

---

# *OPEN GAME DEFINITIONS STANDARD: QuickGDS v0.1*

---

By Arda Erdikmen, Sep 2009

DRAFT

---

## **Table of Contents**

---

Table of Contents .....	1
Overview.....	2
Goals.....	2
QuickGDS and FullGDS.....	2
Why XML?.....	2
Terminology.....	3
Structure.....	3
GDS data, GDS File type and usage.....	3
Types.....	3
GameDefinitionStandard (Root element).....	4
GameDefinition (GameDefinitionStandard >).....	4
Platform (GameDefinitionStandard > GameDefinition >).....	4
GameFileName (GameDefinitionStandard > GameDefinition > Platform >).....	5
Mapping Controls (GameDefinitionStandard > GameDefinition > Platform >).....	5
ALT (Platform >).....	5
MAP (Platform > ALT >).....	6
ASSIGN (Platform> ALT > MAP >).....	7
SWITCH (Platform > ALT > MAP >).....	7
KEY Constants.....	7
Basic Controls.....	8
Additional Buttons.....	8
Virtual Keys.....	8
Planned Virtual Keys.....	9
Emulator functions.....	10
Game Information Group.....	10
Game Information.....	10
Name (BoundedString, optional) (Platform > Game Information >).....	11

Description (String, Optional) (Platform > Game Information >).....	11
ReleaseDate (BoundedString, optional) (Platform > Game Information >).....	11
Version(TwoPartNumber, optional) (Platform > Game Information >).....	11
Developers,Publishers(BoundedString, optional)(Platform > Game Information >).....	11
Infoseek ID (Number, Optional) (Platform > Game Information >).....	11
People (Bounded String, optional) (Platform > Game Information >).....	12
Example GDS Files .....	12
Minimalistic Example .....	12
Detailed Example.....	13
A GDS Example with Multiple Game Entries.....	14
Creating Multiple Alternative Controls for Same game with Runtime Switching.....	16

## Overview

---

Game definition standard (GDS) is a standard for creating key and button definitions for emulators. When you use an emulator on a console or a cell phone, there is always a problem about controls. Even on a PC with a hundred keys on the keyboard, setting up controls remains as a problem. You have to configure host machine's control buttons, keys or joysticks to emulated machine's accessories. Even this might not be enough, because not every game uses same way to control a character, navigate into menus or start the game.

GDS provides convenient method for generalize controls over different host platforms and on different emulated software by using XML engine.

## Goals

---

The main goal of this project is to bring a standard to the emulation scene, when playing games on different hosts and on emulated platforms with varying control methods. With GDS, all games on all platforms will play on host machine's native controls without needing any prior configuration. You just load the game and start playing immediately regardless that you are using a pc, a handheld, a console or a cell phone.

## QuickGDS and FullGDS

---

The standard itself divided into two parts: QuickGDS and GDS. This document only covers QuickGDS which covers only the basic parts. Note that a QGDS file is a GDS file with fewer features. Both standards are fully compatible with each other.

## Why XML?

---

While harder to parse and integrate into emulators, XML is a widely used format. To cope with parsing problem, there are numerous open source xml parsers out there.

We use XML because while it maintains human readability, it also provides multiple configurations in one gds file.

## Terminology

---

**\*to do\*\***

## Structure

---

A GDS file is an XML file, and conforms to XML standards which can be found at <http://www.w3.org/TR/REC-xml/>. Following elements, attributes and values are defined to use in a game definitions standard file:

### GDS data, GDS File type and usage

---

GDS, "Game Definition Standard" is not only a key assignment library, but it's also stores comprehensive information about a game.

GDS files must conform to XML 1.0 standard.

A GDS file may contain one or more games in it. Common usage of the GDS files like this:

```
It_came_from_desert.tzx
It_came_from_desert.gds
```

You provide a GDS file for every game image in the folder. We call this usage as "GDS pairs".

