managers the example grid generator scene.	24
CaveExploration.EnableColliderOnTriggerExit	Disables a colliders trigger (i.e. enables collisions with a collider) when the collider exits the specified trigger. Useful for shooting a projectile to ensure it does not collide with the enemy/player that shot the projectile.	24
CaveExploration.EnemiesPlaced		25
CaveExploration.EnemyAI	Abstract base class for all enemy AI.	26
CaveExploration.EnemyHealth	Handles enemy health and death (including audio and animation).	27

CaveExploration.EnemyKillCountSpeech	
Shows character speech on enemy killed.	28
CaveExploration.EnemyKilled	28
CaveExploration.EnemyProjectile	
Responsible to updating enemy projectiles. And applying damage on collision.	29
CaveExploration.EnemySpawner	
Responsible for spawning enemies at level start.	30
CaveExploration.Events	31
CaveExploration.Explode	
Spawns an explosion on mouse click at the objects position.	31
CaveExploration.Explosion	
Attach to an explosion. Kills an enemy in explosion radius.	32
CaveExploration.FollowPlayer	
Attach to camera. Used to follow player once spawned.	33
CaveExploration.GameEvent	
All game events inherit this abstract base class.	34
CaveExploration.GridManager	
Singleton. Handles level generation.	34
CaveExploration.Health	
The players health.	38
CaveExploration.IntroductorySpeech	
Introductory speech. For example, can be used to introduce a level, the game, controls etc. . .	40
CaveExploration.IntroductorySpeechFinishedEvent	
Raised when the players intro speech has finished.	41
CaveExploration.JetAudio	
Handles playing audio on jet pack use.	41
CaveExploration.Jetpack	
Responsible for controlling the players jetpack.	42
CaveExploration.JetpackBar	
Visual representation of the jetpack fuel.	43
CaveExploration.LevelCompleteEvent	
Raised when the player completes a level.	45
CaveExploration.LevelEnd	
Added to the exit block for the current level. When the character enters the blocks collider Level←	
Complete is set to true. This is queried in the Director Class.	46
CaveExploration.LevelGeneratedEvent	
Raised when a level has finished being generated.	46
CaveExploration.LevelGeneratedSpeechRequired	
Raised when a level has been generated and player speech is required.	47
CaveExploration.LightDecreaseEvent	
Raised when the players light should be decreased e.g. on damage.	47
CaveExploration.LightIntensityPulse	
Pulses a lights intensity between two values over time.	48
CaveExploration.LightPulse	
Used to pulse a light variable.	49
CaveExploration.LightRangePulse	
Pulses a light range between two values over time.	50
CaveExploration.Node	
Logical representation of a block. Holds the node type (e.g. wall, floor etc), it's coordinates (a pointer to the node in a 2d array - not it's position on screen), it's position on screen and path finding variables.	50
CaveExploration.NodeCluster	
Represents a group of neighbouring floor nodes i.e. a cavern in the environment.	53
CaveExploration.NodeClusterManager	
Singleton. Identifies, manages and holds reference to the different node clusters. Can connect un-connected clusters using path finding.	54

CaveExploration.NodeList	Encapsulates 2D array of all active nodes in game. Provides helper methods such as Add and Contains that mimick there list counterparts.	55
CaveExploration.ObjectManager	Manages adding and retrieving objects from the object pool.	58
CaveExploration.ObjectPool	Object pool system.	60
CaveExploration.PathManager	Creates a path between two nodes.	62
CaveExploration.Player	Handles player movement.	63
CaveExploration.PlayerAnimation	Responsible for updating the players animation based on movement.	64
CaveExploration.PlayerAudio	65
CaveExploration.PlayerDamagedEvent	Raised when the player takes damage.	67
CaveExploration.PlayerKilledEvent	Raised when the player is killed.	68
CaveExploration.PlayerLight	Responsible for updating the players light.	68
CaveExploration.PlayerSpawnedEvent	Raised when the player is spawned.	70
CaveExploration.Projectile	Attach to players light projectile. Handles seeking enemies, collision, and returning projectile to pool.	71
CaveExploration.SeekPlayer	AI for the "SeekPlayer" enemy. Handles animation, audio, and AI states.	72
CaveExploration.ShootPlayer	AI for the "ShootPlayer" enemy. Handles animation, audio, and AI states.	74
CaveExploration.SideCheck	Checks whether the player is colliding on the sides with a wall tile.	75
SortingLayer	Changes a mesh renderers sorting layer.	76
CaveExploration.TexturePack	Singleton. Provides a centralised location for block tetures. Provides a method to retrieve a texture based on a node type.	77
CaveExploration.ThrowLight	Handles the players ability to throw light.	79
CaveExploration.TopCheck	Checks whether the player is colliding on the top with a wall tile.	80
CaveExploration.Utilities	Singleton. Provides centralised access to methods used by different classes.	81
CaveExploration.WaypointManager	Holds list of waypoints and allows iteration over waypoints.	84

Chapter 4

Namespace Documentation

4.1 CaveExploration Namespace Reference

Classes

- class [AudioPlayer](#)
Handles playing audio clips, includes methods to handle volume, pitch, looped status, and playing songs forwards/backwards.
- class [BackgroundAudio](#)
Plays a background audio clip on start.
- class [BobSprite](#)
"Bobs" sprite up and down.
- class [BottomCheck](#)
Used to determine if the player is grounded and apply fall damage.
- class [Bounce](#)
AI for the "bounce" enemy. Handles animation, audio and AI states.
- class [CameraBackground](#)
Handles changing of background image based on currently used texture pack.
- class [CharacterSpeech](#)
Responsible for showing character speech (text) on screen.
- class [Collectible](#)
Attach to collectible in game. Handles seeking player, and giving player a light boost on collection.
- class [CollectiblePickedUpEvent](#)
- class [CollectibleSpawner](#)
Spawns collectibles on round start.
- class [DamagePlayerOnCollision](#)
Attach to any object to damage the player when a collision with this object occurs.
- class [DirectorFullGame](#)
Manages the game scene. Starts level, player, and collectible placement/creation. Handles player dead and level complete events.
- class [DirectorGridGenerator](#)
Managers the example grid generator scene.
- class [EnableColliderOnTriggerExit](#)
Disables a colliders trigger (i.e. enables collisions with a collider) when the collider exits the specified trigger. Useful for shooting a projectile to ensure it does not collide with the enemy/player that shot the projectile.
- class [EnemiesPlaced](#)
- class [EnemyAI](#)
Abstract base class for all enemy AI.

- class [EnemyHealth](#)
Handles enemy health and death (including audio and animation).
- class [EnemyKillCountSpeech](#)
Shows character speech on enemy killed.
- class [EnemyKilled](#)
- class [EnemyProjectile](#)
Responsible to updating enemy projectiles. And applying damage on collision.
- class [EnemySpawner](#)
Responsible for spawning enemies at level start.
- class [Events](#)
- class [Explode](#)
Spawns an explosion on mouse click at the objects position.
- class [Explosion](#)
Attach to an explosion. Kills an enemy in explosion radius.
- class [FollowPlayer](#)
Attach to camera. Used to follow player once spawned.
- class [GameEvent](#)
All game events inherit this abstract base class.
- class [GridManager](#)
Singleton. Handles level generation.
- class [Health](#)
The players health.
- class [IntroductorySpeech](#)
Introductory speech. For example, can be used to introduce a level, the game, controls etc.
- class [IntroductorySpeechFinishedEvent](#)
Raised when the players intro speech has finished.
- class [JetAudio](#)
Handles playing audio on jet pack use.
- class [Jetpack](#)
Responsible for controlling the players jetpack.
- class [JetpackBar](#)
Visual representation of the jetpack fuel.
- class [LevelCompleteEvent](#)
Raised when the player completes a level.
- class [LevelEnd](#)
Added to the exit block for the current level. When the character enters the blocks collider LevelComplete is set to true. This is queried in the Director Class.
- class [LevelGeneratedEvent](#)
Raised when a level has finished being generated.
- class [LevelGeneratedSpeechRequired](#)
Raised when a level has been generated and player speech is required.
- class [LightDecreaseEvent](#)
Raised when the players light should be decreased e.g. on damage.
- class [LightIntensityPulse](#)
Pulses a lights intensity between two values over time.
- class [LightPulse](#)
Used to pulse a light variable.
- class [LightRangePulse](#)
Pulses a light range between two values over time.
- class [Node](#)

Logical representation of a block. Holds the node type (e.g. wall, floor etc), it's coordinates (a pointer to the node in a 2d array - not it's position on screen), it's position on screen and path finding variables.

- class [NodeCluster](#)

Represents a group of neighbouring floor nodes i.e. a cavern in the environment.

- class [NodeClusterManager](#)

Singleton. Identifies, manages and holds reference to the different node clusters. Can connect un-connected clusters using path finding.

- class [NodeList](#)

Encapsulates 2D array of all active nodes in game. Provides helper methods such as Add and Contains that mimic there list counterparts.

- class [ObjectManager](#)

Manages adding and retrieving objects from the object pool.

- class [ObjectPool](#)

Object pool system.

- class [PathManager](#)

Creates a path between two nodes.

- class [Player](#)

Handles player movement.

- class [PlayerAnimation](#)

Responsible for updating the players animation based on movement.

- class [PlayerAudio](#)

- class [PlayerDamagedEvent](#)

Raised when the player takes damage.

- class [PlayerKilledEvent](#)

Raised when the player is killed.

- class [PlayerLight](#)

Responsible for updating the players light.

- class [PlayerSpawnedEvent](#)

Raised when the player is spawned.

- class [Projectile](#)

Attach to players light projectile. Handles seeking enemies, collision, and returning projectile to pool.

- class [SeekPlayer](#)

AI for the "SeekPlayer" enemy. Handles animation, audio, and AI states.

- class [ShootPlayer](#)

AI for the "ShootPlayer" enemy. Handles animation, audio, and AI states.

- class [SideCheck](#)

Checks whether the player is colliding on the sides with a wall tile.

- class [TexturePack](#)

Singleton. Provides a centralised location for block tetures. Provides a method to retrieve a texture based on a node type.

- class [ThrowLight](#)

Handles the players ability to throw light.

- class [TopCheck](#)

Checks whether the player is colliding on the top with a wall tile.

- class [Utilities](#)

Singleton. Provides centralised access to methods used by different classes.

- class [WaypointManager](#)

Holds list of waypoints and allows iteration over waypoints.

Enumerations

- enum **AIState** { **Idle**, **PlayerInSight** }
Enemy AI state.
- enum **NodeType** {
Invalid = -1, **Wall**, **WallTopLeft**, **WallTopMiddle**,
WallTopRight, **WallMiddleLeft**, **WallMiddle**, **WallMiddleRight**,
WallBottomLeft, **WallBottomMiddle**, **WallBottomRight**, **Background**,
Entry, **Exit**, **Collectible**, **Max** }
The cell type.

Functions

- delegate void **InitWaypointDel** (List< **Node** > path)
Delegate used by waypoint manager.

4.1.1 Enumeration Type Documentation

4.1.1.1 enum **CaveExploration.AIState** [strong]

Enemy AI state.

4.1.1.2 enum **CaveExploration.NodeType** [strong]

The cell type.

4.1.2 Function Documentation

4.1.2.1 delegate void **CaveExploration.InitWaypointDel** (List< **Node** > *path*)

Delegate used by waypoint manager.

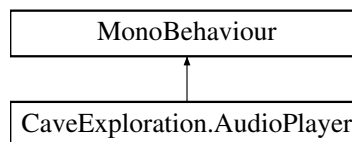
Chapter 5

Class Documentation

5.1 CaveExploration.AudioPlayer Class Reference

Handles playing audio clips, includes methods to handle volume, pitch, looped status, and playing songs forwards/backwards.

Inheritance diagram for CaveExploration.AudioPlayer:



Public Member Functions

- bool `IsPlaying` ()
Determines whether this instance is playing an audio clip.
- void `PlaySound` (AudioClip clip, float volume)
plays the sound clip at the specified volume.
- void `PlaySound` (AudioClip clip, float volume, bool looped)
plays the sound clip at the specified volume.
- void `PlaySound` (AudioClip clip, float volume, float pitch)
plays the sound clip at the specified volume and pitch.

5.1.1 Detailed Description

Handles playing audio clips, includes methods to handle volume, pitch, looped status, and playing songs forwards/backwards.

5.1.2 Member Function Documentation

5.1.2.1 bool CaveExploration.AudioPlayer.IsPlaying ()

Determines whether this instance is playing an audio clip.

Returns

`true` if this instance is playing; otherwise, `false`.

5.1.2.2 void CaveExploration.AudioPlayer.PlaySound (AudioClip *clip*, float *volume*)

plays the sound clip at the specified volume.

Parameters

<i>clip</i>	Clip.
<i>volume</i>	Volume.

5.1.2.3 void CaveExploration.AudioPlayer.PlaySound (AudioClip *clip*, float *volume*, bool *looped*)

plays the sound clip at the specified volume.

Parameters

<i>clip</i>	Clip.
<i>volume</i>	Volume.
<i>looped</i>	If set to <code>true</code> the clip is looped.

5.1.2.4 void CaveExploration.AudioPlayer.PlaySound (AudioClip *clip*, float *volume*, float *pitch*)

plays the sound clip at the specified volume and pitch.

Parameters

<i>clip</i>	Clip.
<i>volume</i>	Volume.
<i>pitch</i>	Pitch.

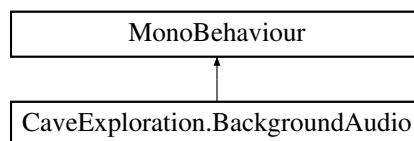
The documentation for this class was generated from the following file:

- `AudioPlayer.cs`

5.2 CaveExploration.BackgroundAudio Class Reference

Plays a background audio clip on start.

Inheritance diagram for CaveExploration.BackgroundAudio:



Public Attributes

- `AudioClip[]` [BackgroundAudioTracks](#)
The background audio tracks. A random track (with no bias) is selected and played.
- `float` [Volume](#) = 0.5f
The volume to play background audio.

5.2.1 Detailed Description

Plays a background audio clip on start.

5.2.2 Member Data Documentation

5.2.2.1 AudioClip [] CaveExploration.BackgroundAudio.BackgroundAudioTracks

The background audio tracks. A random track (with no bias) is selected and played.

5.2.2.2 float CaveExploration.BackgroundAudio.Volume = 0.5f

The volume to play background audio.

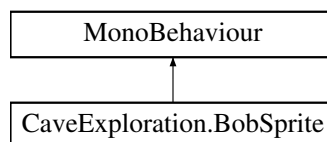
The documentation for this class was generated from the following file:

- BackgroundAudio.cs

5.3 CaveExploration.BobSprite Class Reference

"Bobs" sprite up and down.

Inheritance diagram for CaveExploration.BobSprite:



Public Attributes

- float **MaxUpDown** = 0.1f
Max up/down movement.
- float **Speed** = 1f
Up/down speed.

Properties

- bool **Enabled** [set]
Sets a value indicating whether this [CaveExploration.BobSprite](#) is enabled.

5.3.1 Detailed Description

"Bobs" sprite up and down.

5.3.2 Member Data Documentation

5.3.2.1 float CaveExploration.BobSprite.MaxUpDown = 0.1f

Max up/down movement.

5.3.2.2 float CaveExploration.BobSprite.Speed = 1f

Up/down speed.

5.3.3 Property Documentation

5.3.3.1 `bool CaveExploration.BobSprite.Enabled` [set]

Sets a value indicating whether this `CaveExploration.BobSprite` is enabled.

`true` if enabled; otherwise, `false`.

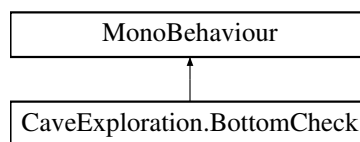
The documentation for this class was generated from the following file:

- `BobSprite.cs`

5.4 CaveExploration.BottomCheck Class Reference

Used to determine if the player is grounded and apply fall damage.

Inheritance diagram for `CaveExploration.BottomCheck`:



Public Attributes

- LayerMask `GroundMask`
The ground mask.
- float `FallingHeightToDamage` = 2f
The height at which the player can fall from and damage applied.
- int `FallDamage` = 1
The damage applied when fall damage is applied.
- ParticleSystem `particle`
The particle system used when the player becomes grounded.

Properties

- bool `IsGrounded` [get, set]
Gets or sets a value indicating whether the player is grounded.

5.4.1 Detailed Description

Used to determine if the player is grounded and apply fall damage.

5.4.2 Member Data Documentation

5.4.2.1 `int CaveExploration.BottomCheck.FallDamage` = 1

The damage applied when fall damage is applied.

5.4.2.2 float CaveExploration.BottomCheck.FallingHeightToDamage = 2f

The height at which the player can fall from and damage applied.

5.4.2.3 LayerMask CaveExploration.BottomCheck.GroundMask

The ground mask.

5.4.2.4 ParticleSystem CaveExploration.BottomCheck.particle

The particle system used when the player becomes grounded.

5.4.3 Property Documentation

5.4.3.1 bool CaveExploration.BottomCheck.IsGrounded [get], [set]

Gets or sets a value indicating whether the player is grounded.

`true` if this instance is grounded; otherwise, `false`.

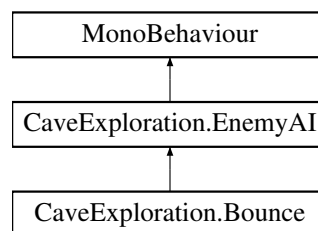
The documentation for this class was generated from the following file:

- BottomCheck.cs

5.5 CaveExploration.Bounce Class Reference

AI for the "bounce" enemy. Handles animation, audio and AI states.

Inheritance diagram for CaveExploration.Bounce:



Public Member Functions

- void [PlayNearbyAudio](#) ()
Plays the audio clip when enemy activates.
- override void [Update](#) ()
Update this instance using a simple state machine.

Public Attributes

- AudioClip [PlayerNearbyAudio](#)
The audio clip to play when enemy activates.

Protected Member Functions

- override void **DecideState** ()
- override void **Idle** ()
- override void **AttackPlayer** ()

Additional Inherited Members

5.5.1 Detailed Description

AI for the "bounce" enemy. Handles animation, audio and AI states.

5.5.2 Member Function Documentation

5.5.2.1 void CaveExploration.Bounce.PlayNearbyAudio ()

Plays the audio clip when enemy activates.

5.5.2.2 override void CaveExploration.Bounce.Update () [virtual]

Update this instance using a simple state machine.

Reimplemented from [CaveExploration.EnemyAI](#).

5.5.3 Member Data Documentation

5.5.3.1 AudioClip CaveExploration.Bounce.PlayerNearbyAudio

The audio clip to play when enemy activates.

