Test design

Setup UserManagmentTest

Name	Class	Stage
Setup1	UserManagment	usm = new UserManagment<>()

Name	Class	Stage
Setup2	UserManagment	Setup1 User<String> newUser1 = new User<>("anderson", 20) User<String> newUser2 = new User<>("juan", 30) User<String> newUser3 = new User<>("carolina", 10) usm.addPlayer(newUser1) usm.addPlayer(newUser2) usm.addPlayer(newUser3)

Name	Class	Stage

Setup3	UserManagment	setup1()
		User<String> newUser1 = new User<>("anderson", 10)
		User<String> newUser2 = new User<>("juan", 20)
		User<String> newUser3 = new User<>("carolina", 30)
		User<String> newUser4 = new User<>("sebastian", 5)
		User<String> newUser5 = new User<>("jhon", 2)
		usm.addPlayer(newUser1)
		usm.addPlayer(newUser2)
		usm.addPlayer(newUser3)
		usm.addPlayer(newUser4)
		usm.addPlayer(newUser5)

Setup AdjListGraphTest

Name	Class	Stage
Setup1	AdjListGraph	adjL = new AdjListGraph<String>(false, true, 0)

Name	Class	Stage

Setup2	AdjListGraph	setup1()
		adjL.addVertex("Bob's House")
		adjL.addVertex("Patrick's House")
		adjL.addVertex("Scissors Shop")
		adjL.addVertex("Card Shop")
		adjL.addEdge("Bob's House", "Patrick's House",20)
		adjL.addEdge("Bob's House", "Squidward's House",35)
		adjL.addEdge("Bob's House", "Card Shop",130)
		adjL.addEdge("Bob's House", "Scissors Shop", 120)

Name	Class	Stage

Setup3	AdjListGraph	adjL = new AdjListGraph<String>(false, true, 9)
		adjL.addVertex("Bob's House")
		adjL.addVertex("Squidward's House")
		adjL.addVertex("Patrick's House")
		adjL.addVertex("Planton's Restaurant")
		adjL.addVertex("Vehicle School")
		adjL.addVertex("Scissors Shop")
		adjL.addVertex("Massage Shop")
		adjL.addVertex("Card Shop")
		adjL.addVertex("Krabby Crustacio")
		adjL.addEdge("Bob's House", "Patrick's House",20)
		adjL.addEdge("Bob's House", "Card Shop",130)
		adjL.addEdge("Bob's House", "Scissors Shop", 120)
		adjL.addEdge("Bob's House", "Squidward's House", 35)
		adjL.addEdge("Squidward's House","Patrick's House",40)

		adjL.addEdge("Squidward's House", "Scissors Shop", 60)
		adjL.addEdge("Patrick's House", "Krabby Crustacio", 140)
		adjL.addEdge("Patrick's House", "Vehicle School", 40)
		adjL.addEdge("Planton's Restaurant", "Krabby Crustacio", 15)
		adjL.addEdge("Card Shop", "Planton's Restaurant", 20)
		adjL.addEdge("Vehicle School", "Card Shop", 50)
		adjL.addEdge("Scissors Shop", "Card Shop", 15)
		adjL.addEdge("Vehicle School","Scissors Shop", 30)
		adjL.addEdge("Scissors Shop", "Krabby Crustacio", 50)
		adjL.addEdge("Scissors Shop", "Planton's Restaurant", 40)
		adjL.addEdge("Massage Shop", "Scissors Shop", 35)
		adjL.addEdge("Massage Shop", "Krabby Crustacio", 45)
		adjL.addEdge("Massage Shop", "Planton's Restaurant", 30)
		adjL.addEdge("Massage Shop","Patrick's House",80)

Setup adjMatrixTest

Name	Class	Stage
Setup1	adjMatrixTest	adjM = new AdjMatrixGraph<String>(false, true, 4)

Name	Class	Stage

Setup2	adjMatrixTest	setup1()
		adjM.addVertex("Bob's House")
		adjM.addVertex("Patrick's House")
		adjM.addVertex("Scissors Shop")
		adjM.addVertex("Card Shop")

Name	Class	Stage

Setup3	adjMatrixTest	setup1()
		adjM.addVertex("Bob's House")
		adjM.addVertex("Patrick's House")
		adjM.addVertex("Scissors Shop")
		adjM.addVertex("Card Shop")
		adjM.addEdge("Bob's House", "Patrick's House",20)
		adjM.addEdge("Bob's House", "Squidward's House",35)
		adjM.addEdge("Bob's House", "Card Shop",130)
		adjM.addEdge("Bob's House", "Scissors Shop", 120)

Name	Class	Stage

Setup4	adjMatrixTest	adjL = new AdjListGraph<String>(false, true, 9)
		
		adjL.addVertex("Bob's House")
		adjL.addVertex("Squidward's House")
		adjL.addVertex("Patrick's House")
		adjL.addVertex("Planton's Restaurant")
		adjL.addVertex("Vehicle School")
		adjL.addVertex("Scissors Shop")
		adjL.addVertex("Massage Shop")
		adjL.addVertex("Card Shop")
		adjL.addVertex("Krabby Crustacio")
		adjL.addEdge("Bob's House", "Patrick's House",20)
		adjL.addEdge("Bob's House", "Card Shop",130)
		adjL.addEdge("Bob's House", "Scissors Shop", 120)
		adjL.addEdge("Bob's House", "Squidward's House", 35)
		adjL.addEdge("Squidward's House","Patrick's House",40)

