Assignment 3
FIT2099
Object Oriented Design And Implementation

Done by:
Sutulova Tatiana 30806151
Elaf Abdullah Saleh Alhaddad 31063977

Work Breakdown Agreement

Our team consists of two team members, which makes it easier to equally separate the workload. Most of the work will be done during the Zoom meetings to ensure that the time spent by both students to complete given tasks is equal and to be able to solve any issues and problems that the student may face, while completing their parts, is discussed and solved immediately.

General task	Small tasks	Deadline	Student
Improvements to existing functionality	Combine FollowBehaviour and FollowFoodBehaviour	26/10/2020	Tatiana
	Test and debug existing functionality	27/10/2020	Elaf
Implementation of the functionality	Agilisaurus, AgilisaurusEgg classes	28/10/2020	Elaf
	Archaeopteryx, ArchaeopteryxEgg class	28/10/2020	Tatiana
	HuntingAction class	28/10/2020	Elaf
	Water class, ThirstyBehaviour, DrinkAction classes	29/10/2020	Tatiana
	Second map implementation	2/11/2020	Tatiana, Elaf

	Adding different game modes	3/11/2020	Tatiana, Elaf
Changes to existing documentation	Class Diagram	4/11/2020	Tatiana
	Sequence Diagrams	4/11/2020	Elaf
	Documentation	4/11/2020	Tatiana, Elaf
Finalising	Testing	4/11/2020	Elaf
	Commenting	4/11/2020	Tatiana
	Report about the possible improvements in engine	6/11/2020	Elaf, Tatiana
Submission	Submit all the required files	6/11/2020	Elaf, Tatiana
Sutulova Tatiana,30806151	I agree with the WBA	Signature: 	
Elaf Abdullah Saleh, Alhaddad 31063977	I agree with the WBA	Signature: 	

Content

1. Description of new classes:

Class Name	Description	Attributes and methods that were added
Water	• Child of the **Ground** class, which is needed for **Dinosaur** to drink from it • Actors except **Archaeopteryx** are unable to step on it • Represented by the '~' character and gives the dinosaur 25 drinking points if the interaction happens.	Attributes: • **DRINK_POINTS** - the amount of points the **Dinosaur** water level will increase by if the dinosaur drinks from it. It has a constant value of 25. Methods: • **canActorEnter()** - allows only an instance of **Archaeopteryx** to enter the **Water** cell • **blocksThrownObjects()** - does not allow to throw any objects on the **Water** cell • **getDRINK_POINTS( )**- Returns the number of **DRINK_POINTS** that water gives.
EndGameAction	Action which is responsible for ending the game. It is called when the user chooses quit the game option from the main menu	Methods: • **execute()** - removes the player from the map to terminate the game

	Ends the game by removing the player from the map.	• **menuDescription() -** returns an informative string about the action
ThirstyBehaviour	All the dinosaurs implement thirsty behaviour if their water level is less than 50 but the food level is prioritised, so the food level must be above 30, otherwise the dinosaur will be implementing **HungryBehaviour.**	**getAction() -**searches for the nearest water source and makes the dinosaur follow it. Once they reach it, the dinosaur will drink the water
DrinkAction	Special action that allows the dinosaur of any species to drink from the water cell.	Attributes: • **Water water,** the instance of water class, that the **DrinkAction** will be conducted on. (the target water cell) Methods: • **execute()** Performs the Action, that allows the dinosaur to drink from the target water pool • **menuDescription()** Returns the informative string.

HuntingAction	This action allows the Allosaur to hunt for the **Agilisaurus** and Allows the **Archaeopteryx** to hunt for all other dinosaurs. This class and its algorithm is explained in detail in its respective sequence diagram.	Attributes: • **target** - Actor class, the subject of the hunt. Methods: • **execute() -** Performs the Action that allows the dinosaur to hunt after its target • **menuDescription() -** Returns the informative string.
DisplayInt	The child class of **Display,** which is responsible for reading integers as a user input.	Methods: • **readInt**
Archaeopteryx	• **Archaeopteryx** is carnivore and implements carnivore behaviours as well as common dinosaur behaviours: ○ eating corpses and eggs ○ Hunting any kind of a dinosaur (except its own), attacking it and killing for food. ○ Searching for food or dinosaurs to attack if the food level is low ○ Searching for water if the water level is low ○ If age, water level and food level is sufficient, looking for a dinosaur of the same specie to breed	Constructor only

