

DOCUMENTATION

GAUNTLET - PASCAL STUDIO

Adresse GIT: git@git.epitech.eu:/caucho_g/cpp_indie_studio/

Membre de l'équipe de développement :

guillaume.cauchois@epitech.eu (master)

quentin.baudet@epitech.eu

robin.grattepanche@epitech.eu

thomas.fossaert@epitech.eu

pierre.staszak@epitech.eu

adrien.warin@epitech.eu

TABLE DES MATIÈRES :

| | |
|----------------------------------|-----------|
| Class - Score | 7 |
| Auteur | 7 |
| Description | 7 |
| Syntaxe du fichier de sauvegarde | 7 |
| Attributs | 7 |
| Méthodes | 7 |
| Class - Position | 9 |
| Auteur: | 9 |
| Description: | 9 |
| Attributs: | 9 |
| Méthodes: | 9 |
| Class - Configuration | 10 |
| Auteur | 10 |
| Description | 10 |
| Attributs | 10 |
| Méthodes | 10 |
| Class - IAudio | 11 |
| Auteur | 11 |
| Description | 11 |
| Méthodes | 11 |
| Class - Music | 12 |
| Auteur | 12 |
| Description | 12 |
| Attributs | 12 |
| Méthodes | 12 |
| Class - SoundSystem | 13 |
| Auteur | 13 |
| Description | 13 |
| Attributs | 13 |
| Méthodes | 13 |
| Class - ParserInputFile | 14 |
| Auteur | 14 |
| Description | 14 |
| Syntaxe du fichier de sauvegarde | 14 |

| | |
|------------------------------|-----------|
| Attributs | 14 |
| Méthodes | 14 |
| Class - Device | 15 |
| Auteur | 15 |
| Description | 15 |
| Attributs | 15 |
| Méthodes | 15 |
| Class - DeviceManager | 17 |
| Auteur | 17 |
| Description | 17 |
| Attributs | 17 |
| Méthodes | 17 |
| Class - foodStack | 18 |
| Auteur: | 18 |
| Description: | 18 |
| Attributs: | 18 |
| Méthode: | 18 |
| Class - goldStack | 19 |
| Auteur: | 19 |
| Description: | 19 |
| Attributs: | 19 |
| Méthode: | 19 |
| Class - Ranged | 19 |
| Auteur: | 19 |
| Description: | 19 |
| Attributs: | 19 |
| Méthodes | 19 |
| Class - Melee | 19 |
| Auteur: | 21 |
| Description: | 21 |
| Attributs: | 21 |
| Méthodes | 21 |
| Class - Tank | 21 |
| Auteur: | 22 |
| Description: | 22 |
| Attributs: | 22 |
| Méthodes | 22 |
| Class - Light | 22 |
| Auteur: | 23 |

| | |
|----------------------------------|-----------|
| Description: | 23 |
| Attributs: | 23 |
| Méthodes | 23 |
| Class - IResources | 25 |
| Auteur | 25 |
| Description | 25 |
| Méthodes | 25 |
| Class - Warrior | 26 |
| Auteur: | 26 |
| Description: | 26 |
| Attributs: | 26 |
| Méthodes | 26 |
| Class - Mage | 26 |
| Auteur: | 27 |
| Description: | 27 |
| Attributs: | 27 |
| Méthodes | 27 |
| Class - Archer | 27 |
| Auteur: | 28 |
| Description: | 28 |
| Attributs: | 28 |
| Méthodes | 28 |
| Class - MapManager | 29 |
| Auteur | 29 |
| Description | 29 |
| Syntaxe du fichier de sauvegarde | 29 |
| Attributs | 29 |
| Méthodes | 29 |
| Class - Character | 32 |
| Auteur | 32 |
| Description | 32 |
| Attributs | 32 |
| Méthodes | 32 |
| Class - GameObject | 34 |
| Auteur | 34 |
| Description | 34 |
| Attributs | 34 |
| Méthodes | 34 |
| Class - Loot | 35 |

