Unleashing Aurora Gt

04: Game Editor Interface

gameloft

Version

Date	Author	Version	Changelog
21/07/08	gaspar.deelias@gameloft.com	1.0.5	Initial Version

Guideline

Topics of this presentation:

- Game Editor Interface:
  - Game Explorer
  - Object Properties Window
  - Preview Window
  - Toolbar
  - Templates Screen
- Different options for layers, levels and camera.
- Blending Layer.

Q: Imagine you are asked to develop a 2D platform game from scratch, what would you do first?

- Start coding some levels, then think about design, and finally kill yourself when the producer tells you that your game is useless.....

- NO WAY

AuroraGT

Introduction

- We need LEVEL DESIGN independent from the CODE.
  (So our code is just a level interpreter.)

- The level editor is a key part in game programming (mostly in side scrolling games).

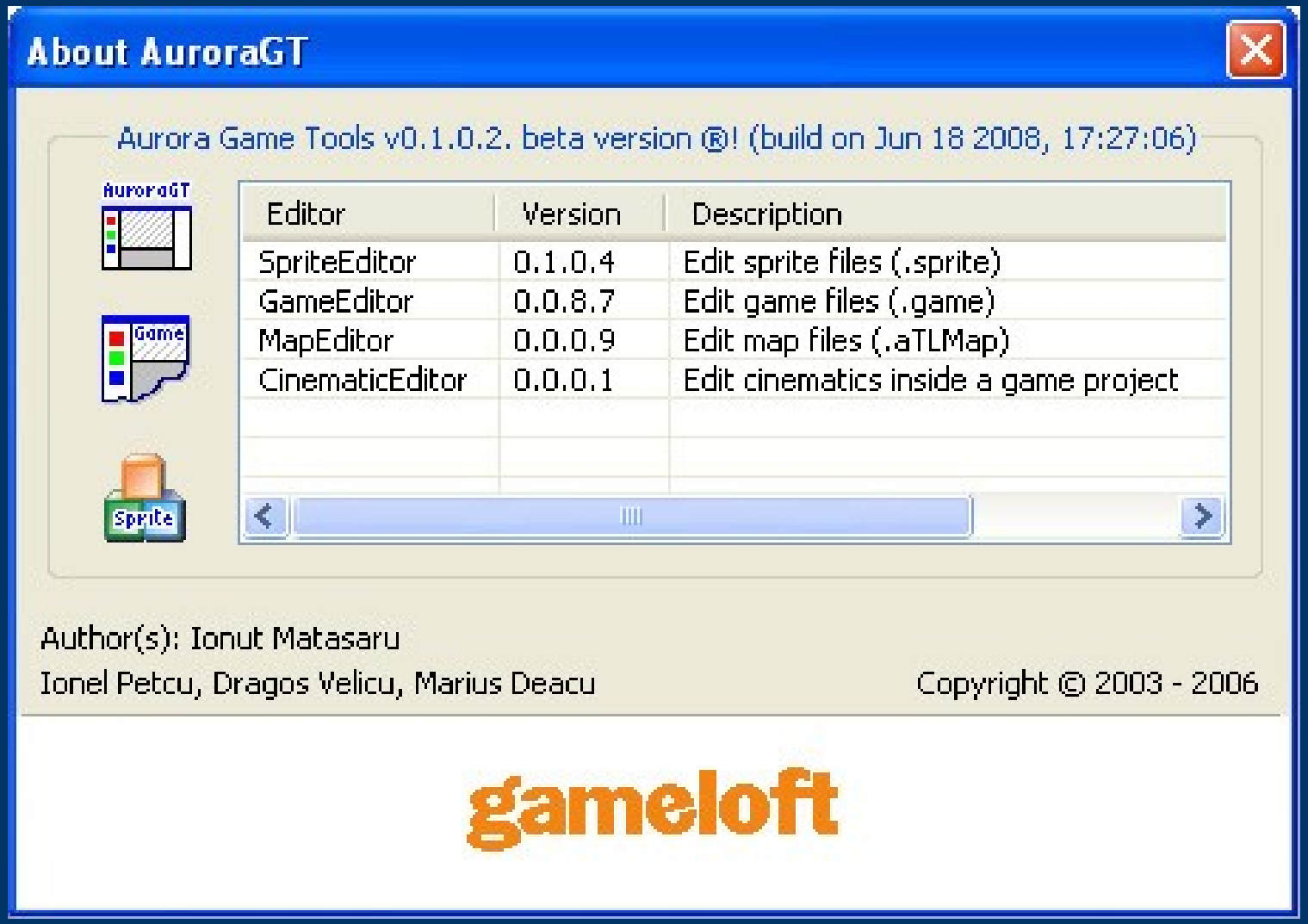
AuroraGT

Introduction

- Importance of LEVEL CREATION TOOL
- (Level modifications will be made till the end of project's life)
- This is where AuroraGT fits...
- AuroraGT is more than a level editor:
  - Sprite editor (.sprite files)
  - Map editor (.aTLMap files)
  - Level editor (.game files)
  - Cinematic editor (.game files)

AuroraGT

Reference Version(1)



[1] https://terminus.mdc.gameloft.org/vc/tools/AuroraGT (r1189)

AuroraGT

vs Other game editors

- Commercial games released with a level editor.
- Used by designers and the actual users.
- AuroraGT is more than that.
- Used for many games! Not just one
- AuroraGT is flexible

Q: Can you imagine where is this flexibility? How can we adapt or prepare AuroraGT for different games?

AuroraGT

Flexibility

A: A template file is defined for this purpose (.gts)

- The .gts file:
  - Created by developers.
  - Every game has its own template file.
  - Is a list of resources for a game.
  - Is mandatory.
  - Is an abstract layer

AuroraGT

Icons and Symbols used through this document

 Game Explorer

 Object Properties

 Template Explorer

 Preview Window

 Toolbar

.GTS file

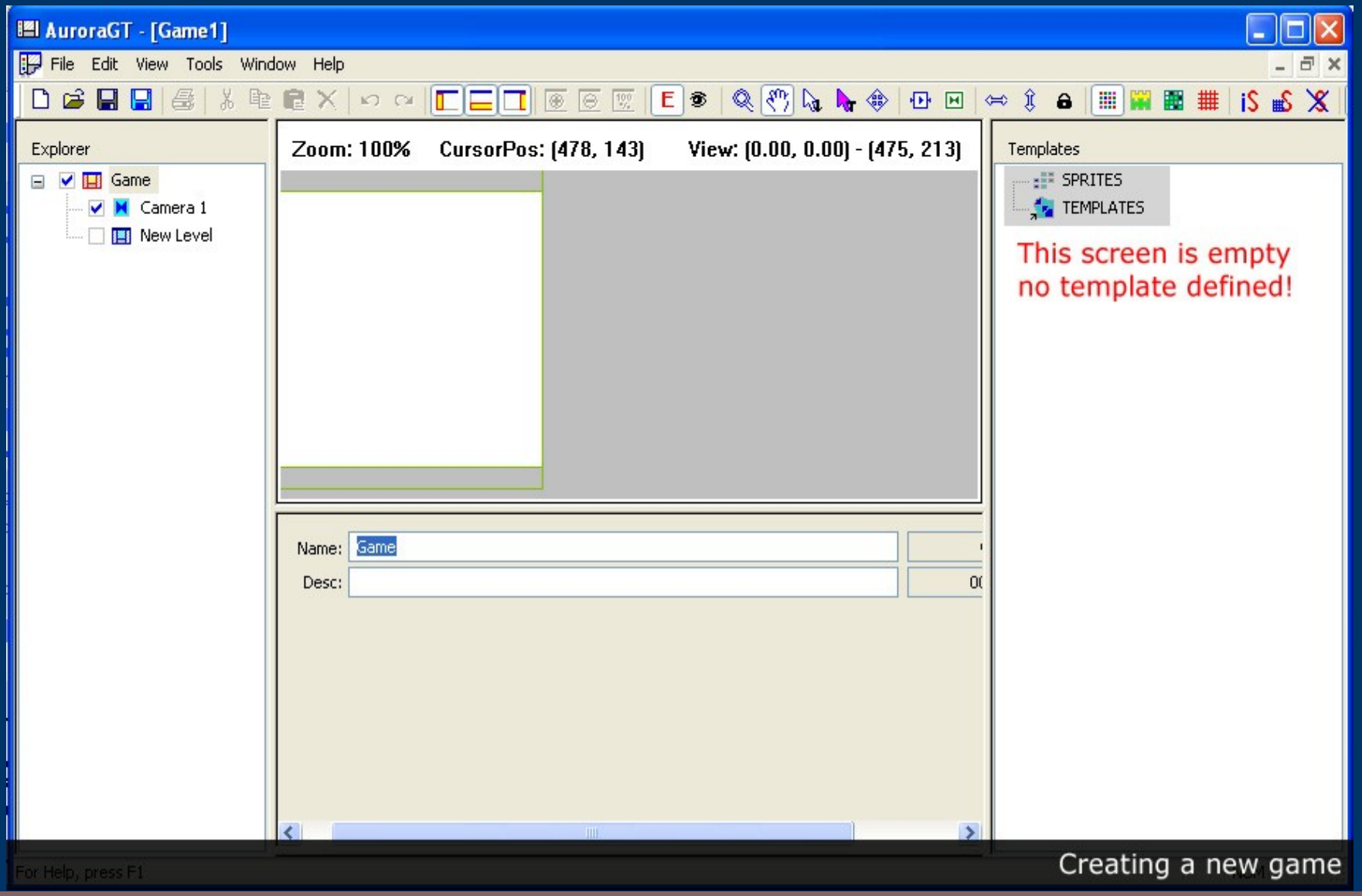
.GAME file

NOTE: From now and on, AuroraGT stands for AuroraGT Game Editor.

AuroraGT

The templates file (.gts)

- This is an empty created game:



AuroraGT

The templates file (.gts)

- This is the .game file saved by AuroraGT:

- In the previous screenshot there was no resources defined, so we couldn't insert objects or maps in the game!
- We need to specify a template to use.
- Let's go to the next slide...

```
{
    FLAGS 0x0000008F

    CAMERA "Camera 1"
    {
        FLAGS 0x0000000B
        POS 0 0
        SIZE 176 204
        LIMIT_AREA 0 0 704 416
        FILL_COLOR 128 128 128 128
        HIDE_TOP 14
        HIDE_BOTTOM 14
    }

    LEVEL "New Level"
    {
        FLAGS 0x0000000A
        POS 0 0
    }
}
```

.game myGame.gts

AuroraGT

The templates file (.gts)

Q: Where is defined this template?

