# Brain Invaders Level Format Structure

The new version of Brain Invaders supports, among other things, new dynamic levels. These levels are loaded in by the game through a XML-file. This document details the structure of that XML-file, so that levels can be written by hand or through a level XML generator. Please note that Brain Invaders is currently hard coded to expect a correct XML structure and will thus crash in cases where the structure is broken.

## File level

The file of a Brain Invaders level should be called “BILevel<x>.xml” where <x> is the number of the level. Brain Invaders starts at level 1 and increases the number after a level is finished. If the next level cannot be found, the game will end, so after level 3, there has to be a level 4 in order to also load level 5. There is also a “BILevelTraining.xml” which is used for the P300-calibration but it is recommended to keep this document as is.

## Basic level information

A Brain Invaders level should always be opened with the tag **<Level>** Within this tag, the entire level description is put. The first children of the level tag should describe the basic information of the level, after which multiple Level Blocks should describe the actual elements.

### TotalBlocks

Use the tag **<TotalBlocks>** to tell the game how many Alien Blocks is can expect to place in the level. This number needs to be the same amount of blocks later described in the level file. Keep in mind that the maximum amount of blocks is currently capped at 25. To use more, update the maximum in the Brain Invaders source code. The value of this tag has to be an integer.

### Movement Speed

Next, the game expects a few tags on movement speed, so blocks can move in the appropriate speed. **<StepSpeed>** gives the speed in seconds it takes for a new step to be taken. **<SpeedIncrease>** gives a value with which the stepspeed will be multiplied after each step. This means that if the value is smaller than 1, the speed will increase and decrease if otherwise. At value one the speed will stay constant. Finally the tag **<MaxSpeed>** gives the maximum speed in seconds. If this speed is reached, the speed increase will stop. Keep in mind that maxSpeed prevents speding up from stopping, so be careful when using values larger than 1 for the Speed Increase.

## Block

As specified above, the **<Level>** tag should contain the same amount of **<Block></Block>** tags as specified in **<TotalBlocks>**. Within the **<Block>** Tag, a block of Aliens is defined. These Aliens move according to the same path, in a relative position from one another. This allows the level builder to specify groups rather than just single aliens (though a single alien can be described by just putting one in a block). The Block-tag consists of **<TotalAliens>** to tell the game how many aliens are in the block, **<Path>** to describe the path of the block and multiple **<Alien>**-tags, describing the same amount of aliens as given in **<TotalAliens>**.

## Path

The **<Path></Path>** tag describes the full path of a Block. The tag is started with **<NumOfSteps>** indicating the amount of steps to be taken by the path, followed by that amount of **<Step>** tags. Within the **<Step>** tag, a **<X>** and **<Y>** value are given, specifying the coordinates between which the path moves. Please refrain from changing both X and Y from step to step, as this may cause out of bound behaviour. The X values should vary between 0 and 1200, the Y value between 0 and 900. Furthermore, keep in mind that Aliens within the same block are spaced a distance of 150 over both X and Y apart and that each step taken is 75.

## Alien

Lastly, a **<Block>** tag should contain the appropriate amound of **<Alien>** tags. These tags describe each individual Alien within the block. An Alien-tag contains a **<Name>** tag, which gives a (unique!) name to the Alien. We recommend using the name “A<x>:<y>“ where x and y are the values within the absolute matrix. This is followed by the tags **<RelPos>** and **<MatPos>**, both containing a X and Y value respectively. The relative position determines the position of the alien within the block. Where 0, 0 is the starting position of the path, and every value of X and Y corresponds to a 150 spacing from that position. Keep in mind that the block can only support 25 by 25 Aliens (which should be more than enough as this allows for out of screen Aliens. The Matrix Position is the position of the Alien within the blinking Matrix of the P300 system. Currently we only support a 6 by 6 matrix, but this should change in the future. Make sure each alien is placed in one unique matrix position. Lastly, there is a **<Type>** tag, specifying the alien type as an integer. Currently the values correspond to:

1. A Regular Alien, small sized
2. A Regular Alien, medium sized
3. A Regular Alien, large sized
4. A Target Alien
5. A Distractor Alien, Target Coloured
6. A Distractor Alien, Different Colour

## Overview

In brief, the XML document should be structured as such:

<Level>

<TotalBlocks>1</TotalBlocks>

<StepSpeed>1.0</StepSpeed>

<SpeedIncrease>1.0</SpeedIncrease>

<MaxSpeed>1.0</MaxSpeed>

<Block>

<TotalAliens>1</TotalAliens>

<Path>

<NumOfSteps>1</NumOfSteps>

<Step>

<X>0</X>

<Y>0</Y>

</Step>

</Path>

<Alien>

<Name>A0:0</Name>

<RelPos>

<X>0</X>

<Y>0</Y>

</RelPos>

<MatPos>

<X>0</X>

<Y>0</Y>

</MatPos>

<Type>0</Type>

</Alien>

</Level>