**WordGame 游戏保存样本**

此样本需要 2016 年 3 月或之后发布的 Xbox One XDK，

或 2018 年 4 月发布的 Windows 10 Update SDK (17134)

# 描述

此样本演示了使用“连接存储 API”构建完整的端到端游戏体验的 Xbox One **XDK** *和* **UWP** 的游戏保存功能。

此示例为以下游戏保存方案提供选项。

* **使用“完全同步”或“按需同步”模式**

出于演示目的，在示例启动时，可以选择使用完全同步 API（在控制台和标题存储服务之间同步所有游戏保存数据）或按需同步 API（仅按需同步游戏保存数据）。  
注意：如果您已经执行完全同步且想要尝试按需同步，则应该使用其他用户登录，或清除游戏保存数据的本地缓存。

* + 要清除 Xbox One 的本地缓存，可以从 XDK 命令提示符运行“xbstorage.exe reset / force”命令。
  + 要清除 Windows 10 电脑的本地缓存，可以从管理员命令提示符运行“gamesaveutil.exe reset”命令。注意：安装 Windows 10 Fall Creators Update SDK 后，可以在以下目录中找到**gamesaveutil.exe**：  
    %ProgramFiles(x86)%\Windows Kits\10\Extension SDKs\XboxLive\1.0\Bin\x64
* **加载、保存和删除游戏保存数据**

使用菜单选项加载游戏板（使用“获取”或“读取”API），予以保存并删除。最多可以保存 9 个不同的游戏板。

* **列出容器和 blob**

使用菜单选项，以枚举容器和 blob。输出显示于游戏屏幕的可滚动调试输出区域。

* **查看上次修改日期和剩余配额**

此信息显示于游戏屏幕标题的正下方。

* **用户退出时自动保存**

如果当前游戏板尚未保存，则在用户退出时将自动保存。

* **暂停时自动保存**对于暂停事件，如果当前游戏板尚未保存，则将自动保存。

# 构建示例

**XDK 构建要求**

* Visual Studio 2015
* Xbox One XDK

**UWP构建要求**

* Visual Studio 2017（15.7 更新）或更高版本
* 2018 年 4 月发布的 Windows 10 Update SDK (17134)
* 2017 年 11 月发布的 Xbox Live SDK 版
  + 注意：从 Visual Studio 构建此样本时，将从 nuget.org 自动还原 Xbox Live SDK NuGet 数据包

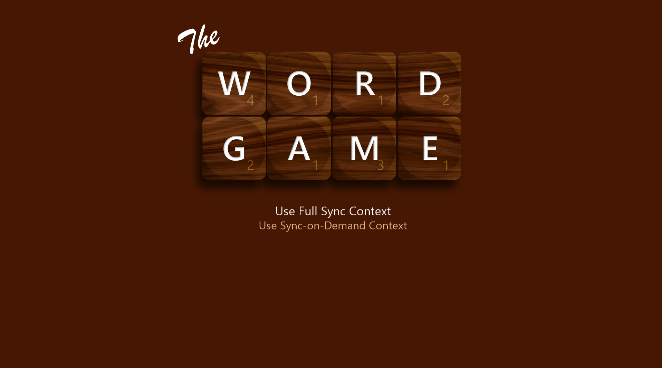
# 使用样本

**Xbox Live 沙盒要求**

* Xbox One devkit：将主机沙盒设为 XDKS.1
* Windows 10：您可以使用默认零售沙盒或将其设置为 XDKS.1，以试验游戏是否在电脑和主机之间保存漫游（有关如何在 Windows 中更改 Xbox Live 沙盒的说明，请参阅 Xbox Live SDK 附带文档）

**启动菜单**

|  |  |  |
| --- | --- | --- |
| 操作 | 游戏手柄 | 键盘/鼠标 |
| 在“完全同步”  模式和  “按需同步”  模式之间选择 | LS 或 D-Pad | 方向键 |
| 选择菜单项 | A 键 | 输入或  鼠标左键单击 |
| 切换全屏 |  | Alt+Enter |