- To specify which template to use, we need to add inside our .game file the command:

LOAD_TEMPLATES “myGame.gts”

```
{
    FLAGS 0x0000008F

    LOAD_TEMPLATES "myGame.gts"

    CAMERA "Camera 1"
    {
        FLAGS 0x0000000B
        POS 0 0
        SIZE 176 204
        LIMIT_AREA 0 0 704 416
        FILL_COLOR 128 128 128 128
        HIDE_TOP 14
        HIDE_BOTTOM 14
    }

    LEVEL "New Level"
    {
        FLAGS 0x0000000A
        POS 0 0
    }
}
```

.game game1.game

- We will explain how to create a .gts file in the next session.

AuroraGT

The Game Editor

- The .game file is an objects bank where all game resources are combined to create gameplay in levels.
- It enables game designers to create a game with a simple graphical tool with no need to know a thing about coding.
- Learning how it works will help us to solve problems and bugs faster than before.

- What's the difference between comercial game editors and Gameloft's AuroraGT game editor?

AuroraGT

The Game Editor

- AuroraGT is a really adaptive editor, it's used in many games, and some of them very different.
- As we produce a big quantity of games, and in general they're small compared with pc/console games, there's no way to have an specific game editor for each game!
- This requirement gives us some pros and some cons.

AuroraGT

Pros 'n Cons

- PROS:
  - No need to redesign a level editor for similar games.
  - Developers get used to it and the learning curve is more efficient since it doesn't have big changes between versions.
  - Aurora gets improved more and more every version.

- CONS:
  - Aurora Game Editor is not completely independent from the code, we need to recompile the game after changing or adding features.
  - The template file .gts needs to be created for every game (this gts file makes AuroraGT portable to different games).
  - This tool gets bigger and some features are unknown due to lack of documentation.

AuroraGT

Interface

- Let's start with AuroraGT game editor interface and its main sections:

1. Template Explorer
2. The Game Explorer
3. Object Properties Window
4. Preview Window
5. Toolbar

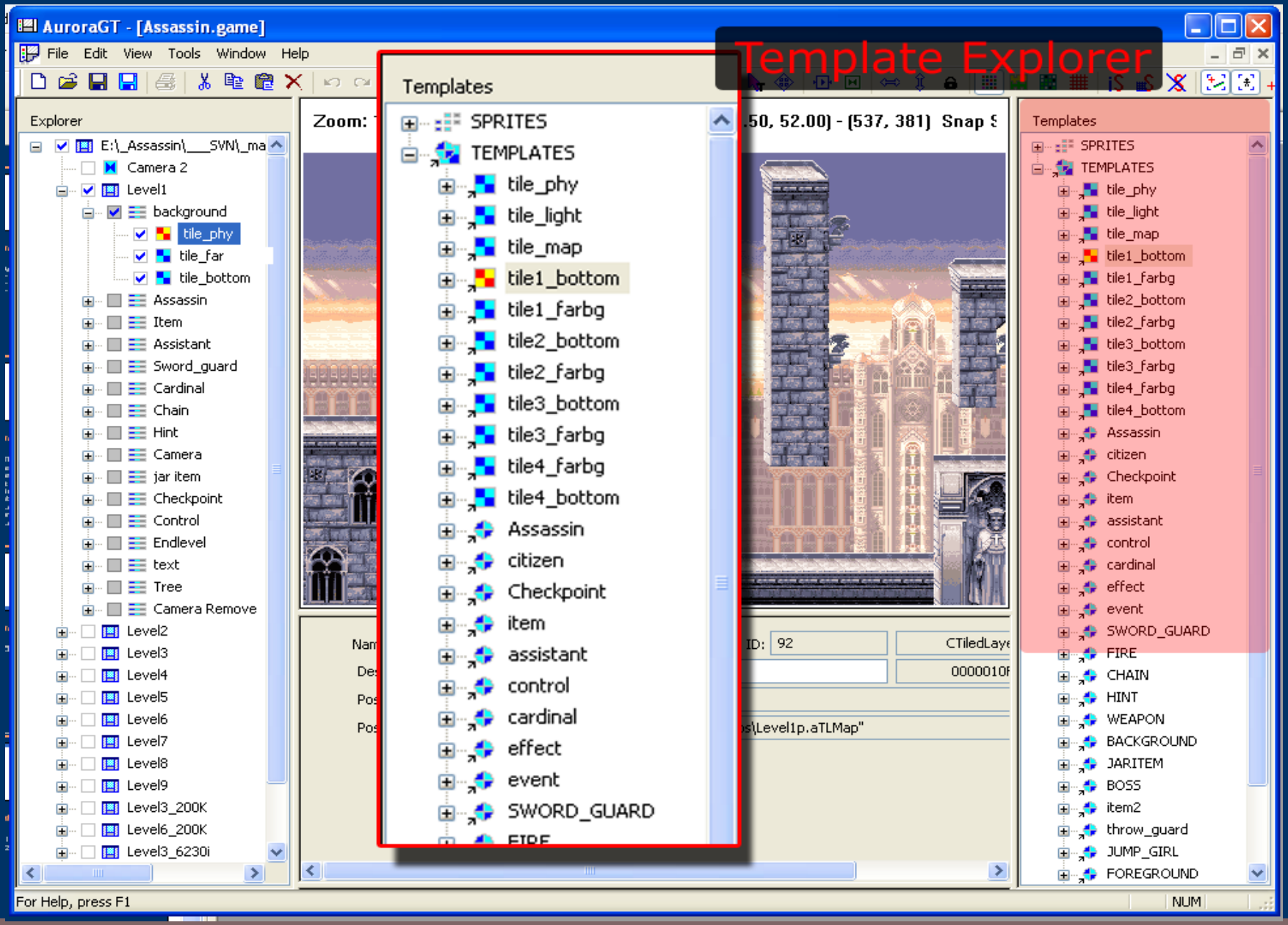
AuroraGT

Interface: Template Explorer

- Template explorer:
  - Usage, examples.
  - Adding an actor.
  - Removing an actor
  - Clonning
  - Actor properties
  - Class vs Instance