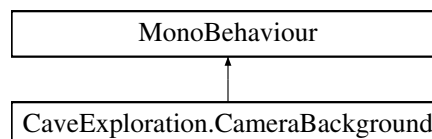
The documentation for this class was generated from the following file:

- Bounce.cs

5.6 CaveExploration.CameraBackground Class Reference

Handles changing of background image based on currently used texture pack.

Inheritance diagram for CaveExploration.CameraBackground:



5.6.1 Detailed Description

Handles changing of background image based on currently used texture pack.

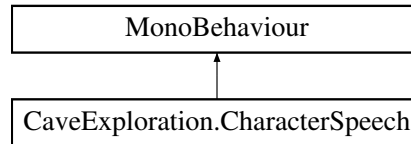
The documentation for this class was generated from the following file:

- CameraBackground.cs

5.7 CaveExploration.CharacterSpeech Class Reference

Responsible for showing character speech (text) on screen.

Inheritance diagram for CaveExploration.CharacterSpeech:



Public Member Functions

- void [Speak](#) (string text)
Shows the specified text at the location. The time the text is shown is based on the text length.

Properties

- bool [Speaking](#) [get, set]
Gets or sets a value indicating whether this [CaveExploration.CharacterSpeech](#) is currently speaking.

5.7.1 Detailed Description

Responsible for showing character speech (text) on screen.

5.7.2 Member Function Documentation

5.7.2.1 void CaveExploration.CharacterSpeech.Speak (string text)

Shows the specified text at the location. The time the text is shown is based on the text length.

Parameters

<i>text</i>	Text.
-------------	-------

5.7.3 Property Documentation

5.7.3.1 bool CaveExploration.CharacterSpeech.Speaking [get], [set]

Gets or sets a value indicating whether this [CaveExploration.CharacterSpeech](#) is currently speaking.

`true` if speaking; otherwise, `false`.

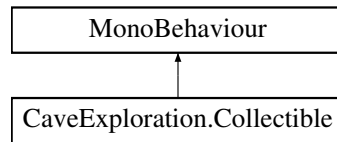
The documentation for this class was generated from the following file:

- CharacterSpeech.cs

5.8 CaveExploration.Collectible Class Reference

Attach to collectible in game. Handles seeking player, and giving player a light boost on collection.

Inheritance diagram for CaveExploration.Collectible:



Public Attributes

- float `LightIntensityAdded` = 1f
The light intensity added to player on collection.
- float `LightRadiusAdded` = 0.2f
The light radius added to player on collection.
- float `SeekDistance` = 1
If collectible within this distance to player, the collectible will move towards players location.
- float `SeekSpeed` = 1f
The speed at which collectible moves towards player.

5.8.1 Detailed Description

Attach to collectible in game. Handles seeking player, and giving player a light boost on collection.

5.8.2 Member Data Documentation

5.8.2.1 float CaveExploration.Collectible.LightIntensityAdded = 1f

The light intensity added to player on collection.

5.8.2.2 float CaveExploration.Collectible.LightRadiusAdded = 0.2f

The light radius added to player on collection.

5.8.2.3 float CaveExploration.Collectible.SeekDistance = 1

If collectible within this distance to player, the collectible will move towards players location.

5.8.2.4 float CaveExploration.Collectible.SeekSpeed = 1f

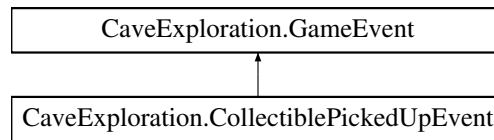
The speed at which collectible moves towards player.

The documentation for this class was generated from the following file:

- Collectible.cs

5.9 CaveExploration.CollectiblePickedUpEvent Class Reference

Inheritance diagram for CaveExploration.CollectiblePickedUpEvent:



Public Member Functions

- **CollectiblePickedUpEvent** (float lightAmount, float radiusAmount)

Properties

- float **LightAmount** [get]
- float **RadiusAmount** [get]

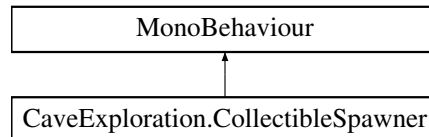
The documentation for this class was generated from the following file:

- CollectiblePickedUpEvent.cs

5.10 CaveExploration.CollectibleSpawner Class Reference

Spawns collectibles on round start.

Inheritance diagram for CaveExploration.CollectibleSpawner:



Public Member Functions

- void **OnLevelGenerated** ([GameEvent](#) e)
Handles the level generated event. Places collectibles.

Public Attributes

- [GameObject](#) **Collectible**
The collectible prefab.
- int **CollectibleWallLimit** = 4
The number of surrounding walls required to spawn collectible.
- float **SpawnChance** = 0.1f
The chance a collectible will spawn when a suitable location is found.
- int **MaxCollectibles** = 40
The maximum number of collectibles to be spawned in any one level.

5.10.1 Detailed Description

Spawns collectibles on round start.

5.10.2 Member Function Documentation

5.10.2.1 void CaveExploration.CollectibleSpawner.OnLevelGenerated (GameEvent e)

Handles the level generated event. Places collectibles.

Parameters

e	E.
---	----

5.10.3 Member Data Documentation

5.10.3.1 GameObject CaveExploration.CollectibleSpawner.Collectible

The collectible prefab.

5.10.3.2 int CaveExploration.CollectibleSpawner.CollectibleWallLimit = 4

The number of surrounding walls required to spawn collectible.

5.10.3.3 int CaveExploration.CollectibleSpawner.MaxCollectibles = 40

The maximum number of collectibles to be spawned in any one level.

5.10.3.4 float CaveExploration.CollectibleSpawner.SpawnChance = 0.1f

The chance a collectible will spawn when a suitable location is found.

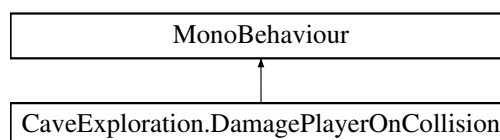
The documentation for this class was generated from the following file:

- CollectibleSpawner.cs

5.11 CaveExploration.DamagePlayerOnCollision Class Reference

Attach to any object to damage the player when a collision with this object occurs.

Inheritance diagram for CaveExploration.DamagePlayerOnCollision:



Public Attributes

- int [DamageAmount](#) = 1
The damage amount.
- float [DamageForce](#) = 50f
The damage knockback force.

5.11.1 Detailed Description

Attach to any object to damage the player when a collision with this object occurs.

5.11.2 Member Data Documentation

5.11.2.1 `int CaveExploration.DamagePlayerOnCollision.DamageAmount = 1`

The damage amount.

5.11.2.2 `float CaveExploration.DamagePlayerOnCollision.DamageForce = 50f`

The damage knockback force.

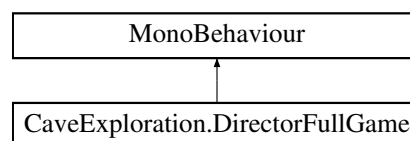
The documentation for this class was generated from the following file:

- `DamagePlayerOnCollision.cs`

5.12 CaveExploration.DirectorFullGame Class Reference

Manages the game scene. Starts level, player, and collectible placement/creation. Handles player dead and level complete events.

Inheritance diagram for `CaveExploration.DirectorFullGame`:



Public Attributes

- `int Seed = 1`
The seed used to generate the level. The same seed and generation settings will always generate the same level.
- `GameObject PlayerPrefab`
The player prefab.
- `GameObject EndCollectiblePrefab`
The end collectible prefab.

5.12.1 Detailed Description

Manages the game scene. Starts level, player, and collectible placement/creation. Handles player dead and level complete events.

5.12.2 Member Data Documentation

5.12.2.1 `GameObject CaveExploration.DirectorFullGame.EndCollectiblePrefab`

The end collectible prefab.

5.12.2.2 `GameObject CaveExploration.DirectorFullGame.PlayerPrefab`

The player prefab.

5.12.2.3 `int CaveExploration.DirectorFullGame.Seed = 1`

The seed used to generate the level. The same seed and generation settings will always generate the same level.

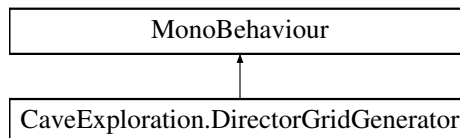
The documentation for this class was generated from the following file:

- `DirectorFullGame.cs`

5.13 `CaveExploration.DirectorGridGenerator` Class Reference

Managers the example grid generator scene.

Inheritance diagram for `CaveExploration.DirectorGridGenerator`:



Public Attributes

- `int Seed = 1`

The seed used to generate the level. The same seed and generation settings will always generate the same level.

5.13.1 Detailed Description

Managers the example grid generator scene.

5.13.2 Member Data Documentation

5.13.2.1 `int CaveExploration.DirectorGridGenerator.Seed = 1`

The seed used to generate the level. The same seed and generation settings will always generate the same level.

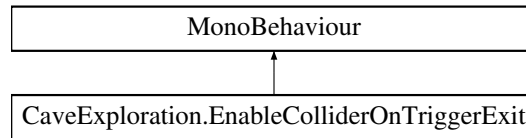
The documentation for this class was generated from the following file:

- `DirectorGridGenerator.cs`

5.14 `CaveExploration.EnableColliderOnTriggerExit` Class Reference

Disables a colliders trigger (i.e. enables collisions with a collider) when the collider exits the specified trigger. Useful for shooting a projectile to ensure it does not collide with the enemy/player that shot the projectile.

Inheritance diagram for `CaveExploration.EnableColliderOnTriggerExit`:



Public Attributes

- bool `MultipleColliders` = true
Enable this is the object contains multiple colliders.
- string `Tag` = "Player"
The tag of the gameobject that will be shooting projectile.

5.14.1 Detailed Description

Disables a colliders trigger (i.e. enables collisions with a collider) when the collider exits the specified trigger. Useful for shooting a projectile to ensure it does not collide with the enemy/player that shot the projectile.

5.14.2 Member Data Documentation

5.14.2.1 bool CaveExploration.EnableColliderOnTriggerExit.MultipleColliders = true

Enable this is the object contains multiple colliders.

5.14.2.2 string CaveExploration.EnableColliderOnTriggerExit.Tag = "Player"

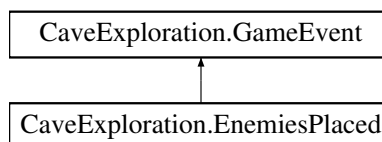
The tag of the gameobject that will be shooting projectile.

The documentation for this class was generated from the following file:

- EnableColliderOnTriggerExit.cs

5.15 CaveExploration.EnemiesPlaced Class Reference

Inheritance diagram for CaveExploration.EnemiesPlaced:



Public Member Functions

- **EnemiesPlaced** (int enemyCount)

Properties

- int **EnemyCount** [get]

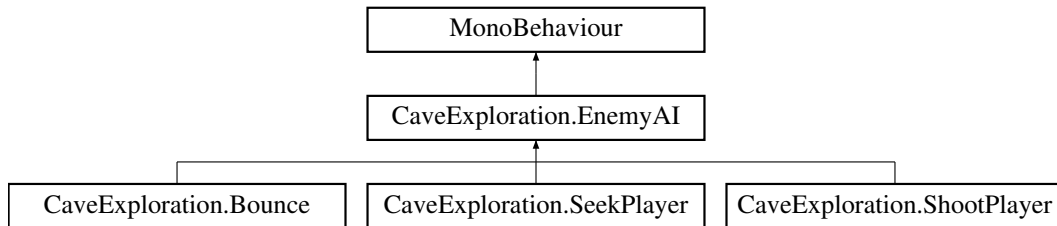
The documentation for this class was generated from the following file:

- EnemiesPlaced.cs

5.16 CaveExploration.EnemyAI Class Reference

Abstract base class for all enemy AI.

Inheritance diagram for CaveExploration.EnemyAI:



Public Member Functions

- virtual void [Update](#) ()
Update this instance using a simple state machine.

Protected Member Functions

- abstract void **Idle** ()
- abstract void **AttackPlayer** ()
- abstract void **DecideState** ()
- bool **PlayerNearby** ()

Protected Attributes

- [AIState](#) **state**
- Transform **player**

5.16.1 Detailed Description

Abstract base class for all enemy AI.

5.16.2 Member Function Documentation

5.16.2.1 virtual void CaveExploration.EnemyAI.Update () [virtual]

Update this instance using a simple state machine.

Reimplemented in [CaveExploration.SeekPlayer](#), [CaveExploration.ShootPlayer](#), and [CaveExploration.Bounce](#).

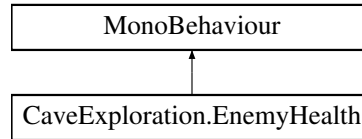
The documentation for this class was generated from the following file:

- EnemyAI.cs

5.17 CaveExploration.EnemyHealth Class Reference

Handles enemy health and death (including audio and animation).

Inheritance diagram for CaveExploration.EnemyHealth:



Public Member Functions

- void [Kill](#) (Vector3 projectilePosition, float force)
Kill the specified enemy and applys knockback force.

Public Attributes

- float [DeathDuration](#) = 0.5f
The duration of the death animation.
- bool [CountTowardsEnemyCount](#) = true
Counts towards number of enemies killed.
- AudioClip [SoundOnDeath](#)
The sound played on death.

5.17.1 Detailed Description

Handles enemy health and death (including audio and animation).

5.17.2 Member Function Documentation

5.17.2.1 void CaveExploration.EnemyHealth.Kill (Vector3 *projectilePosition*, float *force*)

Kill the specified enemy and applys knockback force.

Parameters

<i>projectilePosition</i>	Projectile position.
<i>force</i>	Force.

5.17.3 Member Data Documentation

5.17.3.1 bool CaveExploration.EnemyHealth.CountTowardsEnemyCount = true

Counts towards number of enemies killed.

5.17.3.2 float CaveExploration.EnemyHealth.DeathDuration = 0.5f

The duration of the death animation.

5.17.3.3 AudioClip CaveExploration.EnemyHealth.SoundOnDeath

The sound played on death.

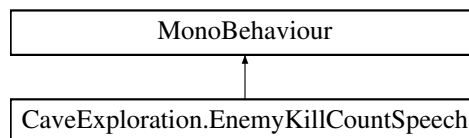
The documentation for this class was generated from the following file:

- EnemyHealth.cs

5.18 CaveExploration.EnemyKillCountSpeech Class Reference

Shows character speech on enemy killed.

Inheritance diagram for CaveExploration.EnemyKillCountSpeech:



Public Attributes

- `string[] EnemyKilledSpeech = {"Got em!", "", "1 Down", "How many are left?", "", "Good Shot"}`
Speech options on enemy killed.

5.18.1 Detailed Description

Shows character speech on enemy killed.

5.18.2 Member Data Documentation

5.18.2.1 `string[] CaveExploration.EnemyKillCountSpeech.EnemyKilledSpeech = {"Got em!", "", "1 Down", "How many are left?", "", "Good Shot"}`

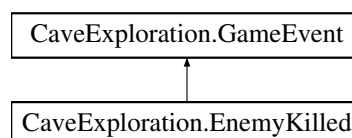
Speech options on enemy killed.

The documentation for this class was generated from the following file:

- EnemyKillCountSpeech.cs

5.19 CaveExploration.EnemyKilled Class Reference

Inheritance diagram for CaveExploration.EnemyKilled:



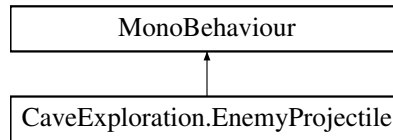
The documentation for this class was generated from the following file:

- EnemyKilled.cs

5.20 CaveExploration.EnemyProjectile Class Reference

Responsible to updating enemy projectiles. And applying damage on collision.

Inheritance diagram for CaveExploration.EnemyProjectile:



Public Member Functions

- void [ReturnToPool](#) ()
Returns to pool.

Public Attributes

- int [DamageAmount](#) = 1
The amount of damage to apply to the player.
- float [DamageForce](#) = 50f
The knockback force of the projectile.
- float [MaxTimeAlive](#) = 2f
The maximum time the projectile can be in the scene. It is added to the pool once this time has been reached.

5.20.1 Detailed Description

Responsible to updating enemy projectiles. And applying damage on collision.

5.20.2 Member Function Documentation

5.20.2.1 void CaveExploration.EnemyProjectile.ReturnToPool ()

Returns to pool.

5.20.3 Member Data Documentation

5.20.3.1 int CaveExploration.EnemyProjectile.DamageAmount = 1

The amount of damage to apply to the player.

5.20.3.2 float CaveExploration.EnemyProjectile.DamageForce = 50f

The knockback force of the projectile.

5.20.3.3 float CaveExploration.EnemyProjectile.MaxTimeAlive = 2f

The maximum time the projectile can be in the scene. It is added to the pool once this time has been reached.

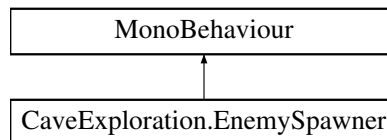
The documentation for this class was generated from the following file:

- EnemyProjectile.cs

5.21 CaveExploration.EnemySpawner Class Reference

Responsible for spawning enemies at level start.

Inheritance diagram for CaveExploration.EnemySpawner:



Public Types

- enum **EnemyPlacementType** { RandomBackgroundTile, RandomFloorTile, RandomTileWithRectSpaceAround, RandomTileWithCircleSpaceAround }

Public Member Functions

- void [OnLevelGenerated](#) ([GameEvent](#) e)
Handles the level generated event. Places the enemies based on placement type.

Public Attributes

- bool [RandomRotation](#) = false
Apply random rotation to enemy when spawned.
- GameObject [Enemy](#)
The enemy prefab to spawn.
- int [MaxEnemies](#) = 10
The maximum number of enemies of this type to spawn.
- EnemyPlacementType [PlacementType](#) = EnemyPlacementType.RandomBackgroundTile
The enemy placement type. Please see Read Me for more information on placement types.

5.21.1 Detailed Description

Responsible for spawning enemies at level start.

5.21.2 Member Function Documentation

5.21.2.1 void CaveExploration.EnemySpawner.OnLevelGenerated ([GameEvent](#) e)

Handles the level generated event. Places the enemies based on placement type.

Parameters

e	E.
---	----

5.21.3 Member Data Documentation

5.21.3.1 GameObject CaveExploration.EnemySpawner.Enemy

The enemy prefab to spawn.

5.21.3.2 int CaveExploration.EnemySpawner.MaxEnemies = 10

The maximum number of enemies of this type to spawn.

5.21.3.3 EnemyPlacementType CaveExploration.EnemySpawner.PlacementType = EnemyPlacementType.RandomBackground↔ Tile

The enemy placement type. Please see Read Me for more information on placement types.