**游戏板**

|  |  |  |
| --- | --- | --- |
| 操作 | 游戏手柄 | 键盘/鼠标 |
| 移动光标 | LS 或 D-Pad | 方向键 |
| 选择菜单项 | A 键 | 输入或  鼠标左键单击 |
| 选择游戏保存槽 | LB/RB 按钮 | 1 - 9 键或  鼠标 |
| 更改光标下的字母图块 | RS 左/右 | A - Z 键 |
| 清除光标下的字母 | X 键 | 删除或空格 |
| 滚动调试输出 | RS 上/下 | 上一页/下一页  和  Home / End 键 |
| 切换全屏 |  | Alt+Enter |



# 游戏菜单备注

* **获取板**使用 **GetAsync** API，为当前游戏保存槽加载游戏板。
* **读取板**  
  使用 **ReadAsync** API（GetAsync 的替代方法），为当前游戏保存槽加载游戏板。
* **保存板**  
  使用 **SubmitUpdatesAsync** API，为当前游戏保存槽保存游戏板。
* **重置板**  
  清除所有字母的板。如果板先前已保存，则会将板标记为“已更新”。如果板尚未保存，则板将不会被标记为“已更新”。
* **删除板**使用 **DeleteContainerAsync** API ，以删除当前游戏保存槽的游戏板。
* **删除板 Blob**  
  使用 **SubmitUpdatesAsync** API，以仅删除当前游戏板的 blob，留下一个空容器。
* **列出容器**  
  使用 **GetContainerInfo2Async** API，以枚举所有容器，并将它们列于游戏屏幕的调试输出区域。
* **列出容器和 Blob**使用 **GetContainerInfo2Async** 和 **GetBlobInfoAsync** API，以枚举所有容器和 blob，并将它们列于游戏屏幕的调试输出区域。

游戏玩法笔记

**游戏玩法**

游戏以 5 x 5 网格进行。可以在网格上的任何位置放置字母。形成公认英语单词的连续字母，无论是沿板横向还是纵向，都将根据单词中每个字母的总分值得分。目标是将分数最大化。每个板上放置的每个字母都有数量限制。剩下的计数可在游戏板正上方进行追踪。

**游戏板加载**

出于演示目的，当游戏板屏幕出现时，或当切换到新游戏保存槽时，游戏板**不要自动加载**。这使开发人员有机会选择加载方法（“获取”或“读取”），并可在特定游戏保存槽完全控制加载和保存操作。

**改变游戏板和自动保存**

当游戏板上的字母有更改时，或者使用“重置”菜单命令时，游戏板将被标记为“已更新”（在屏幕顶部的板名称后面用星号表示）。在以下条件下，将自动保存已更新的游戏板：

* 切换到不同的游戏板（游戏手柄 LB/RB）
* 用户退出
* 游戏暂停

# 实施说明

**GameSaveManager** 类将管理游戏的游戏保存操作。**InitializeForUser()** 方法为玩家设置连接的存储保存上下文。还有加载、保存、枚举和删除保存数据等方法。有关类中每个方法的使用说明，请参阅标头文件中的注释。

游戏使用了两类游戏数据结构：索引和游戏板。模板化 **GameSave** 类提供了 GameSaveManager 用于为任何类型的游戏数据一般加载和保存数据的方法。由 GameSaveManager.h 中 **GameBoardIndex** 结构定义的索引主要用于跟踪玩家上次所用的保存槽（“活动板”）。游戏板数据由 GameBoard.h 中的 **GameBoard** 结构表示。

# 已知的问题

没有。

# 更新历史记录

**初始发行：*2016 年 2 月***

与 2016 年 2 月的 XDK 一起发布，仅适用于Xbox One XDK

**2016 年 3 月更新：**

新功能：

* 在游戏中添加 UWP 版本（单个 Visual Studio 项目现在可用于 XDK 和 UWP），支持游戏手柄、键盘和鼠标输入
* 从 UWP 版本的挂起中恢复后，可显示自上次同步后经过更改的任何容器（如果容器是在挂起时被加载到内存中，则应在保存任何进一步更改之前重新加载）
* UWP 版本使用 1602 版 Xbox Live SDK 和 UWP 的 Xbox Live Platform Extensions

与游戏保存过程相关的更改：

* 重构 GameSaveManager Suspend() 方法并添加 OnSignOut() 方法
* 除了保存脏的活动游戏板之外，如果游戏板是脏的，用户注销现在将保存游戏板索引

其他变化：

* 单词列表文件相当小，并且在单词列表继续在后台加载时将显示游戏板

Bug 修复：

* 修复了恢复游戏时出现的崩溃 bug，用户在挂起时已退出(null deref)

