**WordGame 게임 저장 샘플**

이 샘플을 사용하려면 Xbox One XDK 2016년 3월 이상

또는 Windows 10 2018년 4월 업데이트 SDK(17134)가 필요합니다.

# 설명

이 샘플은 Xbox One **XDK** *및* **UWP** 에 대한 게임 저장 기능을 보여주며 연결된 스토리지 API를 사용하여 완전한 엔드 투 엔드 게임 경험을 구축합니다.

이 샘플은 다음 게임 저장 시나리오에 대한 옵션을 제공합니다.

* **"전체 동기화"또는 "주문형 동기화"모드 사용**

데모 목적으로 샘플 실행시 (콘솔과 제목 저장 서비스간에 모든 게임 저장 데이터를 동기화하는) 전체 동기화 API 또는 (필요에 따라 게임 저장 데이터 만 동기화하는) 동기화 요청 API 중 하나를 사용하도록 선택할 수 있습니다.  
알림: 이미 전체 동기화를 수행했으며 요청시 동기화를 시도하려면 다른 사용자로 로그인하거나 게임 저장 데이터의 로컬 캐시를 지워야 합니다.

* + Xbox One의 로컬 캐시를 지우려면 XDK 명령 프롬프트에서 "xbstorage.exe reset/force” 명령을 실행하십시오.
  + Windows 10 PC의 로컬 캐시를 지우려면 관리자 명령 프롬프트에서 "gamesaveutil.exe reset"명령을 실행하세요. 알림: **gamesaveutil.exe**는 Windows 10 Fall Creators Update SDK를 설치하면 다음 디렉토리에서 찾을 수 있습니다:  
    %ProgramFiles(x86)%\Windows Kits\10\Extension SDKs\XboxLive\1.0\Bin\x64
* **게임 저장 데이터 로드, 저장 및 삭제**

메뉴 옵션을 사용하여 (Get 또는 Read API를 사용하여) 게임 보드를 로드하고 저장하고 삭제하십시오. 최대 9개의 다른 보드를 저장할 수 있습니다.

* **목록 컨테이너 및 블로브**

메뉴 옵션을 사용하여 컨테이너와 블로브를 열거하십시오. 출력은 게임 화면의 스크롤 가능한 디버그 출력 영역에 표시됩니다.

* **최종 수정 날짜 및 남은 저장 용량 보기**

이 정보는 게임 화면의 제목 바로 아래에 표시됩니다.

* **사용자 로그 아웃시 자동 저장**

현재 게임 보드가 아직 저장되지 않은 경우 사용자가 로그 아웃하면 자동으로 저장됩니다.

* **일시 중지시 자동 저장**일시 중지 이벤트에 대한 응답으로 현재 게임 보드가 아직 저장되지 않은 경우 자동으로 저장됩니다.

# 샘플 빌드

**XDK 빌드 요구 사항**

* Visual Studio 2015
* Xbox One XDK

**UWP 빌드 요구 사항**

* Visual Studio 2017 (15.7 업데이트) 이상
* Windows 10 2018년 4월 SDK 업데이트 (17134)
* Xbox Live SDK 버전 2017.11
  + 알림: Visual Studio에서 이 샘플을 빌드할 때 Xbox Live SDK NuGet 패키지가 nuget.org에서 자동으로 복원됩니다.

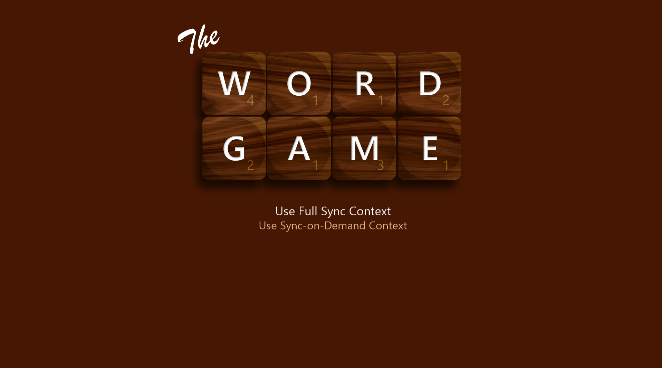
# 샘플 사용하기

**Xbox Live Sandbox 요구 사항**

* Xbox One devkit : 콘솔의 샌드 박스를 XDKS.1로 설정하십시오.
* 윈도우 10: 기본 소매 샌드박스를 사용하거나 XDKS.1로 설정하여 PC와 콘솔 간의 게임 저장 로밍을 테스트할 수 있습니다 (Windows에서 Xbox Live 샌드박스를 변경하는 방법에 대한 지침은 Xbox Live SDK에 포함된 문서를 참조하십시오)

