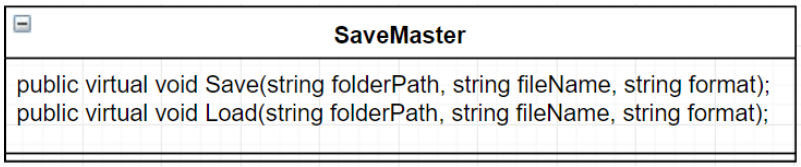
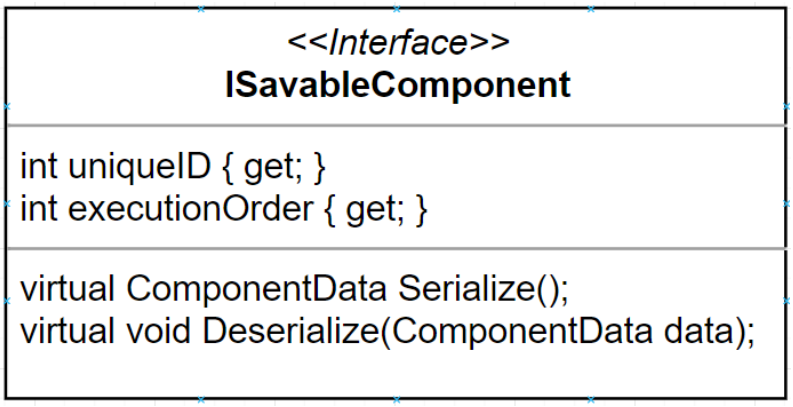
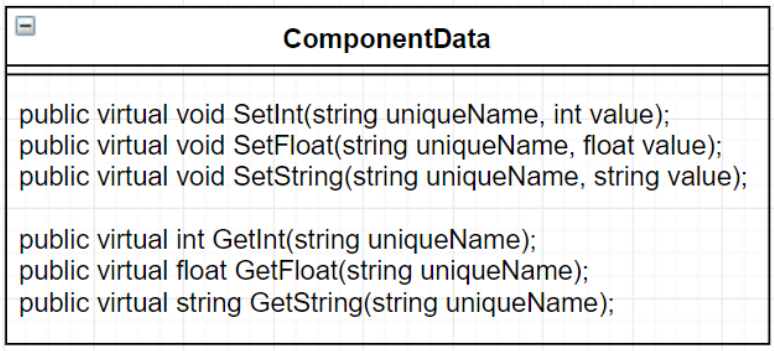
To see a detailed tutorial, follow the link : https://youtu.be/H7WuyoXlNZE

The system consists of three entities : **SaveMaster, ISavableComponent, ComponentData**