| | |
|------------------------------|-----------|
| Auteur | 36 |
| Description | 36 |
| Attributs | 36 |
| Méthodes | 36 |
| Class - NPC | 36 |
| Auteur | 37 |
| Description | 37 |
| Attributs | 37 |
| Méthodes | 37 |
| Class - RenderManager | 38 |
| Auteur | 38 |
| Description | 38 |
| Attributs | 38 |
| Méthodes | 38 |
| Class - Skeleton | 40 |
| Auteur | 40 |
| Description | 40 |
| Attributs | 40 |
| Méthodes | 40 |
| Class - Script | 40 |
| Auteur | 41 |
| Description | 41 |
| Attributs | 41 |
| Méthodes | 41 |
| Class - Spawner | 42 |
| Auteur | 42 |
| Description | 42 |
| Méthodes | 42 |
| Class - BOSS | 42 |
| Auteur | 43 |
| Description | 43 |
| Attributs | 43 |
| Méthodes | 43 |
| Class - Zombie | 43 |
| Auteur | 44 |
| Description | 44 |
| Attributs | 44 |
| Méthodes | 44 |
| Class - Particle | 45 |

| | |
|--------------------------|-----------|
| Auteur | 45 |
| Description | 45 |
| Attributs | 45 |
| Méthodes | 45 |
| Class - Animation | 46 |
| Auteur | 46 |
| Description | 46 |
| Attributs | 46 |
| Méthodes | 46 |

Class - Score

Auteur

guillaume.cauchois@epitech.eu

Description

Cette classe a pour but de manipuler les scores obtenus dans le jeu GAUNTLET par les joueurs. Cette classe permet la génération de sauvegarde dans un fichier et de la même manière de charger des scores depuis un même fichier

Syntaxe du fichier de sauvegarde

Pseudo_du_joueur=42

où :

- Pseudo_du_joueur est le pseudo que le joueur a choisi dans le menu de choix des personnages
- 42 est le score réalisé par le joueur

Attributs

| Nom de l'attribut | Description |
|---------------------------|---|
| <code>_leaderboard</code> | Il s'agit d'une map sous la forme de couple de <i>[PSEUDO, SCORE]</i> |

Méthodes

| Nom de la méthode | Description |
|----------------------------------|---|
| <code>getLeaderBoard</code> | Permet d'obtenir le contenu de la variable <code>_leaderboard</code> |
| <code>addEntry</code> | Permet de créer un nouveau couple de score " <i>joueur -> score</i> " et de l'ajouter à la variable <code>_leaderboard</code> |
| <code>addLineSaveToScores</code> | Permet d'appeler la méthode <code>addEntry</code> en fonction d'une ligne du fichier de sauvegarde respectant la syntaxe défini ci-dessus. |
| <code>saveInFile</code> | Permet de générer un fichier de sauvegarde respectant la syntaxe ci-dessus. |
| <code>loadFromSave</code> | Permet de lire un fichier de sauvegarde respectant la syntaxe ci-dessus et d'écraser l'intégralité des scores enregistré dans le <i>this->_leaderboard</i> par le contenu du |

| | |
|--|------------------------|
| | fichier de sauvegarde. |
|--|------------------------|

Class - Position

Auteur:

adrien.warin@epitech.eu

Description:

Cette classe a pour but de manipuler les différents axes de positions (X, Y, Z), des objets, personnages etc..

Cette classe encapsule l'objet de la librairie Ogre3D : "Ogre::vector3()"

Attributs:

| Nom de l'attribut | Description |
|-------------------|--|
| _x | de type INT, représente l'axe X |
| _y | de type INT, représente l'axe Y |
| _z | de type INT, représente l'axe Z |
| _vector | de type Ogre::Vector3 qui est un vecteur de positions, il contient nos 3 positions X, Y et Z |

Méthodes:

| Nom de la méthode | Description |
|-----------------------------|--|
| get[X,Y,Z]Position() const; | Renvois la position actuelle de l'axe X, Y ou Z |
| getVector() const; | renvois la position actuelle de tous les axes contenus dans le vecteur _vector |
| set[X,Y,Z]Position(int); | Attribut l'argument de type INT à l'axe X, Y ou Z |
| setPosition(int, int, int) | Attribut les arguments de type INT aux trois axes X, Y et Z |

Class - Configuration

Auteur

guillaume.cauchois@epitech.eu

Description

Cette classe a pour but dans le cas d'une utilisation dans le menuCore de générer la configuration nécessaire au lancement du gameCore. Par exemple, il contiendra l'ensemble des joueurs (PC) avec le choix du personnage, la configuration de ces inputs, et le reste de la configuration du menu. Cette configuration sera donc envoyé sous la forme d'un pointeur au gameCore pour être utilisé pour le jeux.