		adjL.addEdge("Squidward's House", "Scissors Shop", 60)
		adjL.addEdge("Patrick's House", "Krabby Crustacio", 140)
		adjL.addEdge("Patrick's House", "Vehicle School", 40)
		adjL.addEdge("Planton's Restaurant", "Krabby Crustacio", 15)
		adjL.addEdge("Card Shop", "Planton's Restaurant", 20)
		adjL.addEdge("Vehicle School", "Card Shop", 50)
		adjL.addEdge("Scissors Shop", "Card Shop", 15)
		adjL.addEdge("Vehicle School","Scissors Shop", 30)
		adjL.addEdge("Scissors Shop", "Krabby Crustacio", 50)
		adjL.addEdge("Scissors Shop", "Planton's Restaurant", 40)
		adjL.addEdge("Massage Shop", "Scissors Shop", 35)
		adjL.addEdge("Massage Shop", "Krabby Crustacio", 45)
		adjL.addEdge("Massage Shop", "Planton's Restaurant", 30)
		adjL.addEdge("Massage Shop","Patrick's House",80)

Test UserManagment

Objt: verify that player has been created and added			
Class	Method	Stage	Result
UserManagment	addPlayer getRoot getRight getLeft	Setup	The player has been added and functionality of the getters

Objt:verify the correct positions of the tree			
Class	Method	Stage	Result
UserManagment	getNickname	Setup2	The name from the players in the correct position

Test AdjListGraph

<table>
<tr><td colspan="4">Objt: Verify the correct functionality of the add vertex and his index</td></tr>
<tr><td>Class</td><td>Method</td><td>Stage</td><td>Result</td></tr>
<tr><td>AdjListGraph</td><td>addVertex
getVertex
getValue
getIndex</td><td>Setup1</td><td>The vertices are created and added</td></tr>
</table>

<table>
<tr><td colspan="4">Objt:verify the correct functionality of the add vertex in another setup and his positions</td></tr>
<tr><td>Class</td><td>Method</td><td>Stage</td><td>Result</td></tr>
<tr><td>AdjListGraph</td><td>addVertex
removeVertex</td><td>Stup2</td><td>The vertices have been deleted and are in the correct positions</td></tr>
</table>

<table>
<tr><td colspan="4">Objt: verify the correct functionality of the dijkstra method</td></tr>
<tr><td>Class</td><td>Method</td><td>Stage</td><td>Result</td></tr>
<tr><td>AdjListGraph</td><td>dijkstra</td><td>Setup3</td><td>Return the correct value from each edge</td></tr>
</table>

<table>
<tr><td colspan="4">Objt: verify the correct functionality of the search adjVertex from vertex and remove edge and vertex</td></tr>
<tr><td>Class</td><td>Method</td><td>Stage</td><td>Result</td></tr>
<tr><td>AdjListGraph</td><td>searchAdjVertex
removeEdge
removeVertex</td><td>Setup2</td><td>Found the adjVertex from vertex, successfully deleted the vertex and edge</td></tr>
</table>

<table>
<tr><td colspan="4">Objt: verify the bfs method</td></tr>
<tr><td>Class</td><td>Method</td><td>Stage</td><td>Result</td></tr>
<tr><td>AdjListGraph</td><td>bfs</td><td>Setup3</td><td>Correct return of the bfs method</td></tr>
</table>

<table>
<tr><td colspan="4">Objt:verify the Kruskal method</td></tr>
<tr><td>Class</td><td>Method</td><td>Stage</td><td>Result</td></tr>
<tr><td>AdjListGraph</td><td>Kruskal</td><td>Setup3</td><td>The correct return of the Kruskal method</td></tr>
</table>

Objt: found the adjVertex from a vertex and compare the weight are correct			
Class	Method	Stage	Result
AdjListGraph	searchAdjVertex getAdjList getWeight	Setup3	The weight in each vertex are correct

Objt: verify the prim mehtod			
Class	Method	Stage	Result
AdjListGraph	Prim IsDirect isWeigthed	Setup3	The correct result from prim method

Objt: verify the Floyd warshall method			
Class	Method	Stage	Result
AdjListGraph	FloydWarshall	Setup	A different initial matrix before to run the method

Test AdjMatrixGraph

Objt: veriy the method addvertex and the vertices have been added			
Class	Method	Stage	Result
AdjMatrixGraph	addVertex getAdjMatrix getNumVertex	Setup1	The vertices have been added and his index are correct

Objt: verify the correct functionality of remove vertex and edge methods			
Class	Method	Stage	Result
AdjMatrixGraph	removeEdge removeVertex	Setup3	The edges and vertices have been removed

Objt: verify the Floyd warshall method			
Class	Method	Stage	Result
AdjMatrixGraph	floydWarshall	Setup3	The matrix are correcet

Objt: verify the kruskal mehtod			
Class	Method	Stage	Result
AdjMatrixGraph	kruskal	Setup3	The return of the method are correct

Objt: verify the Dijkstra method			
Class	Method	Stage	Result
AdjMatrixGraph	dijkstra	Setup4	The return are correct with the vertices

Objt: verify the bfs and dfs method			
Class	Method	Stage	Result
AdjMatrixGraph	Bfs dfs	Setup4	The return are correct with the vertices

Objt: verify the prim method			
Class	Method	Stage	Result
AdjMatrixGraph	prim	Setup4	The return are correct with the vertices

Objt: verify the searchInGraph method			
Class	Method	Stage	Result
AdjMatrixGraph	searchInGraph	Setup4	The vertex has been found