5.21.3.4 bool CaveExploration.EnemySpawner.RandomRotation = false

Apply random rotation to enemy when spawned.

The documentation for this class was generated from the following file:

- EnemySpawner.cs

5.22 CaveExploration.Events Class Reference

Public Member Functions

- delegate void **EventDelegate**< T > (T e)
- void **AddListener**< T > (EventDelegate< T > del)
- void **RemoveListener**< T > (EventDelegate< T > del)
- void **Raise** ([GameEvent](#) e)

Properties

- static [Events](#) **instance** [[get](#)]

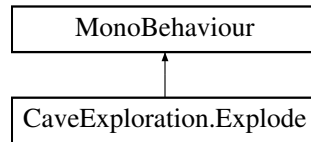
The documentation for this class was generated from the following file:

- Events.cs

5.23 CaveExploration.Explode Class Reference

Spawns an explosion on mouse click at the objects position.

Inheritance diagram for CaveExploration.Explode:



Public Member Functions

- void **Execute** ()

Public Attributes

- GameObject **Explosion**

5.23.1 Detailed Description

Spawns an explosion on mouse click at the objects position.

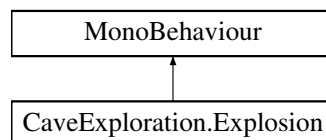
The documentation for this class was generated from the following file:

- Explode.cs

5.24 CaveExploration.Expllosion Class Reference

Attach to an explosion. Kills an enemy in explosion radius.

Inheritance diagram for CaveExploration.Expllosion:



Public Member Functions

- void [PlayExplosionSound](#) ()
Plays the explosion sound.
- void [ReturnToPool](#) ()
Returns the explosion to the object pool. Called by animation.

Public Attributes

- float [ExplosionForce](#) = 25f
The force used to knockback objects in radius.
- AudioClip [ExplosionClip](#)
The audio clip to play on explosion.

5.24.1 Detailed Description

Attach to an explosion. Kills an enemy in explosion radius.

5.24.2 Member Function Documentation

5.24.2.1 void CaveExploration.Exlosion.PlayExplosionSound ()

Plays the explosion sound.

5.24.2.2 void CaveExploration.Exlosion.ReturnToPool ()

Returns the explosion to the object pool. Called by animation.

5.24.3 Member Data Documentation

5.24.3.1 AudioClip CaveExploration.Exlosion.ExlosionClip

The audio clip to play on explosion.

5.24.3.2 float CaveExploration.Exlosion.ExlosionForce = 25f

The force used to knockback objects in radius.

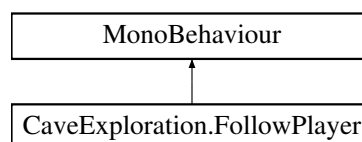
The documentation for this class was generated from the following file:

- Explosion.cs

5.25 CaveExploration.FollowPlayer Class Reference

Attach to camera. Used to follow player once spawned.

Inheritance diagram for CaveExploration.FollowPlayer:



Public Attributes

- Vector3 [Displacement](#)
The displacement from the player.

5.25.1 Detailed Description

Attach to camera. Used to follow player once spawned.

5.25.2 Member Data Documentation

5.25.2.1 Vector3 CaveExploration.FollowPlayer.Displacement

The displacement from the player.

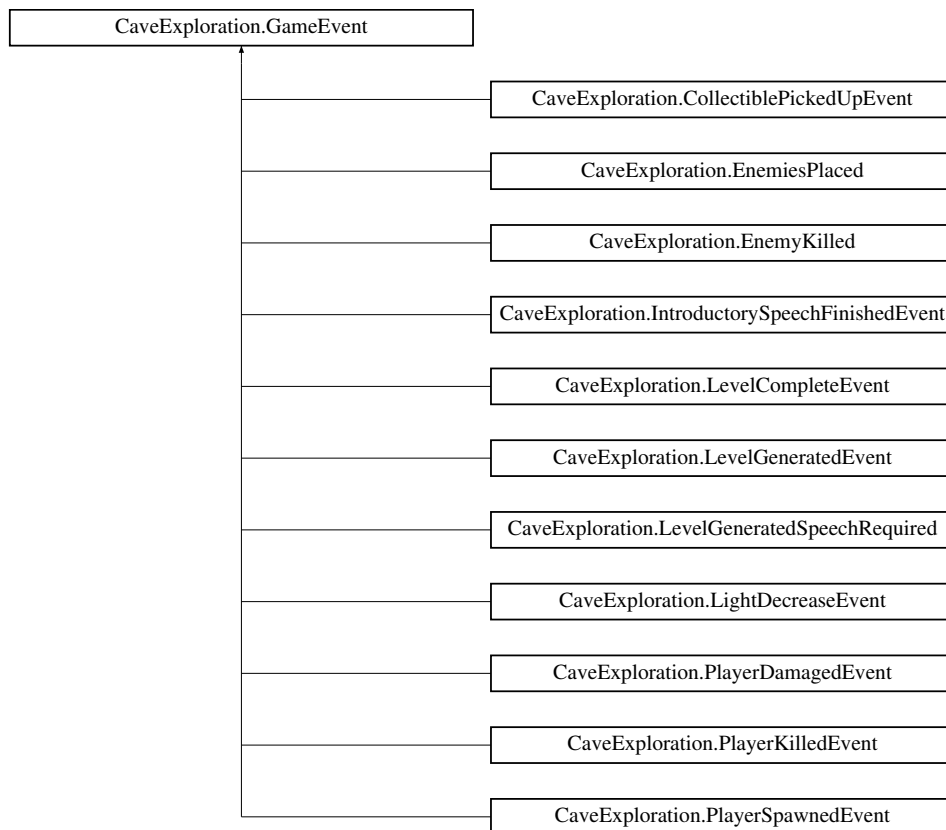
The documentation for this class was generated from the following file:

- FollowPlayer.cs

5.26 CaveExploration.GameEvent Class Reference

All game events inherit this abstract base class.

Inheritance diagram for CaveExploration.GameEvent:



5.26.1 Detailed Description

All game events inherit this abstract base class.

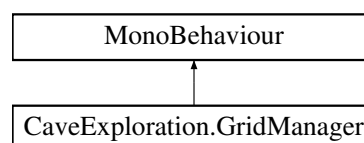
The documentation for this class was generated from the following file:

- GameEvent.cs

5.27 CaveExploration.GridManager Class Reference

Singleton. Handles level generation.

Inheritance diagram for CaveExploration.GridManager:



Public Member Functions

- void [GenerateWithSeed](#) (int seed, bool firstLevel)
Generates the environment with specified seed. If you pass same value for seed it will generate same environment.
- [Node GetFloorNodeMaxDistanceFromStartNode](#) (float maxDistance, [Node](#) startNode)
Gets the floor node at a distance less than max distance from start node.
- [Node GetFloorNodeMinDistanceFromStartNode](#) (float minDistance, [Node](#) startNode)
Gets the floor node at a distance greater than min distance from start node.
- [Node GetRandomBackgroundNode](#) ()
Gets a random background node.
- List< [Node](#) > [GetBackgroundNodes](#) ()
Returns a list of all background nodes.
- [Node GetRandomFloorNode](#) ()
Gets a random floor node.
- int [CountWallMooreNeighbours](#) (Vector2 coord)
Returns a count of the number of neighbours of the cell at the specified coord that are walls.

Public Attributes

- Rect [GridSize](#)
The size of the grid.
- int [NumberOfTransistionSteps](#) = 0
The number of transistion steps to perform when generating the level. The higher this number the more the the specified rules are applied.
- float [ChanceToBecomeWall](#) = 0.40f
The chance for a floor tile to become a wall tile when first generating the level.
- int [FloorsToWallConversion](#) = 4
If a tile has a number of neighbours higher than this then it too will be changed into a wall tile.
- int [WallsToFloorConversion](#) = 3
If a wall tile has less than this number of neighbours that are wall tiles then it is converted into a background tile.
- GameObject [Cell](#)
The tile prefab.

Properties

- [NodeList Grid](#) [get, set]
Gets or sets the grid.
- [Node StartNode](#) [get, set]
Gets or sets the start node (where the player is spawned).
- [Node EndNode](#) [get, set]
Gets or sets the end node (where the end object is spawned).
- [TexturePack TexturePack](#) [get]
Gets the current texture pack.
- static [GridManager instance](#) [get]
Gets the instance.

5.27.1 Detailed Description

Singleton. Handles level generation.

5.27.2 Member Function Documentation

5.27.2.1 `int CaveExploration.GridManager.CountWallMooreNeighbours (Vector2 coord)`

Returns a count of the number of neighbours of the cell at the specified coord that are walls.

Returns

The wall moore neighbours.

Parameters

<i>coord</i>	Coordinate.
--------------	-------------

5.27.2.2 `void CaveExploration.GridManager.GenerateWithSeed (int seed, bool firstLevel)`

Generates the environment with specified seed. If you pass same value for seed it will generate same environment.

Parameters

<i>seed</i>	Seed.
<i>firstLevel</i>	If set to <code>true</code> then player speech is started.

5.27.2.3 `List<Node> CaveExploration.GridManager.GetBackgroundNodes ()`

Returns a list of all background nodes.

Returns

The background nodes.

5.27.2.4 `Node CaveExploration.GridManager.GetFloorNodeMaxDistanceFromStartNode (float maxDistance, Node startNode)`

Gets the floor node at a distance less than max distance from start node.

Returns

The floor node.

Parameters

<i>maxDistance</i>	Maximum distance between start node and returned node.
<i>startNode</i>	Start node.

5.27.2.5 `Node CaveExploration.GridManager.GetFloorNodeMinDistanceFromStartNode (float minDistance, Node startNode)`

Gets the floor node at a distance greater than min distance from start node.

Returns

The floor node.

Parameters

<i>minDistance</i>	Minimum distance between start node and returned node.
<i>startNode</i>	Start node.

5.27.2.6 Node CaveExploration.GridManager.GetRandomBackgroundNode ()

Gets a random background node.

Returns

The random background node.

5.27.2.7 Node CaveExploration.GridManager.GetRandomFloorNode ()

Gets a random floor node.

Returns

The random floor node.

5.27.3 Member Data Documentation**5.27.3.1 GameObject CaveExploration.GridManager.Cell**

The tile prefab.

5.27.3.2 float CaveExploration.GridManager.ChanceToBecomeWall = 0.40f

The chance for a floor tile to become a wall tile when first generating the level.

5.27.3.3 int CaveExploration.GridManager.FloorsToWallConversion = 4

If a tile has a number of neighbours higher than this then it too will be changed into a wall tile.

5.27.3.4 Rect CaveExploration.GridManager.GridSize

The size of the grid.

5.27.3.5 int CaveExploration.GridManager.NumberOfTransistionSteps = 0

The number of transistion steps to perform when generating the level. The higher this number the more the the specified rules are applied.

5.27.3.6 int CaveExploration.GridManager.WallsToFloorConversion = 3

If a wall tile has less than this number of neighbours that are wall tiles then it is converted into a background tile.

5.27.4 Property Documentation

5.27.4.1 Node `CaveExploration.GridManager.EndNode` `[get]`, `[set]`

Gets or sets the end node (where the end object is spawned).

The end node.

5.27.4.2 `NodeList` `CaveExploration.GridManager.Grid` `[get]`, `[set]`

Gets or sets the grid.

The grid.

5.27.4.3 `GridManager` `CaveExploration.GridManager.instance` `[static]`, `[get]`

Gets the instance.

The instance.

5.27.4.4 Node `CaveExploration.GridManager.StartNode` `[get]`, `[set]`

Gets or sets the start node (where the player is spawned).

The start node.

5.27.4.5 `TexturePack` `CaveExploration.GridManager.TexturePack` `[get]`

Gets the current texture pack.

The texture pack.

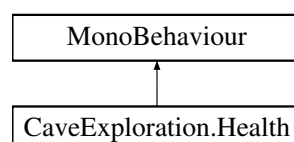
The documentation for this class was generated from the following file:

- `GridManager.cs`

5.28 `CaveExploration.Health` Class Reference

The players health.

Inheritance diagram for `CaveExploration.Health`:



Public Member Functions

- void `OnDamage` (`PlayerDamagedEvent` e)

Handles the `PlayerDamagedEvent`. Applies damage, shows speech, and kills player (if health less than zero).

Public Attributes

- int `MaxHealth` = 5
The players maximum health.
- string[] `HurtSpeech`
Speech options to show when the player is hurt.
- string[] `LowHealthSpeech`
Speech options to show when the player has low health.
- int `LowHealthThreshold` = 2
The low health threshold. Speech is shown when the players health is below the threshold.
- Color `HurtColour` = Color.red
The colour to change the players sprite to when damaged.
- float `DamageAnimationTime` = 0.5f
The damage animation time.

5.28.1 Detailed Description

The players health.

5.28.2 Member Function Documentation

5.28.2.1 void CaveExploration.Health.OnDamage (PlayerDamagedEvent e)

Handles the `PlayerDamagedEvent`. Applies damage, shows speech, and kills player (if health less than zero).

Parameters

<code>e</code>	E.
----------------	----

5.28.3 Member Data Documentation

5.28.3.1 float CaveExploration.Health.DamageAnimationTime = 0.5f

The damage animation time.

5.28.3.2 Color CaveExploration.Health.HurtColour = Color.red

The colour to change the players sprite to when damaged.

5.28.3.3 string [] CaveExploration.Health.HurtSpeech

Speech options to show when the player is hurt.

5.28.3.4 string [] CaveExploration.Health.LowHealthSpeech

Speech options to show when the player has low health.

5.28.3.5 int CaveExploration.Health.LowHealthThreshold = 2

The low health threshold. Speech is shown when the players health is below the threshold.

5.28.3.6 `int CaveExploration.Health.MaxHealth = 5`

The players maximum health.

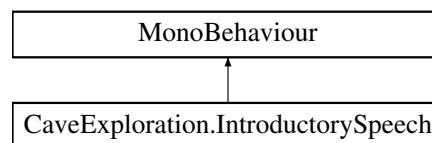
The documentation for this class was generated from the following file:

- Health.cs

5.29 CaveExploration.IntroductorySpeech Class Reference

Introductory speech. For example, can be used to introduce a level, the game, controls etc.

Inheritance diagram for CaveExploration.IntroductorySpeech:



Public Attributes

- `string[] Speech`
Speech options. These options are shown one at a time.

Properties

- `bool InProgress` `[get]`
Gets a value indicating whether this `CaveExploration.IntroductorySpeech` is in progress.
- `static IntroductorySpeech instance` `[get]`
Gets the instance.

5.29.1 Detailed Description

Introductory speech. For example, can be used to introduce a level, the game, controls etc.

5.29.2 Member Data Documentation

5.29.2.1 `string[] CaveExploration.IntroductorySpeech.Speech`

Speech options. These options are shown one at a time.

5.29.3 Property Documentation

5.29.3.1 `bool CaveExploration.IntroductorySpeech.InProgress` `[get]`

Gets a value indicating whether this `CaveExploration.IntroductorySpeech` is in progress.

`true` if in progress; otherwise, `false`.

5.29.3.2 IntroductorySpeech CaveExploration.IntroductorySpeech.instance [static], [get]

Gets the instance.

The instance.

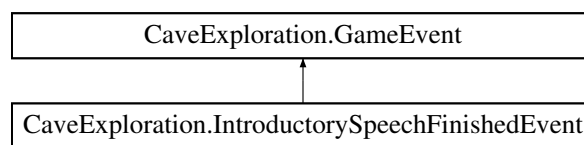
The documentation for this class was generated from the following file:

- IntroductorySpeech.cs

5.30 CaveExploration.IntroductorySpeechFinishedEvent Class Reference

Raised when the players intro speech has finished.

Inheritance diagram for CaveExploration.IntroductorySpeechFinishedEvent:



5.30.1 Detailed Description

Raised when the players intro speech has finished.

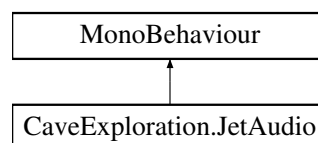
The documentation for this class was generated from the following file:

- IntroductorySpeechFinishedEvent.cs

5.31 CaveExploration.JetAudio Class Reference

Handles playing audio on jet pack use.

Inheritance diagram for CaveExploration.JetAudio:



Public Attributes

- AudioClip [JetAudioClip](#)
The audio clip to play when the jetpack is in use.
- float [Volume](#) = 0.2f
The volume to play the jet pack audio clip.

5.31.1 Detailed Description

Handles playing audio on jet pack use.

5.31.2 Member Data Documentation

5.31.2.1 AudioClip CaveExploration.JetAudio.JetAudioClip

The audio clip to play when the jetpack is in use.

5.31.2.2 float CaveExploration.JetAudio.Volume = 0.2f

The volume to play the jet pack audio clip.

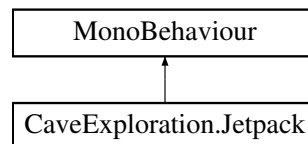
The documentation for this class was generated from the following file:

- JetAudio.cs

5.32 CaveExploration.Jetpack Class Reference

Responsible for controlling the players jetpack.

Inheritance diagram for CaveExploration.Jetpack:



Public Member Functions

- bool [FuelRecharging](#) ()
Gets or sets a value indicating whether this [CaveExploration.Jetpack](#) is currently recharging the fuel.

Public Attributes

- float [JetForce](#) = 30f
The force applied when jet pack is active.
- float [MaxForce](#) = 1.5f
The maximum force applied to the player.
- float [JetFuel](#) = 600f
The maximum jet fuel.
- float [JetFuelWaste](#) = 2f
The jet fuel waste per second.

Properties

- bool [UsingJet](#) [get, set]
Gets or sets a value indicating whether this [CaveExploration.Jetpack](#) is using the jetpack.
- bool [CanJet](#) [get, set]
Gets or sets a value indicating whether this instance can use the jetpack.

5.32.1 Detailed Description

Responsible for controlling the players jetpack.

5.32.2 Member Function Documentation

5.32.2.1 bool CaveExploration.Jetpack.FuelRecharging ()

Gets or sets a value indicating whether this [CaveExploration.Jetpack](#) is currently recharging the fuel.

Returns

`true`, if recharging was fueled, `false` otherwise.

5.32.3 Member Data Documentation

5.32.3.1 float CaveExploration.Jetpack.JetForce = 30f

The force applied when jet pack is active.

5.32.3.2 float CaveExploration.Jetpack.JetFuel = 600f

The maximum jet fuel.

5.32.3.3 float CaveExploration.Jetpack.JetFuelWaste = 2f

The jet fuel waste per second.

5.32.3.4 float CaveExploration.Jetpack.MaxForce = 1.5f

The maximum force applied to the player.