Attributs

| Nom de l'attribut | Description |
|-------------------|--|
| _players | Il s'agit d'un vecteur d'objet Pc contenant les players correctement configuré pour l'utilisation dans le jeux en lui même |
| _score | c'est un entier qui représente le score du joueur qu'il obtient dans le jeu |

Méthodes

| Nom de la méthode | Description |
|-------------------|--|
| getPlayers | Permet d'obtenir le contenu de la variable <i>_players</i> |
| setPlayers | Permet d'assigner la variable <i>_players</i> |
| addScorePoint | Permet de modifier la valeur <i>_score</i> |
| getScore | Permet de récupérer le Score du jeu |
| getKey | Permet de savoir si la clef a été récupérée |

Class - IAudio

Auteur

pierre.staszak@epitech.eu

Description

Cette classe est une interface regroupant toute les fonctions utiles pour charger et lire des musiques, ainsi que des bruitages.

Méthodes

| Nom de la méthode | Description |
|-------------------|---|
| loadAudio | Permet de charger le son |
| playAudio | Permet de lire le son. Cette fonction existe en deux versions : une sans paramètres, nécessitant donc d'avoir chargé le son au préalable. Et une autre prenant en paramètre le chemin du son, pour pouvoir le charger et le lire dans la même fonction. |
| stopAudio | Utile pour les musiques. Permet d'arrêter la lecture d'une musique en cours. |
| setAudioVolume | Permet de changer le volume du son |
| getAudioVolume | Permet d'obtenir le volume actuel du son |
| getStatus | permet de savoir si le son est joué ou non |
| getCurrentName | permet d'obtenir le nom de la musique en cours |
| setCurrentName | permet de configurer le nom de la musique en cours |
| setLoop | permet de lire en boucle la musique |

Class - Music

Auteur

pierre.staszak@epitech.eu

Description

Cette classe hérite de l'interface IAudio et représente une musique.

Attributs

| Nom de l'attribut | Description |
|-------------------|--|
| _music | Il s'agit d'un attribut propre à la librairie SFML (sf::Music) qui contient la musique sous le format .ogg |
| _currMusic | représente la musique en cours |
| _musicFilePath | représente le chemin du dossier de la musique |

Méthodes

Celles de la classe IAudio

Class - SoundSystem

Auteur

pierre.staszak@epitech.eu

Description

Cette classe hérite de l'interface IAudio et représente un bruitage.

Attributs

| Nom de l'attribut | Description |
|-------------------|--|
| _volume | Représente le volume sonore du bruitage |
| _buffer | Il s'agit d'un attribut propre à la librairie SFML (sf::SoundBuffer) qui précharge le bruitage sous le format .ogg |
| _sound | Contient le bruitage préchargé au préalable |
| _currSoundName | représente le nom du son |

Méthodes

Celles de la classe IAudio

Class - ParserInputFile

Auteur

guillaume.cauchois@epitech.eu

Description

Cette classe a pour but de lire un fichier de configuration de binding d'input. C'est-à-dire qu'elle permet de lire des fichiers contenant la configuration des claviers, manettes pour un joueur donné.

Syntaxe du fichier de sauvegarde

Id_enum_eventType=id_enum_OISKeyCode

où :

- *Id_enum_eventType* correspond à la valeur de l'énumération associée à l'événement pour lequel on souhaite conserver / récupérer une touche associée.
- *id_enum_OISKeyCode* correspond à la valeur de l'énumération associée à la touche dans l'enum `OIS::KeyCode` associé à un événement donné.

Attributs

| Nom de l'attribut | Description |
|-----------------------|--|
| <code>_binding</code> | Il s'agit d'un buffer utilisé pendant la génération de la map de mappage. Elle ne doit pas être utilisée en dehors de cette classe |

Méthodes

| Nom de la méthode | Description |
|--------------------------------------|--|
| <code>getArrayBindingFromFile</code> | Permet de générer une map de mappage en fonction du contenu d'un fichier généré en fonction du nom passé en paramètre. |
| <code>saveBindingInFile</code> | Permet de mettre à jour le fichier de configuration du mappage en fonction de la map passée en paramètre Remarque : Mise à jour de l'attribut <code>_binding</code> avec la map passée en paramètre |

Class - DeviceManager

Auteur

guillaume.cauchois@epitech.eu

Description

Cette classe permet la gestion des Devices, pour la création notamment.

