Van de Scene zijn objecten het meest cachable en groundinfo. Non-interactable objects vallen daaronder evenals groundinformation.

Widgets in runescape 2 zijn gedeeltelijk cachable in de zin van scherm coördinaten, afmeting etc. *Het is hoogstwaarschijnlijk efficienter om de widgets id’s te cachen, en deze id’s te linken aan een beschrijving van de inhoud, mogelijk met behulp van #defines.*

Mogelijk is het om een bank object te creëren dat een representatie is van het in runescape begevende object. Naast widgets kent runescape ook het Menu element dat acties bevat met een eventueel gespecificeerd object of een optie.

* Characters (NPC’s, Player)
* GroundItems
* Veranderlijke objecten (tree, ore, door, etc)

Solution: Clusterbased 64x64.

Object indexed caching met bijbehorende collisionflag map

Widgets cachen we in een hash op basis van ID. Deze caching zal plaats vinden met behulp van hit-ratio. Dat willen zeggen dat de widgets die maar weinig worden opgehaald het eerst uit de cache gegooid zullen worden als de cache te groot wordt.

Ook cachen:

* Quests . Tasks, daily tasks
  + *Sowieso voor (veel) later, dus deze kan genegeerd worden.*
* Bank
* Equipment
* Stats

Een cluster bevat dus zowel collision map alsmede de object map

Travel algoritme in Runescape:

* Dijkstra variant
  + kan gebruik maken van clusters
  + kan gebruik maken van requirements
  + kan gebruik maken van eerder gelopen paden
    - *Dit kost relatief veel geheugenruimte (opslaan van paden e.d.) ten opzichte van de hit ratio van een dergelijk pad. Het runnen van Dijkstra kost niet zo veel tijd.*