When talking about handheld devices, storage is another problem. So GDS provides another solution to storing GDS data: it's possible to merge all those files to form a single big "GDS archive". A gds data is formed to create an information hierarchy to make it easy to separate one game's information from another in an archive. When merged into one file, the gds archive must be named as "gds\_archive.gds".

When emulator checks if there is a GDS file associated to a game, "GDS pairs" always supersedes a "GDS archive". That means, if there is a "pair" for a game, an emulator never searches for a "GDS archive" unless the pair is broken or invalid. If there is no pair for a game, emulator first checks if there is a "gds\_archive.gds" file in the game directory, if it's there, it will search for the game entry by checking either game name (if that information is available to emulator) or game file names. Those fields are only mandatory fields in a GDS file.

## Types

---

Following data types are used in the GDS specification:

Type	Description
<b>TwoPartNumber</b>	A simple type contains version information eg. "1.2" (0-65535.0-65535)
<b>path</b>	Windows based path string. In a path string <code>\*?:"&lt;&gt; </code> characters not allowed.
<b>BoundedString</b>	A string which preserves white space and max. 512 chars long.
<b>Number</b>	A simple type contains an integer number.
<b>String</b>	A string which preserves white space and max. Up to 65535 chars long.
<b>Bool</b>	A Boolean value, either "False" or "True".

## GameDefinitionStandard (Root element)

---

GameDefinitionStandard, is the root element and it contains one or more game definitions.

```
< GameDefinitionStandard>
  ...One or more game definitions...
</ GameDefinitionStandard>
```

## GameDefinition (GameDefinitionStandard >)

---

This is the main element that holds all the data about a game and key bindings. This element has two attributes:

GameName, Version

Name	Type	Mandatory	Description
<b>GameName</b>	Bounded String	Yes	GameName is uniquely identifies your game to the emulator. It may be taken from a big database, or may be pre-set by the creator of the game. Stay true to the game in question's name. Sometimes games have different names across platforms. In this case, try to find oldest/original version's name, and put other names to <gameinformation> block.
<b>Version</b>	TwoPart Number	Yes	GDS definition version for this entry. It starts from 0.1 as of this document.
<b>Creator</b>	Bounded String	No	Creator of GDS file
<b>Generator</b>	Bounded String	No	The Tool used when creating GDS file

### Example

```
<GameDefinition GameName="Chuckie Egg" Version="0.1">
```

## Platform (GameDefinitionStandard > GameDefinition >)

---

This element specifies the definition file is specially to be played on a specific emulated machine. It must be declared before any MAP binding.

It has a set of optional sub items with additional attributes: "Hardware", "Model", "Version". Declaring one of those attributes makes that definition only used by that specified hardware or model. If PLATFORM attributes are empty, the definition is shared by different platforms and models.

Name	Type	Mandatory	Description
<b>Hardware</b>	Bounded String	No	Defines the hardware brand, e.g. Commodore, Sinclair, Amstrad, Nintendo, Gamepark.
<b>Model</b>	Bounded String	No	Defines the hardware model. E.g. Amiga, ZX Spectrum +, CPC464, GP32.
<b>Version</b>	Bounded String	No	Hardware or OS version. E.g. 3.5, 3.9, 1.0B. Note that this attribute is a string.

## Examples

---

```
<Platform Brand="Sinclair">
```

... definition is declared for any computer produced under Sinclair brand...

```
</Platform>
```

```
<Platform Model="Commodore 64">
```

...definition is declared only for commodore 64 model...

```
</platform>
```

```
<Platform>
```

...definitions will be shared on all platforms...

```
</platform>
```

## GameFileName (GameDefinitionStandard > GameDefinition > Platform >)

This attribute may be required to identify game to the emulator, yet may be ignored by it. This attribute is not only recommended, it's mandatory. GDS files may be merged together to form a big definition file, so this entry separates every entry. A Platform element may include more than one GameFileName element under it.