5.32.4 Property Documentation

5.32.4.1 bool CaveExploration.Jetpack.CanJet [get], [set]

Gets or sets a value indicating whether this instance can use the jetpack.

`true` if this instance can jet; otherwise, `false`.

5.32.4.2 bool CaveExploration.Jetpack.UsingJet [get], [set]

Gets or sets a value indicating whether this [CaveExploration.Jetpack](#) is using the jetpack.

`true` if using jetpack; otherwise, `false`.

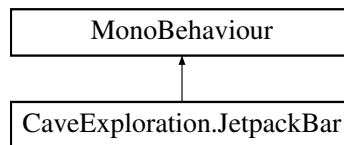
The documentation for this class was generated from the following file:

- Jetpack.cs

5.33 CaveExploration.JetpackBar Class Reference

Visual representation of the jetpack fuel.

Inheritance diagram for CaveExploration.JetpackBar:



Public Member Functions

- void [Disable](#) ()
Disable the jetpack fuel renderer.
- void [Enable](#) ()
Enables the jetpack fuel renderer.
- void [UpdateColour](#) (float currentFuel)
Updates the bars colour based on the current fuel amount.
- void [UpdateLocalScaleY](#) (float scaleY)
Updates the bars y scale based on current fuel amount.

Public Attributes

- Color [ColourWhenFullFuel](#)
The colour when jetpack fuel is full.
- Color [ColourWhenEmptyFuel](#)
The colour when jetpack fuel is empty.

5.33.1 Detailed Description

Visual representation of the jetpack fuel.

5.33.2 Member Function Documentation

5.33.2.1 void CaveExploration.JetpackBar.Disable ()

Disable the jetpack fuel renderer.

5.33.2.2 void CaveExploration.JetpackBar.Enable ()

Enables the jetpack fuel renderer.

5.33.2.3 void CaveExploration.JetpackBar.UpdateColour (float *currentFuel*)

Updates the bars colour based on the current fuel amount.

Parameters

<i>currentFuel</i>	Current fuel.
--------------------	---------------

5.33.2.4 void CaveExploration.JetpackBar.UpdateLocalScaleY (float *scaleY*)

Updates the bars y scale based on current fuel amount.

Parameters

<i>scaleY</i>	Scale y.
---------------	----------

5.33.3 Member Data Documentation

5.33.3.1 Color CaveExploration.JetpackBar.ColourWhenEmptyFuel

The colour when jetpack fuel is empty.

5.33.3.2 Color CaveExploration.JetpackBar.ColourWhenFullFuel

The colour when jetpack fuel is full.

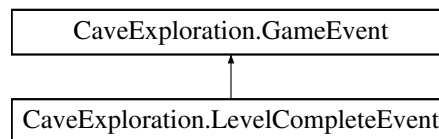
The documentation for this class was generated from the following file:

- JetpackBar.cs

5.34 CaveExploration.LevelCompleteEvent Class Reference

Raised when the player completes a level.

Inheritance diagram for CaveExploration.LevelCompleteEvent:



Public Member Functions

- [LevelCompleteEvent](#) (int nextSeed)
Initializes a new instance of the [CaveExploration.LevelCompleteEvent](#) class.

Properties

- int [NextSeed](#) [get]
Gets the seed to be used when generating the next level.

5.34.1 Detailed Description

Raised when the player completes a level.

5.34.2 Constructor & Destructor Documentation

5.34.2.1 CaveExploration.LevelCompleteEvent.LevelCompleteEvent (int nextSeed)

Initializes a new instance of the [CaveExploration.LevelCompleteEvent](#) class.

Parameters

<i>nextSeed</i>	Next seed.
-----------------	------------

5.34.3 Property Documentation

5.34.3.1 `int CaveExploration.LevelCompleteEvent.NextSeed` [get]

Gets the seed to be used when generating the next level.

The next seed.

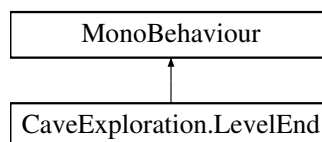
The documentation for this class was generated from the following file:

- LevelCompleteEvent.cs

5.35 CaveExploration.LevelEnd Class Reference

Added to the exit block for the current level. When the character enters the blocks collider LevelComplete is set to true. This is queried in the Director Class.

Inheritance diagram for CaveExploration.LevelEnd:



5.35.1 Detailed Description

Added to the exit block for the current level. When the character enters the blocks collider LevelComplete is set to true. This is queried in the Director Class.

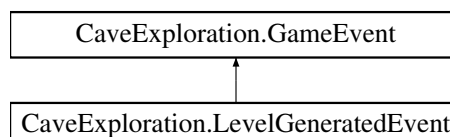
The documentation for this class was generated from the following file:

- LevelEnd.cs

5.36 CaveExploration.LevelGeneratedEvent Class Reference

Raised when a level has finished being generated.

Inheritance diagram for CaveExploration.LevelGeneratedEvent:



5.36.1 Detailed Description

Raised when a level has finished being generated.

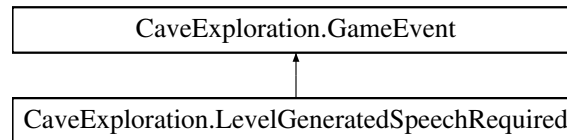
The documentation for this class was generated from the following file:

- LevelGeneratedEvent.cs

5.37 CaveExploration.LevelGeneratedSpeechRequired Class Reference

Raised when a level has been generated and player speech is required.

Inheritance diagram for CaveExploration.LevelGeneratedSpeechRequired:



5.37.1 Detailed Description

Raised when a level has been generated and player speech is required.

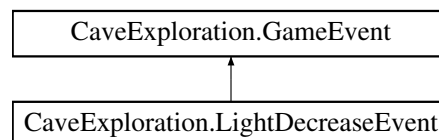
The documentation for this class was generated from the following file:

- LevelGeneratedSpeechRequired.cs

5.38 CaveExploration.LightDecreaseEvent Class Reference

Raised when the players light should be decreased e.g. on damage.

Inheritance diagram for CaveExploration.LightDecreaseEvent:



Public Member Functions

- [LightDecreaseEvent](#) (float?lightAmount=null, float?radiusAmount=null)
Initializes a new instance of the [CaveExploration.LightDecreaseEvent](#) class.

Properties

- float [LightAmount](#) [get]
The amount to decrease the players light.
- float [RadiusAmount](#) [get]
The amount to decrease the players radius.

5.38.1 Detailed Description

Raised when the players light should be decreased e.g. on damage.

5.38.2 Constructor & Destructor Documentation

5.38.2.1 CaveExploration.LightDecreaseEvent.LightDecreaseEvent (float? *lightAmount* = null, float? *radiusAmount* = null)

Initializes a new instance of the [CaveExploration.LightDecreaseEvent](#) class.

Parameters

<i>lightAmount</i>	Amount to decrease the players light intensity.
<i>radiusAmount</i>	Amount to decrease the players light radius.

5.38.3 Property Documentation

5.38.3.1 float CaveExploration.LightDecreaseEvent.LightAmount [get]

The amount to decrease the players light.

The light amount.

5.38.3.2 float CaveExploration.LightDecreaseEvent.RadiusAmount [get]

The amount to decrease the players radius.

The radius amount.

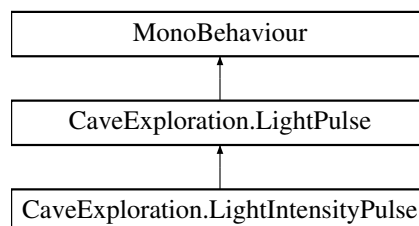
The documentation for this class was generated from the following file:

- LightDecreaseEvent.cs

5.39 CaveExploration.LightIntensityPulse Class Reference

Pulses a lights intensity between two values over time.

Inheritance diagram for CaveExploration.LightIntensityPulse:



Additional Inherited Members

5.39.1 Detailed Description

Pulses a lights intensity between two values over time.

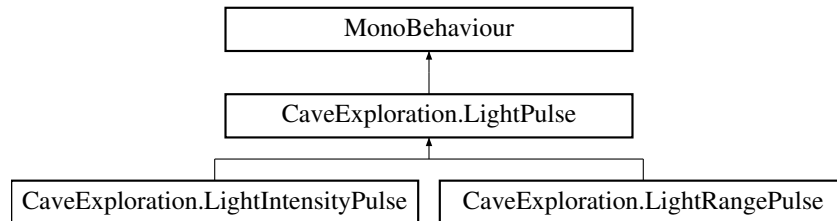
The documentation for this class was generated from the following file:

- LightIntensityPulse.cs

5.40 CaveExploration.LightPulse Class Reference

Used to pulse a light variable.

Inheritance diagram for CaveExploration.LightPulse:



Public Attributes

- float `Speed` = 10f
The speed to pulse variable.
- float `Minimum` = 1f
The minimum value.
- float `Maximum` = 8f
The maximum value.

Protected Attributes

- `Light_light`

5.40.1 Detailed Description

Used to pulse a light variable.

5.40.2 Member Data Documentation

5.40.2.1 float CaveExploration.LightPulse.Maximum = 8f

The maximum value.

5.40.2.2 float CaveExploration.LightPulse.Minimum = 1f

The minimum value.

5.40.2.3 float CaveExploration.LightPulse.Speed = 10f

The speed to pulse variable.

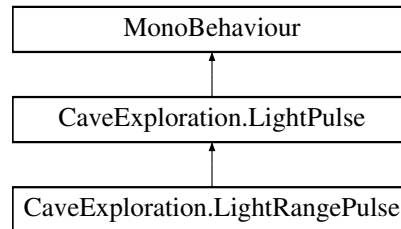
The documentation for this class was generated from the following file:

- `LightPulse.cs`

5.41 CaveExploration.LightRangePulse Class Reference

Pulses a light range between two values over time.

Inheritance diagram for CaveExploration.LightRangePulse:



Additional Inherited Members

5.41.1 Detailed Description

Pulses a light range between two values over time.

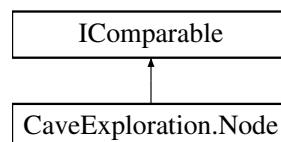
The documentation for this class was generated from the following file:

- LightRangePulse.cs

5.42 CaveExploration.Node Class Reference

Logical representation of a block. Holds the node type (e.g. wall, floor etc), it's coordinates (a pointer to the node in a 2d array - not it's position on screen), it's position on screen and path finding variables.

Inheritance diagram for CaveExploration.Node:



Public Member Functions

- [Node](#) ()
Initializes a new instance of the [CaveExploration.Node](#) class.
- [Node](#) (Vector2 coordinates)
Initializes a new instance of the [CaveExploration.Node](#) class.
- [Node](#) (Vector2 coordinates, [NodeType](#) state)
Initializes a new instance of the [CaveExploration.Node](#) class.
- [Node](#) (Vector2 coordinates, [NodeType](#) state, bool isOccupied)
Initializes a new instance of the [CaveExploration.Node](#) class.
- float [GetFScore](#) ()
Total score. Returns GScore + HScore
- int [CompareTo](#) (object obj)
Used to compare the movement cost of two nodes for pathfinding.

Properties

- [NodeType NodeState](#) [get, set]
Gets or sets the state of the node.
- [Vector2 Coordinates](#) [get, set]
Gets or sets the node coordinates (i.e. location within the 2d array as opposed to actual position in world).
- [Vector2 Position](#) [get, set]
Gets or sets the nodes position in the scene.
- [bool IsOccupied](#) [get, set]
Gets or sets a value indicating whether this node is occupied by an enemy/player.
- [float GScore](#) [get, set]
The cost to move into this node used for path finding.
- [float HScore](#) [get, set]
Estimated cost to move from this node to end node.
- [Node Parent](#) [get, set]
Used when traversing a path.
- [bool IsObstacle](#) [get]
Gets a value indicating whether this instance is obstacle and cannot be traversed.

5.42.1 Detailed Description

Logical representation of a block. Holds the node type (e.g. wall, floor etc), it's coordinates (a pointer to the node in a 2d array - not it's position on screen), it's position on screen and path finding variables.

5.42.2 Constructor & Destructor Documentation

5.42.2.1 CaveExploration.Node.Node ()

Initializes a new instance of the [CaveExploration.Node](#) class.

5.42.2.2 CaveExploration.Node.Node (Vector2 coordinates)

Initializes a new instance of the [CaveExploration.Node](#) class.

Parameters

<i>coordinates</i>	Coordinates.
--------------------	--------------

5.42.2.3 CaveExploration.Node.Node (Vector2 coordinates, NodeType state)

Initializes a new instance of the [CaveExploration.Node](#) class.

Parameters

<i>coordinates</i>	Coordinates.
<i>state</i>	State.

5.42.2.4 CaveExploration.Node.Node (Vector2 coordinates, NodeType state, bool isOccupied)

Initializes a new instance of the [CaveExploration.Node](#) class.

Parameters

<i>coordinates</i>	Coordinates.
<i>state</i>	State.
<i>isOccupied</i>	If set to <code>true</code> is occupied by a player or enemy.

5.42.3 Member Function Documentation

5.42.3.1 `int CaveExploration.Node.CompareTo (object obj)`

Used to compare the movement cost of two nodes for pathfinding.

Returns

Relative position.

Parameters

<i>obj</i>	Object.
------------	---------

5.42.3.2 `float CaveExploration.Node.GetFScore ()`

Total score. Returns GScore + HScore

5.42.4 Property Documentation

5.42.4.1 `Vector2 CaveExploration.Node.Coordinates` `[get], [set]`

Gets or sets the node coordinates (i.e. location within the 2d array as opposed to actual position in world).

The coordinates.

5.42.4.2 `float CaveExploration.Node.GScore` `[get], [set]`

The cost to move into this node used for path finding.

The G score.

5.42.4.3 `float CaveExploration.Node.HScore` `[get], [set]`

Estimated cost to move from this node to end node.

The H score.

5.42.4.4 `bool CaveExploration.Node.IsObstacle` `[get]`

Gets a value indicating whether this instance is obstacle and cannot be traversed.

`true` if this instance is obstacle; otherwise, `false`.

5.42.4.5 `bool CaveExploration.Node.IsOccupied` `[get], [set]`

Gets or sets a value indicating whether this node is occupied by an enemy/player.

`true` if this instance is occupied; otherwise, `false`.

5.42.4.6 NodeType CaveExploration.Node.NodeState [get], [set]

Gets or sets the state of the node.

The state of the node.

5.42.4.7 Node CaveExploration.Node.Parent [get], [set]

Used when traversing a path.

The parent.

5.42.4.8 Vector2 CaveExploration.Node.Position [get], [set]

Gets or sets the nodes position in the scene.

The position.

The documentation for this class was generated from the following file:

- Node.cs

5.43 CaveExploration.NodeCluster Class Reference

Represents a group of neighbouring floor nodes i.e. a cavern in the environment.

Public Member Functions

- [NodeCluster](#) ()
Initializes a new instance of the [CaveExploration.NodeCluster](#) class.

Properties

- List< [Node](#) > [Nodes](#) [get, set]
Gets or sets the nodes contained by this cluster.

5.43.1 Detailed Description

Represents a group of neighbouring floor nodes i.e. a cavern in the environment.

5.43.2 Constructor & Destructor Documentation

5.43.2.1 CaveExploration.NodeCluster.NodeCluster ()

Initializes a new instance of the [CaveExploration.NodeCluster](#) class.

5.43.3 Property Documentation

5.43.3.1 List<Node> CaveExploration.NodeCluster.Nodes [get], [set]

Gets or sets the nodes contained by this cluster.

The nodes.

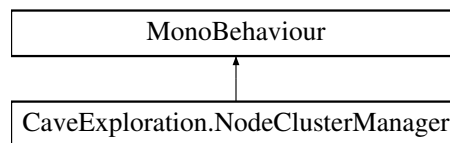
The documentation for this class was generated from the following file:

- NodeCluster.cs

5.44 CaveExploration.NodeClusterManager Class Reference

Singleton. Identifies, manages and holds reference to the different node clusters. Can connect un-connected clusters using path finding.

Inheritance diagram for CaveExploration.NodeClusterManager:



Public Member Functions

- void [IdentifyClusters](#) ([NodeList](#) nodes, Rect size)
Identifies the clusters. Uses a flood-fill algorithm to identify neighbouring floor tiles.
- void [ConvertDisconnectedClustersToNodeType](#) ([NodeType](#) nodeType)
Converts the type of the disconnected nodes. Can be used to convert all disconnect nodes to walls.
- void [ConnectClusters](#) ()
Uses A algorithm to find a path from disconnected clusters to main cluster and convert nodes on that path to floors.*
- int [CalculateMainCluster](#) ()
Iterates through each cluster and returns the index of the cluster with the largest size.

Properties

- List< [NodeCluster](#) > [Clusters](#) [get, set]
Gets or sets the clusters.
- [NodeCluster MainCluster](#) [get]
Gets the main cluster. The main cluster contains the largest number of nodes.

5.44.1 Detailed Description

Singleton. Identifies, manages and holds reference to the different node clusters. Can connect un-connected clusters using path finding.

5.44.2 Member Function Documentation

5.44.2.1 int CaveExploration.NodeClusterManager.CalculateMainCluster ()

Iterates through each cluster and returns the index of the cluster with the largest size.

Returns

The main cluster index.

5.44.2.2 void CaveExploration.NodeClusterManager.ConnectClusters ()

Uses A* algorithm to find a path from disconnected clusters to main cluster and convert nodes on that path to floors.

5.44.2.3 void CaveExploration.NodeClusterManager.ConvertDisconnectedClustersToNodeType (NodeType nodeType)

Converts the type of the disconnected nodes. Can be used to convert all disconnect nodes to walls.

Parameters

<i>nodeType</i>	The type of node to convert to.
-----------------	---------------------------------

5.44.2.4 void CaveExploration.NodeClusterManager.IdentifyClusters (NodeList nodes, Rect size)

Identifies the clusters. Uses a flood-fill algorithm to identify neighbouring floor tiles.

Parameters

<i>nodes</i>	A list of all active nodes.
<i>size</i>	The size of the level.

5.44.3 Property Documentation

5.44.3.1 List<NodeCluster> CaveExploration.NodeClusterManager.Clusters [get], [set]

Gets or sets the clusters.

The clusters.

5.44.3.2 NodeCluster CaveExploration.NodeClusterManager.MainCluster [get]

Gets the main cluster. The main cluster contains the largest number of nodes.

The main cluster.

The documentation for this class was generated from the following file:

- NodeClusterManager.cs

5.45 CaveExploration.NodeList Class Reference

Encapsulates 2D array of all active nodes in game. Provides helper methods such as Add and Contains that mimic there list counterparts.

