# Save Manager

## Documentation

Created by:
Saulius

# Table of content

# Save Manager

A multi-platform (Mac/Windows/Web Player/Android/iOS) save system solution for Unity. It allows Unity developers to implement save system to their applications (games, simulations, etc.). System is suitable for Client and Server.

## Requirements

Unity3D application.

## Features

- The Save Manager is fully integrated in the Unity editor (Mac and Win).
- Supports any Mac and Windows standalone applications. Plus Web Player, Android, iOS.
- All code is commented as much as possible, to be easy understandable for user.
- Very easy to use.
- Demonstration scenes included.

DISCLAIMER: Use this package at your own risk. We have tested the package sufficiently in our development environments but we cannot be held responsible for your usage of this package.

## Supported types

| Primitive Types | Unity Types | Collections |
|---|---|---|
| •string<br>•float<br>•double<br>•int<br>•bool<br>•byte<br>•char<br>•long<br>•short<br>•uint<br>•ulong<br>•ushort | •Vector2<br>•Vector3<br>•Vector4<br>•Quaternion<br>•Color<br>•Color32<br>•Texture2D | •List<> |

NOTICE: 1. Then saving class you can only user primitives and unity types. Saving List<> is not supported instead save List itself.
2. You can only save collections containing primitive types.

# Save Manager Setup

1. Attach "App" script to any game object from SaveManager folder and that’s it.

# Scripting

## AddField

AddField ( field : string, value : type ) : void

**Description**

Adds field to update list . Field is name of the variable you’re saving (which will be used to retrieve variable later) and second is the variable it self ( you can use any supported type ).

To access write: App.Save. AddField (youVariableName, yourVariable);

Example:

```
using UnityEngine;

using System.Collections;

public class Demo : MonoBehaviour {

        private string name = "";

                // Update is called once per frame

        void OnGUI () {

                name = GUI.TextArea(new Rect(200,200,150,30),name,25);

                        if(GUI.Button(new Rect(200,280,150,30),"Save")){

                        App.Save.AddField(“name”, name);

                }

        }

}
```

## AddClass

AddClass<Type> ( field : string, value : type ) : void

**Description**

Adds whole class to update list . Field is name of the class you’re saving (which will be used to retrieve class later) and second is the variable it self ( your class ).

Only these variables which have setters and getters will be saved. And class can only contain primitives and unity types!

To access write: App.Save. AddClass<YourClass> (youVariableName, yourVariable);

Example:

```
using UnityEngine;

using System.Collections;

public class Demo : MonoBehaviour {


                // Update is called once per frame

        void OnGUI () {

                        User user = new User(“name”,18,true);

                        if(GUI.Button(new Rect(200,280,150,30),"Save")){

                        App.Save.AddClass<User>(user.Name, user);

                }

        }

}
```

## GetVariable

Get**[supportedType]** (string fieldName)

**Description**

Returns your saved variable.

To access write: App.Save. Get**[Float,Int,String,Vector3,Texture2D, etc.]** (youVariableName);

Example:

```
using System.Collections;

public class Demo : MonoBehaviour {

        float myVariable = 0;

                // Update is called once per frame

        void OnGUI () {

                        if(GUI.Button(new Rect(200,280,150,30),"Load")){

                        myVariable = App.Save.GetFloat (MyFloat");

                        Debug.Log(myVariable);

                }
        }
}
```

## GetClass

GetClass<yourClass> (string fieldName)

**Description**

Returns your saved class.

To access write: App.Save. GetClass<yourClass> (youVariableName);

Example:

```
using System.Collections;
public class Demo : MonoBehaviour {
        User user = new User("user0",18, false);
                // Update is called once per frame
        void OnGUI () {
                        User user = new User(“name”,18,true);
                        if(GUI.Button(new Rect(200,280,150,30),"Save")){
                        users = App.Save.GetClass<User>("user0");
                        Debug.Log(user.name);
                }
        }
}
```

# Contact info

Email : thegame1617@yahoo.com or support@survivetilltomorrow.com

Website : http://assets.survivetilltomorrow.com

Skype : sauliusxxl1