### Example

---

```
<Platform Brand="Sinclair" Model="ZX Spectrum + Peripherals="Multiface 3"
Version="Issue 3B">
```

```
  <GameFileName>somegame.tzx</Gamefilename>
```

```
  <GameFileName>somegame.z80</Gamefilename>
```

```
  <GameFileName>somegame.sna</Gamefilename>
```

```
  <GameFileName>somegame.szx</Gamefilename>
```

... Definitions go here

```
</Platform>
```

## Mapping Controls (GameDefinitionStandard > GameDefinition > Platform >)

### ALT (Platform >)

---

GDS supports different setups and ways to cycle between them. ALT tag sets a new set of controls to each host button. Every mapping set starts with an "ALT" tag. Attributes follows. While there is no limit on creating alternate key configurations, we should keep in mind that mobile devices usually are low on resources, and possibly there may be not enough memory to store alternate setups. To stay on the safe side, try to limit your alternate setup count to four.

Name	Type	Mandatory	Description
ID	BoundedString	Yes	A unique id for the button set.

<b>Description</b>	BoundedString	No	A description of the set of keys.
<b>Default</b>	Boolean	No	This attribute makes the alternative set as default controls. It will be the active control set when launching a game. You should specify at least one default attribute. If you don't specify the default attribute in the any of the alternate setup it will be up to the emulator to choose one of them, you cannot guarantee which control set will be used.

## Examples

```

<ALT id="main" description="Movement" Default="True">
  <MAP button="B_FIRE" description="Fire Gun" >
    <ASSIGN key=VK_SPACE/>
  </MAP>
</ALT>
<ALT id="inv" description="inventory keys">
  <MAP button="B_FIRE" description="Pickup Item">
    <ASSIGN key=VK_T/>
  </MAP>
</ALT>

```

## MAP (Platform > ALT >)

Map tag, assigns one or more virtual key to one or more host button.

Name	Type	Mandatory	Description
<b>Button</b>	Button	Yes	Host button to assign. You can find all available button constants at the next part of this document.
<b>Description</b>	BoundedString	No	Assigns a description to assigned key. If you don't provide a description, default one will be used. Default descriptions are the same as the element name but without the "B_" prefix. Eg. "B_START"'s default description is "START", "B_UP"'s is "UP" ...

## Examples

```

<MAP button="B_UP">
  ...mapped virtual key assignments and/or macros...
</MAP>

<MAP button="B_FIRE1" Description="Smart Bomb">
  ....assignments...
</MAP>

```

## ASSIGN (Platform > ALT > MAP >)

---

Copies the host button behavior to virtual key. If user presses the button for 1 sec, Virtual key will be pressed for exactly 1 second. This is the most basic assignment command. "Assign" generally is not used in the macros as it creates an interactive assignment and may produce unexpected results. (Macros will be introduced in the next version of this standard).

Name	Type	Mandatory	Description
<b>Key</b>	Vkey	Yes	Virtual key to be assigned

### Example

---

```
<MAP button="B_UP">  
  <ASSIGN key="VK_Q"/>  
</MAP>
```

## SWITCH (Platform > ALT > MAP >)

---

This tag switches settings between alternate setups. Switch is a way to assign a button to jump to a different "ALT".

Name	Type	Mandatory	Description
<b>Alt</b>	String	Yes	Alternate Location ID
<b>Clear</b>	Bool	No	Clear attribute clears all assignments to GDS_DUMMY before making a switch. Default setting for this attribute is "False". So assignments will be carried to alternate setups if they are not set in the alternatives.

### Examples

---

```
<ALT id="main" description="Movement" default="true">  
  <MAP button="B_SELECT" description=" switch to inventory keys">  
    <SWITCH alt="inv"/>  
  </MAP>  
</ALT>  
<ALT id="inv" description="inventory keys">  
  <MAP button="B_SELECT" description=" switch to Movement">  
    <SWITCH alt="main"/>  
  </MAP>  
</ALT>
```

## KEY Constants

---

## Basic Controls

---

The basic controls include a directional pad, two action buttons, a select and a start button.