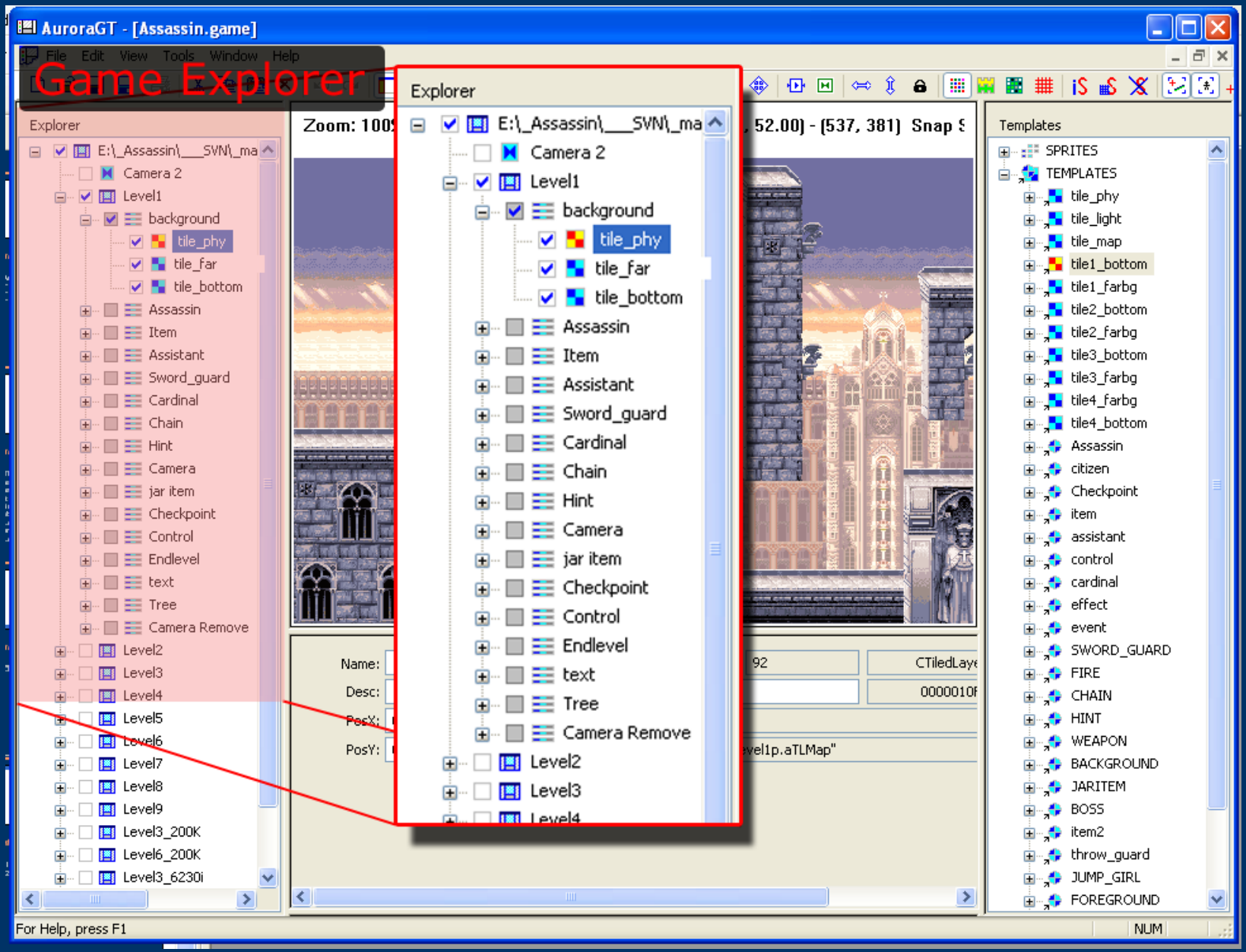
AuroraGT

Interface: Template Explorer



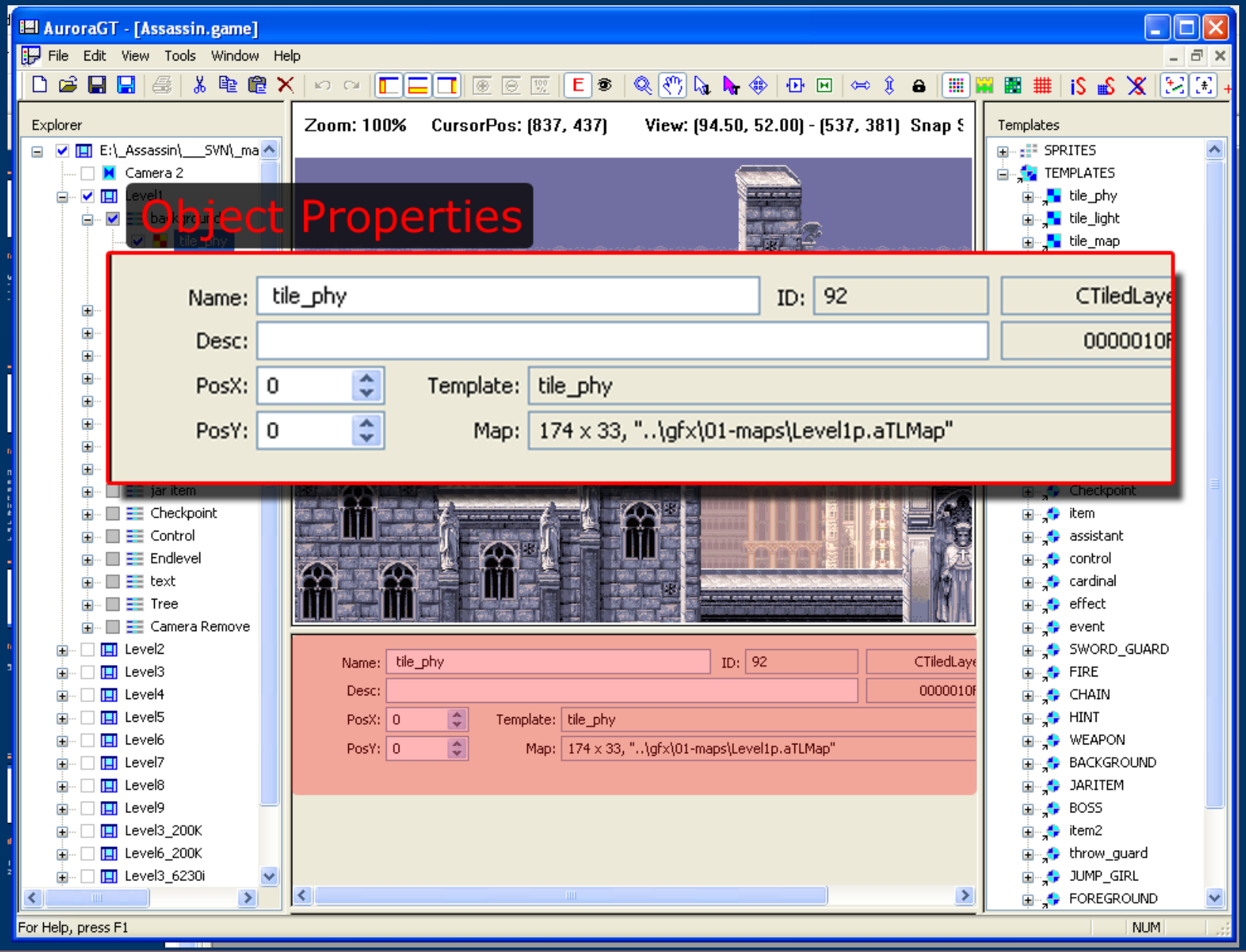
AuroraGT

Interface: Game Explorer



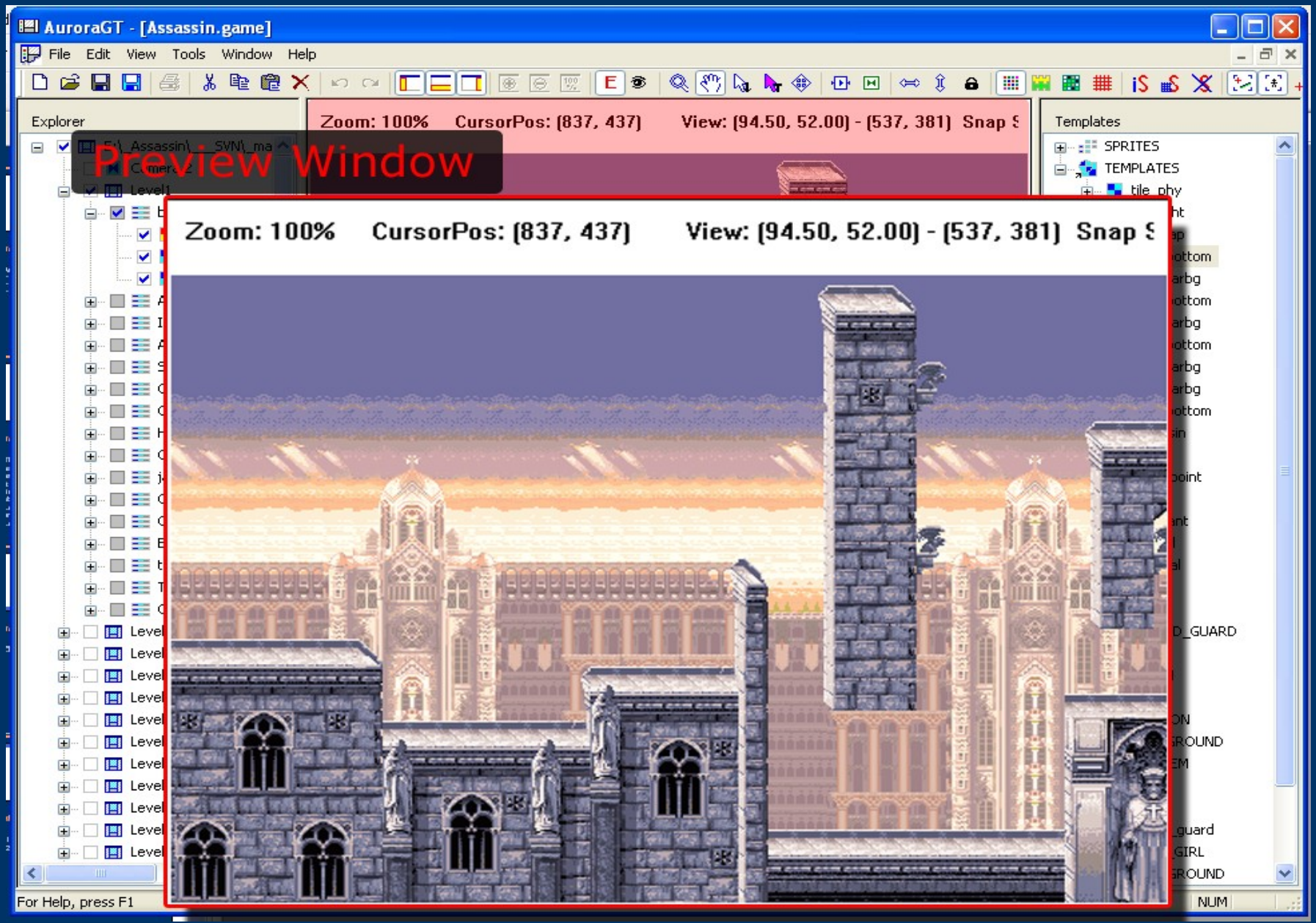
AuroraGT

Interface: Object Properties



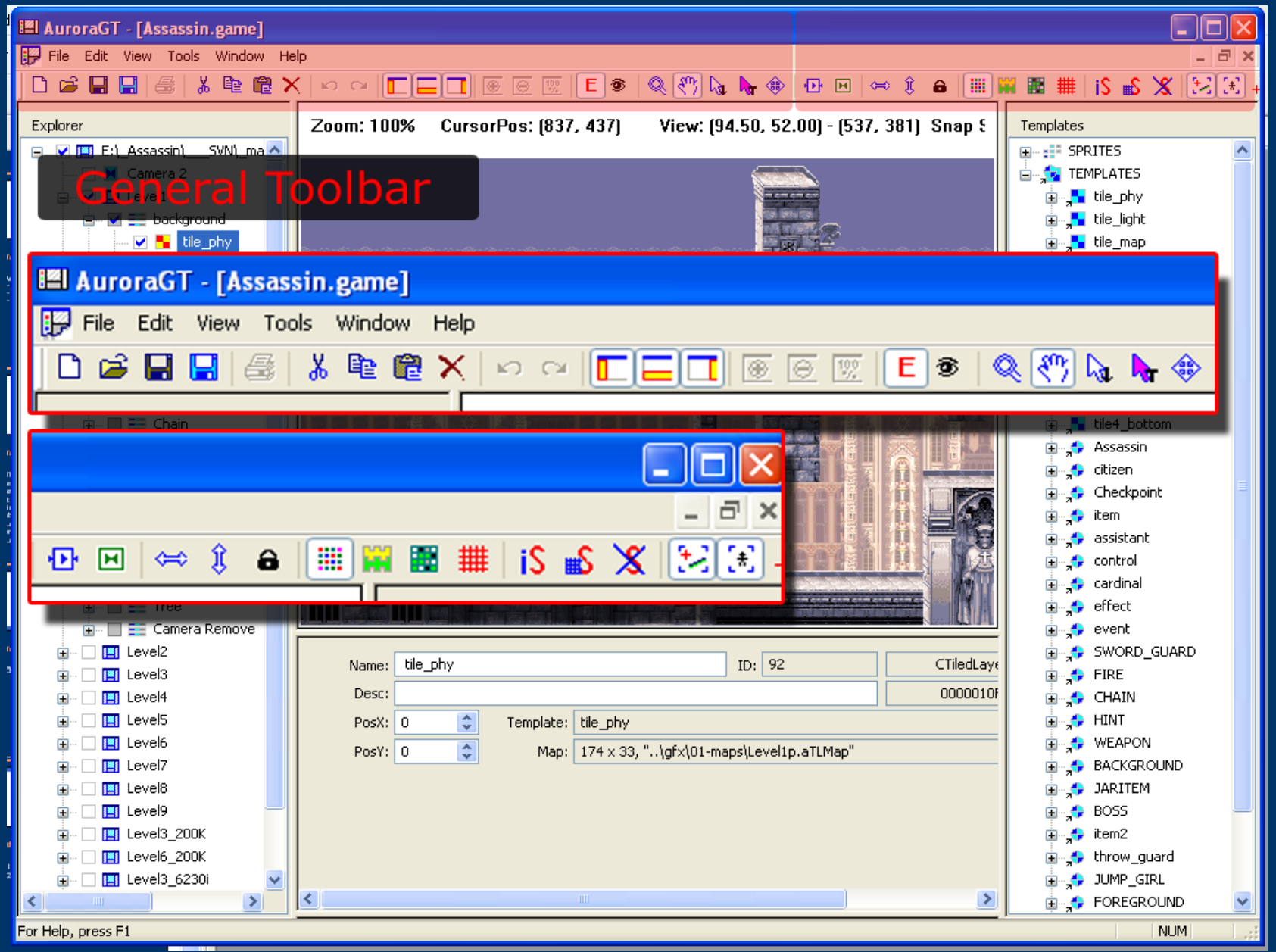
AuroraGT

Interface: Preview Window



AuroraGT

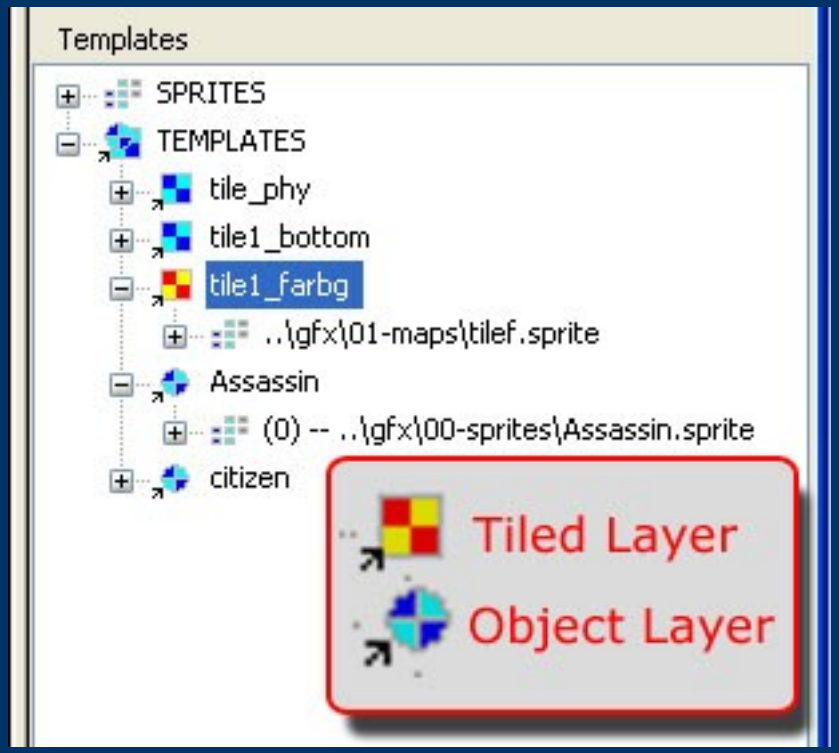
Interface: Toolbar



Interface

1. Template Explorer

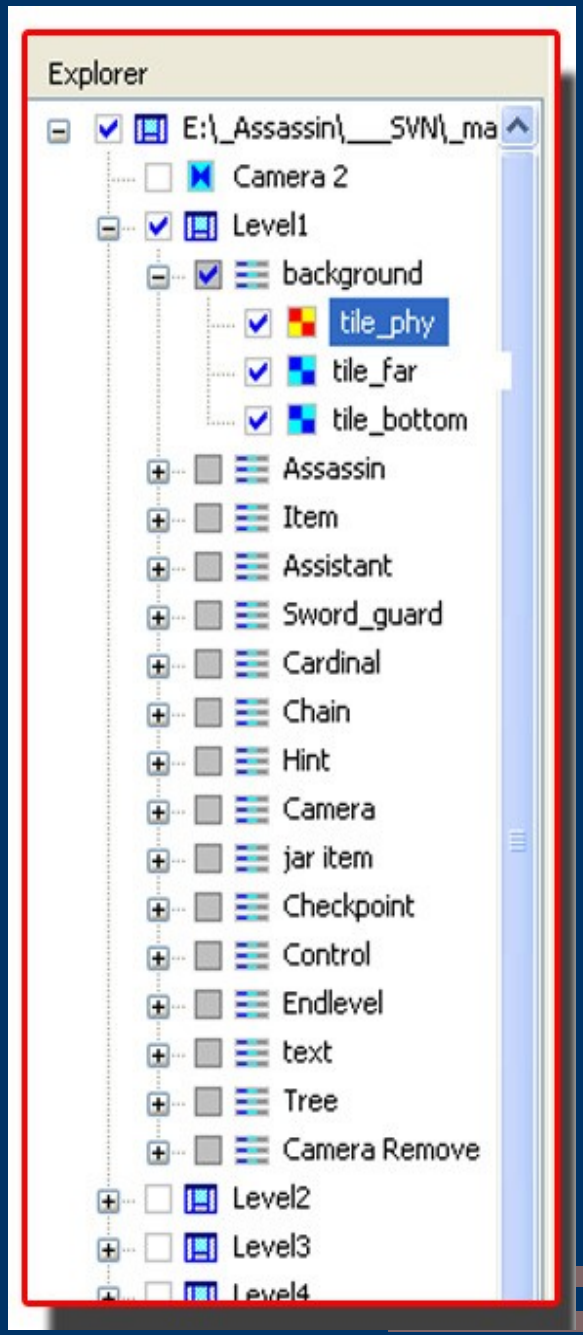
- Is a list of all available resources for our game.
- If a sprite or tileset is not listed here, we can not use it in our game.
- It also defines object properties.
- Is a tree view of our game's template (.gts)



– Contains:
  Tileset Layers: includes tilesets used by maps (.aTLMap files).
  Object Layers: the rest of the available sprites are listed here.
  Sprites List: Every sprite listed in the Templates file is also listed here.

Interface

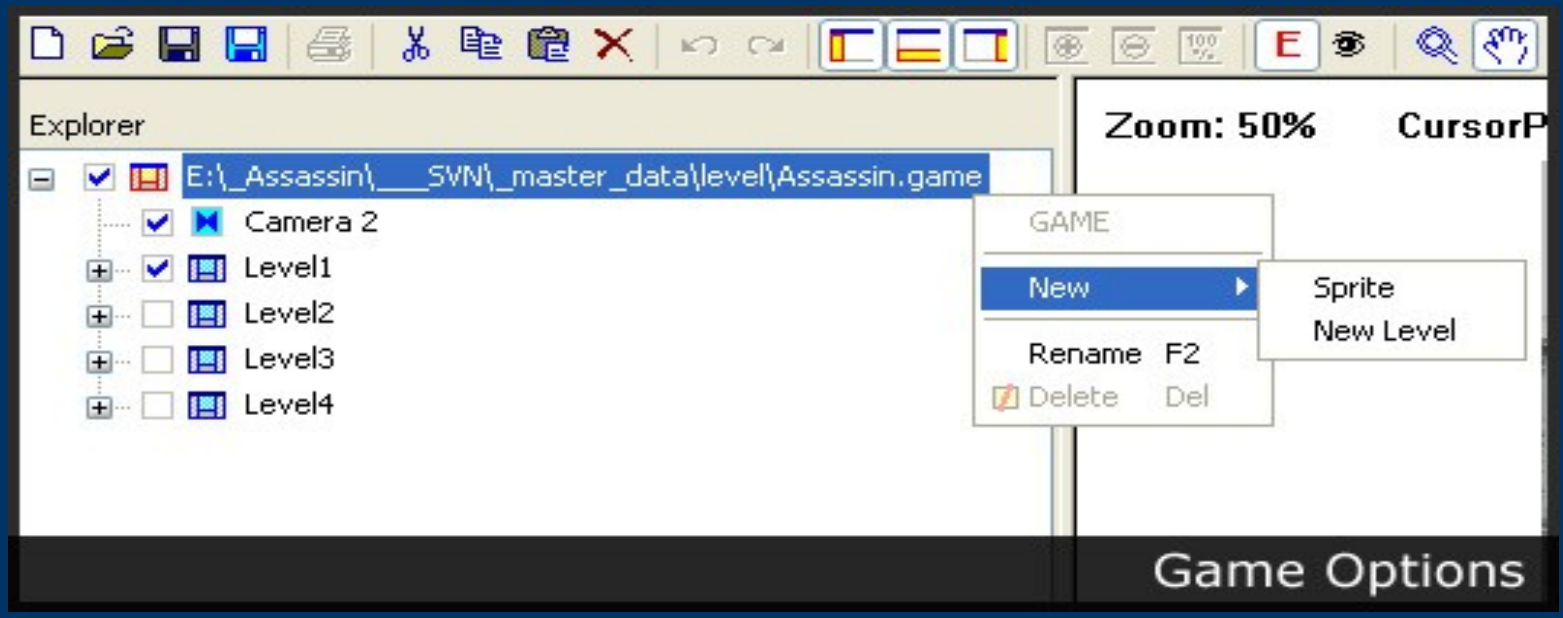
2. Game Explorer



- Organized in a three view.
- One camera only
- One or more levels contain:
  - Tiled Layers
  - Object Layers
  - Cinematics
- Layer group
- Painting order

Game Explorer

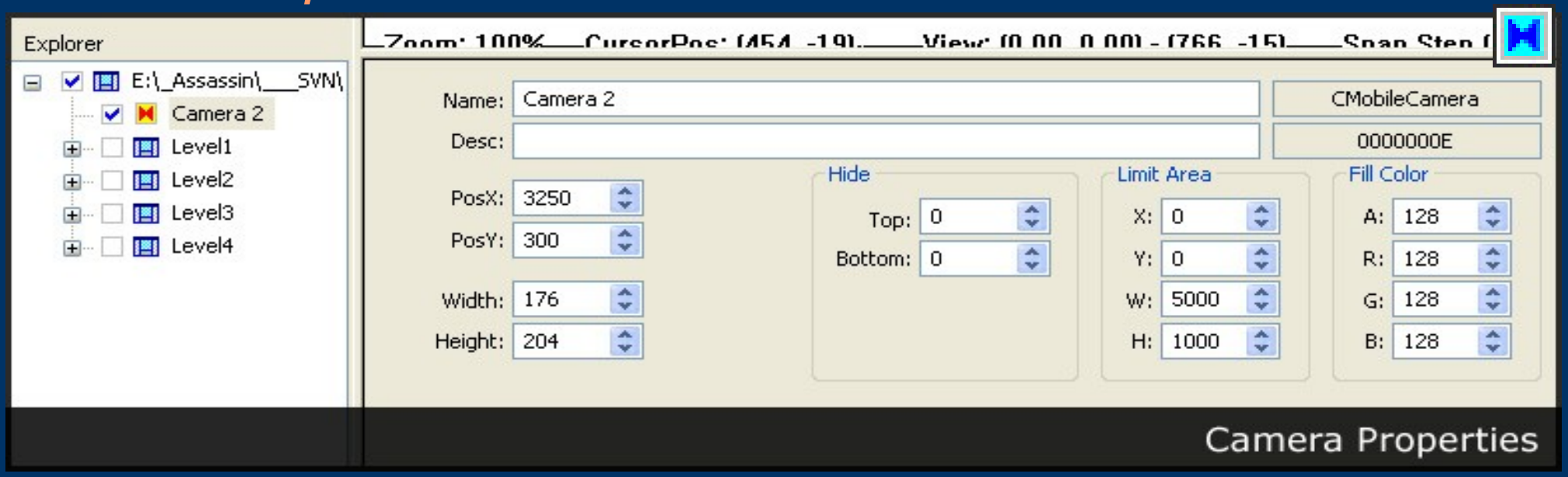
Game



- .game Options:
  - New->sprite: not implemented
  - New->Level: create a level
  - Rename: Change name
- Every game is intended to have at least one level.
- All game objects are inserted inside a level, except for the camera.