**시작 메뉴**

|  |  |  |
| --- | --- | --- |
| 액션 | 게임 패드 | 키보드 / 마우스 |
| '전체 동기화'모드와 '주문형 동기화'모드를 선택하세요 | LS 또는 D 패드 | 방향 키 |
| 메뉴 항목 선택 | A 버튼 | Enter 또는  마우스 좌클릭 |
| 전체 화면 전환 |  | Alt+Enter |



**게임 보드**

|  |  |  |
| --- | --- | --- |
| 액션 | 게임 패드 | 키보드 / 마우스 |
| 커서 이동 | LS 또는 D 패드 | 방향 키 |
| 메뉴 항목 선택 | A 버튼 | Enter 또는  마우스 좌클릭 |
| 게임 저장 슬롯 선택 | LB/RB 버튼 | 1 - 9 까지의 키 또는  마우스 |
| 커서 아래 문자 타일 변경 | RS 좌/우 | A - Z 키 |
| 커서 아래의 문자 지우기 | X 버튼 | 삭제 또는 스페이스 |
| 스크롤 디버그 출력 | RS 위/아래 | Page Up/Dn  그리고  Home/End |
| 전체 화면 전환 |  | Alt+Enter |



# 게임 메뉴 노트

* **보드 가져 오기  
  GetAsync** API를 사용하여 현재 게임 저장 슬롯을 위한 게임 보드를 로드합니다.
* **보드 읽기**  
  **ReadAsync**API(GetAsync의 다른 방법)를 사용하여 현재 게임 저장 슬롯에 게임 보드를 로드합니다.
* **보드 저장**  
  **SubmitUpdatesAsync** API를 사용하여 현재 게임 저장 슬롯을 위한 게임 보드를 저장합니다.
* **보드 재설정**  
  모든 문자 보드를 지웁니다. 보드가 이전에 저장되었다면 보드에 "더티 (dirty)"라고 표시됩니다. 보드가 아직 저장되지 않은 경우, 보드는 "더티 (dirty)"로 표시되지 않습니다.
* **보드 삭제  
  DeleteContainerAsync** API를 사용하여 현재 게임 저장 슬롯에 대한 게임 보드를 삭제합니다.
* **보드 블로브 삭제**  
  **SubmitUpdatesAsync** API를 사용하여 빈 컨테이너를 남겨두고 현재 게임 보드에 대한 블로브를 삭제합니다.
* **목록 컨테이너**  
  **GetContainerInfo2Async** API를 사용하여 모든 컨테이너를 열거하고 게임 화면의 디버그 출력 영역에 나열합니다.
* **목록 컨테이너 및 블로브  
  GetContainerInfo2Async** 과 **GetBlobInfoAsync** API를 사용하여 모든 컨테이너와 blob을 열거하고 게임 화면의 디버그 출력 영역에 나열합니다.

게임 플레이 노트

**게임 플레이**

게임은 5 x 5 그리드로 재생됩니다. 그리드의 아무 곳에나 문자를 배치할 수 있습니다. 보드를 가로 질러 또는 아래에 인식된 영어 단어를 형성하는 연속된 글자는 단어의 각 글자에 대한 점수 값의 합계에 기초하여 점수를 얻습니다. 목표는 점수를 최대화하는 것입니다. 각 문자에는 제한된 수의 글자를 사용해야 합니다. 나머지 카운트는 게임 보드 바로 위에서 추적됩니다.

**게임 보드 로딩**

데모 용으로 게임 보드는 게임 보드 화면이 나타나거나 새로운 게임 저장 슬롯으로 전환 할 때 **자동으로 로드되지 않습니다.** 이렇게 하면 개발자가 특정 게임 저장 슬롯에서로드 메소드 (가져 오기 또는 읽기)를 선택하고 로드 및 저장 조작을 완전히 제어할 수 있습니다.