Name	Recommended Usage
<b>B_SELECT</b>	Switching between alternate configurations, options, menus or pause.
<b>B_START</b>	To start a game. Provide a macro if needed.
<b>B_UP</b>	Cursor/Character Movement
<b>B_DOWN</b>	Cursor/Character Movement
<b>B_LEFT</b>	Cursor/Character Movement
<b>B_RIGHT</b>	Cursor/Character Movement
<b>B_FIRE1</b>	Main Action Button
<b>B_FIRE2</b>	Secondary action button
<b>B_TOOL1</b>	A tool button for assign utility keys (eg. QuickSave)
<b>B_TOOL2</b>	Another tool button (eg.QuickLoad)

The best binding is made by assigning all buttons (including additional buttons) to some function/virtual key. But in many cases this is not possible. But a binding never complete without all of the above buttons have an assignment.

## Additional Buttons

---

Use additional buttons if the game only and ultimately requires more buttons, even in this case, try to fit most basic functions to above keys. Keep in mind handheld devices usually does not have those extra keys. And using those keys usually renders your GDS definition unusable by those devices.

B\_FIRE3...31  
B\_LTHUMB1  
B\_LTHUMB2  
B\_LTHUMB3  
B\_RTHUMB1  
B\_RTHUMB2  
B\_RTHUMB3

### CELL PHONE SPECIFIC CONTROLS

B\_YES  
B\_NO  
B\_CPAD\_UP  
B\_CPAD\_DOWN  
B\_CPAD\_LEFT  
B\_CPAD\_RIGHT  
B\_MENU

### THE DEVICES WITH KEYBOARD

B\_KEY\_A...Z  
B\_KEY\_0...9

## Virtual Keys

---

These keys represent the emulated key, for example, a key press event on a emulated machine. There may be different keys in different machines, e.g. Apple key on an old Macintosh computer. Or set of control keys on an atari800XL or c64 Platform. Those keys will be added in later revisions of this document.

As first version intended to be used in Zx Spectrum Platform Emulators, only special spectrum keys are listed here:

Name	Description
VK_0..9	Numbers on keyboard above the letters, not the numeric part of the keyboard.
VK_A..Z	Keys of the keyboard, A to Z (ANSI Characters only)
VK_CAPSSHIFT	Caps Shift key on ZX Spectrum
VK_SYMBOLSHIFT	Symbol Shift key on ZX Spectrum
VK_ENTER	Enter Key on Zx Spectrum
VK_KEYCODE_0...255	Keyboard input codes on emulated machine. Those constants are for special keys, not listed on this listing.
VK_JOY1_UP	Joystick PORT 1 on emulated machine (Sinclair in this case)
VK_JOY1_DOWN	"
VK_JOY1_LEFT	"
VK_JOY1_RIGHT	"
VK_JOY1_FIRE1	"
VK_JOY1_FIRE2	"
VK_JOY2_UP	Joystick PORT 2 on emulated machine (Sinclair in this case)
VK_JOY2_DOWN	"
VK_JOY2_LEFT	"
VK_JOY2_RIGHT	"
VK_JOY2_FIRE1	"
VK_JOY2_FIRE2	"
VK_KEMPSTON_UP	Kempston Joystick on a ZX Spectrum
VK_KEMPSTON_DOWN	"
VK_KEMPSTON_LEFT	"
VK_KEMPSTON_RIGHT	"
VK_KEMPSTON_FIRE1	"
VK_KEMPSTON_FIRE2	"
VK_KEMPSTON_FIRE3	"
VK_MOUSE_X	Generic Mouse on a emulated machine X axis
VK_MOUSE_Y	", Y axis
VK_MOUSE_L	Left mouse button
VK_MOUSE_R	Right mouse button
VK_MOUSE_M	Middle mouse button

## Planned Virtual Keys

Following keys are not need to be supported right now, but it will be added to the later revisions of GDS format. You can use them if you need any.

