Chowdren

Chowdren converts the EXE build of an MMF2 application to C++.
This means that Chowdren is divided into two parts:

- **Converter** written in **Python**
- **Runtime** written in **C++**

The idea is that you never change the generated C++ code, but always make any required changes in the MFA, and then use the converter again.

To port a complete game using Chowdren, you will have to port any missing

- **Extension modules**
- **Movements**
- **Events**
- **Objects**
- **Shaders**

This includes changing both the converter and runtime to handle these.
Usually, the converter only needs a minimal amount data to “understand” new features.
However, the runtime will need the implementation itself in C++/GLSL, which can be large in scope depending on the feature/extension.

Chowdren has a lot of these already implemented, either partially or completely.

The list of native objects is

- **Active**
- **String**
- **Backdrop**
- **Quick Backdrop**
- **Counter**
- **RTF**
- **Lives**
- **Sub Application**

The list of extensions is

- **Advanced Direction object**
- **Alpha Channel object**
- **Associate Array**
- **Phizix object (Box2D)**
- **Background Images object**
- **Binary**
- **Binary Array object**
- **Capture**
- **CharImage**
- **Clickteam Movement Controller object**
- **Colorizer**
- **Console object**
- **Control X**
- **Dialog Box**
- **Easing**
- **MMF2 Params object**
- **ForEach**
- **Global Store X**
- **Image Manipulator**
- **INI++**
- **Joystick 2 object**
- **Array object**
- **Active System Box**
- **Background System Box**
- **Date and Time**
- **Direction Calculator**
- **File object**
- **Fast Loop object**
- **Window Focus**
- **INI object**
- **List object**
- **Active Picture**
- **Picture**
- **Vitalize! object**
- **Popup Object**
- **Window Control**
- **Key object**
- **Keyboard object**
- **Layer object**
- **Masked Edit**
- **String Parser**
- **Perspective object**
- **Platform object**
- **Python object**
- **Sound Player**
- **Steam object**
- **Steamworks object**
- **Stochastic Utility**
- **String Replace**
- **String Tokenizer**
- **Surface object**
- **Text Blitter**
- **Ultimate Fullscreen**
- **Value Add**
- **Viewport**
- **XBOX Gamepad**

The list of movements is

- **Ball**
- **Path**
- **Eight Directions**
- **Pinball**
- **Static**

The list of native actions (excluding the ones from extensions) is

- **Create Object**
- **Shoot**
- **Start Loop**
- **Stop Loop**
- **Set X**
- **Set Y**
- **Set Alterable Value**
- **Add To Alterable**
- **Spread Value**
- **Subtract From Alterable**
- **Set Alterable String**
- **Add Counter Value**
- **Subtract Counter Value**
- **Set Counter Value**
- **Set Maximum Value**
- **Set Minimum Value**
- **Set Global String**
- **Set Global Value**
- **Add Global Value**
- **Subtract Global Value**
- **Set String**
- **Set Bold**
- **Hide**
- **Show**
- **Set Paragraph**
- **Lock Channel**
- **Stop Channel**
- **Resume Channel**
- **Pause Channel**
- **Set Channel Position**
- **Set Channel Pan**
- **Set Channel Volume**
- **Play Looping Channel File Sample**
- **Play Channel File Sample**
- **Play Channel Sample**
- **Play Looping Channel Sample**
- **Play Looping Sample**
- **Play Sample**
- **Set Channel Frequency**
- **Set Direction**
- **Set R G B Coefficient**
- **Set Angle**
- **Deactivate Group**
- **Activate Group**
- **Center Display X**
- **Center Display Y**
- **Center Display**
- **End Application**
- **Restart Application**
- **Look At**
- **Set Position**
- **Execute Evaluated Program**
- **Hide Cursor**
- **Show Cursor**
- **Fullscreen Mode**
- **Next Frame**
- **Previous Frame**
- **Move To Layer**
- **Jump To Frame**
- **Restart Frame**
- **Set Alpha Coefficient**
- **Set Semi Transparency**
- **Set X Scale**
- **Set Y Scale**
- **Set Scale**
- **Force Animation**
- **Restore Animation**
- **Force Frame**
- **Force Speed**
- **Restore Frame**
- **Set Ink Effect**
- **Set Effect**
- **Add To Debugger**
- **Set Frame Rate**
- **Destroy**
- **Bring To Back**
- **Bring To Front**
- **Delete All Created Backdrops**
- **Delete Created Backdrops**
- **Set Effect Parameter**
- **Set Effect Image**
- **Set Frame Background Color**
- **Add Backdrop**
- **Paste Active**
- **Move In Front**
- **Move Behind**
- **Force Direction**
- **Restore Direction**
- **Stop Animation**
- **Start Animation**
- **Restore Speed**
- **Set Main Volume**
- **Stop All Samples**
- **Pause All Sounds**
- **Resume All Sounds**
- **Stop Sample**
- **Set Sample Pan**
- **Set Sample Position**
- **Set Sample Volume**
- **Set Sample Frequency**
- **Next Paragraph**
- **Pause Application**
- **Set Random Seed**
- **Set Timer**
- **Set Loop Index**
- **Ignore Controls**
- **Restore Controls**
- **Change Control Type**
- **Flash During**
- **Set Maximum Speed**
- **Set Speed**
- **Bounce**
- **Start**
- **Stop**
- **Set Directions**
- **Go To Node**
- **Select Movement**
- **Next Movement**
- **Enable Flag**
- **Disable Flag**
- **Toggle Flag**
- **Reverse**
- **Replace Color**
- **Set Lives**
- **Set Score**
- **Subtract Lives**
- **Add Lives**
- **Enable Vsync**
- **Disable Vsync**
- **Set Gravity**
- **Load Active Frame**
- **Set Clipboard**
- **Set Frame Effect**
- **Set Frame Effect Parameter**
- **Set Frame Alpha Coefficient**
- **Pause Debugger**
- **Jump Sub Application Frame**
- **Set Text Color**
- **Set Frame Height**