**게임 보드 변경 및 자동 저장**

게임 보드에서 글자가 변경되거나 메뉴 재설정 명령을 사용하면 게임 보드가 "더티"(화면 상단의 보드 이름 뒤에 별표로 표시)로 표시됩니다. 더러운 게임 보드는 다음 조건에서 자동 저장됩니다:

* 다른 게임 보드 (게임 패드 LB/RB)로 전환
* 사용자 로그 아웃
* 게임 일시 중지

# 구현 정보

**GameSaveManager** 클래스는 게임에 대한 게임 저장 작업을 관리합니다. **InitializeForUser()** 메소드는 플레이어에 대한 연결된 스토리지 저장 컨텍스트를 설정합니다. 저장 데이터를 로드, 저장, 열거 삭제하는 메소드도 있습니다. 클래스의 각 메소드에 대한 사용법은 헤더 파일의 주석을 참조하십시오.

게임에서 사용하는 게임 데이터 구조에는 인덱스와 게임 보드의 두 가지 유형이 있습니다. 템플리트화 된 게임저장 (**GameSave**) 클래스는 GameSaveManager에서 모든 유형의 게임 데이터에 대해 데이터를 로드하고 저장하는 데 사용할 메서드를 제공합니다. 색인은 GameSaveManager.h에서 **GameBoardIndex** struct로 정의되며 주로 플레이어 ( "활성 보드")가 사용한 마지막 저장 슬롯을 추적하는 데 사용됩니다. 게임 보드 데이터는 GameBoard.h 의 게임 보드 (**GameBoard**) struct표시됩니다.

# 알려진 문제점

없음.

# 업데이트 기록

**초판: *2016년 2월***

Xbox One XDK 전용으로 2016년 2월 XDK와 함께 출시되었습니다.

**2016 년 3 월 업데이트:**

새로운 기능:

* 추가된 게임의 UWP 버전(별도의 Visual Studio 프로젝트가 XDK 및 UWP 용으로 제공됨), 게임 패드, 키보드 및 마우스 입력 지원.
* UWP 버전의 일시 중지에서 재개하면 마지막 동기화 이후 변경된 컨테이너가 표시됩니다(일시 중지된 상태에서 메모리에 로드된 경우 추가 변경 사항을 저장하기 전에 다시 로드해야 함)
* UWP 버전은 Xbox Live SDK 버전 1602와 UWP용 Xbox Live 플랫폼 확장을 사용합니다.

게임 저장 프로세스와 관련된 변경 사항:

* GameSaveManager Suspend() 메서드를 리팩토링하고 OnSignOut() 메서드를 추가했습니다.
* 이제 사용자 로그아웃은 게임 보드 색인이 어지러워진 경우 게임 보드 색인을 저장하고 더러워진 활성 게임 보드도 저장합니다.

기타 변경 사항:

* 단어 목록 파일은 상당히 작으며 단어 목록이 배경에서 계속 로드되는 동안 게임 보드가 표시됩니다

버그 수정:

* 일시 중지된 상태에서 게임을 다시 재개하고 사용자가 로그 아웃했을 때의 충돌 버그 수정 (null deref)

**2016년 4월 업데이트:**

변경사항:

* 분리된 GameSaveManager 구현은 UWP와 XDK 소스 파일로 구현되어 나란히 비교하기 쉽습니다.
* 업데이트된 DeviceResources (UWP 및 XDK 버전)를 최신 버전으로 업데이트
* 개선된 PLM 일시 중지 처리 :
  + 게임은 초기화와 동기화를 수행하는 동안 일시 중지 이벤트가 발생하면 게임 저장 색인을 로드하거나(또는 재설정) 게임 저장 메타 데이터를 로드하려고 시도하지 않습니다.
  + 작업 실행 중에 일시 중지 이벤트가 발생하면 비동기 게임 저장 초기화 작업이 끝날 때 게임이 게임 상태를 InGame으로 전환하려고 시도하지 않습니다.