VK_SHIFT	Generic Shift key
VK_F1..F12	Generic F-keys
VK_START	Start button on emulated machine
VK_SELECT	Select button on a emulated machine

VK_RESET	Reset button on emulated machine
VK_OPTION	Option button on emulated machine
VK_HELP	Help button on emulated machine
VK_ESC	Esc button on emulated machine
VK_CAPSLOCK	Caps Lock button on emulated machine
VK_IN(0..65535)_(0..255)	PORT Commands. (eg. VK_IN_31_255 )

## Emulator functions

These functions may not be supported by all emulators!

Name	Description	Recommended Button
GDS_QUICKSAVE	Makes emulator to make a quick save operation	B_TOOL1
GDS_QUICKLOAD	Makes emulator to make a quick load operation	B_TOOL2
GDS_SAVE	Makes emulator to execute a save operation	
GDS_LOAD	Makes emulator to execute a save operation	
GDS_OPTIONS	Makes emulator to show it's options dialog	
GDS_ESC	Makes emulator to performs an ESC action (eg. While in menu's)	
GDS_OK	Accept button for dialogues on emulator	
GDS_CANCEL	Cancel button for dialogues on emulator	
GDS_SELECT	Select button on emulator	
GDS_BACK	Makes emulator to performs an BACK action (eg. While in menu's)	
GDS_QUIT	Makes emulator to quit.	
GDS_RESET	Makes emulator to perform a reset.	
GDS_VIEW_DESCRIPTIONS	Makes emulator to show key descriptions on screen.	
GDS_DUMMY	A dummy keyword. Does nothing.	

\*These functions are not yet defined.

## Game Information Group

While not comprehensive, Open Game Definition Standard supports to include basic information about a game.

### Game Information

This group's root element is <GameInformation>.

#### Example

```
<GameInformation>
  ...informational elements...
</ GameInformation>
```

## **Name (BoundedString, optional) (Platform > Game Information >)**

---

Declares game's official name. While this element is also optional, declaring a name for a game is recommended.

### [Example](#)

---

```
<Name>Chuckie Egg</Name>
```

## **Description (String, Optional) (Platform > Game Information >)**

---

A description about game.

### [Example](#)

---

```
<Description>Fast paced platformer</Description>
```

## **ReleaseDate (BoundedString, optional) (Platform > Game Information >)**

---

### [Example](#)

---

```
<ReleaseDate>2006-08-07</ReleaseDate>
```

## **Version(TwoPartNumber, optional) (Platform > Game Information >)**

---

Specifies the version of the defined game.

### [Example](#)

---

```
<Version>1.0 </Version>
```

## **Developers,Publishers(BoundedString, optional)(Platform > Game Information >)**

---

These two tags may/may not have URLs to respective sites.

### [Example](#)

---

```
<Developers>  
<Developer URL="HTTP://www.somesite.com" >SomeSoftware House</Developer>  
</Developers>
```

```
<Publishers>  
<Publisher>Imagine</Publisher>  
</Publishers>
```

## **Infoseek ID (Number, Optional) (Platform > Game Information >)**

---

For zx spectrum platform, while not mandatory, infoseek ID is highly recommended.

### [Example](#)

---

<Infoseek>004087</Infoseek>

## People (Bounded String, optional) (Platform > Game Information >)

---

The people who worked on the game.