Public Member Functions

- **NodeList** (Rect gridSize)
- bool **Contains** (Node node)
Contains the specified node.
- bool **IsValidCoordinate** (Vector2 coord)
Determines whether the specified coords are valid.
- bool **ContainsNodeTypeAtPosition** (Vector2 coord, NodeType nodeType)
Determines if the node at position is of the specified type.
- void **Add** (Node node)

- Add the specified node to array.*
- **Node First** ()
*Returns first **Node**.*
- void **Replace** (**Node** origNode, **Node** newNode)
Replace the specified origNode with newNode.
- void **Replace** (Vector2 origNodeCoord, **Node** newNode)
Replace the node with origNodeCoord, with the node newNode.
- List< **Node** > **GetAdjacentNodes** (Vector2 cellCoordinate, bool includeObstacles)
Returns a list of adjacent nodes with or without obstacles.
- **Node GetNodeFromPosition** (Vector2 position)
Returns a node for a specified on-screen position.
- **Node GetNodeFromGridCoordinate** (Vector2 coord)
Returns a node with a specified grid coordinate.

Properties

- int **Count** [get]
Gets the number of nodes contained in this list.

5.45.1 Detailed Description

Encapsulates 2D array of all active nodes in game. Provides helper methods such as Add and Contains that mimic their list counterparts.

5.45.2 Member Function Documentation

5.45.2.1 void CaveExploration.NodeList.Add (**Node** node)

Add the specified node to array.

Parameters

<i>node</i>	Node .
-------------	---------------

5.45.2.2 bool CaveExploration.NodeList.Contains (**Node** node)

Contains the specified node.

Parameters

<i>node</i>	Node .
-------------	---------------

5.45.2.3 bool CaveExploration.NodeList.ContainsNodeTypeAtPosition (Vector2 coord, **NodeType** nodeType)

Determines if the node at position is of the specified type.

Returns

`true`, if node at position is of specified type, `false` otherwise.

Parameters

<i>coord</i>	Coordinate.
<i>nodeType</i>	Node type.

5.45.2.4 Node CaveExploration.NodeList.First ()

Returns first [Node](#).

5.45.2.5 List<Node> CaveExploration.NodeList.GetAdjacentNodes (Vector2 cellCoordinate, bool includeObstacles)

Returns a list of adjacent nodes with or without obstacles.

Returns

The adjacent nodes.

Parameters

<i>cellCoordinate</i>	Cell coordinate or original node.
<i>includeObstacles</i>	If set to <code>true</code> include obstacles.

5.45.2.6 Node CaveExploration.NodeList.GetNodeFromGridCoordinate (Vector2 coord)

Returns a node with a specified grid coordinate.

Returns

The node from grid coordinate.

Parameters

<i>coord</i>	Coordinate.
--------------	-------------

5.45.2.7 Node CaveExploration.NodeList.GetNodeFromPosition (Vector2 position)

Returns a node for a specified on-screen position.

Returns

The node from position.

Parameters

<i>position</i>	Position.
-----------------	-----------

5.45.2.8 bool CaveExploration.NodeList.IsValidCoordinate (Vector2 coord)

Determines whether the specified coords are valid.

Returns

`true` if the specified coordinate are valid; otherwise, `false`.

Parameters

<i>coord</i>	Coordinate.
--------------	-------------

5.45.2.9 void CaveExploration.NodeList.Replace (Node origNode, Node newNode)

Replace the specified origNode with newNode.

Parameters

<i>origNode</i>	Original node.
<i>newNode</i>	New node.

5.45.2.10 void CaveExploration.NodeList.Replace (Vector2 origNodeCoord, Node newNode)

Replace the node with origNodeCoord, with the node newNode.

Parameters

<i>origNodeCoord</i>	Original node coordinate.
<i>newNode</i>	New node.

5.45.3 Property Documentation

5.45.3.1 int CaveExploration.NodeList.Count [get]

Gets the number of nodes contained in this list.

The count.

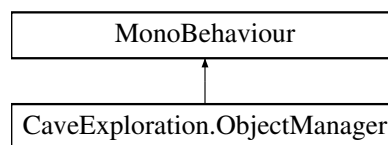
The documentation for this class was generated from the following file:

- NodeList.cs

5.46 CaveExploration.ObjectManager Class Reference

Manages adding and retrieving objects from the object pool.

Inheritance diagram for CaveExploration.ObjectManager:



Public Member Functions

- GameObject [GetObject](#) (string prefabName, Vector2 position, Quaternion rotation)
Gets the object from the object pool. Moves to position with rotation and enables object.
- GameObject [GetObject](#) (string prefabName, Vector2 position)
Gets the object from the object pool. Moves to position and enables object.
- GameObject [GetObject](#) (string prefabName, Vector2 position, bool onlyPooledObjects)
Gets the object from the object pool. Moves to position and enables object.

- void [RemoveObject](#) (GameObject obj)
Removes the object from the scene and adds it to the pool.
- void [RemoveObjects](#) ()
Removes all spawned objects and clears spawn list.

Protected Attributes

- List< GameObject > **objects** = new List<GameObject> ()

Properties

- static [ObjectManager instance](#) [get]
Gets the instance.

5.46.1 Detailed Description

Manages adding and retrieving objects from the object pool.

5.46.2 Member Function Documentation

5.46.2.1 GameObject CaveExploration.ObjectManager.GetObject (string *prefabName*, Vector2 *position*, Quaternion *rotation*)

Gets the object from the object pool. Moves to position with rotation and enables object.

Returns

The object.

Parameters

<i>prefabName</i>	Prefab name.
<i>position</i>	Position.
<i>rotation</i>	Rotation.

5.46.2.2 GameObject CaveExploration.ObjectManager.GetObject (string *prefabName*, Vector2 *position*)

Gets the object from the object pool. Moves to position and enables object.

Returns

The object.

Parameters

<i>prefabName</i>	Prefab name.
<i>position</i>	Position.

5.46.2.3 GameObject CaveExploration.ObjectManager.GetObject (string *prefabName*, Vector2 *position*, bool *onlyPooledObjects*)

Gets the object from the object pool. Moves to position and enables object.

Returns

The object.

Parameters

<i>prefabName</i>	Prefab name.
<i>position</i>	Position.
<i>onlyPooledObjects</i>	If set to <code>true</code> only pooled objects will be returned. If no object is found in pool then null is returned.

5.46.2.4 void CaveExploration.ObjectManager.RemoveObject (GameObject obj)

Removes the object from the scene and adds it to the pool.

Parameters

<i>obj</i>	Object.
------------	---------

5.46.2.5 void CaveExploration.ObjectManager.RemoveObjects ()

Removes all spawned objects and clears spawn list.

5.46.3 Property Documentation**5.46.3.1 ObjectManager CaveExploration.ObjectManager.instance [static],[get]**

Gets the instance.

The instance.

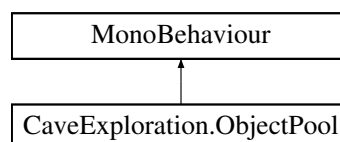
The documentation for this class was generated from the following file:

- ObjectManager.cs

5.47 CaveExploration.ObjectPool Class Reference

Object pool system.

Inheritance diagram for CaveExploration.ObjectPool:

**Public Member Functions**

- GameObject [GetObjectForType](#) (string objectType, bool onlyPooled)
Gets a new object for the name type provided. If no object type exists or if onlypooled is true and there is no objects of that type in the pool then null will be returned.
- void [PoolObject](#) (GameObject obj)
Pools the object specified. Will not be pooled if there is no prefab of that type.

Public Attributes

- `GameObject[]` [objectPrefabs](#)
The object prefabs which the pool can handle.
- `List< GameObject >[]` [pooledObjects](#)
The pooled objects currently available.
- `int[]` [amountToBuffer](#)
The amount of objects of each type to buffer.
- `int` **defaultBufferAmount** = 3

Protected Attributes

- `GameObject` [containerObject](#)
The container object that we will keep unused pooled objects so we dont clog up the editor with objects.

5.47.1 Detailed Description

Object pool system.

5.47.2 Member Function Documentation

5.47.2.1 `GameObject` CaveExploration.ObjectPool.GetObjectForType (`string objectType`, `bool onlyPooled`)

Gets a new object for the name type provided. If no object type exists or if onlypooled is true and there is no objects of that type in the pool then null will be returned.

Returns

The object for type.

Parameters

<i>objectType</i>	Object type.
<i>onlyPooled</i>	If true, it will only return an object if there is one currently pooled.

5.47.2.2 `void` CaveExploration.ObjectPool.PoolObject (`GameObject obj`)

Pools the object specified. Will not be pooled if there is no prefab of that type.

Parameters

<i>obj</i>	Object to be pooled.
------------	----------------------

5.47.3 Member Data Documentation

5.47.3.1 `int []` CaveExploration.ObjectPool.amountToBuffer

The amount of objects of each type to buffer.

5.47.3.2 `GameObject` CaveExploration.ObjectPool.containerObject [protected]

The container object that we will keep unused pooled objects so we dont clog up the editor with objects.

5.47.3.3 `GameObject [] CaveExploration.ObjectPool.objectPrefabs`

The object prefabs which the pool can handle.

5.47.3.4 `List<GameObject> [] CaveExploration.ObjectPool.pooledObjects`

The pooled objects currently available.

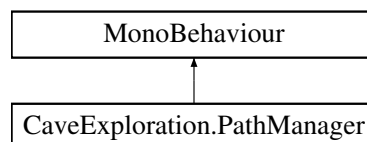
The documentation for this class was generated from the following file:

- `ObjectPool.cs`

5.48 `CaveExploration.PathManager` Class Reference

Creates a path between two nodes.

Inheritance diagram for `CaveExploration.PathManager`:



Public Member Functions

- `List< Node > GetShortestPath (Node orig, Node dest, float wallMovementCost, bool includeObstacles)`
Gets the shortest path using A.*

5.48.1 Detailed Description

Creates a path between two nodes.

5.48.2 Member Function Documentation

5.48.2.1 `List<Node> CaveExploration.PathManager.GetShortestPath (Node orig, Node dest, float wallMovementCost, bool includeObstacles)`

Gets the shortest path using A*.

Returns

A list of nodes between start and end that represents the shortest path.

Parameters

<i>orig</i>	Original cell.
<i>dest</i>	Destination cell.
<i>wallMovementCost</i>	Wall movement cost.

<code>includeObstacles</code>	If set to <code>true</code> include obstacles.
-------------------------------	--

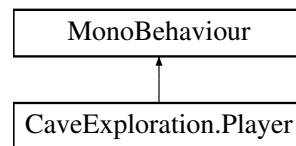
The documentation for this class was generated from the following file:

- PathManager.cs

5.49 CaveExploration.Player Class Reference

Handles player movement.

Inheritance diagram for CaveExploration.Player:



Public Attributes

- bool `IsFacingRight`
Determines whether the player is facing right. Used to face the sprite in the movement direction.
- float `MaxWalkSpeed` = 2f
The maximum walk speed.
- float `MaxRunSpeed` = 4f
The maximum run speed.
- float `JumpForce` = 500f
The jump force.

Properties

- bool `IsDead` [get, set]
Gets or sets a value indicating whether this instance is dead.
- bool `HasJumped` [get, set]
Gets or sets a value indicating whether this instance has jumped. Used for double jumping.
- float `MoveSpeed` [get]
Gets the current movement speed.

5.49.1 Detailed Description

Handles player movement.

5.49.2 Member Data Documentation

5.49.2.1 bool CaveExploration.Player.IsFacingRight

Determines whether the player is facing right. Used to face the sprite in the movement direction.

5.49.2.2 float CaveExploration.Player.JumpForce = 500f

The jump force.

5.49.2.3 float CaveExploration.Player.MaxRunSpeed = 4f

The maximum run speed.

5.49.2.4 float CaveExploration.Player.MaxWalkSpeed = 2f

The maximum walk speed.

5.49.3 Property Documentation

5.49.3.1 bool CaveExploration.Player.HasJumped [get], [set]

Gets or sets a value indicating whether this instance has jumped. Used for double jumping.

`true` if this instance has jumped; otherwise, `false`.

5.49.3.2 bool CaveExploration.Player.IsDead [get], [set]

Gets or sets a value indicating whether this instance is dead.

`true` if this instance is dead; otherwise, `false`.

5.49.3.3 float CaveExploration.Player.MoveSpeed [get]

Gets the current movement speed.

The move speed.

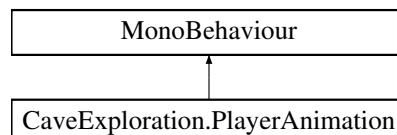
The documentation for this class was generated from the following file:

- Player.cs

5.50 CaveExploration.PlayerAnimation Class Reference

Responsible for updating the players animation based on movement.

Inheritance diagram for CaveExploration.PlayerAnimation:



Public Member Functions

- void `FinishedSpawning` ()
Sets spawning as finished.

Properties

- bool [IsSpawning](#) [get]
Gets a value indicating whether this instance is spawning.

5.50.1 Detailed Description

Responsible for updating the players animation based on movement.

5.50.2 Member Function Documentation

5.50.2.1 void CaveExploration.PlayerAnimation.FinishedSpawning ()

Sets spawning as finished.

5.50.3 Property Documentation

5.50.3.1 bool CaveExploration.PlayerAnimation.IsSpawning [get]

Gets a value indicating whether this instance is spawning.

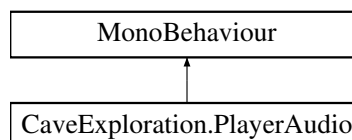
`true` if this instance is spawning; otherwise, `false`.

The documentation for this class was generated from the following file:

- PlayerAnimation.cs

5.51 CaveExploration.PlayerAudio Class Reference

Inheritance diagram for CaveExploration.PlayerAudio:



Public Member Functions

- void [PlayStepAudio](#) ()
Plays random clip on player step.
- void [OnPlayerDamaged](#) ()
Plays sound when player damaged.
- void [PlaySpawnSound](#) ()
Plays spawn sound.

Public Attributes

- AudioClip[] [StepClips](#)
Audio to be played on player taking a step.
- AudioClip [FallingAudio](#)

Audio to be played on player falling.

- AudioClip [JumpAudio](#)

Audio to be played on player jumping.

- AudioClip [HurtAudio](#)

Audio to be played on player hurt.

- AudioClip [SpawnSound](#)

Audio to be played on player spawned.

5.51.1 Member Function Documentation

5.51.1.1 void CaveExploration.PlayerAudio.OnPlayerDamaged ()

Plays sound when player damaged.

5.51.1.2 void CaveExploration.PlayerAudio.PlaySpawnSound ()

Plays spawn sound.

5.51.1.3 void CaveExploration.PlayerAudio.PlayStepAudio ()

Plays random clip on player step.

5.51.2 Member Data Documentation

5.51.2.1 AudioClip CaveExploration.PlayerAudio.FallingAudio

Audio to be played on player falling.

5.51.2.2 AudioClip CaveExploration.PlayerAudio.HurtAudio

Audio to be played on player hurt.

5.51.2.3 AudioClip CaveExploration.PlayerAudio.JumpAudio

Audio to be played on player jumping.

5.51.2.4 AudioClip CaveExploration.PlayerAudio.SpawnSound

Audio to be played on player spawned.

5.51.2.5 AudioClip [] CaveExploration.PlayerAudio.StepClips

Audio to be played on player taking a step.

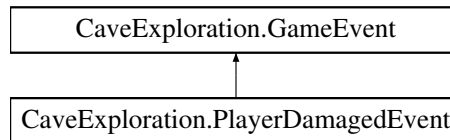
The documentation for this class was generated from the following file:

- PlayerAudio.cs

5.52 CaveExploration.PlayerDamagedEvent Class Reference

Raised when the player takes damage.

Inheritance diagram for CaveExploration.PlayerDamagedEvent:



Public Member Functions

- [PlayerDamagedEvent](#) (int damageAmount)
Initializes a new instance of the [CaveExploration.PlayerDamagedEvent](#) class.
- [PlayerDamagedEvent](#) (int damageAmount, Vector2?forceDirection, float force)
Initializes a new instance of the [CaveExploration.PlayerDamagedEvent](#) class.
- Vector2 [DamageForce](#) ()
The force to apply as knockback when player receives damage.

Properties

- int [DamageAmount](#) [get]
The amount to reduce the players health.

5.52.1 Detailed Description

Raised when the player takes damage.

5.52.2 Constructor & Destructor Documentation

5.52.2.1 CaveExploration.PlayerDamagedEvent.PlayerDamagedEvent (int damageAmount)

Initializes a new instance of the [CaveExploration.PlayerDamagedEvent](#) class.

Parameters

<i>damageAmount</i>	Damage amount.
---------------------	----------------

5.52.2.2 CaveExploration.PlayerDamagedEvent.PlayerDamagedEvent (int damageAmount, Vector2? forceDirection, float force)

Initializes a new instance of the [CaveExploration.PlayerDamagedEvent](#) class.

Parameters

<i>damageAmount</i>	Damage amount.
<i>forceDirection</i>	Force direction.

<i>force</i>	Force.
--------------	--------

5.52.3 Member Function Documentation

5.52.3.1 Vector2 CaveExploration.PlayerDamagedEvent.DamageForce ()

The force to apply as knockback when player receives damage.

Returns

The force.

5.52.4 Property Documentation

5.52.4.1 int CaveExploration.PlayerDamagedEvent.DamageAmount [get]

The amount to reduce the players health.

The damage amount.

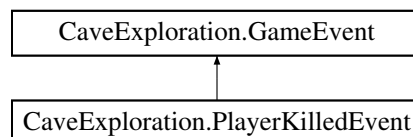
The documentation for this class was generated from the following file:

- PlayerDamagedEvent.cs

5.53 CaveExploration.PlayerKilledEvent Class Reference

Raised when the player is killed.

Inheritance diagram for CaveExploration.PlayerKilledEvent:



5.53.1 Detailed Description

Raised when the player is killed.

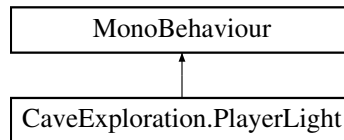
The documentation for this class was generated from the following file:

- PlayerKilledEvent.cs

5.54 CaveExploration.PlayerLight Class Reference

Responsible for updating the players light.

Inheritance diagram for CaveExploration.PlayerLight:



Public Attributes

- float `LightDecreaseOverTime` = 0.001f
The light intensity decrease per fixed update.
- float `RadiusDecreaseOverTime` = 0.0002f
The radius decrease per fixed update.
- AudioClip `PickUpAudio`
The audio clip to play when light collectible picked up.
- bool `RestartLevelOnEmptyLight` = false
Sets whether the level should be reset when light reaches zero.
- string[] `LightLowSpeech`
Speech options when light reaches threshold.
- float `LowLightThreshold` = 2f
The low light threshold (LowLightSpeech is shown when the light reaches this threshold).
- Color `LightDecreaseColour` = Color.black
The light colour when player has been hit.
- float `DamageAnimationTime` = 0.5f
The time to change the light colour.
- string[] `LightDecreaseSpeech`
Speech options to show then the light decreases.