Attributs

| Nom de l'attribut | Description |
|-------------------|--|
| _iManager | Pointeur vers l'inputManager de la librairie OIS |

Méthodes

| Nom de la méthode | Description |
|-------------------|--|
| getIManager | Obtenir le pointeur vers le manager de la librairie OIS. |
| createObject | Permet de créer un OIS::Object |

Class - foodStack

Auteur:

quentin.baudet@epitech.eu

Description:

Cette classe permet la gestion du power up food qui increase la health et qui hérite de loot

Attributs:

Aucun

Méthode:

| | |
|----------------------|------------------------------------|
| increaseHealth(void) | Augmenter l'attribut score des PCs |
|----------------------|------------------------------------|

Class - goldStack

Auteur:

quentin.baudet@epitech.eu

Description:

Cette classe permet la gestion du power up gold qui increase le score et qui hérite de loot

Attributs:

Aucun

Méthode:

| | |
|---------------------|-------------|
| increaseScore(void) | Augmenter l |
|---------------------|-------------|

Class - Ranged

Auteur:

adrien.warin@epitech.eu

Description:

Cette classe a pour but de représenter les personnages qui ont la particularité d'attaquer à distance (Mage / Archer)

Attributs:

Récupère les attributs de Pc.

Méthodes

Récupère les méthodes de Pc.

Class - Melee

Auteur:

adrien.warin@epitech.eu

Description:

Cette classe a pour but de représenter les personnages qui ont la particularité d'attaquer au corps à corps. Elle hérite de la classe Pc.

Attributs:

Récupère les attributs de Pc.

Méthodes

Récupère les méthode de Pc.

Class - Tank

Auteur:

adrien.warin@epitech.eu

Description:

Cette classe représente le personnage de type "Tank" il hérite de la classe "Melee" c'est un personnage avec beaucoup de point de vie qui peut encaisser beaucoup de dégâts.

Attributs:

Récupère les attributs de Melee.

Méthodes

Récupère les méthode de Melee.

Class - Light

Auteur:

robin.grattepanche@epitech.eu

Description:

Cette classe est une encapsulation de la class Ogre::Light, elle permet de gérer la lumière comme son type, sa couleur, sa position ou encore sa direction.

Attributs:

| Nom de l'attribut | Description |
|-------------------|--|
| _light | C'est l'entité lumière en elle-même |
| _scene | La scène à laquelle la lumière est rattachée |
| _name | Le nom de la lumière dans la scène |
| _type | Le type de la lumière > LT_POINT (point de lumière portant dans toutes les directions) > LT_SPOTLIGHT (émet un cône de lumière dans une direction) > LT_DIRECTIONNAL (lumière lointaine émettant des rayons parallèles) |
| _diffColour | La couleur des rayons lumineux |
| _specColour | Le couleur du centre de la lumière |
| _position | La position de la lumière (indisponible pour un type LT_DIRECTIONNAL) |
| _direction | L'orientation de la lumière (inutile pour un type LT_POINT) |

Méthodes

| Nom de la méthode | Description |
|-------------------|---------------------------------|
| setDiffuseColour | change la couleur de la lumière |

| | |
|-------------------|--|
| setSpecularColour | permet de changer la couleur du centre de la lumière |
| setPosition | permet de changer la position de la lumière |
| setDirection | permet de changer l'orientation de la lumière |

Class - IResources

Auteur

quentin.baudet@epitech.eu

Description

Cette classe est une interface regroupant toute les fonctions utiles pour gérer les ressources

Méthodes

| Nom de la méthode | Description |
|-------------------|---------------------------------|
| hide | Permet de cacher le GameObject |
| display | Permet d'afficher un GameObject |

Class - Warrior

Auteur:

adrien.warin@epitech.eu

Description:

Représente un personnage qui a une attaque plus puissante que le Tank mais qui a moins de vie que ce dernier. Il hérite de la classe Melee.

Attributs:

Récupère les attributs de Melee.

Méthodes

Récupère les méthode de Melee.

Class - Mage

Auteur:

quentin.baudet@epitech.eu

Description:

Personnage joueur avec 80 HP et 175% d'attaque, hérité de Ranged

Attributs:

Récupère les attributs de Ranged.

Méthodes

Récupère les méthode de Ranged.

