

## Projet LU2IN002 - 2022-2023

Numéro du groupe de TD/TME : Groupe 11

Nom : CAMPAN	Nom : KACI CHAOUCHE
Prénom : Ina	Prénom : Sarah
N° étudiant : 21108406	N° étudiant : 21108956

Thème choisi (en 2 lignes max.)

Simulation de l'extinction des dinosaures

Schéma UML des classes vision fournisseur (dessin "à la main" scanné ou photo acceptés)

Vous le trouverez dans les pages suivantes

Checklist des contraintes prises en compte:	Nom(s) des classe(s) correspondante(s)
Classe contenant un tableau ou une ArrayList	Simulation
Classe avec membres et méthodes statiques	Agent, Dinosaur, Simulation, TerrainAgent
Classe abstraite et méthode abstraite	Agent, Dinosaur, DinosaurAvecAiles, DinosaurSansAiles,
Interface	Voler, Carnivore, Herbivore
Classe avec un constructeur par copie ou clone()	Meteorite
Définition de classe étendant Exception	FormatException
Gestion des exceptions	TestSimulation

Présentation brève de votre projet (max. 10 lignes) : texte libre expliquant en quoi consiste votre projet.

Notre projet effectue la simulation d'une scène qui vous plonge dans la fin de l'ère mésozoïque, lorsque la météorite Chicxulub a anéanti les noms les plus anciens de la zoologie; comme le Tyrannosaurus Rex et le Diplodocus. Pourtant, de ces espèces émergent nos compagnons actuels les oiseaux, ce qui atteste de la survie de quelques

dinosaures. En particulier, les fossiles du dinosaure *Serikornis sungei*, aussi surnommé Silky, nous montrent que les dinosaures ayant survécu étaient de ceux qui possédaient des ailes, mais qui ne pouvaient pas voler.

Nous avons ainsi modélisé un terrain hypothétique où ces dinosaures existaient entourés de ressources situées aléatoirement sur le territoire. Grâce à une classe Test Simulation, nous pouvons suivre l'évolution dans le temps du nombre d'agents et de ressources; de même que l'on peut déterminer si Silky est une des espèces survivantes.

*Copier / coller vos classes et interfaces à partir d'ici :*

*Agent*

*Carnivore*

*DinosaureAvecAiles*

*DinosaureSansAiles*

*Diplodocus*

*FormatException*

*Fougere*

*Herbivore*

*Insecte*

*Meteorite*

*Oiseau*

*Pterodactyle*

*Silky*

*Simulation*

*TerrainAgent*

*TRex*

*Voler*

# «Joua Vars»

## Simulation

- terrain: Terrain
  - terrainAgent: TerrainAgent
  - tabRessources: ArrayList<Resource>
  - tabAgents: ArrayList<Agent>
- F. nbMax Ressources: int
- F. nbMax Agents: int
- cpt Ressources: int
  - cpt Agents: int
  - nbLignes: int
  - nb colonnes: int

S.O.F PERIODICITE METEORITE

S.O.F NB MAX ETAPES

- ◇ valide (int, int): boolean
- ◇ Simulation (int, int, int, int)
- ◇ Simulation (int, int)
- ◇ verifierTousAgents(): void
- ◇ verifierTousRessources(): void
- ◇ etape(): void

- ◇ remove (Agent): Agent
- ◇ remove (Resource): Resource
- ◇ affiche TousAgents(): String
- ◇ affiche TousRessources(): String
- ◇ tombe Meteorite (Meteorite): void
- ◇ getNbLignes(): int
- ◇ getColonnes(): int