The sub elements are:

Code, Additional Code, Graphics, Sound, Music, Concept, Design

```
<People>
  <Code>James Cook</Code>
  <Code>Matt Winston</Code>
  <Design>Rosetta Rock</Design>
  <Music>Eth Baltaci</Music>
  <Graphics>Rim Kavcar</Graphics>
</People>
```

## Example GDS Files

---

### Minimalistic Example

---

```
<GameDefinitionFile>
  <GameDefinition GameName="Some Game" Version="0.1" Creator="Jack Black">
    <Platform>
      <GameFileName>somegame.tzx</GameFileName>
      <ControlAssignment>
        <ALT ID="1">
          <MAP button="B_UP" >
            <ASSIGN key="VK_Q"/>
          </MAP>
          <MAP button="B_DOWN" >
            <ASSIGN key="VK_A"/>
          </MAP>
          <MAP button="B_LEFT" >
            <ASSIGN key="VK_O"/>
          </MAP>
          <MAP button="B_RIGHT" >
            <ASSIGN key="VK_P"/>
          </MAP>
          <MAP button="B_FIRE1" >
            <ASSIGN key="VK_M"/>
          </MAP>
          <MAP button="B_FIRE2" >
            <ASSIGN key="VK_SPACE"/>
          </MAP>
          <MAP button="B_START" >
            <ASSIGN key="VK_1"/>
          </MAP>
        </ALT>
      </ControlAssignment>
    </Platform>
  </GameDefinition>
</GameDefinitionFile>
```

```

    <MAP button="B_SELECT" >
      <ASSIGN key="VK_H"/>
    </MAP>
    <MAP button="B_TOOL1" >
      <ASSIGN key="GDS_QUICKSAVE"/>
    </MAP>
    <MAP button="B_TOOL2" >
      <ASSIGN key="GDS_QUICKLOAD"/>
    </MAP>

  </ALT>
</ControlAssignment>
</Platform>
</GameDefinition>
</GameDefinitionFile>

```

## Detailed Example

---

```

<GameDefinitionFile>
<GameDefinition GameName="Some Game 2" Version="0.1">
  <Platform Brand="Sinclair" Model="Zx Spectrum+">

    <GameFileName>somegame2.tzx</GameFileName>
    <GameFileName>somegamewithdifferentname.sna</GameFileName>

    <GameInformation>

      <Name>Some Game 2</Name>
      <Description>Sequel to Fast paced platformer</Description>
      <ReleaseDate>2008-08-07</ReleaseDate>
      <Version>1.0 </Version>
      <Developers>
        <Developer URL="HTTP://www.somesite.com" >SomeSoftware
House</Developer>
      </Developers>
      <Publishers>
        < Publisher URL="HTTP://www.somesite.com" >SomeSoftware
House</ Publishers>
      </Publishers>
      <People>
        <Code>James Cook</Code>
        <Code>Matt Winston</Code>
        <Design>Rosetta Rock</Design>
        <Music>Eth Baltaci</Music>
        <Graphics>Rim Kavcar</Graphics>
      </People>
    </GameInformation>

  <ControlAssignment>

```

```

    <ALT ID="1">
      <MAP button="B_UP" >
        <ASSIGN key="VK_Q"/>
      </MAP>
      <MAP button="B_DOWN" >
        <ASSIGN key="VK_A"/>
      </MAP>
      <MAP button="B_LEFT" >
        <ASSIGN key="VK_O"/>
      </MAP>
      <MAP button="B_RIGHT" >
        <ASSIGN key="VK_P"/>
      </MAP>
      <MAP button="B_FIRE1" >
        <ASSIGN key="VK_M"/>
      </MAP>
      <MAP button="B_FIRE2" >
        <ASSIGN key="VK_SPACE"/>
      </MAP>
      <MAP button="B_START" >
        <ASSIGN key="VK_ENTER"/>
      </MAP>
      <MAP button="B_SELECT" >
        <ASSIGN key="VK_R"/>
      </MAP>
    </ALT>
  </ControlAssignment>
</Platform>
</GameDefinition>

</GameDefinitionFile>

