

Törmäystyökalujen ohjeet

Ilkka Pokkinen ja Topias Paljakka

Sisällys

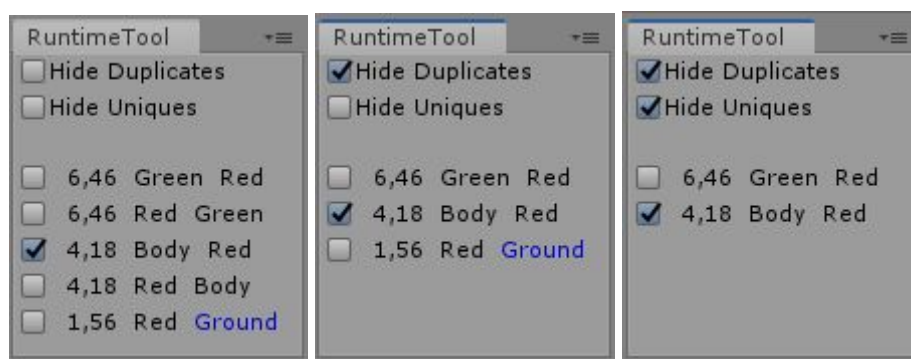
- 2 Editoriskriptiä
 - RuntimeTool
 - CollisionUITool
- 2 MonoBehaviouria
 - VisualSingleton
 - Visuals
- 1 Apuluokka
 - DictionaryGuard
- 1 Shadertiedosto
 - TransparentHighlight-shader

RuntimeTool

Editor ikkuna, joka aukeaa yläpalkista Mevea > Tools > RuntimeTool.

Työkalu pitää lokia törmäyksistä ja antaa mahdollisuuden uudelleen visualisoida jo tapahtuneita törmäyksiä. Lokista voidaan suodattaa pois duplikaatit ja uniikit törmäykset (Kuva 1).

Uniikkeja ovat törmäykset joissa toisella peliobjektilla ei ole CollisionDetector-komponenttia, ne merkitään sinisellä lokissa, kuvassa "Ground" on peliobjekti jossa ei ole CollisionDetector-komponenttia. Suurin osa törmäyksistä ei ole uniikkeja eli molemmissa peliobjekteissa on CollisionDetector-komponentti, josta johtuen ne tuottavat lokiin kaksi merkintää, joista toisen voi halutessaa suodattaa pois.

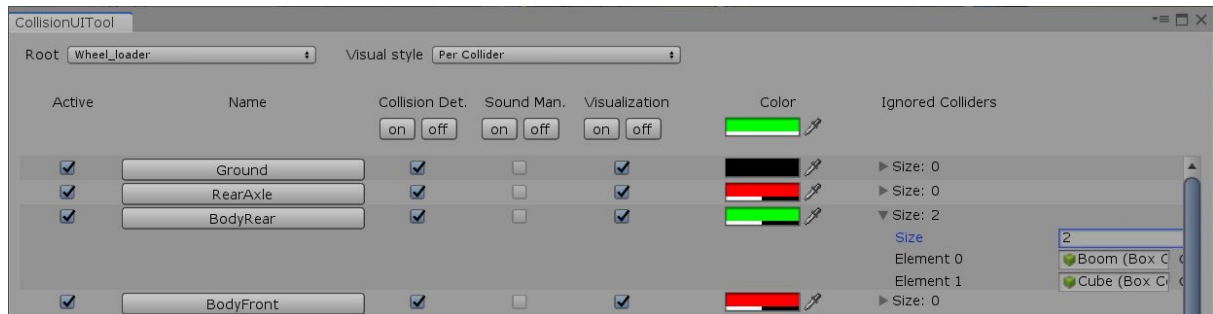


Kuva 1: RuntimeTool-komponentin toiminnallisuus

CollisionUITool

Editorikkuna, joka aukeaa yläpalkista Mevea > Tools > CollisionUITool.

Malleja sisältävissä peliobjekteissa tulee olla "AssetRoot"-tagi. Näiden peliobjektien lapsiobjekteista tulee kantaobjekteja (Kuva 2, vasen yläreuna). Työkalu etsii kantaobjekteista CollisionDetector-komponentilla varustettuja lapsiobjekteja, jotka näkyvät työkalun alaosassa. Lapsiobjektin nimeä klikkaamalla voidaan valita objekti ja tuplaklikkaamalla scenekamera kohdistuu objektiin.