Class - Archer

Auteur:

quentin.baudet@epitech.eu

Description:

Personnage joueur avec 90 HP et 110% d'attaque, hérité de Ranged

Attributs:

Récupère les attributs de Ranged

Méthodes

Récupère les méthode de Ranged

Class - MapManager

Auteur

guillaume.cauchois@epitech.eu

Description

Cette classe a pour but de générer des entités au chargement de la map. La map est générée automatiquement suite au parsing d'un fichier map.cfg (default path : /dist/bin/map.cfg)

Le fichier de configuration de cette carte doit respecter une syntaxe donnée.

Syntaxe du fichier de sauvegarde

```
# map.cfg - Map configuration - Gauntlet
[OPTIONS]
TEXTURE_GROUND path/to_ground/texture.png
[OBJECTS]
TYPE_ENTITY MyEntity1 X,Y,Z    X,Y,Z    /path/to/texture
```

Attributs

| Nom de l'attribut | Description |
|-------------------|---|
| _path | PRIVATE Il s'agit du path vers le fichier de configuration de la map (default : dist/bin/map.cfg) |
| _buffer | PRIVATE Il s'agit d'un buffer qui utilisé pour stocké toutes les lignes utilisés pour la génération de la map. Elle contient en réalité le contenu du fichier de configuration sans les lignes vides et les commentaires |
| _tree | PRIVATE Il s'agit du pointeur vers la tête de l'arbre abstrait. L'arbre abstrait est généré après lecture de la variable buffer et permettra de générer, après lecture, toutes les entités à poser dans la map. |
| _options | PRIVATE Il s'agit d'une map d'option, elle contient sous la forme de tableau "clef/valeur" des options spécifiées dans la sections [OPTIONS] du fichier. |

Méthodes

| Nom de la méthode | Description |
|------------------------|---|
| computeAbstractTree | Permet de générer l'abstract tree en fonction du contenu de la variable buffer. |
| generateMap | Permet en fonction du contenu de la variable _tree, de générer toutes les entités demandés par le fichier map.cfg La fonction prend en paramètre un RenderManager qui sera utilisé pour générer / stocker toutes ces entités |
| getOptions | Permet de récupérer une option spécifiée dans le fichier de configuration dans la section [OPTIONS]. |
| saveInFile | Permet de générer un fichier de sauvegarde respectant la syntaxe ci-dessus. |
| feed | PRIVATE Cette méthode permet d'ajouter au buffer une ligne par rapport à la ligne passée en paramètre |
| loadFromFile | PRIVATE Cette fonction appelle en boucle la méthode feed pour toutes les lignes utiles à la générations des entités (hors lignes vides et commentaires). |
| isSectionDeclaration | PRIVATE Cette fonction permet de savoir si une ligne passée en paramètre respecte le format d'une ligne de SECTION. Format: <i>[SECTION_NAME]</i> |
| isParameterDeclaration | PRIVATE Cette fonction permet de savoir si une ligne passée en paramètre respecte le format d'une ligne d'option. Format: KEY VALUE |
| isObjectDeclaration | PRIVATE Cette fonction permet de savoir si une ligne passée en paramètre respecte le format d'une ligne de déclaration d'entité. Format: <i>TYPE_ENTITY MyEntity1 X,Y,Z X,Y,Z /path/to/texture</i> |
| addObjectASTNode | PRIVATE Permet d'ajouter au noeud de la section OBJECTS une nouvelle objet |
| addParameterASTNode | PRIVATE Permet d'ajouter au noeud OPTIONS une nouvelle |

| | |
|-------------------------------|--|
| | option |
| addSectionASTNode | PRIVATE Permet d'ajouter une nouvelle section à l'arbre abstrait |
| addObjectFieldASTNode | PRIVATE Permet d'ajouter à une node d'objet, une propriété comme un NOM, un TYPE, ou une POSITION. |
| getFieldFromObjectDeclaration | PRIVATE Permet en fonction d'une ligne de déclaration d'objet de récupérer la valeur d'une propriété donnée en paramètre. Par exemple dans la ligne : <i>TYPE_ENTITY MyEntity1 X,Y,Z X,Y,Z /path/to/texture</i> <i>Si je passe en paramètre TYPE, je vais pouvoir récupérer une std::string("TYPE_ENTITY")</i> |
| getSectionName | PRIVATE Permet pour au passage d'une ligne déclaration de section de récupérer le nom effectif de cette nouvelle section. Par exemple pour la ligne [SECTION_NAME] la fonction me renverra une std::string("SECTION_NAME") |
| ObjASTNodeToGameObj | PRIVATE Permet de générer un gameObject grâce au RenderManager passé en paramètre et une node d'AST. |