Game Explorer

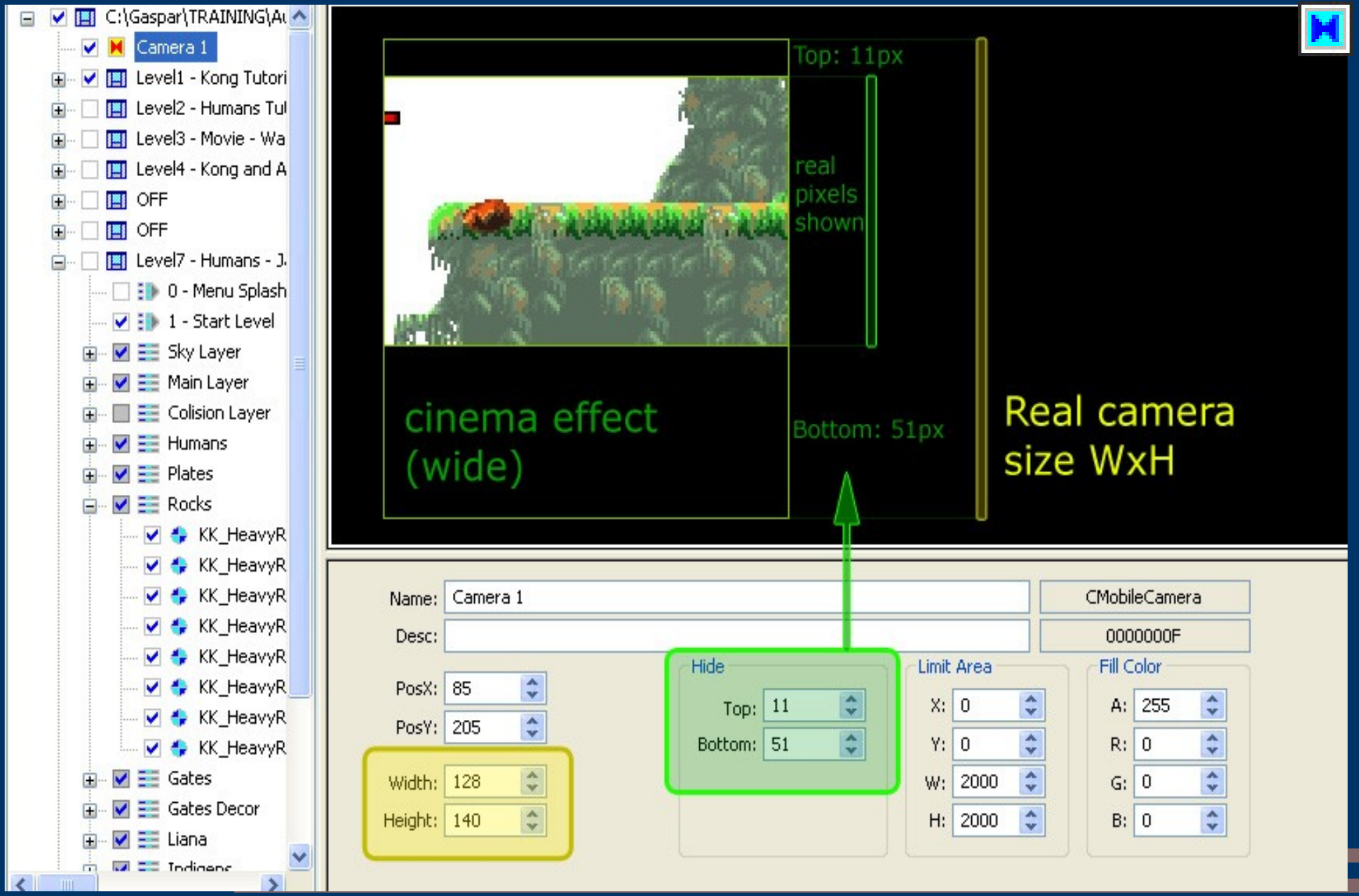
Camera Properties



- PosX, PosY: Initial camera positions
- Width, Height: Size of camera (it's usually the screen size)
- Hide: Will reduce camera's area to create a wide screen effect. King Kong uses this feature to simulate a movie in cinematics.
- Limit Area: Is our game limits, the camera will move inside this limits
- Fill Color: Areas not shown by the camera will use this color in Aurora to give an idea of what will you see on the phone's screen (ARGB)