The list of native conditions (excluding the ones from extensions) is

- **Compare Alterable Value**
- **Compare Alterable String**
- **Compare Global Value**
- **Compare Global String**
- **Compare Counter**
- **Compare X**
- **Compare Y**
- **Compare**
- **Is Overlapping**
- **On Collision**
- **Object Visible**
- **Object Invisible**
- **While Mouse Pressed**
- **Mouse On Object**
- **Always**
- **Mouse Clicked**
- **Object Clicked**
- **Player Key Down**
- **Player Key Pressed**
- **Key Down**
- **Key Pressed**
- **On Group Activation**
- **Facing In Direction**
- **Animation Playing**
- **Chance**
- **Compare Fixed Value**
- **Inside Playfield**
- **Outside Playfield**
- **Is Obstacle**
- **Is Overlapping Background**
- **On Background Collision**
- **Pick Random**
- **Objects In Zone**
- **Pick Objects In Zone**
- **Pick Alterable Value**
- **Number Of Objects**
- **Group Activated**
- **Not Always**
- **Animation Frame**
- **Channel Not Playing**
- **Sample Not Playing**
- **Once**
- **Every**
- **Timer Equals**
- **Timer Greater**
- **Timer Less**
- **Is Bold**
- **Is Italic**
- **Movement Stopped**
- **Path Finished**
- **Node Reached**
- **Compare Speed**
- **Flag On**
- **Flag Off**
- **Near Window Border**
- **Animation Finished**
- **Start Of Frame**
- **Never**
- **Number Of Lives**
- **Any Key Pressed**
- **Repeat**
- **Restrict For**
- **Sub Application Finished**
- **Leaving Playfield**
- **On Loop**

The list of native expressions (excluding the ones from extensions) is

- **Speed**
- **String**
- **To Number**
- **To Int**
- **Abs**
- **To String**
- **Get RGB**
- **Long**
- **Double**
- **End Parenthesis**
- **Plus**
- **Multiply**
- **Divide**
- **Minus**
- **Virgule**
- **Parenthesis**
- **Modulus**
- **AND**
- **OR**
- **XOR**
- **Random**
- **Application Path**
- **Alterable Value**
- **Alterable Value Index**
- **Alterable String Index**
- **Alterable String**
- **Global String**
- **Global Value**
- **Global Value Expression**
- **Y Position**
- **X Position**
- **Action X**
- **Action Y**
- **Get Paragraph**
- **Paragraph Count**
- **Current Paragraph Index**
- **Loop Index**
- **Current Text**
- **X Mouse**
- **Y Mouse**
- **Min**
- **Max**
- **Sin**
- **Cos**
- **Exp**
- **Log**
- **Get Angle**
- **Frame Height**
- **Frame Width**
- **String Length**
- **Find**
- **Reverse Find**
- **Lower String**
- **Upper String**
- **Right String**
- **Mid String**
- **Left String**
- **Fixed Value**
- **Animation Frame**
- **Object Left**
- **Object Right**
- **Object Top**
- **Object Bottom**
- **Get Direction**
- **Get X Scale**
- **Get Y Scale**
- **Power**
- **Square Root**
- **Asin**
- **Atan2**
- **Atan**
- **Alpha Coefficient**
- **Semi Transparency**
- **Effect Parameter**
- **Floor**
- **Round**
- **Animation Number**
- **Ceil**
- **Get Main Volume**
- **Get Channel Position**
- **Get Sample Position**
- **Get Sample Duration**
- **Get Channel Volume**
- **Get Channel Duration**
- **Get Channel Frequency**
- **Object Layer**
- **New Line**
- **X Left Frame**
- **X Right Frame**
- **Y Bottom Frame**
- **Y Top Frame**
- **Object Count**
- **Counter Maximum Value**
- **Application Directory**
- **Application Drive**
- **Timer Value**
- **Timer Hundreds**
- **Counter Value**
- **Current Frame**
- **Get Flag**
- **Get Command Item**
- **Display Mode**
- **Get Clipboard**
- **Total Object Count**
- **Frame Rate**
- **Temporary Path**
- **Get Collision Mask**
- **Font Color**
- **RGB Coefficient**
- **Movement Number**
- **Frame Background Color**

