

Code

Modifier les spells de base

Projet Codinsa2015/Server/Entities/EntityHero.cs -> constructeur. Il y a les 4 spells initialisés par défaut pour le debug. On peut les changer :

```
Spells = new List<Spell>();
Spells.Add(Server.Spells.SpellFactory.Meteor(this));
Spells.Add(Server.Spells.SpellFactory.BroForce(this));
Spells.Add(Server.Spells.SpellFactory.Stasis(this));
Spells.Add(Server.Spells.SpellFactory.MaximumGravity(this));
```

Ajouter des spells actifs

Cf le fichier Codinsa2015/Server/Spells/SpellFactory.cs => plein d'exemples.

La plupart des spells sont réalisables sans avoir à créer de sous-classe de BasicSpell.

Ajouter de l'équipement

Cf le fichier Codinsa2015/Server/Equip/ShopDatabase.cs => plein d'exemples.

Passer par le lobby / phase de pick pour tester des IAs

Dans Codinsa2015/Server/Scene.cs => mettre SKIP_PICKS = false.

Puis lancer des IAs.

Répertoires des IAs

IA c# : Projet Codinsa2015.Client => Program.cs

IA C++ / Java : cf le dossier Codinsa2015.Client\Codinsa2015.Client\Views\Clank => y'a les projets C++ et Java dans des dossiers dedans. /!\ La quasi-totalité du code là dedans est auto-générée, pas toucher !

Contrôleur visuel de debug

Petit topo

Pour lancer le jeu avec le contrôleur visuel de debug, il faut exécuter le projet Codinsa2015.DebugHumanControler.

Quand on lance le jeu avec le contrôleur visuel de debug, on a un aperçu du jeu entrain de se dérouler + la main sur un joueur avec des contrôles types LoL (clic droit pour se déplacer, AZER pour les sorts, 1 2 pour les consommables). Pour passer en mode édition, on peut appuyer sur CTRL (celui de gauche).

Le mode édition

Appuyer sur Tab => switch entre les modes 'Land' et 'Entities' (cf bouton en haut à gauche de l'écran)

- Mode land :
 - clic gauche / droit : ajouter supprimer du terrain.
 - I, O : agrandir / rétrécir la largeur du pinceau
- Mode entities
 - Clic droit sur une entité => affiche un menu

- **L : charge la map dans le fichier “map.txt” (super utile => charge la map de test)**
- M : sauvegarde la map en cours dans le fichier “map.txt”
- Pleins d’autres raccourcis : cf plus loin.

En mode édition, une console est aussi disponible (on peut l’agrandir avec le petit bouton “+” en haut à droite, et la déplacer en la drag & dropant). Pour taper du texte, il faut cliquer dans la zone de texte située en bas de la console.

Commandes utiles:

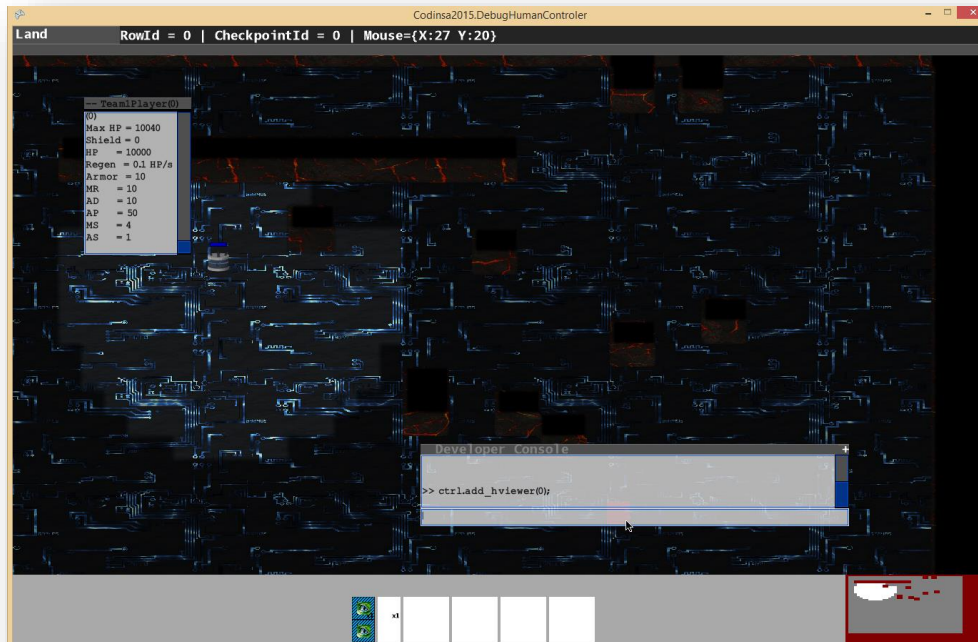
- “ctrl.help();” => affiche un message d’aide qui indique d’autres commandes sympas.
- “ctrl.print_commands();” => affiche des commandes plutôt pratiques

Le langage : c’est un genre de C# interprété maison. Il peut interopérer avec les objets existants etc...

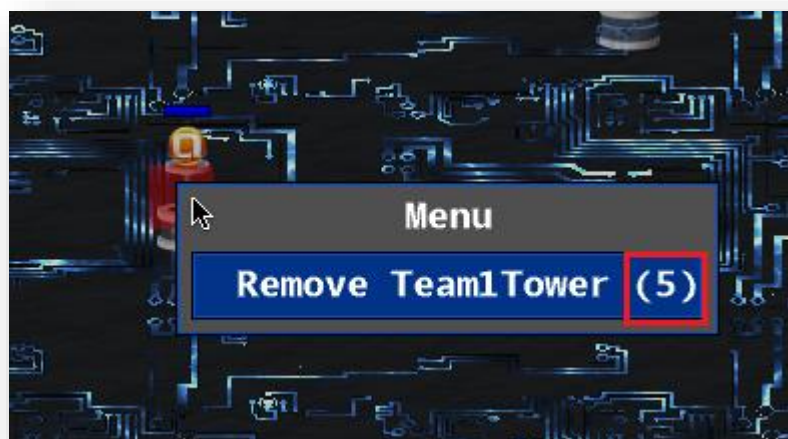
Suivre l'évolution des stats du héros

Console :

```
>> ctrl.add_hviewer(0) ; // h pour héros
```



Pour suivre l'état d'une entité en particulier, clic droit dessus => son id est affiché.



Puis :

```
>> ctrl.add_viewer(<id>);
```