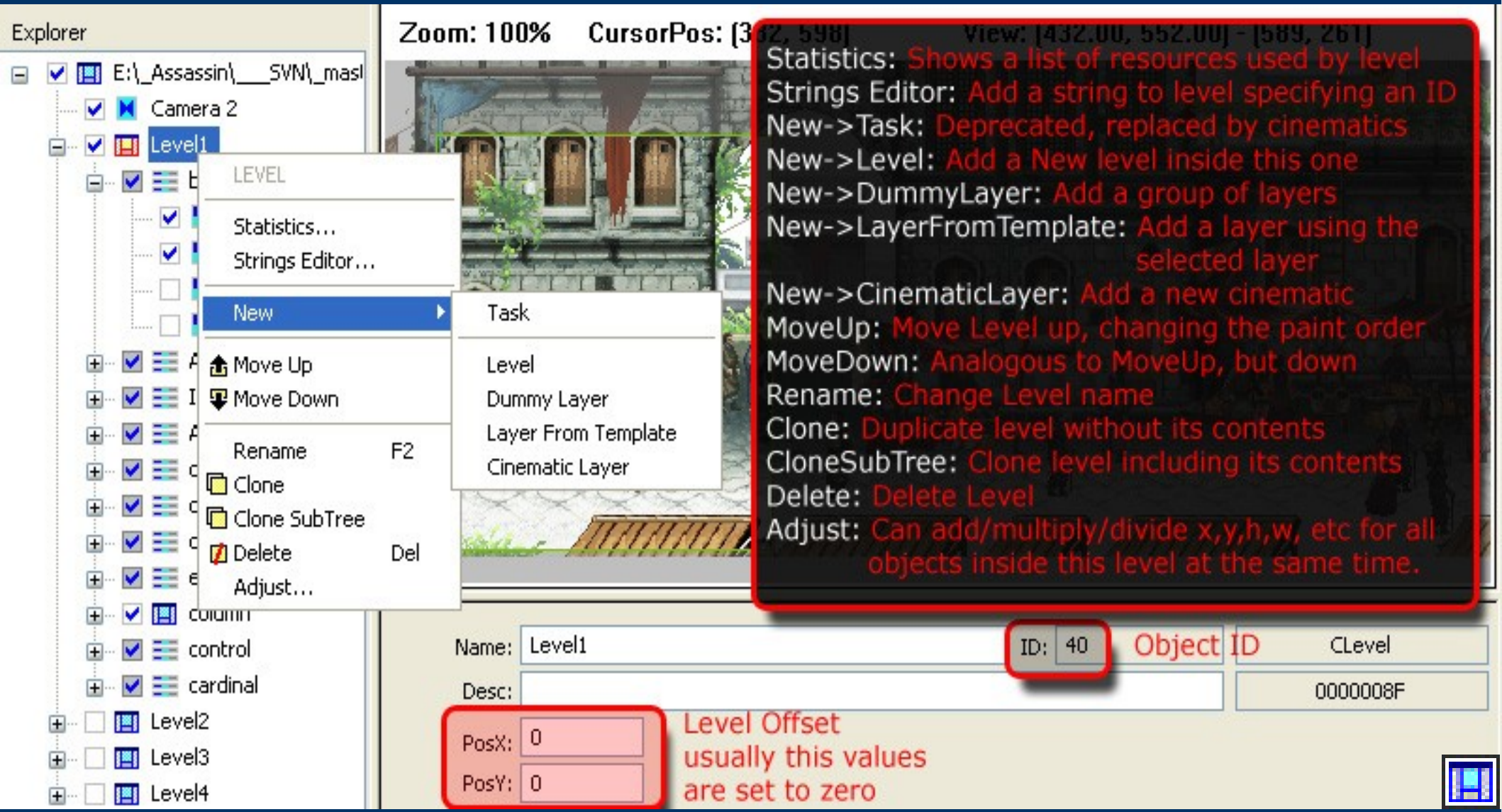
Game Explorer

Camera Properties



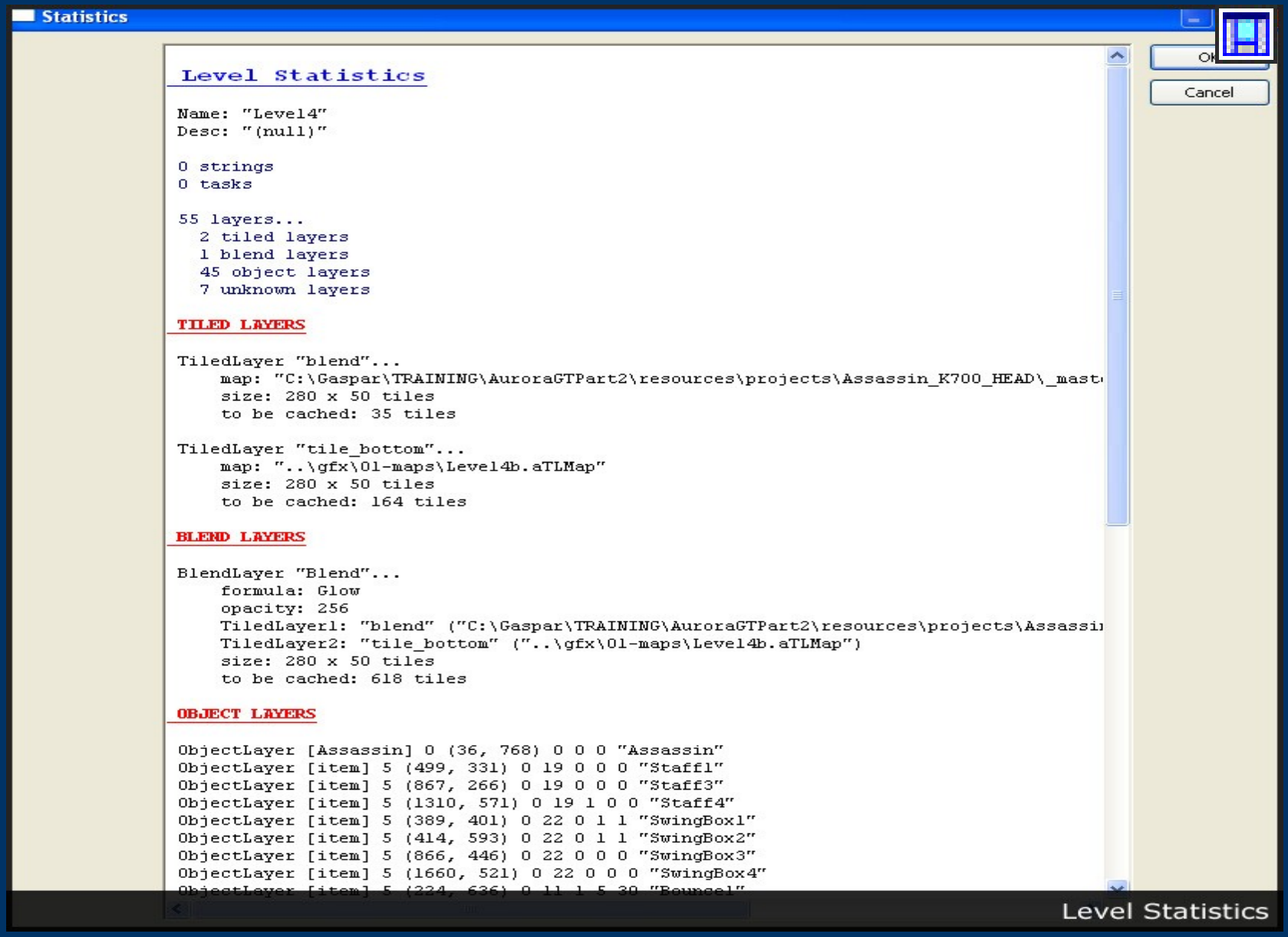
Game Explorer

Level Menu



Game Explorer

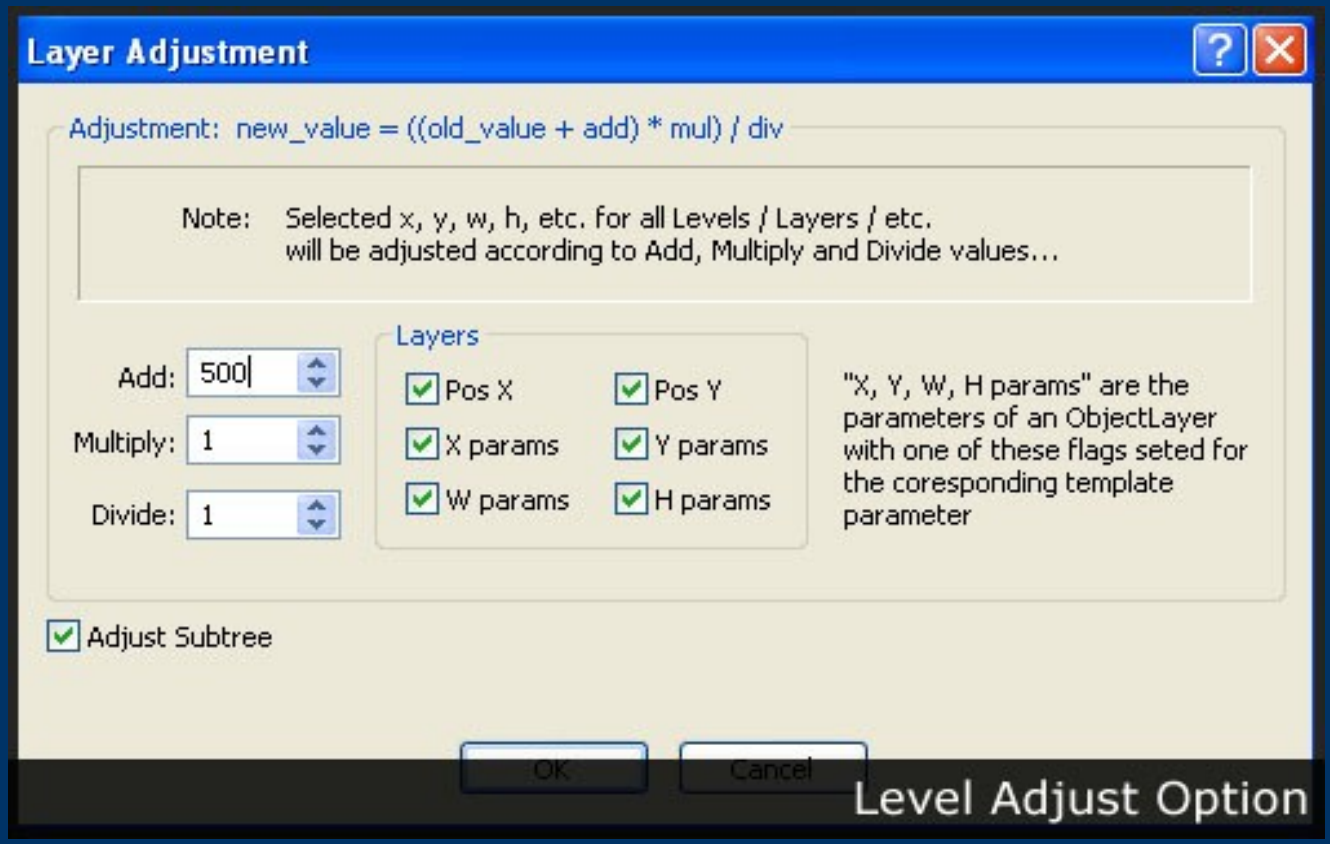
Level Statistics



Game Explorer

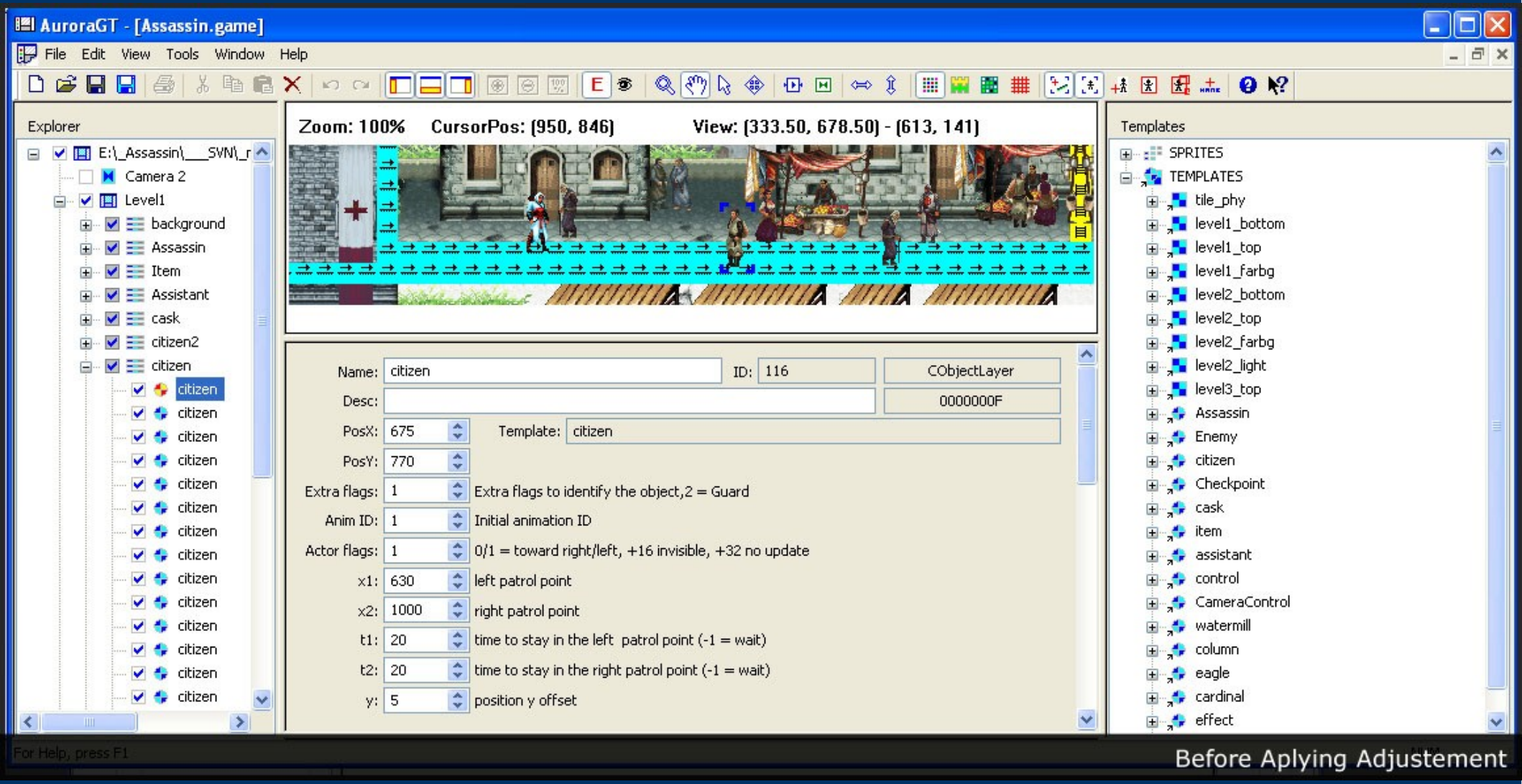
Level Adjustment

- Layer adjustment is used mainly for level scaling
- Layer Adjustment: Let's add 500 to every variable and see it's consequences.