Class - Character

Auteur

thomas.fossaert@epitech.eu

Description

Cette classe représente tous les personnages possibles dans le jeu (Warrior, Tank, Mage et Archer)

Attributs

| Nom de l'attribut | Description |
|-------------------|---|
| _health | entier qui représente les points de vie du joueur |
| _attack | entier qui représente les points d'attaque du joueur |
| _cooldown | entier qui représente le temps entre deux attaques du joueur |
| _speed | entier qui représente la vitesse de déplacement du joueur |
| _range | entier qui représente la distance à laquelle le joueur peut toucher un objet ou un ennemi |
| _state | Etat du personnage (Standing, Walking, Attacking..) |
| _csound | liste des sons du personnage |

Méthodes

| Nom de la méthode | Description |
|-------------------|--|
| takeDamage | Permet d'enlever des points de vie au joueur quand il subit des dégâts |
| attack | Permet de faire attaquer le personnage et d'enlever des points de vie à la cible en argument |

| | |
|-----------------|--|
| isAlive | Permet de savoir si le personnage est toujours en vie (si ses points de vie sont supérieur à 0) après avoir subi une attaque |
| getDistance | Permet de savoir quelle est la distance entre le personnage et une entité quelconque |
| getRange | Permet de savoir la distance dont le personnage a besoin pour pouvoir attaquer |
| launchAnimation | Lance une animation spécifique au personnage |

Class - GameObject

Auteur

thomas.fossaert@epitech.eu

Description

Cette classe permet la création d'objet dans le jeu (personnage, ennemi, environnement etc..)

Attributs

| Nom de l'attribut | Description |
|-------------------|---|
| mAnimationState | permet de savoir le type d'animation |
| mEntity | permet de savoir quelle est l'entité |
| mNode | représente un noeud dans une scène |
| mScript | représente le script de l'intelligence artificielle |
| mPosition | représente la position dans l'espace |
| mAnimation | pointeur sur la classe animation (cf Classe-Animation) |
| mNodeName | représente le nom du noeud |
| _aura | particule pour différencier les personnage |
| _unset | particule étant utilisée pendant l'obtention des bonus (food et gold) |

Méthodes

| Nom de la méthode | Description |
|-------------------|---|
| setOgreBase | Permet d'initialiser une entité dans l'espace |
| launchScript | Permet de lancer le script de l'intelligence artificielle |
| Animate | Permet de lancer une animation |

| | |
|-------------|---|
| unsetEntity | Permet de retirer une entité |
| getPosition | Permet de connaître la position de l'objet |
| hide | Permet de cacher une entité sans la retirer |
| getNodeName | Permet de connaître le nom du noeud |

Class - Loot

Auteur

thomas.fossaert@epitech.eu

Description

Cette classe représente les bonus que le joueur peut ramasser tout au long du jeu

Attributs

| Nom de l'attribut | Description |
|-------------------|-------------------------------|
| _id | Entier qui représente le loot |

Méthodes

| Nom de la méthode | Description |
|-------------------|---|
| isTaken | Permet de savoir si le joueur à ramasser ou non le loot |

Class - NPC

Auteur

thomas.fossaert@epitech.eu

Description

Classe qui représente tout personnage non joueur (ennemis, boss..)

Attributs

| Nom de l'attribut | Description |
|-------------------|------------------------------|
| _id | Entier qui représente le NPC |

Méthodes

Pas de méthode.

Class - RenderManager

Auteur

guillaume.cauchois@epitech.eu

Description

Cette classe a pour but la génération de toutes les entités du Gauntlet de type GameObject, cela comprends donc en partie la génération des Npc, des murs, des Pc. Celle-ci génère automatiquement l'attribution des id pour les objets.

