Generators

Common base

Data oriented approach

With generators we have 2 types of Entities interacting with each other. There will always be an order of magnitude less generators than affected Entities. For example, there is 1 planet and 100 rolling stones on it. So affected objects are iterated linearly, few generators are accessed non-linearly. Each affected Entity stores a buffer of affecting generators and can iterate through them to calculate the effect of all generators.

RangedAffectGenerators Job system

If you want to perform some Jobs before or after the ranged affect generators update, you can sync with the RangedAffectGenerators JobSystem.

Common base class

All ranged generators inherit from the abstract GeneratorBaseComponent.

This component uses a spherical spatial trigger to register affected Entities. GeneratorBaseComponent takes care of the initialization and disposal of the spatial trigger, enabling/disabling the generator. It also provides OnEntityEntered and OnEntityExitedAllGenerators callbacks which can be implemented by the inherited generator to provide a custom behavior for the affected Entities.

GeneratorBaseComponent also provides UpdateGeneratorsEffect method to update all affected entities by the generator.

Gravity generator

Produced data

GravityGeneratorComponent produces GravityEffectData for affected Entities.

If the affected Entity is simulated(has IsSimulatedData), GravityEffectData is translated to CustomGravity by GravityGeneratorComponent.

Gravity shape type

There are three types of gravity shapes: field, spherical and cylindrical. The field shape means that the affected Entities move down. With spherical gravity shape the affected Entities move towards the center of the gravity generator. Cylindrical gravity shape allows the gravity to pull the Entities away or towards an imaginary line at the center of the gravity generator.

Terminal

Gravity strength

In terminal UI gravity strength is just GravitationalAcceleration / g.

Accordingly, lower limit of gravity strength slider = GravityGeneratorDefinition.MinGravitationalAcceleration / g and upper limit of gravity strength slider = GravityGeneratorDefinition.MaxGravitationalAcceleration / g.

So, if you want to change the range of gravity strength slider in the terminal, you should configure GravityGeneratorDefinition.MinGravitationalAcceleration and GravityGeneratorDefinition.MaxGravitationalAcceleration.

Gravity types

For designers

AffectedByGravityComponent allows Entity to be affected by gravity generators. Each GravitySetDefinition lists gravity types that should affect this Entity.

It allows us to configure which types of entities should be affected by the gravity generator(grids, character, floating objects, ...). All gravity types supported by this generator should be added to GravityGeneratorDefinition.GravityTypes. When Entity has AffectedByGravityComponent attached and its definition is added to the list of gravity types supported by the generator, it will be affected.

Technical details

Now the game supports up to 32 gravity types due to the number of bits in the bit field of GravityGeneratorDefinition.

GravityTypesBitField of GravityGeneratorDefinition

Bit field is generated at runtime for all gravity types of this definition. Each set bit represents a separate gravity type.

GravityUpdate Job system

If you want to perform some Jobs before or after the gravity update, you can sync with the GravityUpdate JobSystem.

Gravitational acceleration formula

$$g = (\frac{AccelerationDistance}{r})^{k} * GravitationalAcceleration$$

- The magnitude of the gravitational acceleration is calculated by this formula, where *r* is the distance between the affected Entity and the gravity generator, $AccelerationDistance$ is the distance at which $GravitationalAcceleration$ is applied when fall off power is not 0, k is the fall off power.

$(AccelerationDistance)^{k} * GravitationalAcceleration$ is equivalent to the gravitational constant multiplied by the mass GM.

Planet gravity

GravityGeneratorComponent is attached to the planet prefab. Planets can be different in size, so there was a need to find a way to scale gravity generator values according to the planet radius.

The solution was to create a GravityGeneratorProcessorComponent that scales affect range of gravity generator and sets AccelerationDistance to the value of the planet radius to keep the configured gravitational acceleration at the planet surface of any radius. GravitationalAcceleration value in the GravityGeneratorObjectBuilder of planet prefab is equivalent to m/s^2 at the surface level.

Don't forget that the max affect range of gravity generator even for planets is still limited by GeneratorBaseDefinition.MaxAffectDistance, so make sure it has quite a large value.

