

SimpleEngine

Project Lucid

1.0.0.1

This class library was designed by Danny Huynh

To use this class library you must right click your Project Solution > Add Reference > Browse for this class library

Make sure to give credit to "Project Lucid" when you use this free class library

Table of Contents

MapDesign	
Tile class -----	2
Map class -----	3
MapSet class -----	4
FilteredMapSet class -----	5
 CharacterDesign	
Character class -----	6
CharacterSet class -----	7
FilteredCharacterSet class -----	8
 CardDesign	
Card class -----	9
Deck class -----	10
 StorageDesign	
Item class -----	11
Storage class -----	12
StorageList class -----	14

using SimpleEngine.MapDesign

Tile class <Tile class, cannot be inherited>

Property	Return Type	Description
Layer	int	Returns the Layer of Tile object
Access	bool	Returns the Access of Tile object
Data	Object	Returns the Data of Tile object

using SimpleEngine.MapDesign

Map class <Map class, cannot be inherited>

Constructor	Description
Map(int ID, String Group, int Xdim, int Ydim)	Creates a Map object with input parameter(ID, Group, Xdim, Ydim)

Method	Return Type	Description
Tile(int X, int Y)	Tile	Returns the Tile object at the specified X and Y coordinates in the Map object

Property	Return Type	Description
ID	int	Returns the ID of Map object
Group	String	Returns the Group of Map object
Xdim	int	Returns the Xdim of Map object
Ydim	int	Returns the Ydim of Map object

using SimpleEngine.MapDesign

MapSet class <MapSet class, cannot be inherited>

Constructor	Description
MapSet()	Creates a MapSet object that holds Map objects

Method	Return Type	Description
Add(Map Map)	void	Adds the Map object to the MapSet and will throw an exception if a Map object with the same ID already exist in the MapSet
Remove(int ID)	void	Removes the Map object with that ID from the MapSet and will throw an exception if no Map object in the MapSet with that ID
Clear()	void	Clears all Map objects from the MapSet
Filter(String Group)	FilteredMapSet	Returns a FilteredMapSet object of the MapSet filtered by Group
Map(int ID)	Map	Returns the Map object with that ID from the MapSet and will throw an exception if no Map object with that ID exist in the MapSet

Property	Return Type	Description
Size	int	Returns the size of the MapSet

using SimpleEngine.MapDesign

FilteredMapSet class <FilteredMapSet class, cannot be inherited>

Method	Return Type	Description
Map(int ID)	Map	Adds the Map object to the FilteredMapSet object

Property	Return Type	Description
Size	int	Returns the size of the FilteredMapSet

using SimpleEngine.CharacterDesign

Character class <Character class, cannot be inherited>

Constructor	Description
Character(int ID, String Group)	Creates a Character object with input parameter(ID, Group)
Character(int ID, String Group, Object Data)	Creates a Character object with input parameter(ID, Group, Data)

Property	Return Type	Description
ID	int	Returns the ID of Character object
Group	String	Returns the Group of Character object
Data	Object	Returns and sets the Data of Character object

using SimpleEngine.CharacterDesign

CharacterSet class <CharacterSet class, cannot be inherited>

Constructor	Description
CharacterSet()	Creates a CharacterSet object that contains Character objects

Method	Return Type	Description
Add(Character Character)	void	Adds a Character object to the CharacterSet and will throw an exception if a Character object with the same ID already exist in the CharacterSet
Remove(int ID)	void	Removes the Character object with that ID from the CharacterSet and will throw an exception if no Character with that ID exist in the CharacterSet
Clear()	void	Clears all Character objects from the CharacterSet
Filter(String Group)	CharacterMapSet	Returns a FilteredCharacterSet object of the CharacterSet filtered by Group
Character(int ID)	Character	Returns the Character object with that ID from the CharacterSet and will throw an exception if no Character object with that ID exist in the CharacterSet

Property	Return Type	Description
Size	int	Returns the size of the CharacterSet

using SimpleEngine.CharacterDesign

FilteredCharacterSet class <FilteredCharacterSet class, cannot be inherited>

Method	Return Type	Description
Character(int ID)	Character	Adds the Character object to the FilteredCharacterSet object

Property	Return Type	Description
Size	int	Returns the size of the FilteredCharacterSet

using SimpleEngine.CardDesign

Card class <Card class, cannot be inherited>

Constructor	Description
Card(int ID, String Group, Object Data)	Creates a Card object with input parameter(ID, Group, Data)

Property	Return Type	Description
ID	int	Returns the ID of Card object
Group	String	Returns the Group of Card object
Data	Object	Returns and sets the Data of Card object

using SimpleEngine.CardDesign

Deck class <Deck class, cannot be inherited>

Constructor	Description
Deck()	Creates a Deck object that holds Card objects

Method	Return Type	Description
Add(Card Card)	void	Adds the Card object to the top of the deck and will throw an exception if a Card object with the same ID already exist in the deck
Remove(int ID)	void	Removes the Card object with that ID from the deck and will throw an exception if no Card object in the deck with that ID
TryRemove(int ID)	bool	Removes the Card object with that ID from the deck and returns true if it is successful, otherwise this method will return false and do nothing
Clear()	void	Clears all Card objects from the deck
Draw()	Card	Returns and removes the top Card object from the deck and will throw an exception if there are no Card objects in the deck
Peak()	Card	Returns the top Card object from the deck and will throw an exception if there are no Card objects in the deck
Search(int ID)	Card	Returns the Card object with that ID from the deck and will throw an exception if there are no Card object in the deck with that ID
Reveal(int CardPosition)	Card	Returns the Card object by that CardPosition (starts with 1, the top) from the deck and will throw an exception if CardPosition is out of bounds
Shuffle()	void	Shuffles the deck