5.54.1 Detailed Description

Responsible for updating the players light.

5.54.2 Member Data Documentation

5.54.2.1 float CaveExploration.PlayerLight.DamageAnimationTime = 0.5f

The time to change the light colour.

5.54.2.2 Color CaveExploration.PlayerLight.LightDecreaseColour = Color.black

The light colour when player has been hit.

5.54.2.3 float CaveExploration.PlayerLight.LightDecreaseOverTime = 0.001f

The light intensity decrease per fixed update.

5.54.2.4 string[] CaveExploration.PlayerLight.LightDecreaseSpeech

Speech options to show then the light decreases.

5.54.2.5 `string [] CaveExploration.PlayerLight.LightLowSpeech`

Speech options when light reaches threshold.

5.54.2.6 `float CaveExploration.PlayerLight.LowLightThreshold = 2f`

The low light threshold (LowLightSpeech is shown when the light reaches this threshold).

5.54.2.7 `AudioClip CaveExploration.PlayerLight.PickUpAudio`

The audio clip to play when light collectible picked up.

5.54.2.8 `float CaveExploration.PlayerLight.RadiusDecreaseOverTime = 0.0002f`

The radius decrease per fixed update.

5.54.2.9 `bool CaveExploration.PlayerLight.RestartLevelOnEmptyLight = false`

Sets whether the level should be reset when light reaches zero.

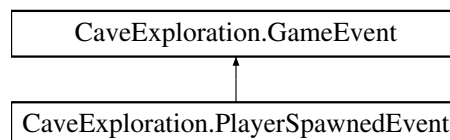
The documentation for this class was generated from the following file:

- `PlayerLight.cs`

5.55 `CaveExploration.PlayerSpawnedEvent` Class Reference

Raised when the player is spawned.

Inheritance diagram for `CaveExploration.PlayerSpawnedEvent`:



Public Member Functions

- [PlayerSpawnedEvent](#) (Transform player)
Initializes a new instance of the [CaveExploration.PlayerSpawnedEvent](#) class.

Properties

- Transform [Player](#) [get]
The players transform.

5.55.1 Detailed Description

Raised when the player is spawned.

5.55.2 Constructor & Destructor Documentation

5.55.2.1 CaveExploration.PlayerSpawnedEvent.PlayerSpawnedEvent (Transform *player*)

Initializes a new instance of the [CaveExploration.PlayerSpawnedEvent](#) class.

Parameters

<i>player</i>	Player .
---------------	--------------------------

5.55.3 Property Documentation

5.55.3.1 Transform CaveExploration.PlayerSpawnedEvent.Player [get]

The players transform.

The player.

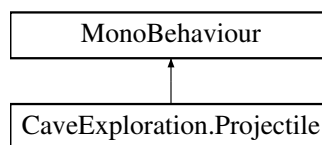
The documentation for this class was generated from the following file:

- PlayerSpawnedEvent.cs

5.56 CaveExploration.Projectile Class Reference

Attach to players light projectile. Handles seeking enemies, collision, and returning projectile to pool.

Inheritance diagram for CaveExploration.Projectile:



Public Attributes

- bool [SeekEnemies](#) = false
Sets whether this projectile should seek an enemy.
- float [SeekSpeed](#) = 2f
The speed to seek enemies.
- float [RotationSpeed](#) = 0.5f
The maximum speed at which the projectile can rotate to move towards an enemy.
- float [MaxTimeAlive](#) = 5f
The maximum time the projectile can be in the scene. It is added to the pool once this time has been reached.

5.56.1 Detailed Description

Attach to players light projectile. Handles seeking enemies, collision, and returning projectile to pool.

5.56.2 Member Data Documentation

5.56.2.1 float CaveExploration.Projectile.MaxTimeAlive = 5f

The maximum time the projectile can be in the scene. It is added to the pool once this time has been reached.

5.56.2.2 float CaveExploration.Projectile.RotationSpeed = 0.5f

The maximum speed at which the projectile can rotate to move towards an enemy.

5.56.2.3 bool CaveExploration.Projectile.SeekEnemies = false

Sets whether this projectile should seek an enemy.

5.56.2.4 float CaveExploration.Projectile.SeekSpeed = 2f

The speed to seek enemies.

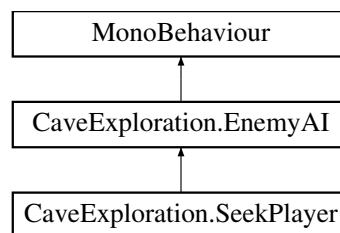
The documentation for this class was generated from the following file:

- Projectile.cs

5.57 CaveExploration.SeekPlayer Class Reference

AI for the "SeekPlayer" enemy. Handles animation, audio, and AI states.

Inheritance diagram for CaveExploration.SeekPlayer:



Public Member Functions

- override void [Update](#) ()
Update this instance using a simple state machine.
- void [BurstMoveTowardsPlayer](#) ()
Burst moves towards the player. Called by animation.

Public Attributes

- float [MoveBurstForce](#) = 1f
The burst force to use when seeking player.
- float [RotationSpeed](#) = 0.5f
The maximum speed at which the enemy can rotate towards player.
- float [TimeBetweenMovementBursts](#) = 0.2f
The time between movement bursts.
- float [MaxForce](#) = 2f
The maximum movement force.
- AudioClip [MoveAudio](#)
Audio clip to play on enemy movement.

Protected Member Functions

- override void **Idle** ()
- override void **AttackPlayer** ()
- override void **DecideState** ()

Additional Inherited Members

5.57.1 Detailed Description

AI for the "SeekPlayer" enemy. Handles animation, audio, and AI states.

5.57.2 Member Function Documentation

5.57.2.1 void CaveExploration.SeekPlayer.BurstMoveTowardsPlayer ()

Burst moves towards the player. Called by animation.

5.57.2.2 override void CaveExploration.SeekPlayer.Update () [virtual]

Update this instance using a simple state machine.

Reimplemented from [CaveExploration.EnemyAI](#).

5.57.3 Member Data Documentation

5.57.3.1 float CaveExploration.SeekPlayer.MaxForce = 2f

The maximum movement force.

5.57.3.2 AudioClip CaveExploration.SeekPlayer.MoveAudio

Audio clip to play on enemy movement.

5.57.3.3 float CaveExploration.SeekPlayer.MoveBurstForce = 1f

The burst force to use when seeking player.

5.57.3.4 float CaveExploration.SeekPlayer.RotationSpeed = 0.5f

The maximum speed at which the enemy can rotate towards player.

5.57.3.5 float CaveExploration.SeekPlayer.TimeBetweenMovementBursts = 0.2f

The time between movement bursts.

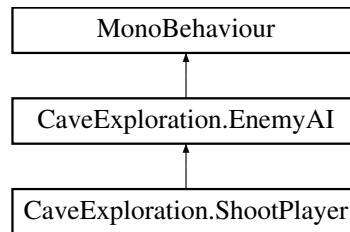
The documentation for this class was generated from the following file:

- SeekPlayer.cs

5.58 CaveExploration.ShootPlayer Class Reference

AI for the "ShootPlayer" enemy. Handles animation, audio, and AI states.

Inheritance diagram for CaveExploration.ShootPlayer:



Public Member Functions

- override void [Update](#) ()
Update this instance using a simple state machine.
- void [ShootProjectile](#) ()
Shoots the projectile in direction of player.

Public Attributes

- float [MoveSpeed](#) = 1f
The maximum movement speed.
- float [ProjectileForce](#) = 1f
The force used to shoot projectiles towards player.
- GameObject [Projectile](#)
The projectile prefab.
- Transform [ProjectileSpawnLocation](#)
The projectile spawn location relative to enemy.
- bool [IsFacingRight](#) = true
Determines whether enemy is facing right. Used to flip sprite to face movement direction.
- AudioClip [ShootAudioClip](#)
Audio clip to play on shoot.

Protected Member Functions

- override void **DecideState** ()
- override void **Idle** ()
- override void **AttackPlayer** ()

Additional Inherited Members

5.58.1 Detailed Description

AI for the "ShootPlayer" enemy. Handles animation, audio, and AI states.

5.58.2 Member Function Documentation

5.58.2.1 void CaveExploration.ShootPlayer.ShootProjectile ()

Shoots the projectile in direction of player.

5.58.2.2 override void CaveExploration.ShootPlayer.Update () [virtual]

Update this instance using a simple state machine.

Reimplemented from [CaveExploration.EnemyAI](#).

5.58.3 Member Data Documentation

5.58.3.1 bool CaveExploration.ShootPlayer.IsFacingRight = true

Determines whether enemy is facing right. Used to flip sprite to face movement direction.

5.58.3.2 float CaveExploration.ShootPlayer.MoveSpeed = 1f

The maximum movement speed.

5.58.3.3 GameObject CaveExploration.ShootPlayer.Projectile

The projectile prefab.

5.58.3.4 float CaveExploration.ShootPlayer.ProjectileForce = 1f

The force used to shoot projectiles towards player.

5.58.3.5 Transform CaveExploration.ShootPlayer.ProjectileSpawnLocation

The projectile spawn location relative to enemy.

5.58.3.6 AudioClip CaveExploration.ShootPlayer.ShootAudioClip

Audio clip to play on shoot.

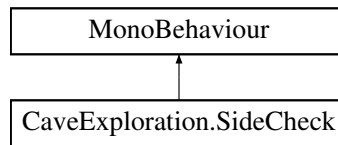
The documentation for this class was generated from the following file:

- ShootPlayer.cs

5.59 CaveExploration.SideCheck Class Reference

Checks whether the player is colliding on the sides with a wall tile.

Inheritance diagram for CaveExploration.SideCheck:



Public Attributes

- LayerMask [GroundMask](#)
The ground mask.
- float [KnockbackForce](#) = 200f
The force to apply when a player walks into a wall.

5.59.1 Detailed Description

Checks whether the player is colliding on the sides with a wall tile.

5.59.2 Member Data Documentation

5.59.2.1 LayerMask CaveExploration.SideCheck.GroundMask

The ground mask.

5.59.2.2 float CaveExploration.SideCheck.KnockbackForce = 200f

The force to apply when a player walks into a wall.

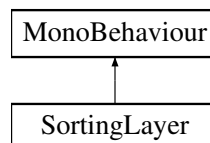
The documentation for this class was generated from the following file:

- SideCheck.cs

5.60 SortingLayer Class Reference

Changes a mesh renderers sorting layer.

Inheritance diagram for SortingLayer:



Public Attributes

- int [SortingOrder](#) = 0
The sorting order.

5.60.1 Detailed Description

Changes a mesh renderers sorting layer.

5.60.2 Member Data Documentation

5.60.2.1 int SortingLayer.SortingOrder = 0

The sorting order.

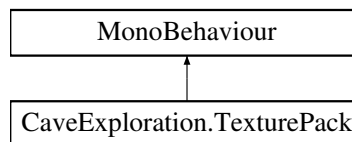
The documentation for this class was generated from the following file:

- SortingLayer.cs

5.61 CaveExploration.TexturePack Class Reference

Singleton. Provides a centralised location for block textures. Provides a method to retrieve a texture based on a node type.

Inheritance diagram for CaveExploration.TexturePack:



Public Member Functions

- Sprite [GetSpriteFromCellType](#) (NodeType cellType)
Returns a texture based on a node type.
- Vector2 [GetSpriteSize](#) (NodeType cellType, Vector2 localScale)
Gets the size of the sprite.

Public Attributes

- bool [Enabled](#) = true
Whether this texture pack can be used in the game.
- Sprite [WallTopLeft](#)
The wall top left sprite.
- Sprite [WallTopMiddle](#)
The wall top middle sprite.
- Sprite [WallTopRight](#)
The wall top right sprite.
- Sprite [WallMiddleLeft](#)
The wall middle left sprite.
- Sprite [WallMiddle](#)
The wall middle sprite.
- Sprite [WallMiddleRight](#)
The wall middle right sprite.
- Sprite [WallBottomLeft](#)
The wall bottom left sprite.
- Sprite [WallBottomMiddle](#)
The wall bottom middle sprite.
- Sprite [WallBottomRight](#)

The wall bottom right sprite.

- Sprite [Background](#)

The background sprite.

5.61.1 Detailed Description

Singleton. Provides a centralised location for block textures. Provides a method to retrieve a texture based on a node type.

5.61.2 Member Function Documentation

5.61.2.1 Sprite CaveExploration.TexturePack.GetSpriteFromCellType (NodeType cellType)

Returns a texture based on a node type.

5.61.2.2 Vector2 CaveExploration.TexturePack.GetSpriteSize (NodeType cellType, Vector2 localScale)

Gets the size of the sprite.

Returns

The sprite size.

Parameters

<i>cellType</i>	Cell type.
<i>localScale</i>	Local scale.

5.61.3 Member Data Documentation

5.61.3.1 Sprite CaveExploration.TexturePack.Background

The background sprite.

5.61.3.2 bool CaveExploration.TexturePack.Enabled = true

Whether this texture pack can be used in the game.

5.61.3.3 Sprite CaveExploration.TexturePack.WallBottomLeft

The wall bottom left sprite.

5.61.3.4 Sprite CaveExploration.TexturePack.WallBottomMiddle

The wall bottom middle sprite.

5.61.3.5 Sprite CaveExploration.TexturePack.WallBottomRight

The wall bottom right sprite.

5.61.3.6 Sprite CaveExploration.TexturePack.WallMiddle

The wall middle sprite.

5.61.3.7 Sprite CaveExploration.TexturePack.WallMiddleLeft

The wall middle left sprite.

5.61.3.8 Sprite CaveExploration.TexturePack.WallMiddleRight

The wall middle right sprite.

5.61.3.9 Sprite CaveExploration.TexturePack.WallTopLeft

The wall top left sprite.

5.61.3.10 Sprite CaveExploration.TexturePack.WallTopMiddle

The wall top middle sprite.

5.61.3.11 Sprite CaveExploration.TexturePack.WallTopRight

The wall top right sprite.

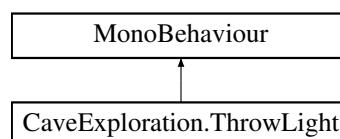
The documentation for this class was generated from the following file:

- TexturePack.cs

5.62 CaveExploration.ThrowLight Class Reference

Handles the players ability to throw light.

Inheritance diagram for CaveExploration.ThrowLight:



Public Attributes

- int `Capacity` = 3
The number of lights held by the pleyer at game start.
- GameObject `Throwable`
The throwable prefab.
- float `Force` = 5f
The force that light is thrown.
- string[] `SpeechOnEmpty`
Speech options to show when there are no more lights to be thrown.

5.62.1 Detailed Description

Handles the players ability to throw light.

5.62.2 Member Data Documentation

5.62.2.1 `int CaveExploration.ThrowLight.Capacity = 3`

The number of lights held by the player at game start.

5.62.2.2 `float CaveExploration.ThrowLight.Force = 5f`

The force that light is thrown.

5.62.2.3 `string [] CaveExploration.ThrowLight.SpeechOnEmpty`

Speech options to show when there are no more lights to be thrown.

5.62.2.4 `GameObject CaveExploration.ThrowLight.Throwable`

The throwable prefab.

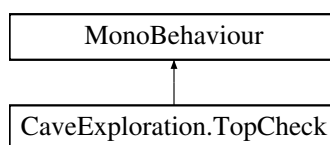
The documentation for this class was generated from the following file:

- `ThrowLight.cs`

5.63 CaveExploration.TopCheck Class Reference

Checks whether the player is colliding on the top with a wall tile.

Inheritance diagram for `CaveExploration.TopCheck`:



Public Attributes

- LayerMask `GroundMask`
The ground mask.
- float `KnockbackForce = 20f`
The force to apply when a player collides with a wall.
- ParticleSystem `particle`
The particle.

5.63.1 Detailed Description

Checks whether the player is colliding on the top with a wall tile.

5.63.2 Member Data Documentation

5.63.2.1 LayerMask CaveExploration.TopCheck.GroundMask

The ground mask.

5.63.2.2 float CaveExploration.TopCheck.KnockbackForce = 20f

The force to apply when a player collides with a wall.

5.63.2.3 ParticleSystem CaveExploration.TopCheck.particle

The particle.

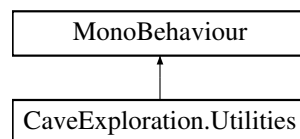
The documentation for this class was generated from the following file:

- TopCheck.cs

5.64 CaveExploration.Utilities Class Reference

Singleton. Provides centralised access to methods used by different classes.

Inheritance diagram for CaveExploration.Utilities:



Public Member Functions

- Vector2 [GetNodePosition](#) (Node node)
Returns world position of a specified node
- Vector2 [GetGridCoordinateForPosition](#) (Vector2 position)
Returns grid coordinates for a node at
- float [IncrementTowards](#) (float n, float target, float acc)
Increments n towards target based on acceleration. Useful in bespoke physics engine to increment a characters position towards their toarget.
- GameObject [FindObjectWithTag](#) (string scriptName, string tag)
Implementation of unitys FindObjectWithTag with additional logging.
- T [FindObjectOfType< T >](#) (string scriptName)
Implementation of unitys FindObjectOfType with additional logging.
- T [GetChildComponent< T >](#) (string scriptName, Transform owner)
Implementation of unitys GetChildComponenet with additional logging.
- T [GetComponent< T >](#) (string scriptName)
Implementation of unitys GetComponentet with additional logging.
- void **UpdateNodeSortingOrder** (string scriptName, GameObject node, int order)
- GameObject [InstantiatePrefabAtPosition](#) (string scriptName, GameObject prefab, Vector2 position)
Implementation of unitys Instantiate with additional logging. Instantiates object at specified position. Preferably use [ObjectManager](#) to initialise objects as this uses the object pool.
- GameObject [InstantiatePrefab](#) (string scriptName, GameObject prefab)

Implementation of unitys Instantiate with additional logging. Preferably use [ObjectManager](#) to initialise objects as this uses the object pool.

- GameObject [GetPrefab](#) (string scriptName, string prefabName)
Returns prefab with specified name. Outputs error if prefab not found.
- List< Ray2D > [CreateRays](#) (Vector2 position, Vector2 heading, float angle, int number)
Creates rays with angle difference.
- List< Ray2D > [CreateRays](#) (Vector2 position, Vector2 heading, float angle)
Creates three rays with angle difference.

Public Attributes

- bool [IsDebug](#) = false
Sets whether the game is in debug mode. When in debug mode extra logs are shown and player can use cheat keys to skip levels.

