Avem clasa Deserialize. In momentul rularii, se apeleaza functia Start() generata de Unity. In functia Start() avem o functie Getlevel().

Functia Getlevel():

XmlDocument xmlDoc = new XmlDocument();

- xmlDoc este un nou document xml.

xmlDoc.Load("path\\format_date.xml");

- pe acesta il luam dintr-o locatie data printr-un path pus ca argument al functiei ".Load".

XmlNodeList floorlist = xmlDoc.GetElementsByTagName("floor");

- Ne folosim de functiile deja existente asociate documentelor Xml.

- Prin ".GetElementsByTagName" anuntam ca urmeaza sa cautam in nodul floor, subnodurile prin floorlist.

```
foreach (XmlNode floorinfo2 in floorlist)
    {...
```

- prin acest mod parcurgem fiecare nod individual

- analog pentru fiecare nod parinte

```
if (roomstuff.Name == "name")
{
        ...
}
```

- Avand ca fii nodurile roomstuff, parcurgem subnodurile prin ".Name"

```
if (roomstuff.Name == "type")
{
	switch (roomstuff.Attributes["name"].Value)
	{
		case "wall":

			XmlNodeList roomdimensions = roomstuff.ChildNodes;
			foreach (XmlNode dimension in roomdimensions)
			{
				if (dimension.Name == "x1")
				{
					string x1_str = dimension.InnerText;
					x1 = float.Parse(x1_str);
				}
				...
```

- In momentul in care ajungem la nodul "type", trebuie sa aflam ce atribut i-am pus. Asta aflam prin ".Attributes["name"].Value", deoarece am notat cu "name" locul unde

stocam atributul.

- Prin "switch" si "case" respectiv, verificam peste ce atribut dam. De exemplu cand ajungem la un "<type name="wall">" executam ce e dupa case-ul respectiv.

- Pentru ca sunt stocate ca si stringuri informatiile dintre tag-uri, le stocam intr-un string cu ".InnerText" iar apoi il transformam in float prin ".Parse".