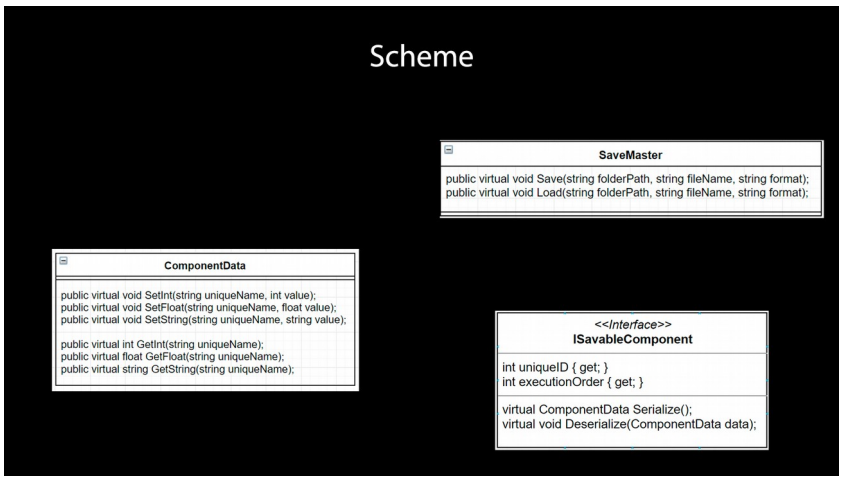


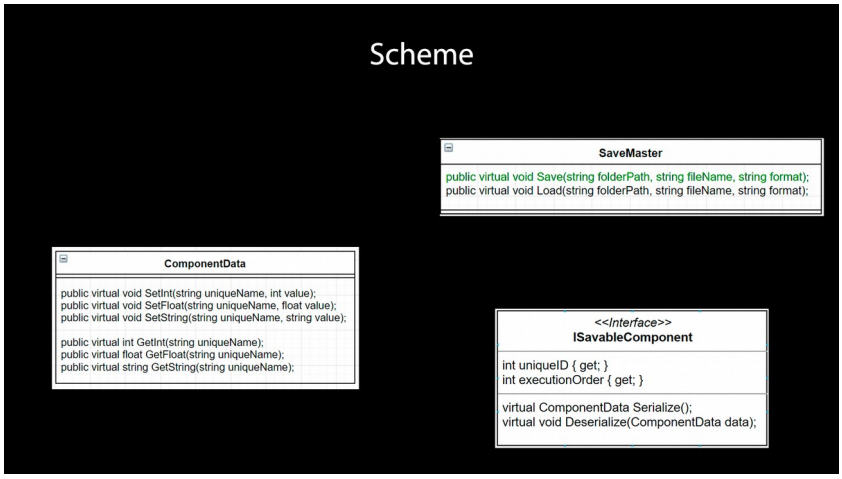


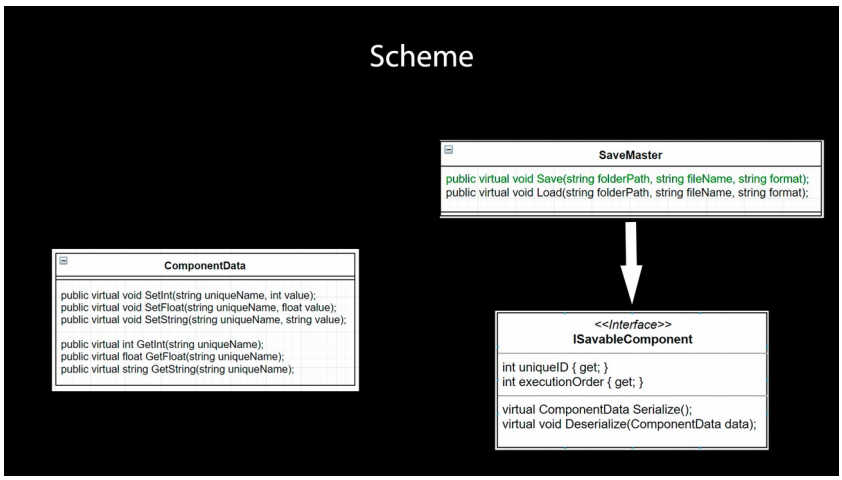
The idea is pretty simple.

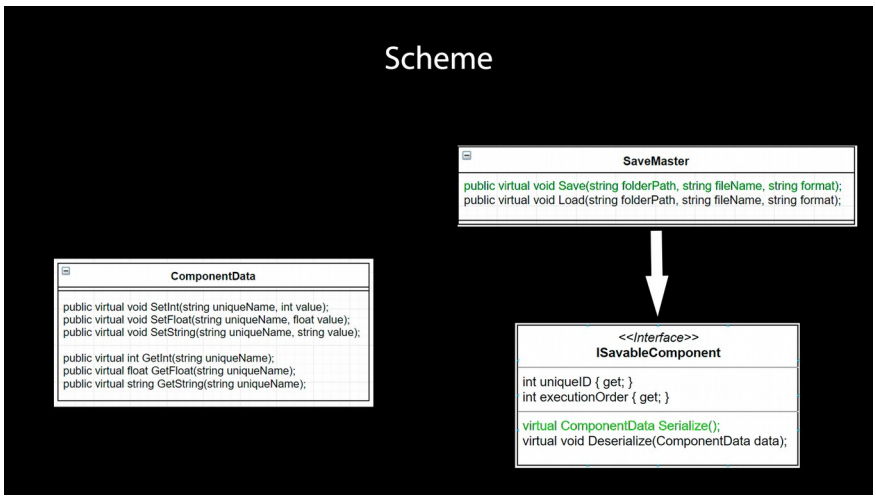
When You call «Save» method on SaveMaster. SaveMaster find all components that implements ISavableComponent interface and call «Serialize» method. This method returns «ComponentData» then SaveMaster store «ComponentData» to disk.

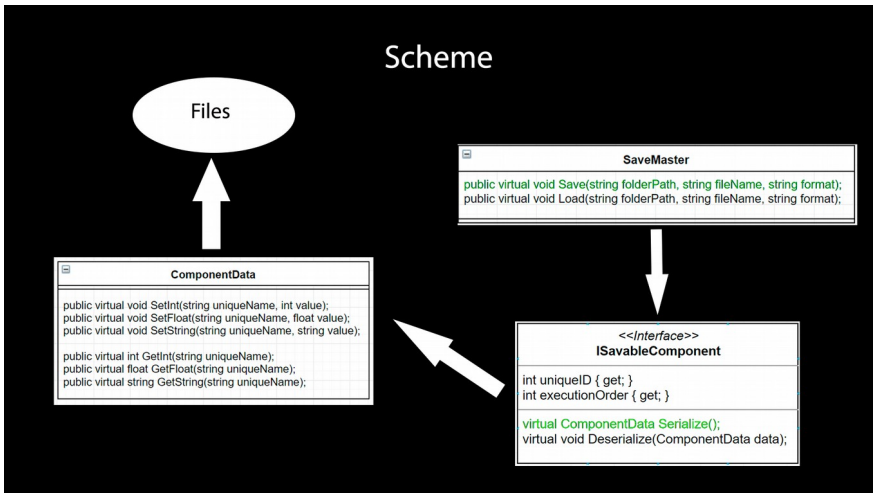
When You call «Load» method on SaveMaster occurs reverse process.

