

DOCUMENTATION PROJETPYTHON

MAIN_MENU.PY :

Classe Menu

`__init__(self)`

- Initialise les variables nécessaires pour le menu.

`run_simulation(self)`

- Quitte le menu et lance la simulation avec les paramètres définis dans le menu.

`main_menu(self)`

- Crée et affiche le menu principal. L'utilisateur peut définir les paramètres de la simulation ici.

GRAPHIQUE.PY :

Classe Affichage_graphique

`__init__(self, world)`

- Initialise l'objet avec le monde donné.

`affichage_bob_food(self, screen)`

- Affiche les Bobs et la nourriture sur l'écran. Les positions sont calculées en fonction de la caméra et du facteur de zoom.

`affichage_grid_iso(self, screen)`

- Affiche une grille isométrique sur l'écran.

`input_box(self, screen)`

- Affiche une boîte de saisie pour l'utilisateur. Utilisé pour changer la taille du monde.

`run(self, tick_interval)`

- Lance la boucle principale de la simulation. Gère l'affichage, les entrées utilisateur et les mises à jour du monde.

`graph(self)`

- Affiche un graphique montrant l'évolution de la population de Bobs et de la nourriture au fil du temps.

WORLD.PY :

Classe World

`__init__(self, size=100, food_per_day=120, food_value=200, maxEnergy=1000)`

- Initialise le monde avec les paramètres donnés.

`spawn(self, mob, nb)`

- Fait apparaître un certain nombre de Bobs ou de nourriture dans le monde.

`tick_update(self)`

- Met à jour le monde à chaque tick. Gère le mouvement des Bobs, la consommation de nourriture, la reproduction, etc.

BOB.PY :

Classe Bob

`__init__(self, x, y, world, velocity=1, mass=1, perception=0, memory_space=0, energy=100, maxEnergy=200)`

- Initialise le Bob avec les paramètres donnés.

`move(self)`

- Déplace le Bob dans une direction aléatoire.

`eat(self)`

- Le Bob essaie de manger de la nourriture s'il est sur la même case qu'elle.

`self_reproduce(self)`

- Le Bob se reproduit, créant un nouveau Bob sur la même case.

`update_tick(self)`

- Met à jour le Bob à chaque tick. Gère le mouvement, la consommation de nourriture, la reproduction, etc.

FOOD.PY :

Classe Food

`__init__(self, x, y, world, food_value=200)`

- Initialise la nourriture avec les paramètres donnés.

`kill(self)`

- Supprime la nourriture du monde.

`beEaten(self, value)`

- La nourriture est consommée par un Bob. Retourne la quantité de nourriture consommée.