Properties

- static [Utilities instance](#) [get]
Gets the instance.

5.64.1 Detailed Description

Singleton. Provides centralised access to methods used by different classes.

5.64.2 Member Function Documentation

5.64.2.1 List<Ray2D> CaveExploration.Utilities.CreateRays (Vector2 position, Vector2 heading, float angle, int number)

Creates rays with angle difference.

Returns

The rays.

Parameters

<i>position</i>	Position of rays.
<i>heading</i>	Heading of central ray.
<i>angle</i>	Angle between each ray.
<i>number</i>	Number of rays to create.

5.64.2.2 List<Ray2D> CaveExploration.Utilities.CreateRays (Vector2 position, Vector2 heading, float angle)

Creates three rays with angle difference.

Returns

The created rays.

Parameters

<i>position</i>	Position of rays.
<i>heading</i>	Heading of central ray.
<i>angle</i>	Angle between each ray.

5.64.2.3 T CaveExploration.Utilities.FindObjectOfType< T > (string *scriptName*)

Implementation of unitys FindObjectOfType with additional logging.

Type Constraints

***T* : Component**

5.64.2.4 GameObject CaveExploration.Utilities.FindObjectWithTag (string *scriptName*, string *tag*)

Implementation of unitys FindObjectWithTag with additional logging.

5.64.2.5 T CaveExploration.Utilities.GetChildComponent< T > (string *scriptName*, Transform *owner*)

Implementation of unitys GetChildComponenet with additional logging.

Type Constraints

***T* : Component**

5.64.2.6 T CaveExploration.Utilities.GetComponent< T > (string *scriptName*)

Implementation of unitys GetComponentet with additional logging.

Type Constraints

***T* : Component**

5.64.2.7 Vector2 CaveExploration.Utilities.GetGridCoordinateForPosition (Vector2 *position*)

Returns grid coordinates for a node at

5.64.2.8 Vector2 CaveExploration.Utilities.GetNodePosition (Node *node*)

Returns world position of a specified node

5.64.2.9 GameObject CaveExploration.Utilities.GetPrefab (string *scriptName*, string *prefabName*)

Returns prefab with specified name. Outputs error if prefab not found.

5.64.2.10 float CaveExploration.Utilities.IncrementTowards (float *n*, float *target*, float *acc*)

Increments n towards target based on acceleration. Useful in bespoke physics engine to increment a characters position towards their toarget.

5.64.2.11 `GameObject CaveExploration.Utilities.InstantiatePrefab (string scriptName, GameObject prefab)`

Implementation of unitys Instantiate with additional logging. Preferably use [ObjectManager](#) to initialise objects as this uses the object pool.

5.64.2.12 `GameObject CaveExploration.Utilities.InstantiatePrefabAtPosition (string scriptName, GameObject prefab, Vector2 position)`

Implementation of unitys Instantiate with additional logging. Instantiates object at specified position. Preferably use [ObjectManager](#) to initialise objects as this uses the object pool.

5.64.3 Member Data Documentation

5.64.3.1 `bool CaveExploration.Utilities.IsDebug = false`

Sets whether the game is in debug mode. WHEN in debug mode extra logs are shown and player can use cheat keys to skip levels.

5.64.4 Property Documentation

5.64.4.1 `Utilities CaveExploration.Utilities.instance [static],[get]`

Gets the instance.

The instance.

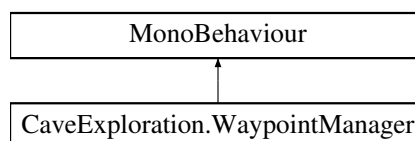
The documentation for this class was generated from the following file:

- Utilities.cs

5.65 CaveExploration.WaypointManager Class Reference

Holds list of waypoints and allows iteration over waypoints.

Inheritance diagram for CaveExploration.WaypointManager:



Public Member Functions

- void [InitialiseWaypointsFromNodes](#) (List< [Node](#) > path)
Initialises the internal waypoints from a list of nodes.
- bool [HasReachedCurrentWaypoint](#) (Vector2 characterPosition)
Determines whether this object has reached current waypoint.
- [Node GetCurrentWaypoint](#) ()
Gets the current waypoint.
- bool [Initialised](#) ()
Gets whether the waypoint manager has been initialised.

- Vector2 [GetNextReactivePosition](#) (Vector2 currentPosition)
Gets the next reactive position. The next position within the players line of sight (LOS) is returned. This iterates from the current waypoint to the next 5 waypoints and returns the last waypoint in the characters LOS.
- Vector2 [GetNextPosition](#) (Vector2 currentPosition)
Gets the next position that character should move towards.

Public Attributes

- bool [IsLooped](#) = false
If not looped will end at last wapoint
- float [WaypointProximity](#) = 0.1f
When entity is within this distance to waypoint it will be registered as visited
- List< [Node](#) > [Waypoints](#) = new List<[Node](#)> ()
List of waypoint objects - initialised in InitialliseWayPoints

Properties

- bool [IsComplete](#) [get, set]
Gets or sets a value indicating whether this instance is complete.

5.65.1 Detailed Description

Holds list of waypoints and allows iteration over waypoints.

5.65.2 Member Function Documentation

5.65.2.1 Node CaveExploration.WaypointManager.GetCurrentWaypoint ()

Gets the current waypoint.

Returns

The current waypoint.

5.65.2.2 Vector2 CaveExploration.WaypointManager.GetNextPosition (Vector2 currentPosition)

Gets the next position that character should move towards.

Returns

The next position.

Parameters

<i>currentPosition</i>	Current position.
------------------------	-------------------

5.65.2.3 Vector2 CaveExploration.WaypointManager.GetNextReactivePosition (Vector2 currentPosition)

Gets the next reactive position. The next position within the players line of sight (LOS) is returned. This iterates from the current waypoint to the next 5 waypoints and returns the last waypoint in the characters LOS.

Returns

The next reactive position.

Parameters

<i>currentPosition</i>	Current position.
------------------------	-------------------

5.65.2.4 **bool** CaveExploration.WaypointManager.HasReachedCurrentWaypoint (**Vector2** *characterPosition*)

Determines whether this object has reached current waypoint.

Returns

`true` if this instance has reached the current waypoint; otherwise, `false`.

Parameters

<i>character↔ Position</i>	Characters current position.
--------------------------------	------------------------------

5.65.2.5 **bool** CaveExploration.WaypointManager.Initialised ()

Gets whether the waypoint manager has been initialised.

5.65.2.6 **void** CaveExploration.WaypointManager.InitialiseWaypointsFromNodes (**List**< **Node** > *path*)

Initialises the internal waypoints from a list of nodes.

Parameters

<i>path</i>	Path.
-------------	-------

5.65.3 Member Data Documentation

5.65.3.1 **bool** CaveExploration.WaypointManager.IsLooped = `false`

If not looped will end at last waypoint

5.65.3.2 **float** CaveExploration.WaypointManager.WaypointProximity = `0.1f`

When entity is within this distance to waypoint it will be registered as visited

5.65.3.3 **List**< **Node** > CaveExploration.WaypointManager.Waypoints = `new List<Node> ()`

List of waypoint objects - initialised in `InitialiseWayPoints`

5.65.4 Property Documentation

5.65.4.1 **bool** CaveExploration.WaypointManager.IsComplete [get], [set]

Gets or sets a value indicating whether this instance is complete.

`true` if this instance is complete; otherwise, `false`.

The documentation for this class was generated from the following file:

- WaypointManager.cs

Index

- AIState
 - CaveExploration, [12](#)
- Add
 - CaveExploration::NodeList, [56](#)
- amountToBuffer
 - CaveExploration::ObjectPool, [61](#)
- Background
 - CaveExploration::TexturePack, [78](#)
- BackgroundAudioTracks
 - CaveExploration::BackgroundAudio, [15](#)
- BurstMoveTowardsPlayer
 - CaveExploration::SeekPlayer, [73](#)
- CalculateMainCluster
 - CaveExploration::NodeClusterManager, [54](#)
- CanJet
 - CaveExploration::Jetpack, [43](#)
- Capacity
 - CaveExploration::ThrowLight, [80](#)
- CaveExploration, [9](#)
 - AIState, [12](#)
 - InitWaypointDel, [12](#)
 - NodeType, [12](#)
- CaveExploration.AudioPlayer, [13](#)
- CaveExploration.BackgroundAudio, [14](#)
- CaveExploration.BobSprite, [15](#)
- CaveExploration.BottomCheck, [16](#)
- CaveExploration.Bounce, [17](#)
- CaveExploration.CameraBackground, [18](#)
- CaveExploration.CharacterSpeech, [19](#)
- CaveExploration.Collectible, [19](#)
- CaveExploration.CollectiblePickedUpEvent, [20](#)
- CaveExploration.CollectibleSpawner, [21](#)
- CaveExploration.DamagePlayerOnCollision, [22](#)
- CaveExploration.DirectorFullGame, [23](#)
- CaveExploration.DirectorGridGenerator, [24](#)
- CaveExploration.EnableColliderOnTriggerExit, [24](#)
- CaveExploration.EnemiesPlaced, [25](#)
- CaveExploration.EnemyAI, [26](#)
- CaveExploration.EnemyHealth, [27](#)
- CaveExploration.EnemyKillCountSpeech, [28](#)
- CaveExploration.EnemyKilled, [28](#)
- CaveExploration.EnemyProjectile, [29](#)
- CaveExploration.EnemySpawner, [30](#)
- CaveExploration.Events, [31](#)
- CaveExploration.Explode, [31](#)
- CaveExploration.Explosion, [32](#)
- CaveExploration.FollowPlayer, [33](#)
- CaveExploration.GameEvent, [34](#)
- CaveExploration.GridManager, [34](#)
- CaveExploration.Health, [38](#)
- CaveExploration.IntroductorySpeech, [40](#)
- CaveExploration.IntroductorySpeechFinishedEvent, [41](#)
- CaveExploration.JetAudio, [41](#)
- CaveExploration.Jetpack, [42](#)
- CaveExploration.JetpackBar, [43](#)
- CaveExploration.LevelCompleteEvent, [45](#)
- CaveExploration.LevelEnd, [46](#)
- CaveExploration.LevelGeneratedEvent, [46](#)
- CaveExploration.LevelGeneratedSpeechRequired, [47](#)
- CaveExploration.LightDecreaseEvent, [47](#)
- CaveExploration.LightIntensityPulse, [48](#)
- CaveExploration.LightPulse, [49](#)
- CaveExploration.LightRangePulse, [50](#)
- CaveExploration.Node, [50](#)
- CaveExploration.NodeCluster, [53](#)
- CaveExploration.NodeClusterManager, [54](#)
- CaveExploration.NodeList, [55](#)
- CaveExploration.ObjectManager, [58](#)
- CaveExploration.ObjectPool, [60](#)
- CaveExploration.PathManager, [62](#)
- CaveExploration.Player, [63](#)
- CaveExploration.PlayerAnimation, [64](#)
- CaveExploration.PlayerAudio, [65](#)
- CaveExploration.PlayerDamagedEvent, [67](#)
- CaveExploration.PlayerKilledEvent, [68](#)
- CaveExploration.PlayerLight, [68](#)
- CaveExploration.PlayerSpawnedEvent, [70](#)
- CaveExploration.Projectile, [71](#)
- CaveExploration.SeekPlayer, [72](#)
- CaveExploration.ShootPlayer, [74](#)
- CaveExploration.SideCheck, [75](#)
- CaveExploration.TexturePack, [77](#)
- CaveExploration.ThrowLight, [79](#)
- CaveExploration.TopCheck, [80](#)
- CaveExploration.Utilities, [81](#)
- CaveExploration.WaypointManager, [84](#)
- CaveExploration::AudioPlayer
 - IsPlaying, [13](#)
 - PlaySound, [13](#), [14](#)
- CaveExploration::BackgroundAudio
 - BackgroundAudioTracks, [15](#)
 - Volume, [15](#)
- CaveExploration::BobSprite
 - Enabled, [16](#)
 - MaxUpDown, [15](#)
 - Speed, [15](#)
- CaveExploration::BottomCheck

- FallDamage, 16
- FallingHeightToDamage, 16
- GroundMask, 17
- IsGrounded, 17
- particle, 17
- CaveExploration::Bounce
 - PlayNearbyAudio, 18
 - PlayerNearbyAudio, 18
 - Update, 18
- CaveExploration::CharacterSpeech
 - Speak, 19
 - Speaking, 19
- CaveExploration::Collectible
 - LightIntensityAdded, 20
 - LightRadiusAdded, 20
 - SeekDistance, 20
 - SeekSpeed, 20
- CaveExploration::CollectibleSpawner
 - Collectible, 22
 - CollectibleWallLimit, 22
 - MaxCollectibles, 22
 - OnLevelGenerated, 22
 - SpawnChance, 22
- CaveExploration::DamagePlayerOnCollision
 - DamageAmount, 23
 - DamageForce, 23
- CaveExploration::DirectorFullGame
 - EndCollectiblePrefab, 23
 - PlayerPrefab, 23
 - Seed, 24
- CaveExploration::DirectorGridGenerator
 - Seed, 24
- CaveExploration::EnableColliderOnTriggerExit
 - MultipleColliders, 25
 - Tag, 25
- CaveExploration::EnemyAI
 - Update, 26
- CaveExploration::EnemyHealth
 - CountTowardsEnemyCount, 27
 - DeathDuration, 27
 - Kill, 27
 - SoundOnDeath, 27
- CaveExploration::EnemyKillCountSpeech
 - EnemyKilledSpeech, 28
- CaveExploration::EnemyProjectile
 - DamageAmount, 29
 - DamageForce, 29
 - MaxTimeAlive, 29
 - ReturnToPool, 29
- CaveExploration::EnemySpawner
 - Enemy, 31
 - MaxEnemies, 31
 - OnLevelGenerated, 30
 - PlacementType, 31
 - RandomRotation, 31
- CaveExploration::Explosion
 - ExplosionClip, 33
 - ExplosionForce, 33
 - PlayExplosionSound, 33
 - ReturnToPool, 33
- CaveExploration::FollowPlayer
 - Displacement, 33
- CaveExploration::GridManager
 - Cell, 37
 - ChanceToBecomeWall, 37
 - CountWallMooreNeighbours, 36
 - EndNode, 38
 - FloorsToWallConversion, 37
 - GenerateWithSeed, 36
 - GetBackgroundNodes, 36
 - GetFloorNodeMaxDistanceFromStartNode, 36
 - GetFloorNodeMinDistanceFromStartNode, 36
 - GetRandomBackgroundNode, 37
 - GetRandomFloorNode, 37
 - Grid, 38
 - GridSize, 37
 - instance, 38
 - NumberOfTransitionSteps, 37
 - StartNode, 38
 - TexturePack, 38
 - WallsToFloorConversion, 37
- CaveExploration::Health
 - DamageAnimationTime, 39
 - HurtColour, 39
 - HurtSpeech, 39
 - LowHealthSpeech, 39
 - LowHealthThreshold, 39
 - MaxHealth, 39
 - OnDamage, 39
- CaveExploration::IntroductorySpeech
 - InProgress, 40
 - instance, 40
 - Speech, 40
- CaveExploration::JetAudio
 - JetAudioClip, 42
 - Volume, 42
- CaveExploration::Jetpack
 - CanJet, 43
 - FuelRecharging, 43
 - JetForce, 43
 - JetFuel, 43
 - JetFuelWaste, 43
 - MaxForce, 43
 - UsingJet, 43
- CaveExploration::JetpackBar
 - ColourWhenEmptyFuel, 45
 - ColourWhenFullFuel, 45
 - Disable, 44
 - Enable, 44
 - UpdateColour, 44
 - UpdateLocalScaleY, 44
- CaveExploration::LevelCompleteEvent
 - LevelCompleteEvent, 45
 - NextSeed, 46
- CaveExploration::LightDecreaseEvent
 - LightAmount, 48

- LightDecreaseEvent, 48
- RadiusAmount, 48
- CaveExploration::LightPulse
 - Maximum, 49
 - Minimum, 49
 - Speed, 49
- CaveExploration::Node
 - CompareTo, 52
 - Coordinates, 52
 - GScore, 52
 - GetFScore, 52
 - HScore, 52
 - IsObstacle, 52
 - IsOccupied, 52
 - Node, 51
 - NodeState, 52
 - Parent, 53
 - Position, 53
- CaveExploration::NodeCluster
 - NodeCluster, 53
 - Nodes, 53
- CaveExploration::NodeClusterManager
 - CalculateMainCluster, 54
 - Clusters, 55
 - ConnectClusters, 54
 - ConvertDisconnectedClustersToNodeType, 55
 - IdentifyClusters, 55
 - MainCluster, 55
- CaveExploration::NodeList
 - Add, 56
 - Contains, 56
 - ContainsNodeTypeAtPosition, 56
 - Count, 58
 - First, 57
 - GetAdjacentNodes, 57
 - GetNodeFromGridCoordinate, 57
 - GetNodeFromPosition, 57
 - IsValidCoordinate, 57
 - Replace, 58
- CaveExploration::ObjectManager
 - GetObject, 59
 - instance, 60
 - RemoveObject, 60
 - RemoveObjects, 60
- CaveExploration::ObjectPool
 - amountToBuffer, 61
 - containerObject, 61
 - GetObjectForType, 61
 - objectPrefabs, 61
 - PoolObject, 61
 - pooledObjects, 62
- CaveExploration::PathManager
 - GetShortestPath, 62
- CaveExploration::Player
 - HasJumped, 64
 - IsDead, 64
 - IsFacingRight, 63
 - JumpForce, 63
 - MaxRunSpeed, 64
 - MaxWalkSpeed, 64
 - MoveSpeed, 64
- CaveExploration::PlayerAnimation
 - FinishedSpawning, 65
 - IsSpawning, 65
- CaveExploration::PlayerAudio
 - FallingAudio, 66
 - HurtAudio, 66
 - JumpAudio, 66
 - OnPlayerDamaged, 66
 - PlaySpawnSound, 66
 - PlayStepAudio, 66
 - SpawnSound, 66
 - StepClips, 66
- CaveExploration::PlayerDamagedEvent
 - DamageAmount, 68
 - DamageForce, 68
 - PlayerDamagedEvent, 67
- CaveExploration::PlayerLight
 - DamageAnimationTime, 69
 - LightDecreaseColour, 69
 - LightDecreaseOverTime, 69
 - LightDecreaseSpeech, 69
 - LightLowSpeech, 69
 - LowLightThreshold, 70
 - PickUpAudio, 70
 - RadiusDecreaseOverTime, 70
 - RestartLevelOnEmptyLight, 70
- CaveExploration::PlayerSpawnedEvent
 - Player, 71
 - PlayerSpawnedEvent, 71
- CaveExploration::Projectile
 - MaxTimeAlive, 71
 - RotationSpeed, 71
 - SeekEnemies, 72
 - SeekSpeed, 72
- CaveExploration::SeekPlayer
 - BurstMoveTowardsPlayer, 73
 - MaxForce, 73
 - MoveAudio, 73
 - MoveBurstForce, 73
 - RotationSpeed, 73
 - TimeBetweenMovementBursts, 73
 - Update, 73
- CaveExploration::ShootPlayer
 - IsFacingRight, 75
 - MoveSpeed, 75
 - Projectile, 75
 - ProjectileForce, 75
 - ProjectileSpawnLocation, 75
 - ShootAudioClip, 75
 - ShootProjectile, 75
 - Update, 75
- CaveExploration::SideCheck
 - GroundMask, 76
 - KnockbackForce, 76
- CaveExploration::TexturePack