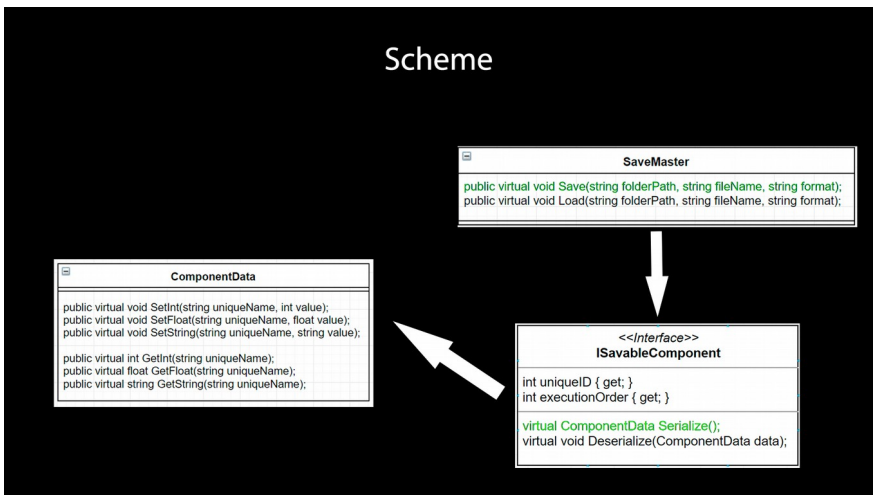

Scheme
SaveMaster
public virtual void Save(string folderPath, string fileName, string format);
public virtual void Load(string folderPath, string fileName, string format);
ComponentData
public virtual void SetInt(string uniqueName, int value);
public virtual void SetFloat(string uniqueName, float value);
public virtual void SetString(string uniqueName, string value);
public virtual int GetInt(string uniqueName);
public virtual float GetFloat(string uniqueName);
public virtual string GetString(string uniqueName);
<<Interface>>
ISavableComponent
int uniqueID { get; }
int executionOrder { get; }
virtual ComponentData Serialize();
virtual void Deserialize(ComponentData data);

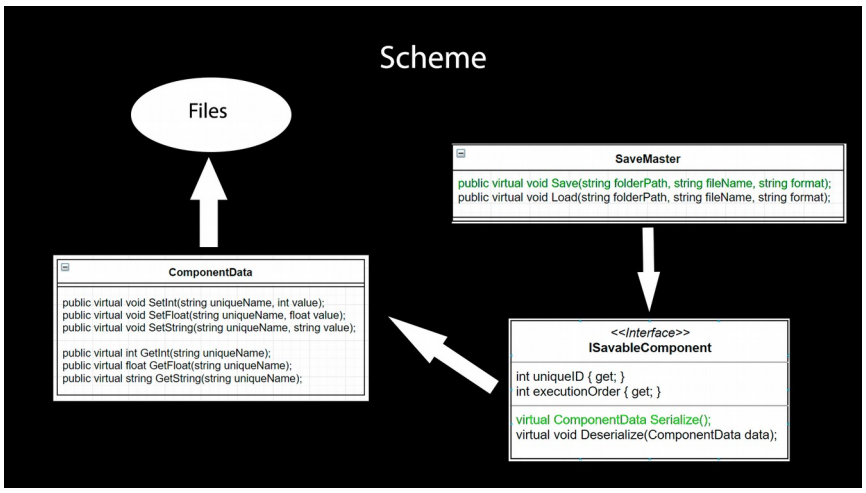

Scheme
SaveMaster
public virtual void Save(string folderPath, string fileName, string format);
public virtual void Load(string folderPath, string fileName, string format);
ComponentData
public virtual void SetInt(string uniqueName, int value);
public virtual void SetFloat(string uniqueName, float value);
public virtual void SetString(string uniqueName, string value);
public virtual int GetInt(string uniqueName);
public virtual float GetFloat(string uniqueName);
public virtual string GetString(string uniqueName);
<<Interface>>
ISavableComponent
int uniqueID { get; }
int executionOrder { get; }
virtual ComponentData Serialize();
virtual void Deserialize(ComponentData data);