Property	Return Type	Description
Count	int	Returns current deck size

using SimpleEngine.StorageDesign

Item class <Item class, cannot be inherited>

Constructor	Description
Item(int ID, String Group, String Name, Object Data, int Quantity)	Creates an Item object with input parameter(ID, Group, Name, Data, Quantity)

Property	Return Type	Description
ID	int	Returns the ID of Item object
Group	String	Returns the Group of Item object
Name	String	Returns the Name of Item object
Data	Object	Returns and sets the Data of Item object
Quantity	int	Returns and sets the Quantity of Item object

using SimpleEngine.StorageDesign

Storage class <Storage class, cannot be inherited>

Constructor	Description
Storage(int Capacity)	Creates an static Storage object that holds Item objects and will throw an exception if Capacity is less than 1

Method	Return Type	Description
Add(Item Item, int Slot)	void	Adds the Item object to the specified slot (Starts from 1) and will throw an exception if that slot already contains an Item object or if Slot is less than 1 or greater than the capacity
Remove(int Slot)	void	Removes the Item object at the specified slot (starts from 1) and will throw an exception if Slot is less than 1, greater than the capacity, or if there is no Item object at that slot
Replace(Item Item, int Slot)	void	Adds the Item object to the specified slot (Starts from 1) and will throw an exception if Slot is less than 1 or greater than the capacity
Use(int Slot)	Item	Decreases Quantity of the Item object in the slot (Starts from 1) of Storage by 1 as well as returning 1 quantity of that Item object and will throw an exception if Slot is less than 1, greater than the capacity, or if the Item object in the slot of Storage is null
Use(int Slot, int Quantity)	Item	Decreases Quantity of the Item object in the slot (Starts from 1) of Storage by input Quantity as well as returning 1 quantity of that Item object and will throw an exception if Slot is less than 1, greater than the capacity, or if the Item object in the slot of Storage is null
Discard(int Slot)	void	Decreases Quantity of the Item object in the slot (Starts from 1) of Storage by 1 and will throw an exception if Slot is less than 1, greater than the capacity, or if the Item object in the slot of Storage is null
Discard(int Slot, int Quantity)	void	Decreases Quantity of the Item object in the slot (Starts from 1) of Storage by input Quantity and will throw an exception if Slot is less than 1, greater than the capacity, or if the Item object in the slot of Storage is null
Switch(int Slot1, int Slot2)	void	Switches the two Item objects between the slots (Starts from 1) and will throw an exception if either slots are less than 1 or greater than the capacity
Clear()	void	Clears all Item objects from Storage
Item(int Slot)	Item	Returns the Item object's ID at the slot (Starts from 1) of Storage and will throw an exception if Slot is less

		than 1, greater than capacity, or if the Slot of Storage is null (Surround this with a Try-Catch Statement to avoid exception when trying to access null Slot of Storage)
--	--	---

Property	Return Type	Description
Capacity	<code>int</code>	Returns the Capacity of Storage

using SimpleEngine.StorageDesign

StorageList class <StorageList class, cannot be inherited>

Constructor	Description
StorageList()	Creates a dynamic StorageList object that holds Item objects

Method	Return Type	Description
Add(Item Item)	void	Adds the Item object to StorageList
Remove(int Index)	void	Removes the Item object at the specified Index (Starts at 1) and will throw an exception if Index is less than 1 or greater than the list's size
Replace(Item Item, int Index)	void	Adds the Item object to the specified Index (Starts from 1) and will throw an exception if Index is less than 1 or greater than the size
Use(int Index)	Item	Decreases Quantity of the Item object in the index (Starts from 1) of StorageList by 1 as well as returning 1 quantity of that Item object and will throw an exception if Index is less than 1, greater than the size, or if the Item object in the index of StorageList is null
Use(int Index, int Quantity)	Item	Decreases Quantity of the Item object in the index (Starts from 1) of StorageList by input Quantity as well as returning 1 quantity of that Item object and will throw an exception if Index is less than 1, greater than the size, or if the Item object in the index of StorageList is null
Discard(int Index)	void	Decreases Quantity of the Item object in the index (Starts from 1) of StorageList by 1 and will throw an exception if Index is less than 1, greater than the size, or if the Item object in the index of StorageList is null
Discard(int Index, int Quantity)	void	Decreases Quantity of the Item object in the index (Starts from 1) of StorageList by input Quantity and will throw an exception if Index is less than 1, greater than the size, or if the Item object in the index of StorageList is null
Switch(int Index1, int Index2)	void	Switches the two Item objects between the 2 index (Starts from 1) and will throw an exception if either index are less than 1 or greater than the size
Clear()	void	Clears all Item objects from StorageList
Item(int Index)	Item	Returns the Item object at the index (Starts from 1) of StorageList and will throw an exception if Index is less than 1 or greater than size

Property	Return Type	Description
Size	int	Returns the Size of StorageList