- Background, 78
- Enabled, 78
- GetSpriteFromCellType, 78
- GetSpriteSize, 78
- WallBottomLeft, 78
- WallBottomMiddle, 78
- WallBottomRight, 78
- WallMiddle, 78
- WallMiddleLeft, 79
- WallMiddleRight, 79
- WallTopLeft, 79
- WallTopMiddle, 79
- WallTopRight, 79
- CaveExploration::ThrowLight
 - Capacity, 80
 - Force, 80
 - SpeechOnEmpty, 80
 - Throwable, 80
- CaveExploration::TopCheck
 - GroundMask, 81
 - KnockbackForce, 81
 - particle, 81
- CaveExploration::Utilities
 - CreateRays, 82
 - FindObjectOfType< T >, 83
 - FindObjectWithTag, 83
 - GetChildComponent< T >, 83
 - GetComponent< T >, 83
 - GetGridCoordinateForPosition, 83
 - GetNodePosition, 83
 - GetPrefab, 83
 - IncrementTowards, 83
 - instance, 84
 - InstantiatePrefab, 83
 - InstantiatePrefabAtPosition, 84
 - IsDebug, 84
- CaveExploration::WaypointManager
 - GetCurrentWaypoint, 85
 - GetNextPosition, 85
 - GetNextReactivePosition, 85
 - HasReachedCurrentWaypoint, 86
 - InitialiseWaypointsFromNodes, 86
 - Initialised, 86
 - IsComplete, 86
 - IsLooped, 86
 - WaypointProximity, 86
 - Waypoints, 86
- Cell
 - CaveExploration::GridManager, 37
- ChanceToBecomeWall
 - CaveExploration::GridManager, 37
- Clusters
 - CaveExploration::NodeClusterManager, 55
- Collectible
 - CaveExploration::CollectibleSpawner, 22
- CollectibleWallLimit
 - CaveExploration::CollectibleSpawner, 22
- ColourWhenEmptyFuel
 - CaveExploration::JetpackBar, 45
- ColourWhenFullFuel
 - CaveExploration::JetpackBar, 45
- CompareTo
 - CaveExploration::Node, 52
- ConnectClusters
 - CaveExploration::NodeClusterManager, 54
- containerObject
 - CaveExploration::ObjectPool, 61
- Contains
 - CaveExploration::NodeList, 56
- ContainsNodeTypeAtPosition
 - CaveExploration::NodeList, 56
- ConvertDisconnectedClustersToNodeType
 - CaveExploration::NodeClusterManager, 55
- Coordinates
 - CaveExploration::Node, 52
- Count
 - CaveExploration::NodeList, 58
- CountTowardsEnemyCount
 - CaveExploration::EnemyHealth, 27
- CountWallMooreNeighbours
 - CaveExploration::GridManager, 36
- CreateRays
 - CaveExploration::Utilities, 82
- DamageAmount
 - CaveExploration::DamagePlayerOnCollision, 23
 - CaveExploration::EnemyProjectile, 29
 - CaveExploration::PlayerDamagedEvent, 68
- DamageAnimationTime
 - CaveExploration::Health, 39
 - CaveExploration::PlayerLight, 69
- DamageForce
 - CaveExploration::DamagePlayerOnCollision, 23
 - CaveExploration::EnemyProjectile, 29
 - CaveExploration::PlayerDamagedEvent, 68
- DeathDuration
 - CaveExploration::EnemyHealth, 27
- Disable
 - CaveExploration::JetpackBar, 44
- Displacement
 - CaveExploration::FollowPlayer, 33
- Enable
 - CaveExploration::JetpackBar, 44
- Enabled
 - CaveExploration::BobSprite, 16
 - CaveExploration::TexturePack, 78
- EndCollectiblePrefab
 - CaveExploration::DirectorFullGame, 23
- EndNode
 - CaveExploration::GridManager, 38
- Enemy
 - CaveExploration::EnemySpawner, 31
- EnemyKilledSpeech
 - CaveExploration::EnemyKillCountSpeech, 28
- ExplosionClip
 - CaveExploration::Explosion, 33

- ExplosionForce
 - CaveExploration::Explosion, [33](#)
- FallDamage
 - CaveExploration::BottomCheck, [16](#)
- FallingAudio
 - CaveExploration::PlayerAudio, [66](#)
- FallingHeightToDamage
 - CaveExploration::BottomCheck, [16](#)
- FindObjectOfType< T >
 - CaveExploration::Utilities, [83](#)
- FindObjectWithTag
 - CaveExploration::Utilities, [83](#)
- FinishedSpawning
 - CaveExploration::PlayerAnimation, [65](#)
- First
 - CaveExploration::NodeList, [57](#)
- FloorsToWallConversion
 - CaveExploration::GridManager, [37](#)
- Force
 - CaveExploration::ThrowLight, [80](#)
- FuelRecharging
 - CaveExploration::Jetpack, [43](#)
- GScore
 - CaveExploration::Node, [52](#)
- GenerateWithSeed
 - CaveExploration::GridManager, [36](#)
- GetAdjacentNodes
 - CaveExploration::NodeList, [57](#)
- GetBackgroundNodes
 - CaveExploration::GridManager, [36](#)
- GetChildComponent< T >
 - CaveExploration::Utilities, [83](#)
- GetComponent< T >
 - CaveExploration::Utilities, [83](#)
- GetCurrentWaypoint
 - CaveExploration::WaypointManager, [85](#)
- GetFScore
 - CaveExploration::Node, [52](#)
- GetFloorNodeMaxDistanceFromStartNode
 - CaveExploration::GridManager, [36](#)
- GetFloorNodeMinDistanceFromStartNode
 - CaveExploration::GridManager, [36](#)
- GetGridCoordinateForPosition
 - CaveExploration::Utilities, [83](#)
- GetNextPosition
 - CaveExploration::WaypointManager, [85](#)
- GetNextReactivePosition
 - CaveExploration::WaypointManager, [85](#)
- GetNodeFromGridCoordinate
 - CaveExploration::NodeList, [57](#)
- GetNodeFromPosition
 - CaveExploration::NodeList, [57](#)
- GetNodePosition
 - CaveExploration::Utilities, [83](#)
- GetObject
 - CaveExploration::ObjectManager, [59](#)
- GetObjectForType
 - CaveExploration::ObjectPool, [61](#)
- GetPrefab
 - CaveExploration::Utilities, [83](#)
- GetRandomBackgroundNode
 - CaveExploration::GridManager, [37](#)
- GetRandomFloorNode
 - CaveExploration::GridManager, [37](#)
- GetShortestPath
 - CaveExploration::PathManager, [62](#)
- GetSpriteFromCellType
 - CaveExploration::TexturePack, [78](#)
- GetSpriteSize
 - CaveExploration::TexturePack, [78](#)
- Grid
 - CaveExploration::GridManager, [38](#)
- GridSize
 - CaveExploration::GridManager, [37](#)
- GroundMask
 - CaveExploration::BottomCheck, [17](#)
 - CaveExploration::SideCheck, [76](#)
 - CaveExploration::TopCheck, [81](#)
- HScore
 - CaveExploration::Node, [52](#)
- HasJumped
 - CaveExploration::Player, [64](#)
- HasReachedCurrentWaypoint
 - CaveExploration::WaypointManager, [86](#)
- HurtAudio
 - CaveExploration::PlayerAudio, [66](#)
- HurtColour
 - CaveExploration::Health, [39](#)
- HurtSpeech
 - CaveExploration::Health, [39](#)
- IdentifyClusters
 - CaveExploration::NodeClusterManager, [55](#)
- InProgress
 - CaveExploration::IntroductorySpeech, [40](#)
- IncrementTowards
 - CaveExploration::Utilities, [83](#)
- InitWaypointDel
 - CaveExploration, [12](#)
- InitialiseWaypointsFromNodes
 - CaveExploration::WaypointManager, [86](#)
- Initialised
 - CaveExploration::WaypointManager, [86](#)
- instance
 - CaveExploration::GridManager, [38](#)
 - CaveExploration::IntroductorySpeech, [40](#)
 - CaveExploration::ObjectManager, [60](#)
 - CaveExploration::Utilities, [84](#)
- InstantiatePrefab
 - CaveExploration::Utilities, [83](#)
- InstantiatePrefabAtPosition
 - CaveExploration::Utilities, [84](#)
- IsComplete
 - CaveExploration::WaypointManager, [86](#)
- IsDead

- CaveExploration::Player, 64
- IsDebug
 - CaveExploration::Utilities, 84
- IsFacingRight
 - CaveExploration::Player, 63
 - CaveExploration::ShootPlayer, 75
- IsGrounded
 - CaveExploration::BottomCheck, 17
- IsLooped
 - CaveExploration::WaypointManager, 86
- IsObstacle
 - CaveExploration::Node, 52
- IsOccupied
 - CaveExploration::Node, 52
- IsPlaying
 - CaveExploration::AudioPlayer, 13
- IsSpawning
 - CaveExploration::PlayerAnimation, 65
- IsValidCoordinate
 - CaveExploration::NodeList, 57
- JetAudioClip
 - CaveExploration::JetAudio, 42
- JetForce
 - CaveExploration::Jetpack, 43
- JetFuel
 - CaveExploration::Jetpack, 43
- JetFuelWaste
 - CaveExploration::Jetpack, 43
- JumpAudio
 - CaveExploration::PlayerAudio, 66
- JumpForce
 - CaveExploration::Player, 63
- Kill
 - CaveExploration::EnemyHealth, 27
- KnockbackForce
 - CaveExploration::SideCheck, 76
 - CaveExploration::TopCheck, 81
- LevelCompleteEvent
 - CaveExploration::LevelCompleteEvent, 45
- LightAmount
 - CaveExploration::LightDecreaseEvent, 48
- LightDecreaseColour
 - CaveExploration::PlayerLight, 69
- LightDecreaseEvent
 - CaveExploration::LightDecreaseEvent, 48
- LightDecreaseOverTime
 - CaveExploration::PlayerLight, 69
- LightDecreaseSpeech
 - CaveExploration::PlayerLight, 69
- LightIntensityAdded
 - CaveExploration::Collectible, 20
- LightLowSpeech
 - CaveExploration::PlayerLight, 69
- LightRadiusAdded
 - CaveExploration::Collectible, 20
- LowHealthSpeech
 - CaveExploration::Health, 39
- LowHealthThreshold
 - CaveExploration::Health, 39
- LowLightThreshold
 - CaveExploration::PlayerLight, 70
- MainCluster
 - CaveExploration::NodeClusterManager, 55
- MaxCollectibles
 - CaveExploration::CollectibleSpawner, 22
- MaxEnemies
 - CaveExploration::EnemySpawner, 31
- MaxForce
 - CaveExploration::Jetpack, 43
 - CaveExploration::SeekPlayer, 73
- MaxHealth
 - CaveExploration::Health, 39
- MaxRunSpeed
 - CaveExploration::Player, 64
- MaxTimeAlive
 - CaveExploration::EnemyProjectile, 29
 - CaveExploration::Projectile, 71
- MaxUpDown
 - CaveExploration::BobSprite, 15
- MaxWalkSpeed
 - CaveExploration::Player, 64
- Maximum
 - CaveExploration::LightPulse, 49
- Minimum
 - CaveExploration::LightPulse, 49
- MoveAudio
 - CaveExploration::SeekPlayer, 73
- MoveBurstForce
 - CaveExploration::SeekPlayer, 73
- MoveSpeed
 - CaveExploration::Player, 64
 - CaveExploration::ShootPlayer, 75
- MultipleColliders
 - CaveExploration::EnableColliderOnTriggerExit, 25
- NextSeed
 - CaveExploration::LevelCompleteEvent, 46
- Node
 - CaveExploration::Node, 51
- NodeCluster
 - CaveExploration::NodeCluster, 53
- NodeState
 - CaveExploration::Node, 52
- NodeType
 - CaveExploration, 12
- Nodes
 - CaveExploration::NodeCluster, 53
- NumberOfTransitionSteps
 - CaveExploration::GridManager, 37
- objectPrefabs
 - CaveExploration::ObjectPool, 61
- OnDamage
 - CaveExploration::Health, 39

- OnLevelGenerated
 - CaveExploration::CollectibleSpawner, [22](#)
 - CaveExploration::EnemySpawner, [30](#)
- OnPlayerDamaged
 - CaveExploration::PlayerAudio, [66](#)
- Parent
 - CaveExploration::Node, [53](#)
- particle
 - CaveExploration::BottomCheck, [17](#)
 - CaveExploration::TopCheck, [81](#)
- PickUpAudio
 - CaveExploration::PlayerLight, [70](#)
- PlacementType
 - CaveExploration::EnemySpawner, [31](#)
- PlayExplosionSound
 - CaveExploration::Explosion, [33](#)
- PlayNearbyAudio
 - CaveExploration::Bounce, [18](#)
- PlaySound
 - CaveExploration::AudioPlayer, [13](#), [14](#)
- PlaySpawnSound
 - CaveExploration::PlayerAudio, [66](#)
- PlayStepAudio
 - CaveExploration::PlayerAudio, [66](#)
- Player
 - CaveExploration::PlayerSpawnedEvent, [71](#)
- PlayerDamagedEvent
 - CaveExploration::PlayerDamagedEvent, [67](#)
- PlayerNearbyAudio
 - CaveExploration::Bounce, [18](#)
- PlayerPrefab
 - CaveExploration::DirectorFullGame, [23](#)
- PlayerSpawnedEvent
 - CaveExploration::PlayerSpawnedEvent, [71](#)
- PoolObject
 - CaveExploration::ObjectPool, [61](#)
- pooledObjects
 - CaveExploration::ObjectPool, [62](#)
- Position
 - CaveExploration::Node, [53](#)
- Projectile
 - CaveExploration::ShootPlayer, [75](#)
- ProjectileForce
 - CaveExploration::ShootPlayer, [75](#)
- ProjectileSpawnLocation
 - CaveExploration::ShootPlayer, [75](#)
- RadiusAmount
 - CaveExploration::LightDecreaseEvent, [48](#)
- RadiusDecreaseOverTime
 - CaveExploration::PlayerLight, [70](#)
- RandomRotation
 - CaveExploration::EnemySpawner, [31](#)
- RemoveObject
 - CaveExploration::ObjectManager, [60](#)
- RemoveObjects
 - CaveExploration::ObjectManager, [60](#)
- Replace
 - CaveExploration::NodeList, [58](#)
- RestartLevelOnEmptyLight
 - CaveExploration::PlayerLight, [70](#)
- ReturnToPool
 - CaveExploration::EnemyProjectile, [29](#)
 - CaveExploration::Explosion, [33](#)
- RotationSpeed
 - CaveExploration::Projectile, [71](#)
 - CaveExploration::SeekPlayer, [73](#)
- Seed
 - CaveExploration::DirectorFullGame, [24](#)
 - CaveExploration::DirectorGridGenerator, [24](#)
- SeekDistance
 - CaveExploration::Collectible, [20](#)
- SeekEnemies
 - CaveExploration::Projectile, [72](#)
- SeekSpeed
 - CaveExploration::Collectible, [20](#)
 - CaveExploration::Projectile, [72](#)
- ShootAudioClip
 - CaveExploration::ShootPlayer, [75](#)
- ShootProjectile
 - CaveExploration::ShootPlayer, [75](#)
- SortingLayer, [76](#)
 - SortingOrder, [77](#)
- SortingOrder
 - SortingLayer, [77](#)
- SoundOnDeath
 - CaveExploration::EnemyHealth, [27](#)
- SpawnChance
 - CaveExploration::CollectibleSpawner, [22](#)
- SpawnSound
 - CaveExploration::PlayerAudio, [66](#)
- Speak
 - CaveExploration::CharacterSpeech, [19](#)
- Speaking
 - CaveExploration::CharacterSpeech, [19](#)
- Speech
 - CaveExploration::IntroductorySpeech, [40](#)
- SpeechOnEmpty
 - CaveExploration::ThrowLight, [80](#)
- Speed
 - CaveExploration::BobSprite, [15](#)
 - CaveExploration::LightPulse, [49](#)
- StartNode
 - CaveExploration::GridManager, [38](#)
- StepClips
 - CaveExploration::PlayerAudio, [66](#)
- Tag
 - CaveExploration::EnableColliderOnTriggerExit, [25](#)
- TexturePack
 - CaveExploration::GridManager, [38](#)
- Throwable
 - CaveExploration::ThrowLight, [80](#)
- TimeBetweenMovementBursts
 - CaveExploration::SeekPlayer, [73](#)

Update

- CaveExploration::Bounce, [18](#)
- CaveExploration::EnemyAI, [26](#)
- CaveExploration::SeekPlayer, [73](#)
- CaveExploration::ShootPlayer, [75](#)

UpdateColour

- CaveExploration::JetpackBar, [44](#)

UpdateLocalScaleY

- CaveExploration::JetpackBar, [44](#)

UsingJet

- CaveExploration::Jetpack, [43](#)

Volume

- CaveExploration::BackgroundAudio, [15](#)
- CaveExploration::JetAudio, [42](#)

WallBottomLeft

- CaveExploration::TexturePack, [78](#)

WallBottomMiddle

- CaveExploration::TexturePack, [78](#)

WallBottomRight

- CaveExploration::TexturePack, [78](#)

WallMiddle

- CaveExploration::TexturePack, [78](#)

WallMiddleLeft

- CaveExploration::TexturePack, [79](#)

WallMiddleRight

- CaveExploration::TexturePack, [79](#)

WallTopLeft

- CaveExploration::TexturePack, [79](#)

WallTopMiddle

- CaveExploration::TexturePack, [79](#)

WallTopRight

- CaveExploration::TexturePack, [79](#)

WallsToFloorConversion

- CaveExploration::GridManager, [37](#)

WaypointProximity

- CaveExploration::WaypointManager, [86](#)

Waypoints

- CaveExploration::WaypointManager, [86](#)