Scheme
SaveMaster
public virtual void Save(string folderPath, string fileName, string format);
public virtual void Load(string folderPath, string fileName, string format);
ComponentData
public virtual void SetInt(string uniqueName, int value);
public virtual void SetFloat(string uniqueName, float value);
public virtual void SetString(string uniqueName, string value);
public virtual int GetInt(string uniqueName);
public virtual float GetFloat(string uniqueName);
public virtual string GetString(string uniqueName);
<<Interface>>
ISavableComponent
int uniqueID { get; }
int executionOrder { get; }
virtual ComponentData Serialize();
virtual void Deserialize(ComponentData data);


Scheme
SaveMaster
public virtual void Save(string folderPath, string fileName, string format);
public virtual void Load(string folderPath, string fileName, string format);
ComponentData
public virtual void SetInt(string uniqueName, int value);
public virtual void SetFloat(string uniqueName, float value);
public virtual void SetString(string uniqueName, string value);
public virtual int GetInt(string uniqueName);
public virtual float GetFloat(string uniqueName);
public virtual string GetString(string uniqueName);
<<Interface>>
ISavableComponent
int uniqueID { get; }
int executionOrder { get; }
virtual ComponentData Serialize();
virtual void Deserialize(ComponentData data);


Scheme
Files
SaveMaster
public virtual void Save(string folderPath, string fileName, string format);
public virtual void Load(string folderPath, string fileName, string format);
ComponentData
public virtual void SetInt(string uniqueName, int value);
public virtual void SetFloat(string uniqueName, float value);
public virtual void SetString(string uniqueName, string value);
public virtual int GetInt(string uniqueName);
public virtual float GetFloat(string uniqueName);
public virtual string GetString(string uniqueName);
<<Interface>>
ISavableComponent
int uniqueID { get; }
int executionOrder { get; }
virtual ComponentData Serialize();
virtual void Deserialize(ComponentData data);


Scheme
SaveMaster
public virtual void Save(string folderPath, string fileName, string format);
public virtual void Load(string folderPath, string fileName, string format);
ComponentData
public virtual void SetInt(string uniqueName, int value);
public virtual void SetFloat(string uniqueName, float value);
public virtual void SetString(string uniqueName, string value);
public virtual int GetInt(string uniqueName);
public virtual float GetFloat(string uniqueName);
public virtual string GetString(string uniqueName);
<<Interface>>
ISavableComponent
int uniqueID { get; }
int executionOrder { get; }
virtual ComponentData Serialize();
virtual void Deserialize(ComponentData data);



All You need to make component «savable» it's just implement «ISavableComponent» interface on this component.

Here is example how to make Transform data savable.

Extended component data class :

```
namespace NGS.ExtendableSaveSystem
{
    [Serializable]
    public class ExtendedComponentData : ComponentData
    {
        public virtual void SetVector3(string uniqueName, Vector3 value)
        {
            SetFloat(uniqueName + ".x", value.x);
            SetFloat(uniqueName + ".y", value.y);
            SetFloat(uniqueName + ".z", value.z);
        }

        public virtual void SetTransform(string uniqueName, Transform transform)
        {
            SetVector3(uniqueName + ".position", transform.position);
            SetVector3(uniqueName + ".rotation", transform.eulerAngles);
            SetVector3(uniqueName + ".scale", transform.localScale);
        }


        public Vector3 GetVector3(string uniqueName)
        {
            return new Vector3(
                GetFloat(uniqueName + ".x"),
                GetFloat(uniqueName + ".y"),
                GetFloat(uniqueName + ".z"));
        }

        public void GetTransform(string uniqueName, Transform transform)
        {
            transform.position = GetVector3(uniqueName + ".position");
            transform.eulerAngles = GetVector3(uniqueName + ".rotation");
            transform.localScale = GetVector3(uniqueName + ".scale");
        }
    }
}
```

TransformSaver class :

```
namespace NGS.ExtendableSaveSystem
{
    public class TransformSaver : MonoBehaviour, ISavableComponent
    {
        [SerializeField] private int _uniqueID;
        [SerializeField] private int _executionOrder;

        public int uniqueID...
        public int executionOrder...


        private void Reset()
        {
            _uniqueID = GetHashCode();
        }

        public ComponentData Serialize()
        {
            ExtendedComponentData data = new ExtendedComponentData();

            data.SetTransform("transform", transform);

            return data;
        }

        public void Deserialize(ComponentData data)
        {
            ExtendedComponentData unpacked = (ExtendedComponentData)data;

            unpacked.GetTransform("transform", transform);
        }
    }
}
```

To see a detailed tutorial, follow the link : https://youtu.be/H7WuyoXlNZE

If You have any questions You can freely contact me via e-mail : andre-orsk@yandex.ru