버그 수정:

* 게임 저장 초기화 중 오류가 발생했을 때 "로딩 중"메뉴 상태에서 게임이 멈추는 버그 수정

**2016년 5월 업데이트:**

변경사항:

* UWP 버전이 Xbox Live SDK 버전 1604를 사용하도록 업데이트되었습니다 (그러나 여전히 버전 1602 및 1603과 코드 호환 가능)
* XDK 버전의 경우 디버그 로그 파일이 T: 대신 D:에 저장되므로 제목이 실행되지 않을 때 사용할 수 있습니다.
* UWP GameSaveManager의 경고가 추가되었습니다 InitializeForUser() 전달된 SCID가 null인 경우(게임 저장 컨텍스트는 오프라인에만 해당)입니다
* 코드 품질 향상

**2017 년 3 월 업데이트:**

변경사항:

* 이제 UWP 버전은 WinRT(C++/CX) XSAPI NuGet 이진 파일을 대신하여 C++ XSAPI NuGet 이진 파일을 사용합니다.
* UWP 버전은 기본적으로 Xbox Live SDK 버전 1703을 사용하지만 대부분의 이전 버전과 호환됩니다

**알림:** Xbox Live SDK 용 NuGet 이전 파일을 nuget.org에서 다운로드할 수 있습니다. Visual Studio에서 이 샘플을 빌드하면 자동으로 다운로드/복원됩니다.

* 사용자 관리를 위해 ATG 표준 "Xbox Live Tool Kit"사용으로 전환되었습니다
* UWP 샘플의 다중 사용자 (MUA) 버전으로의 전환 지원 (C++ XSAPI가 이를 완전히 지원하면, pch.h와 appx 매니페스트의 주석 처리 제거로 활성화될 수 있습니다)
* 단일 사용자 UWP 버전의 스위치 계정 기능을 제거하여 UWP 용 Xbox Live SDK 1703 버전의 switch\_account () API 제거와 호환됩니다.
* 입력 컨트롤 개선: 게임 커서 및 메뉴가 왼쪽 스틱, D 패드 또는 키보드 커서 화살표의 모든 조합을 통해 제어됩니다.
* 사용자 관리, UI 흐름과 일시 중지 동작으로부터의 재개 개선 기능을 포함합니다.

**2017년 4월 업데이트:**

변경사항:

* UWP 버전은 현재 Xbox에서 실행될 때 기본적으로 다중 사용자 앱(MUA)입니다.
* UWP 버전을 이제 PC의 기본 (소매) 샌드박스에서 실행할 수 있습니다.

**2017년 9월 업데이트:**

변경사항:

* Visual Studio 2017(15.1 업데이트)와 Windows 10 Creators SDK(15063)로 UWP 프로젝트를 업데이트했습니다
* UWP를 빌드하려면 새로운 Xbox Live Extensions SDK 15063이 필요합니다

**2017년 10월 업데이트:**

변경사항:

* 오류 처리 개선 및 디버그 출력 향상

**2017년 12월 업데이트:**

변경사항:

* Visual Studio 2017(15.4 업데이트) 및 Windows 10 Fall Creators Update SDK(16299)로 UWP 프로젝트를 업데이트했습니다.
* 연결 기억 영역 API 용 winmd가 이제 Windows 10 Fall Creators Update SDK에 포함되어 있으므로 UWP 빌드에 Xbox Live Extensions SDK가 더 이상 필요하지 않습니다.
* Xbox Live SDK 버전 2017.11 NuGet 패키지를 참조하도록 UWP 버전이 업데이트되었습니다.
* 동기화/로드/저장 비동기 호출 지속 시간 기록 추가

버그 수정:

* GetRemainingBytesInQuotaAsync()가 내부적으로 쓰로우하는 경우 UWP 버전에서 충돌 수정
* 처음 로드되었을 때 잘못된 저장 슬롯이 화면 상단에 표시될 수 있는 버그 수정

**2018년 5월 업데이트:**

변경사항:

* Visual Studio 2017(15.7 업데이트)와 Windows 10 2018년 4월 SDK(17134)로 UWP 프로젝트를 업데이트했습니다.

# 개인정보처리방침

샘플을 컴파일하고 실행할 때 샘플 실행 파일의 이름이 Microsoft로 보내져 샘플 사용을 추적 할 수 있습니다. 이 데이터 수집을 거부하려면 Main.cpp에서 "샘플 사용 텔레메트리"라는 코드 블록을 제거하면 됩니다.

Microsoft의 개인 정보 취급 방침에 대한 일반적인 내용은 [Microsoft 개인 정보 취급 방침](https://privacy.microsoft.com/ko-kr/privacystatement/)을 참조하십시오.