De La Salle University
COLLEGE OF COMPUTER STUDIES

MCO1 Test Cases

Submitted by:
Benavidez, Joseph Royce
Sayat, John Christian
CCPROG3 - S14

Tile Class

Method	#	Test Description	Sample Input Data	Expected Output	Actual Output	P/F
isOccupied()	1	Determines if tile is empty	Tile with no animal	false	false	P
	2	Determines if a tile has an animal	Tile with “Tiger” Object	true	true	P
isWater()	1	Determines if the tile is water(lake)	Tile(1, 3, “Lake”)	true	true	P
isDen()	1	Check if a tile is base den	Tile(3, 0, “Den”)	true	true	P

Animal Class

Method	#	Test Description	Sample Input Data	Expected Output	Actual Output	P/F
canCapture()	1	Determine if a stronger animal can capture a weaker one	Lion(7), Cat (3) on normal tile	true	true	P
	2	Checks if a weaker animal cannot capture a stronger one	Cat(3), Tiger(6)	false	false	P
	3	Checks if Rat can capture Elephant	Rat(1), Elephant(8)	true	true	P
	4	Checks if weaker animal cannot capture stronger one	Cat(3), Tiger(6)	false	false	P
isSwimmer()	1	Determines if an animal can swim	Rat(1), Tiger(6), Elephant(8)	True, true, false	True, true, false	P
getRank()	1	Return the rank of the animal	Lion(7)	7	7	P
getSpecies()	1	Returns the species of the animal	Tiger(6)	“Tiger”	“Tiger”	P
move(int x, int y)	1	Moves an animal to a new coordinate	Tiger(6, 2, 3) → move(3, 4)	(3, 4)	(3, 4)	P

Board Class

Method	#	Test Description	Sample Input Data	Expected Output	Actual Output	P/F
isValidMove()	1	Determines if an animal can move to an empty tile	Tiger(6) → move(1,1)	true	true	P
	2	Prevents an animal from moving out of bounds	Lion(7) → move(-1, 0)	true	true	P
	3	Prevents a non-swimming animal from moving into water	Dog(3) → move (1, 3) (lake)	false	false	P
	4	Allows a swimming animal to move into water	Rat(1) → move(1, 3) (lake)	true	true	P
	5	Allows an animal to capture a weaker one	Tiger(6) → move(1, 1,) (Cat)	true	true	P
	6	Prevents a weaker animal from capturing a stronger one	Cat(3) → move (1, 1) (Tiger)	false	false	P
moveAnimal()	1	Moves an animal to a valid tile	Elephant(8, 0, 2) → move(1, 2)	(1, 2)	(1, 2)	P
moveAnimal()	2	Moves an animal to an occupied tile after capture	Tiger(6) → move(1, 1) (cat)	(1, 1)	(1, 1)	P
getTile()	1	Returns the correct tile type	(3, 0) → Den	“Den”	“Den”	P
	2	Returns if a tile is water	(1, 3) → Lake	true	true	P

Player Class

Method	#	Test Description	Sample Input Data	Expected Output	Actual Output	P/F
addAnimal()	1	Adds an animal to the player's list	player.addAnimal(new Animal("Lion", 7, 6, 0, "Li1))	animals.size() == 1	1	P
	2	Can add multiple animals in a player's list	Add "Elephant" and "Leopard"	animals.size() == 2	2	P
winCondition()	1	Determines if a player reaches the opponent's den/base, the player wins	"Tiger" piece moves to (3,8) for left-side player	true	true	P
	2	Determines when a player does not win if no animal reaches the base	No animals at (3,8) or (3,0)	false	false	P
	3	Right-side player wins	"Lion(Li2)" moves to (3,0)	true	true	P
getAnimalSymbol()	1	Gets an existing animal	player.getAnimalSymbol ("Ti1")	Tiger object	Tiger object	P
	2	Gets a non-existent animal	player.getAnimalSymbol ("El2") -> not owned by player	null	null	P
getName()	1	Gets the player's name	player.getName()	"John"	"John"	P
getAnimals()	1	Gets the player's list of animals after adding	player.getAnimals() -> Lion, Leopard	["Lion", "Leopard"]	["Lion", "Leopard"]	P

Game Class

Method	#	Test Description	Sample Input Data	Expected Output	Actual Output	P/F
preGame()	1	Players pick an animal from a shuffled list	1(Rat) → 2(Cat)	Players assigned chosen animals	Players assigned chosen animals	P
	2	Determines first player based on animal rank	Tiger(6) vs. Wolf(4)	Tiger(6) starts	Tiger(6) starts	P
startGame()	1	Moves an animal to an empty tile	Tiger(6) → D	Tiger moves right	Tiger moves right	P
	2	Prevents an invalid move (out of bounds)	Lion(7) → W (top edge)	“Invalid move”	“Invalid move”	P
	3	Allows valid attack on weaker animals	Tiger(6) → Cat(2)	“Tiger captured Cat”	“Tiger captures Cat”	P
	4	Prevents invalid attack on stronger animal	Cat(2) → Tiger(6)	“Invalid attack”	“Invalid attack”	P
	5	Ends game when a player reaches the base	Rat(1) → (Base)	“Player X wins!”	“Player X wins!”	P
setupAnimals()	1	Assigns animals to both players correctly	Player1 → 8 animals, player2 → 8 animals	8 animals per player	8 animals per player	P