```

## A GDS Example with Multiple Game Entries

---

(Note that Game Definitions from different versions of GDS/QGDS can be used together in a GDS container.)

```

<GameDefinitionFile>

  <GameDefinition GameName="Some game" Version="0.1">
    <Platform>
      <GameFileName>somegame.tzx</GameFileName>
      <ControlAssignment>
        <ALT ID="1">
          <MAP button="B_UP" >
            <ASSIGN key="VK_Q"/>
          </MAP>
          <MAP button="B_DOWN" >
            <ASSIGN key="VK_A"/>
          </MAP>
          <MAP button="B_LEFT" >

```

```

        <ASSIGN key="VK_O"/>
    </MAP>
    <MAP button="B_RIGHT" >
        <ASSIGN key="VK_P"/>
    </MAP>
    <MAP button="B_FIRE1" >
        <ASSIGN key="VK_N"/>
    </MAP>
    <MAP button="B_FIRE2" >
        <ASSIGN key="VK_M"/>
    </MAP>
    <MAP button="B_START" >
        <ASSIGN key="VK_S"/>
    </MAP>
    <MAP button="B_SELECT" >
        <ASSIGN key="VK_D"/>
    </MAP>
</ALT>
</ControlAssignment>
</Platform>
</GameDefinition>

<GameDefinition GameName="Another Game" Version="0.4">
    <Platform>
        <GameFileName>anothergame.tzx</GameFileName>
        <ControlAssignment>
            <ALT ID="SomeID">
                <MAP button="B_UP" >
                    <ASSIGN key="VK_1"/>
                </MAP>
                <MAP button="B_DOWN" >
                    <ASSIGN key="VK_Q"/>
                </MAP>
                <MAP button="B_LEFT" >
                    <ASSIGN key="VK_9"/>
                </MAP>
                <MAP button="B_RIGHT" >
                    <ASSIGN key="VK_0"/>
                </MAP>
                <MAP button="B_FIRE1" >
                    <ASSIGN key="VK_Z"/>
                </MAP>
                <MAP button="B_FIRE2" >
                    <ASSIGN key="VK_M"/>
                </MAP>
                <MAP button="B_START" >
                    <ASSIGN key="VK_R"/>
                </MAP>
                <MAP button="B_SELECT" >
                    <ASSIGN key="VK_T"/>
                </MAP>
            </ALT>
        </ControlAssignment>
    </Platform>
</GameDefinition>

```

```

    </ControlAssignment>
  </Platform>
</GameDefinition>

</GameDefinitionFile>

```

## Creating Multiple Alternative Controls for Same game with Runtime Switching

---

```

<GameDefinitionFile>
  <GameDefinition GameName="Some game" Version="0.1">
    <Platform>
      <GameFileName>somegame.tzx</GameFileName>
      <ControlAssignment>

        <ALT id="main" description="Movement">
          <MAP button="B_SELECT" description=" switch to inventory keys">
            <SWITCH alt="inv"/>
          </MAP>
          <MAP button="B_LEFT" >
            <ASSIGN key="VK_9"/>
          </MAP>
          <MAP button="B_RIGHT" >
            <ASSIGN key="VK_0"/>
          </MAP>
          <MAP button="B_FIRE1" >
            <ASSIGN key="VK_Z"/>
          </MAP>

        </ALT>

        <ALT id="inv" description="inventory keys">
          <MAP button="B_SELECT" description=" switch to Movement">
            <SWITCH alt="main"/>
          </MAP>
          <MAP button="B_LEFT" >
            <ASSIGN key="VK_Z"/>
          </MAP>
          <MAP button="B_RIGHT" >
            <ASSIGN key="VK_X"/>
          </MAP>
          <MAP button="B_FIRE1" >
            <ASSIGN key="VK_SPACE"/>
          </MAP>

        </ALT>

      </ControlAssignment>
    </Platform>
  </GameDefinition>

```

</GameDefinitionFile>