← Terrain <C>

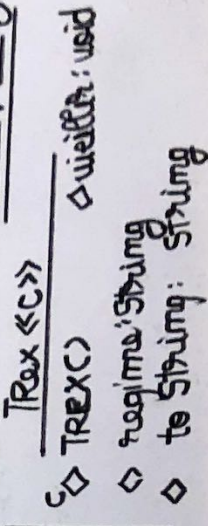
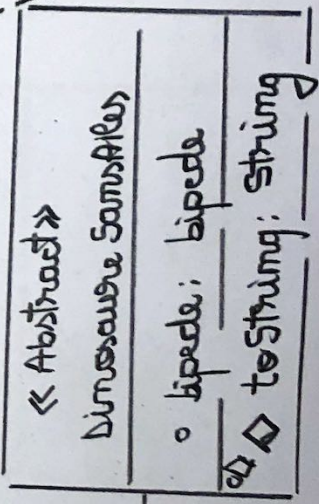
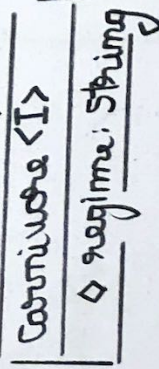
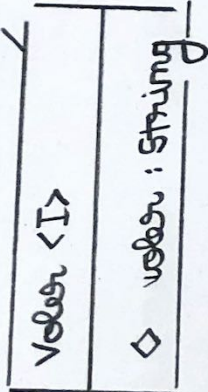
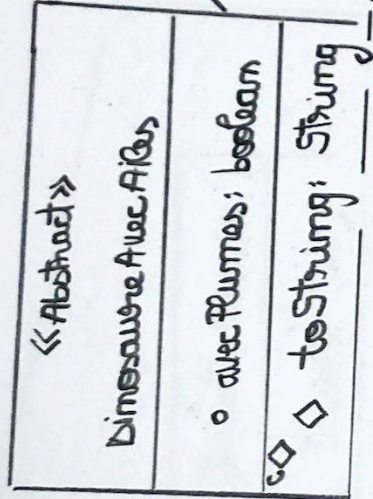
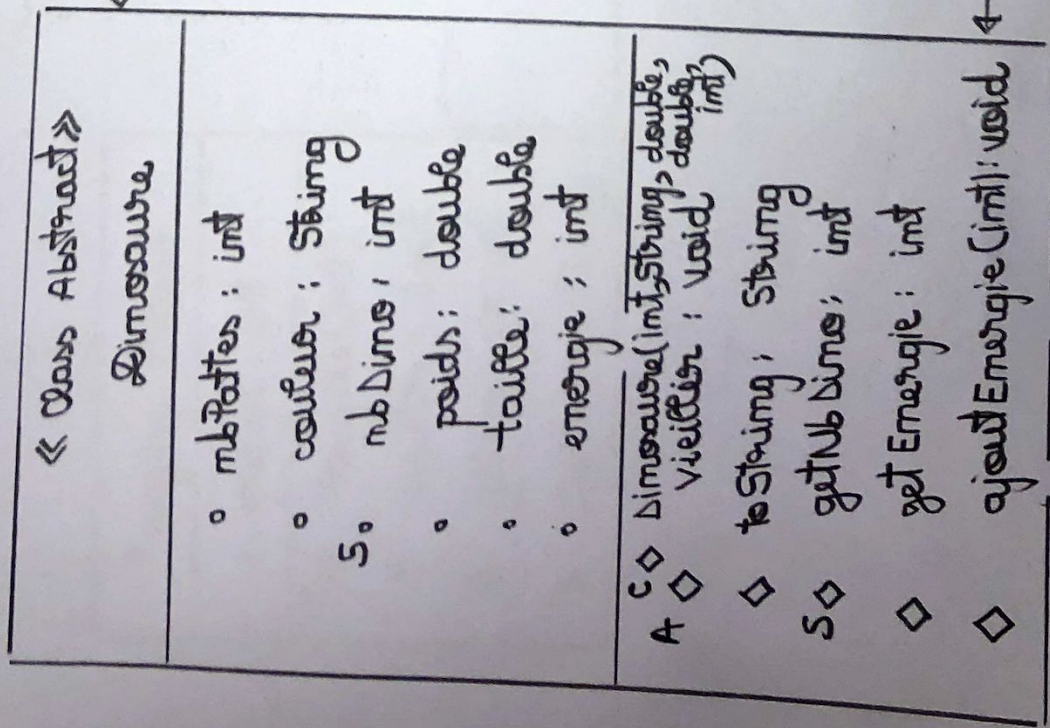
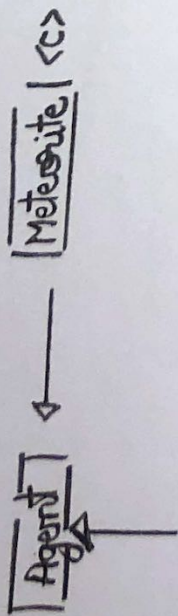
← TerrainAgent <C>



« Class » Agent	
<ul style="list-style-type: none"> <li>• x: int</li> <li>• y: int</li> <li>• nbAgent: int</li> <li>• id: int</li> </ul>	
<ul style="list-style-type: none"> <li>◇ Agent()</li> <li>◇ initialization(): void</li> <li>◇ distance(int, int): double</li> <li>◇ replace(int, int): void</li> <li>◇ toString(): String</li> <li>◇ coordPossibles(int, int): boolean</li> <li>◇ getNbAgents(): int</li> <li>◇ getX(): int</li> <li>◇ getY(): int</li> <li>◇ getId(): int</li> </ul>	

« Class » TerrainAgent	
<ul style="list-style-type: none"> <li>• terrain: Agent[][]</li> <li>• NBLIGNESMAX: int</li> <li>• NBCOLONNESMAX: int</li> <li>• nbLignes: int</li> <li>• nbColonnes: int</li> </ul>	
<ul style="list-style-type: none"> <li>◇ setValides(int, int): boolean</li> <li>◇ TerrainAgent(int, int)</li> <li>◇ estVide(int, int): boolean</li> <li>◇ getCase(int, int): Agent</li> <li>◇ setCase(int, int, Agent): boolean</li> <li>◇ VideCase(int, int): Agent</li> <li>◇ getLignes(): int</li> <li>◇ getColonnes(): int</li> </ul>	







Ressource

<C>

Fougere «Class»
◦ energie : int
◊ Fougere(int)
◊ vieillir : void
◊ getEnergie : int
◊ toString : String

Insecte «Class»
◊ Insecte(int)

Gisement «Class»
◦ estOuef : boolean
◦ age : int
◦ energie : int
◊ Gisement(int)
◊ echelon() : void
◊ vieillir() : void
◊ toString : String
◊ getAge() : int
◊ getEtat() : boolean
◊ getEnergie() : int

«Java Class» Meteorite
◦ poids : double
◦ diametre : double
◦ x : int
◦ y : int
◊ Meteorite(double, double)
◊ clone() : Meteorite
◊ atterrir Meteorite(int, int) : void
◊ toString : String
◊ getX() : int
◊ getY() : int
◊ diametre() : double
◊ poids : double

[\*]

<C>

Agent