Attributs

| Nom de l'attribut | Description |
|-------------------|--|
| _factory | PRIVATE Il s'agit d'un vecteur de pointeur sur membre, qui contient l'ensemble des fonction de génération des GameObjects |
| _entites | PRIVATE Il s'agit d'un vecteur contenant l'intégralité des gameObject généré dans le cadre de notre RenderManager (id auto-incrément) |

Méthodes

| Nom de la méthode | Description |
|---------------------|---|
| createGameObject | Permet la génération d'une entité demandé |
| createZombieObject | Permet la génération d'une entité "ZOMBIE" |
| createBossObject | Permet la génération d'une entité "BOSS" |
| createSkeltonObject | Permet la génération d'une entité "SKELTON" |
| createSpawnerObject | Permet la génération d'une entité "SPAWNER" |
| createDoodadObject | Permet la génération d'une entité "DOUDAD" |
| createFoodObject | Permet la génération d'une entité "FOOD" |

| | |
|---------------------|---|
| createGoldObject | Permet la génération d'une entité "GOLD" |
| createWarriorObject | Permet la génération d'une entité "WARRIOR" |
| createMageObject | Permet la génération d'une entité "MAGE" |
| createArcherObject | Permet la génération d'une entité "ARCHER" |
| createTankObject | Permet la génération d'une entité "TANK" |

Class - Skeleton

Auteur

thomas.fossaert@epitech.eu

Description

Classe qui représente le skeleton qui est l'ennemi de base qui attaque à distance dans jeu. Il hérite de la classe NPC.

Attributs

Pas d'attribut.

Méthodes

Hérite des méthodes de la classe NPC.

Class - Script

Auteur

thomas.fossaert@epitech.eu

Description

Classe qui regroupe les différents scripts pour les IA de NPC

Attributs

Pas d'attribut.

Méthodes

| Nom de la méthode | Description |
|-------------------|------------------------|
| ZombieScript | Script du NPC Zombie |
| SkeletonScript | Script du NPC Skeleton |
| BossScript | Script du NPC Boss |

Class - Spawner

Auteur

guillaume.cauchois@epitech.eu

Description

Cette entité peut faire spawner des entités Npc autour d'elle.

Méthodes

| Nom de la méthode | Description |
|-----------------------|--|
| makeSpawnRandomEntity | Permet de faire spawner à côté du spawner une entité aléatoire |
| makeSpawnZombie | Permet de faire spawner à côté du spawner un zombie |
| makeSpawnSkeleton | Permet de faire spawner à côté du spawner un Skeleton |
| makeSpawnBoss | Permet de faire spawner à côté du spawner un boss |

Class - BOSS

Auteur

thomas.fossaert@epitech.eu

Description

Classe qui représente le boss d'un niveau. Elle hérite de la classe NPC.

Attributs

Hérite des attributs de NPC

Méthodes

Hérite des Méthodes de la classe NPC

Class - Zombie

Auteur

thomas.fossaert@epitech.eu

Description

Cette classe représente le zombie qui est un ennemi de base au corps à corps. Il hérite de la classe NPC.

Attributs

Hérite des attributs de NPC

Méthodes

Hérite des Méthodes de la classe NPC

Class - Particle

Auteur

adrien.warin@epitech.eu

Description

Cette classe permet de créer des particules et les attacher à des objets.

Attributs

| Nom de l'attribut | Description |
|-------------------|--|
| _name | Nom de la particule |
| _path | Chemin du fichier de configuration de la particule |
| _ps | particule systeme |

Méthodes

| Nom de la méthode | Description |
|-------------------|--|
| setParticle | cette méthode permet de créer la particule |
| setObjet | cette méthode permet de rattacher une particule à un objet (exemple: particule de flamme à un objet) |
| destroy | permet de détruire la particule |

Class - Animation

Auteur

pierre.staszak@epitech.eu

Description

Classe qui représente

Attributs

| Nom de l'attribut | Description |
|-------------------|--|
| _name | Nom de l'animation |
| _looping | Détermine si une animation est en train de loop |
| _dist | Représente la distance parcourue par l'objet lors de l'animation |
| _speed | Représente la vitesse de l'animation |
| _castingTime | Représente le temps que mets une animation à se lancer |
| _animationState | Pointeur sur Ogre::AnimationState, représente l'état d'une animation |

Méthodes

| Nom de la méthode | Description |
|-------------------|---|
| isLooping | Renvoie TRUE si l'animation est train de tourner |
| isCasted | Renvoie TRUE si l'animation est finie d'être lancée |
| addTime | Modifie la position du temps |
| disable | Désactive l'animation |
| enable | Active l'animation |

| | |
|-------------------|--|
| setSpeed | Modifie la vitesse de l'animation |
| getName | Renvoie le nom de l'animation |
| launch | Lance l'animation |
| hasEnded | Renvoie TRUE si l'animation est terminée |
| getAnimationState | Renvoie le animationState de l'animation |

Class - IndieExeption

Auteur:

robin.grattepanche@epitech.eu

Description:

Permet de gérer nos messages d'exeption

Attributs:

pas d'attribut.