Game Explorer

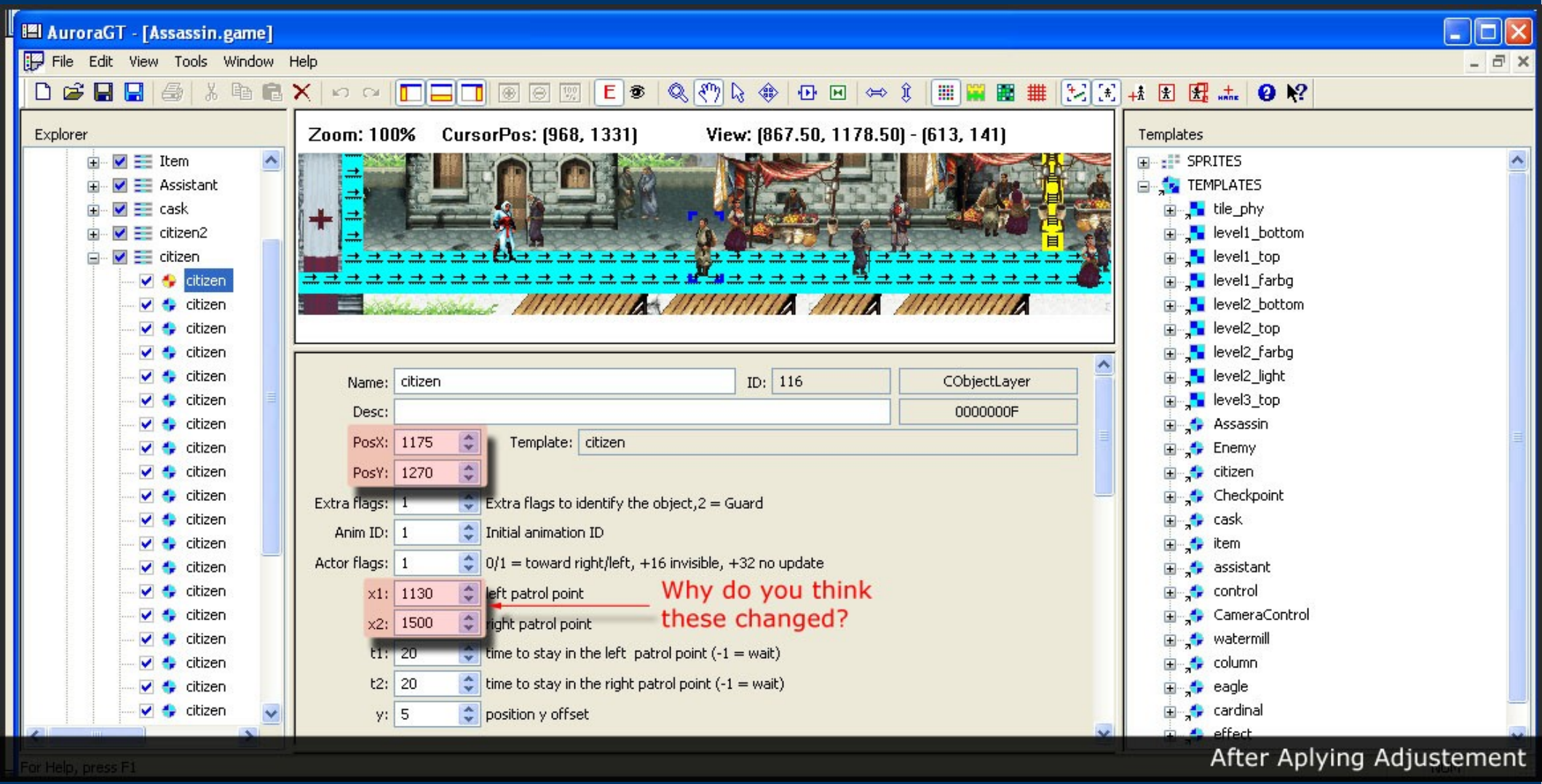
Level



Before Aplying Adjustement

Game Explorer

Level



Game Explorer

Level

.gts

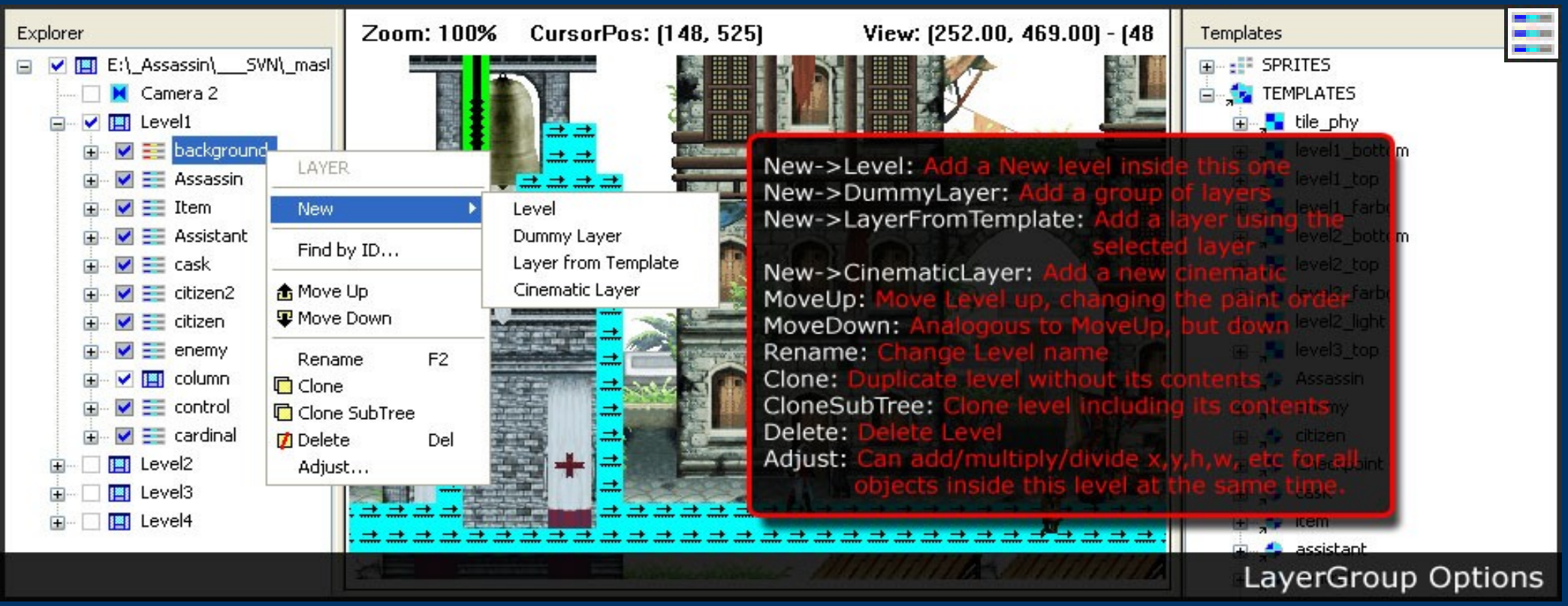
```
TEMPLATE OBJECT_LAYER "citizen"
{
    ID 2
    SPRITE "..\gfx\00-sprites\citizen.sprite"

    SET SPRITE 0
    SET MODULE -1
    SET FRAME -1
    SET AFRAME 0
    SET ANIM PARAM[1]
    SET FLAGS PARAM[2] 0x0001

    PARAMS
    {
        1 "Extra flags" "Extra flags to identify the object,2 = Guard"
        0 "Anim ID" "Initial animation ID"
        0 "Actor flags" "0/1 = toward right/left, +16 invisible, +32 no update"
        x-10 "x1"     "left patrol point" FLAGS { X }
        x+10 "x2"     "right patrol point" FLAGS { X }
        20   "t1"     "time to stay in the left  patrol point (-1 = wait)"
        20   "t2"     "time to stay in the right patrol point (-1 = wait)"
        0    "y"      "position y offset"
    }
}
```

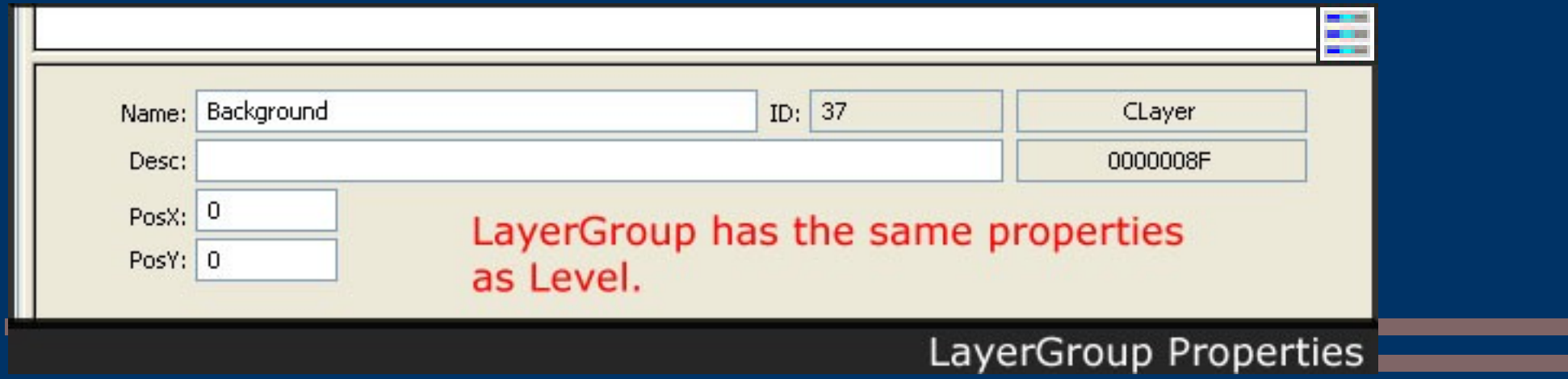
Two parameters modify the X coordinate of the object, so making and adjust to the values changes also the default value of this param.

Template File (.gts)

Game Explorer

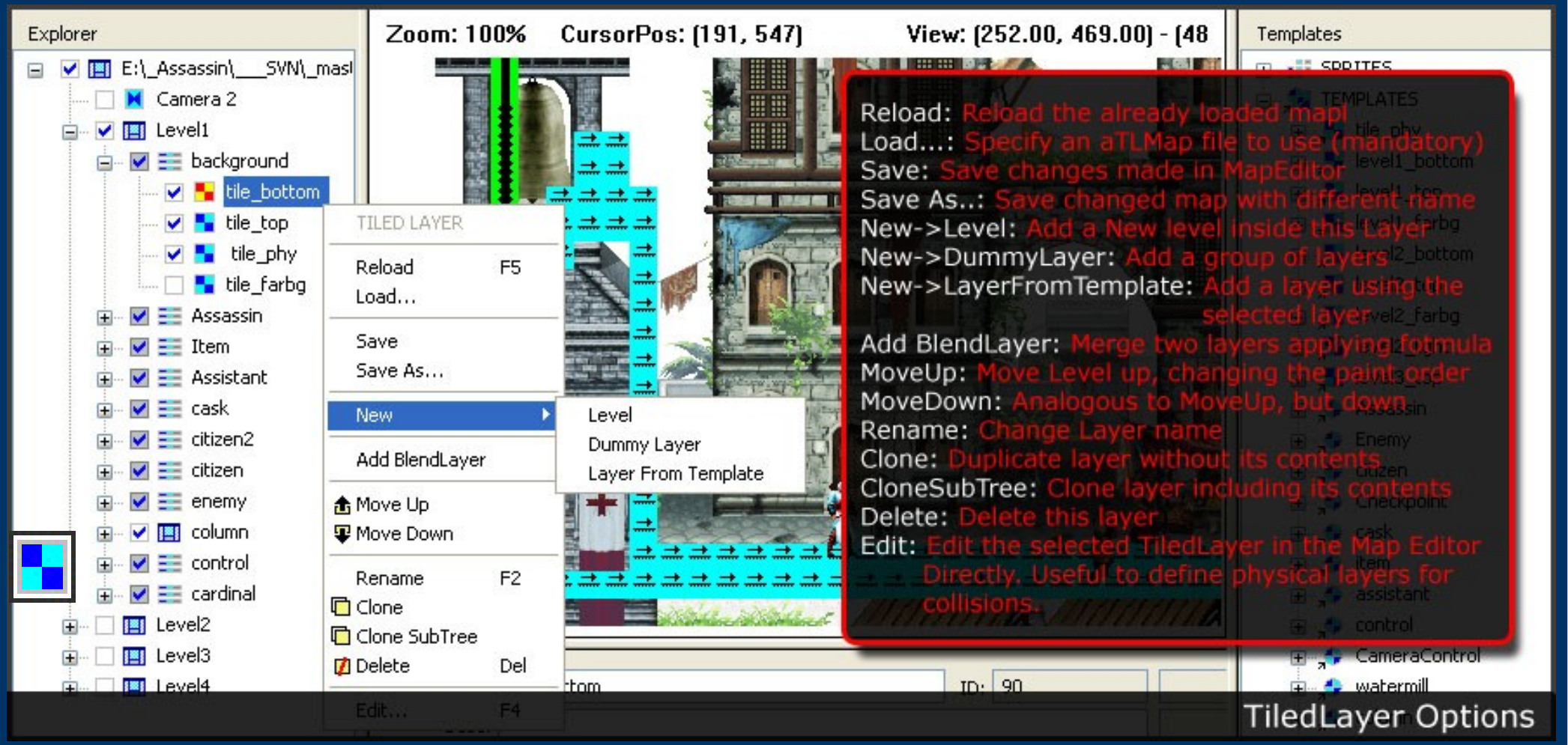
Layer Group (Dummy Layer)