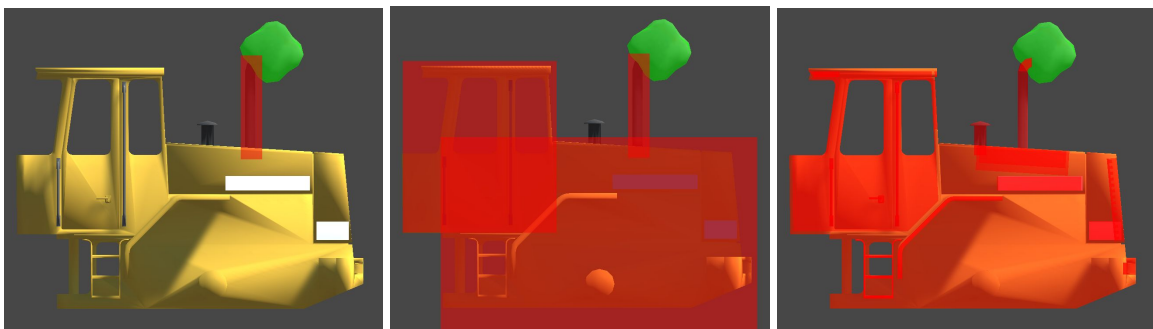


Kuva 2: CollisionUITool

CollisionUITool-työkalulla voidaan aktivoida ja deaktivoida CollisionDetector, SoundManager ja Visuals -komponentteja, muuttaa törmäyksen visualisoinnin väriä ja hallinnoida CollisionDetector-komponentin sivuuttamia collidereita.

Visual Style -valikolla muutetaan törmäyksen visualisoinnin tyyliä (Kuva 3)

- Per Collider: törmäävät colliderit visualisoidaan
- Compound: kaikki törmäävissä peliobjekteissa kiinni olevat colliderit visualisoidaan
- Mesh: Törmäävien peliobjektien meshit visualisoidaan



Kuva 3: Törmäyksen visualisointi tyylit: Per Collider, Compound ja Mesh.

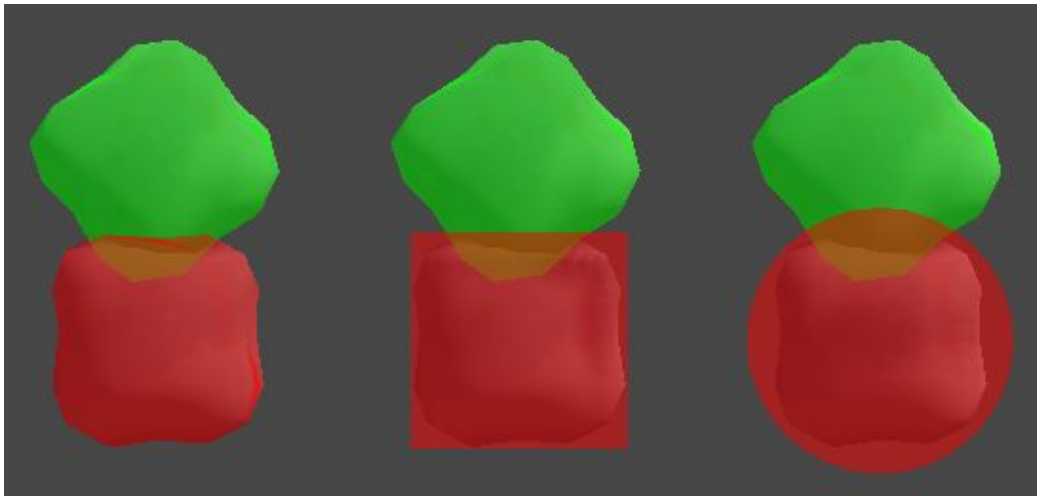
VisualSingleton

Singleton MonoBehaviour joka pitää kirjaa (DictionaryGuard) kaikista sen hetkisistä törmäys visualisoinneista (CollisionEventArgs) ja visualisoi ne halutulla tavalla.

VisualSingletonilla on kaksi moodia: "Gizmos" ja "Build". Visualisointi toteutetaan "Gizmos"-moodissa Unityn visuaalisella virheenkorjaus työkalulla (Gizmos) ja

“Build”-moodissa luomalla uusia peliobjekteja, joille annetaan materiaaliksi annettu visualisointi materiaali. Visualisoinnin väri riippuu yleensä Visuals-komponentista mutta jos törmäys on uniikki visualisointi on sininen kuten RuntimeToolin lokimerkintä.

Gizmot ja “Build”-moodissa luodut peliobjektit tarvitsevat Mesh-objektin, joka riippuen valitusta visualisointi tyylistä (VisualStyle) saadaan päätettyä peliobjektin Collider- tai MeshFilter-komponenteista. Huom. Collider visualisointi toimii vain Sphere-, Box- ja MeshColliderien kanssa (Kuva 4).



Kuva 4: Törmäys visualisointi Mesh-, Box- ja SphereCollidereilla.

Visuals

MonoBehaviour joka törmätessä kääntää VisualSingletonin visualisoimaan tapahtunutta törmäystä annetun ajan verran.

DictionaryGuard

Apuluokka, joka on käytännössä Dictionary, mutta poistaa solun vain jos soluun ei enään viitata muualta. Käytetään VisualSingletonissa, jottei törmäyksiä tarvitse visualisoida useaan kertaan ja jos samaan törmäykseen viitataan useista lähteistä (esim kahdesta eri Visuals-komponentista) visualisointi poistetaan vain kun kaikki lähteet poistavat sen.

TransparentHighlight-shader

Yksinkertainen shaderi (Kuvat 3 ja 4), jota voi halutessaan käyttää visualisointiin “Build”-moodissa. Shaderillä saa aikaan Gizmon tapaisen läpinäkyvän törmäys visualisoinnin, joka on aina päällimmäisenä eli näkyy muiden peliobjektien läpi.