	○ Implements wander behaviour	
ArchaeopteryxEgg **AgilisaurusEgg**	● The egg is created (instantiated) if: ○ Dinosaurs of the same kind breed in the female dinosaur inventory ○ Bought from the **VendingMachine** by the **Player** ● Children of the **EggItem** class	Method: ● **hatchEgg()-** removes the egg item from the location and adds the dinosaur instance (based on the type of the egg) at the same location or at the nearest empty location.
Agilisaurus	● Omnivorous and implements both vegetarian and carnivore behaviours: ○ eats hay, grass and fruits, as well as corpses and eggs ○ Searching for food if the food level is low ○ Searching for water if the water level is low ○ If age, water level and food level is sufficient, looking for a dinosaur of the same specie to breed ○ Implements wander behaviour ● The main target for the **Allosaur** hunting.	Method: ● **playTurn()** allows the dinosaur to perform ● **GrazingGrassAction** if it is standing on the grass cell

2. Description of changes implemented to existing classes:

Class Name	Changes

Player	The player class was adapted to work with the Challenge mode of the game. New attributes that were added: ● **moveChoice -** represents how many rounds the player has chosen the game to last ● **ecoPointsChoice -** represents the number of eco points the player wants to reach till the end of the game ● **turns -** counter, which records how many rounds of the game passed from the beginning Methods that were added or improved: ● **setChoice() -** is called in application class when the **Player** chooses the Challenge mode. It sets the values for **moveChoice** and **ecoPointsChoice,** according to the player's choice. ● **playTurn() -** was improved by adding functionalities allowing the user to quit the game and if the user plays for the proper number of rounds (according to their choice), it calculates whether the player won or lost.
Application	In the **main()** method: ● second map implementation, which allows movement from the north edge of the first map to the south edge of the second map and backwards ● **Player** is allowed to choose the mode of the game: ○ SandBox mode ○ Challenge mode ● **Player** is allowed to quit the game at any time during the game. ● The app will not terminate until the user chooses to do so and all the invalid inputs are properly treated by representing the appropriate message and allowing to re-enter the value.
VendingMachine	In the method **menuVendingMachine()** there were two new types of eggs added for sale: **ArchaeopteryxEgg** and **AgilisaurusEgg.**

FeedAction	In the method **execute()** player is able to feed the dinosaur with another two types of carnivorous food: **ArchaeopteryxEgg** and **AgilisaurusEgg.**
BreedingAction	Adapted to make sure that new types of dinosaurs: **Agilisaurus, Archaeopteryx** are able to breed.
Dinosaur	Since Dinosaurs are supposed to be able to experience thirst, seek for water sources and drink the following attributes and methods were added: ● Attributes: ○ **waterLevel** - which is an integer that represents the dinosaur water level. Every dinosaur initially has a water level of 50. ● Methods: ○ **getWaterLevel()**- returns the current water level ○ **decreaseFoodWaterLevel()** - improved version of **decreaseFoodLevel()** method, which decreases both **waterLevel** and **foodLevel** by 1 ○ **drink()** - allows the dinosaur to drink water, which increases its water points ○ **isConcious()** - besides other factors such as low **hitPoints** and low **foodLevel** , if dinosaur's water level is 0 it becomes unconscious ○ **playTurn()** - if dinosaurs' **foodLevel** is more than 30 but the **waterLevel** is less than 30, the dinosaur implements **ThirstyBehaviour**, seeking for the water source. Since dinosaurs may be herbivorous, carnivorous or omnivorous (which is both) we decided to add two attributes: ● **is_carnivore** ● **is_vegetarian**

	They are booleans that makes it easier to check what kind of food suits the dinosaur. Getters to get the values of these attributes were added as well: • **getIs_carnivore()** • **getIs_vegetarian()**