Méthode:

| Nom de la méthode | Description |
|-------------------|--------------------------------------|
| what | permet d'envoyer le message d'erreur |

Class - Key

Auteur:

thomas.fossaert@epitech.eu

Description:

Cette classe représente les clés que le personnage doit ramasser pour avancer

Attributs:

Aucun

Méthode:

Hérite de Loot

Class - Doodad

Auteur:

thomas.fossaert@epitech.eu

Description:

Cette classe représente les objets de l'environnement (mur, poteaux, tables etc..)

Attributs:

| Nom de l'attribut | Description |
|-------------------|-------------------------------------|
| _id | id de l'objet |
| _Sx | représente la position sur l'axe x |
| _Sy | représente la position sur l'axe y |
| _Sz | représente la position sur l'axe z |
| _orientation | représente l'orientation de l'objet |
| _deco | représente le nom de l'objet |

Méthode:

Hérite de GameObject

Class - HUD

Auteur:

robin.grattepanche@epitech.eu

Description:

Cette classe représente l'HUD (Head Up Display) du jeu c'est à dire les éléments étant affichés à l'écran mais en dehors de la scène du jeu, par exemple : la vie des joueurs, le score...

Attributs:

| Nom de l'attribut | Description |
|-------------------|--|
| _overlayManager | Manager des différents éléments |
| | |
| _panel | la zone consacrée à la vie des joueur et au score |
| _panelKey | la zone pour l'affichage de la clé |
| _panelEnd | représente les différents écrans de fin du jeu |
| _overlay | Element permettant de relier toutes les zones à afficher |
| _vecText | vecteur représentant les différents joueurs. |
| _vecHp | vecteur représentant la vie des joueurs |
| _hudScore | élément représentant l'affichage du score |

Méthode:

| Nom de la méthode | Description |
|-------------------|--|
| createOnePlayer | créer l'affichage d'un joueur unique |
| addLife | Met en place l'affichage du score |
| setupPanel | Met en place l'affichage du background |

| | |
|----------------|--|
| | pour la vie des joueurs |
| setupPanelKey | Met en place de la case contenant l'affichage de la clé |
| setupPanelEnd | Met en place l'élément afin d'afficher les écrans de fin du jeu |
| initTextPlayer | Remplis le vecteur "vecText" de chaque joueur |
| createPlayers | Met en place la partie de l'overlay contenant la vie des joueurs |
| initScore | Met en place la partie de l'overlay contenant le score |
| initLife | Initialise la vie des différents joueurs |
| updateLife | Met à jour la vie des joueurs |
| updateScore | Met à jour le score |
| updateKey | Met la clé en couleur si elle a été récupérée durant la partie |
| getPlayerHp | remplis le vecteur "_vecHp" avec la vie des joueurs (début de la partie) |
| showHUD | affiche l'HUD pendant le jeu |
| showEnd | affiche l'Ecran de fin donné en paramètre |

Fichier Examples.particle

Auteur:

adrien.warin@epitech.eu

Description:

Ce fichier regroupe tous les différents scripts des particules

Exemple d'un script de particule (particule de la boule de feu du mage):

```
particle_system Examples/Fireball
{
    material           Examples/Flare2
    particle_width      700
    particle_height     700
    cull_each           false
    quota               10000
    billboard_type      oriented_self

    // Area emitter
    emitter Point
    {
        angle           1
        emission_rate    50
        time_to_live     0.3
        duration         0.1
        direction        1 0 0
        position         0 50 0
        velocity_min     2500
        velocity_max     2500
        colour_range_start 0.9 0.4 0 1
        colour_range_end 0.8 0.4 0 1
        width            60
        height           60
        depth            150
    }

    // Gravity
    affector LinearForce
    {
        force_vector     0 -100 0
        force_application add
    }

    // Fader
    affector ColourFader
    {
        red -0.25
        green -0.25
        blue -0.25
    }
}
```