Game Explorer

Tiled Layer Menu

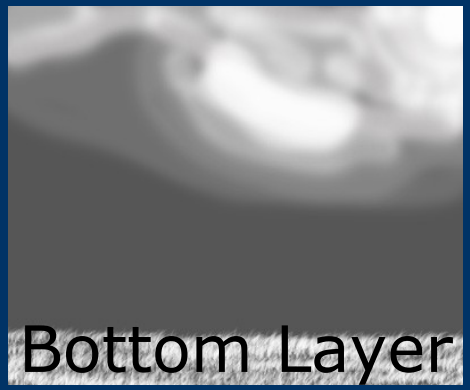


Game Explorer

Blend Layer

- (to) blend is to mix smoothly and inseparably together.
- Some blend Modes:

+

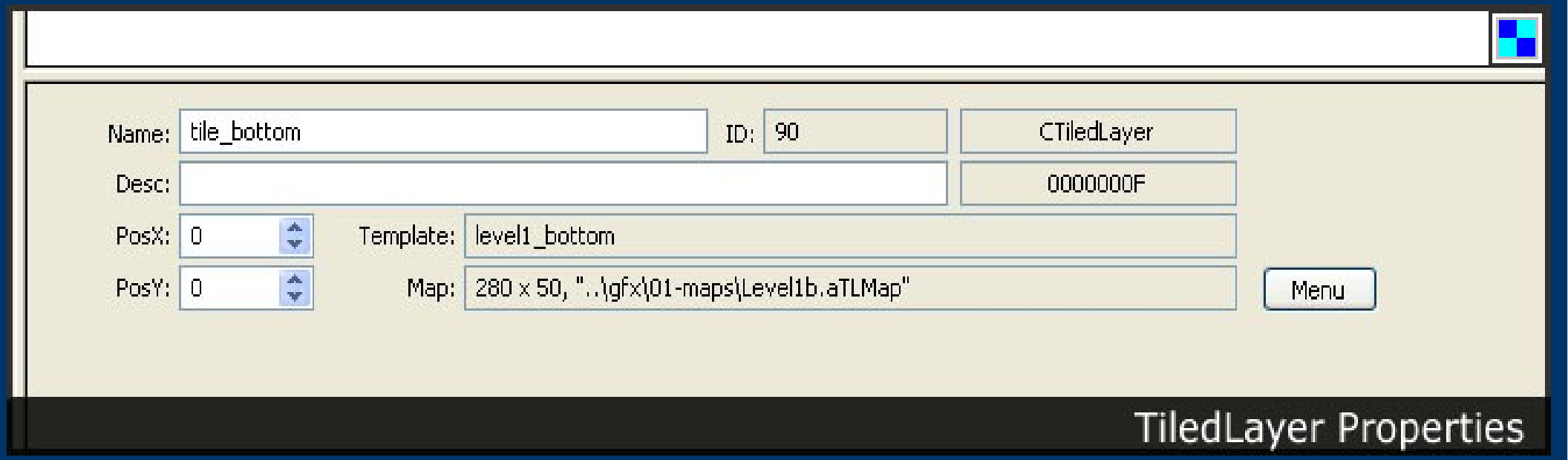


- See video: Adding a Blending Layer

Game Explorer

Tiled Layer Properties

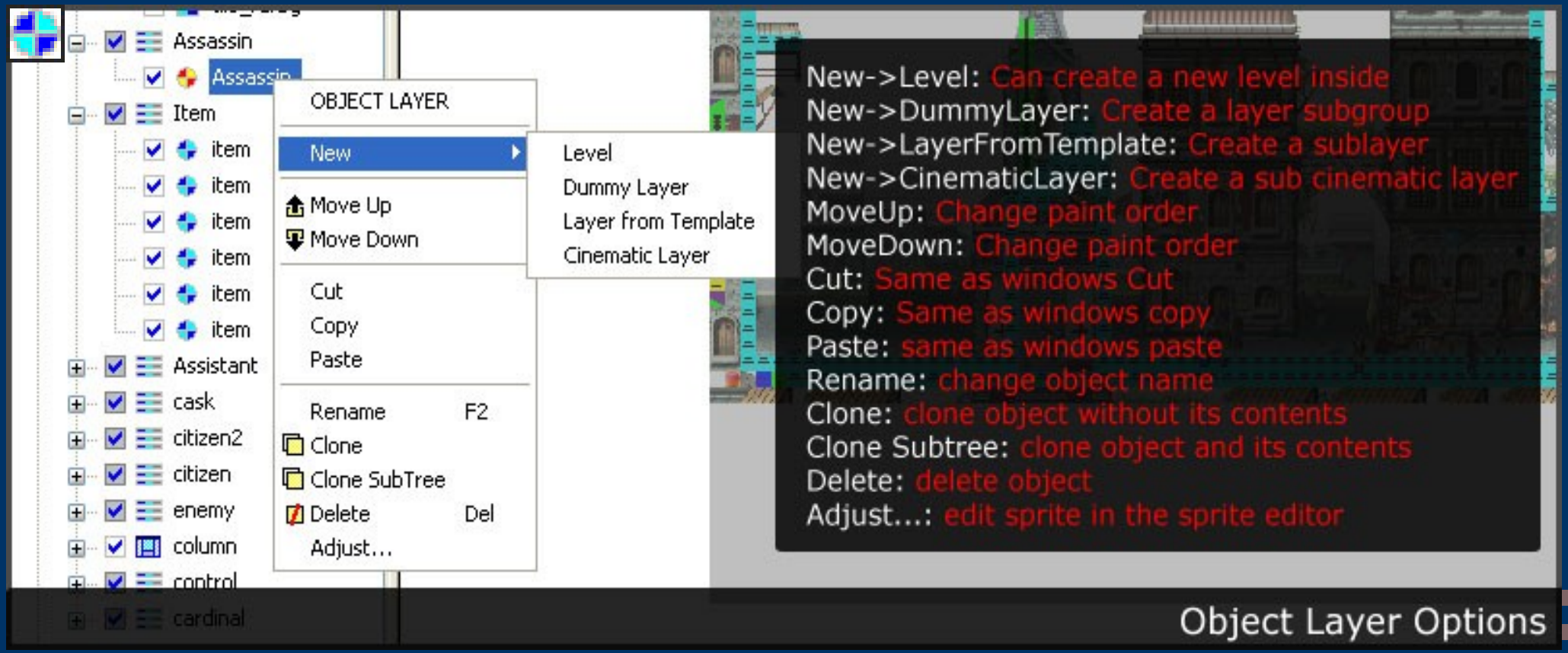
- Same fields as LayerGroup
- Extra field: Map, that indicates the map file and map size loaded for this layer. Every tiled layer needs an already loaded map to work.



Game Explorer

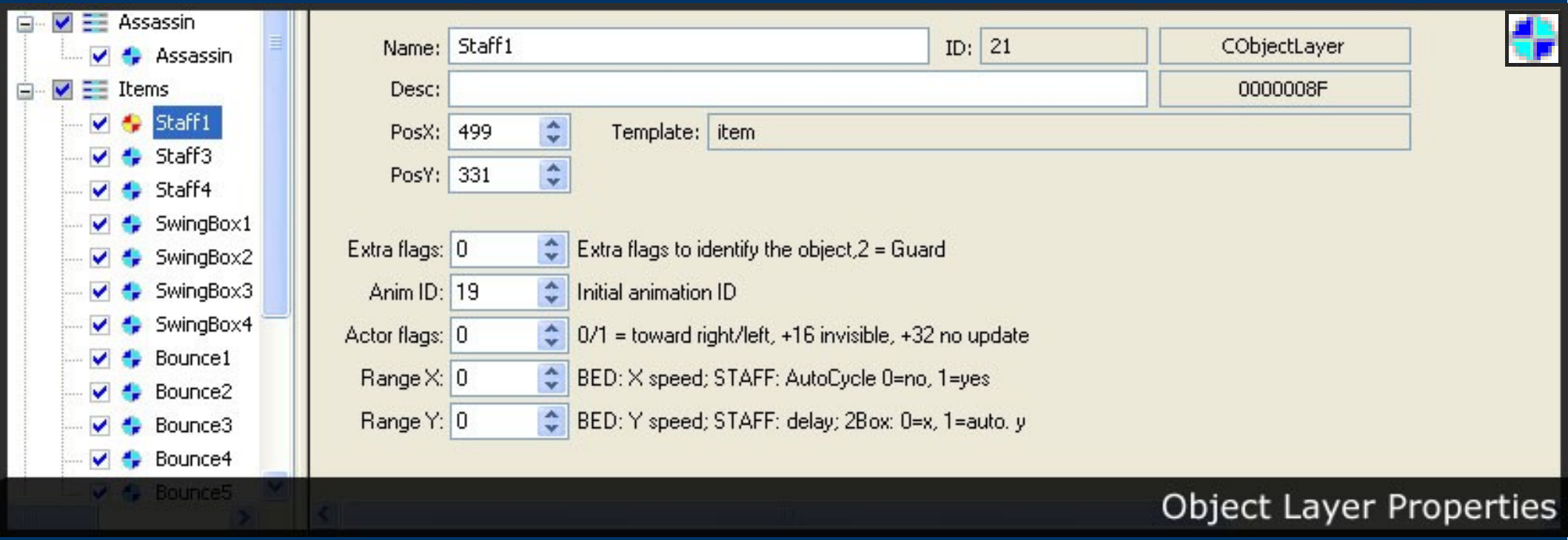
Object Layer Menu

- Many of the options are the same as in level and tiled layer.
- Usually Object Layers work as tree leaves and there aren't other items inside it.



Game Explorer

Object Layer Properties




```
PARAMS
{
    0 "Extra flags" "Extra flags to identify the object,2 = Guard"
    0 "Anim ID" "Initial animation ID"
    0 "Actor flags" "0/1 = toward right/left, +16 invisible, +32 no update"
    0 "Range X" "BED: X speed; STAFF: AutoCycle 0=no, 1=yes"
    0 "Range Y" "BED: Y speed; STAFF: delay; 2Box: 0=x, 1=auto. y"
}
```

.gts file: “item” Object Template

AuroraGT

Conclusion

- Aurora as a Game editor.
- Basic Aurora interface.
- The gts file and its usage.
- A general overview.
- Initial options for creating games from scratch

AuroraGT

Bibliography

- **AuroraGT official repository**
  https://terminus.mdc.gameloft.org/vc/tools/AuroraGT
- **AuroraGT main wiki**
  https://wiki.gameloft.org/twiki/bin/view/Main/AuroraGT

AuroraGT

Contact us

- Please, we look forward for any suggestions or bug found:
  - send us a mail to World-AuroraSuggestions@gameloft.com