Which Entities are affected by GravityGeneratorComponent

GravityGeneratorComponent is added to both client and server Prefabs of Entities to affect not only entities present on the server, but also client-only Entities such as debris fractures.

The CanBeAffectedByGravity Data is required for Entities that should be affected by the gravity generator. Only root Entities are affected.

Entity is affected by the gravity generator only when CanBeAffectedByGravity.GravityTypeBitField overlaps with GravityGeneratorDefinition.GravityTypesBitField. These bit fields are generated automatically at runtime.

Changes of CanBeAffectedByGravity Data are not observed. If CanBeAffectedByGravity Data is updated while Entity is in the affected range of the gravity generator, the generator won't start affecting this Entity.

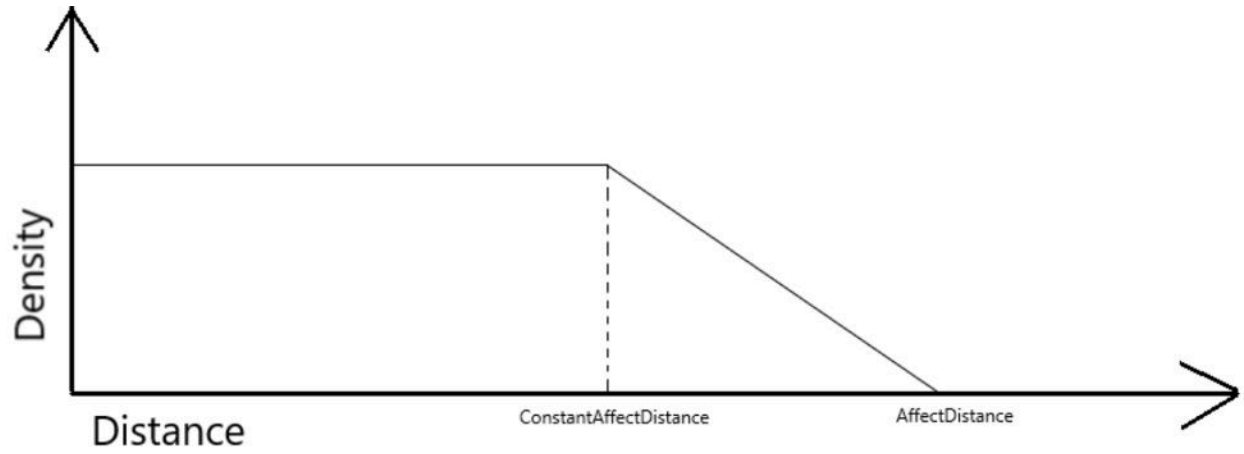
In addition, the gravity is applied only to simulated Entities, so the affected Entity must have IsSimulatedData present, as the Gravitate Job that is invoked for affected Entities and calculates gravitational acceleration, uses WhenSimulatedAttribute.

Atmosphere generator

AtmosphereUpdate Job group

If you want to perform some Jobs before or after the atmosphere update, you can sync with the AtmosphereUpdate JobGroup.

Density calculations formula



The density at a given point of affect area of the atmosphere generator is constant and equal to the value generated by the atmosphere generator if the distance between this point and the atmosphere generator is smaller than ConstantAffectDistance. Then the density begins to decrease linearly and when the distance reaches AffectDistance, the density has a value of 0.

Which Entities are affected by AtmosphereGeneratorComponent

The AirData Data is required for Entities that should be affected by the atmosphere generator. Only root Entities are affected.

In addition, the atmosphere update is applied only to simulated Entities, so the affected Entity must have IsSimulatedData present, as the UpdateAtmosphere Job that is invoked for affected Entities and calculates overall density, uses WhenSimulatedAttribute.

Planet atmosphere

AtmosphereGeneratorComponent is attached to the planet prefab. Fields ConstantAffectDistance and AffectDistance of AtmosphereGeneratorObjectBuilder are multiplied by planet radius in AtmosphereGeneratorProcessorComponent to scale atmosphere generator values according to the planet radius. Just like with the planet gravity, max affect range of planet atmosphere is limited by GeneratorBaseDefinition.MaxAffectDistance.