**2016 年 4 月更新：**

更改：

* 将 GameSaveManager 实现分离为 UWP 和 XDK 源文件，以便于进行并排比较
* 将 DeviceResources（UWP 和 XDK 版）更新到最新版本
* 改进 PLM 暂停处理：
  + 如果在首次执行初始化和同步时发生挂起事件，游戏既不会尝试加载（并可能会重置）游戏保存索引，也不会尝试加载游戏保存元数据
  + 如果在任务执行期间发生挂起事件，游戏将不会尝试在异步游戏保存初始化任务结束时将游戏状态切换到 InGame

Bug 修复：

* 修正了在游戏保存初始化过程中出现错误时游戏可能陷入“加载”菜单状态的 bug

**2016 年 5 月更新：**

更改：

* UWP 版本已更新为使用 1604 版 Xbox Live SDK（但是，仍与 1602 版和 1603 版代码兼容）
* 对于 XDK 版本，调试日志文件现在保存到 D: 而不是 T: 中，因此，在标题未运行时，仍可用
* 如果传入的 SCID 为空，则在 UWP GameSaveManager’s InitializeForUser() 中添加警告（游戏保存上下文仅为脱机）
* 代码质量改进

**2017 年 3 月更新：**

更改：

* UWP 版本现在使用 C XSAPI NuGet 二进制文件，而不是 WinRT (C++/CX) XSAPI NuGet 二进制文件
* UWP 版本默认使用 1703 版 Xbox Live SDK，但与大多数早期版本兼容

**注意**：现在可以从 nuget.org 下载用于 Xbox Live SDK 的 NuGet 二进制文件。从 Visual Studio 构建此样本时，应自动为您下载/恢复。

* 切换到使用 ATG 标准“Xbox Live Tool Kit”，以进行用户管理
* 已添加对切换到 UWP 样本的多用户(MUA)版本的支持（一旦 C++ XSAPI 完全支持此功能，可通过取消 pch.h 和 appx 清单中的更改注释来激活）
* 已删除单用户 UWP 版本中的“切换帐户”功能，以便与 1703 版 Xbox的Xbox Live SDK 中删除的switch\_account() API 兼容
* 输入控制得到改善：游戏光标和菜单可通过“左摇杆”、D-Pad、或键盘光标箭头的任意组合进行控制
* 包括对用户管理、UI 流程、以及从挂起行为恢复等方面的改进

**2017 年 4 月更新：**

更改：

* 在 Xbox 上运行时，UWP 版本现在默认为多用户应用程序(MUA)
* UWP 版本现在可以在电脑上默认(RETAIL)沙盒中运行

**2017 年 9 月更新：**

更改：

* 已更新 Visual Studio 2017（15.1 更新）和 Windows 10 Creators SDK (15063)的 UWP 项目
* UWP 构建需要新的 Xbox Live Extensions SDK 15063

**2017 年 10 月更新：**

更改：

* 对错误处理进行改进，调试输出已增强

**2017 年 12 月更新：**

更改：

* 已更新 Visual Studio 2017（15.4 更新）和 Windows 10 Fall Creators Update SDK (16299) 的 UWP 项目
* 因为 Windows 10 Fall Creators Update SDK 已包含“连接存储 API”，因此，UWP 构建不再需要 Xbox Live Extensions SDK
* UWP 版本已更新，以引用 2017 年 11 月发布的 Xbox Live SDK 的 NuGet 数据包
* 添加同步/加载/保存异步调用持续时间的日志记录

Bug 修复：

* 已修复 UWP 版本中的崩溃，前提是 GetRemainingBytesInQuotaAsync() 在内部抛出
* 已修复首次加载时屏幕顶部可能显示错误存档槽的问题

**2018 年 5 月更新：**

更改：

* 已更新 Visual Studio 2017（15.7 更新）和 2018 年 4 月发布的 Windows 10 SDK (17134) 的 UWP 项目

# 隐私声明

编译和运行示例时，示例可执行文件的文件名将发送给Microsoft以帮助跟踪示例使用情况。要选择退出此数据收集，您可以删除Main.cpp中标记为“Sample Usage Telemetry”的代码块。

有关 Microsoft 隐私政策的更多信息，请参阅 [Microsoft 隐私声明](https://privacy.microsoft.com/zh-cn/privacystatement/)。