Finally, the list of shaders is

- **Subtract**
- **Add**
- **Color Mixer**
- **Looki Offset**
- **Hue**
- **Dodge Blur**
- **Monochrome**
- **Blend**
- **Subtract**
- **Hard Mix**
- **Overlay**
- **Lens**
- **Linear Dodge**
- **Soft Light**
- **Pin Light**
- **Invert**
- **Grain PS2**
- **Multiply**
- **Hard Light**
- **Tint**
- **Channel Blur**
- **Bg Bloom**
- **Under Water**
- **Rotate Sub**
- **Simple Mask**
- **Offsetstationary**
- **Pattern Overlay**
- **Sub Px**
- **Col Dir Blur**
- **Overlay Alpha**
- **Gradient**
- **Zoom Offset**

You can review the list of implemented features and cross-check with the MFA (there is a "Data Elements" tab in MMF2 which displays some of this information).
However, this will not work for native events, movements and objects.

Converter

To set up the converter

1) Install **Python 2.7.8**
2) Install a **native compiler**
3) Check out the following repository from Git:
   https://github.com/matpow2/anaconda
4) Build all Python modules using **build_all.bat**.

Finally, to run the converter on the EXE, run the following in a console in the “Chowdren” subdirectory:

```
C:\Python27\python -m chowdren.run MyGame.exe mygamesrc
```

This will give a lot of console output, with some statistics at the end like the following:

```
unimplemented generated groups in 'TITLE screen': [('SystemBox*', 3), 'PlayerKeyPressed', 'Never', (
'Undefined*', 2)]

stats:
ACTIONS
[('surface_transform_resize_13', 2), ('surface_transform_resize_canvas_78', 1), ('ctrlx_player_set_d
own_8', 1), ('surface_input_output_load_from_file_15', 1), ('ctrlx_enable_disable_disable_alt_tab_ct
rl_esc_and_ctrl_alt_del_15', 1), ('valueadd_set_value_0', 1), ('ctrlx_player_set_up_7', 1), ('ctrlx_
player_set_right_10', 1), ('ctrlx_simulate_keyboard_simulate_key_value_down_33', 1), ('ctrlx_player_
set_fire_1_11', 1), ('ctrlx_player_set_fire_4_39', 1), ('ctrlx_player_set_fire_2_12', 1), ('kcwctrl_
17', 1), ('ctrlx_player_set_left_9', 1), ('ctrlx_player_set_fire_3_38', 1)]

CONDITIONS
[('kcfile_file_is_readable_2', 5), ('PlayerKeyPressed', 2)]

EXPRESSIONS
[('stringreplace_automatic_replace_1', 14), ('txtblt_alignment_get_horizontal_alignment_9', 4), ('in
i_extra_functions_hash_string_string_26', 2), ('kcwctrl_0', 2), ('kcwctrl_1', 2), ('DisplayMode', 1)
]
```

You can use this to get an overview over what events and extensions have yet to be ported into Chowdren.

To give you an idea of what the converter outputs, here is an excerpt of some generated code:

```
void Frames::event_func_2099()
{
    // event 901_6
    {
        beholder4_372_instances.clear_selection();
        for (ObjectIterator it(beholder4_372_instances); !it.end(); ++it) {
            if (!((((Active*)(*it))->alterables->values.get(2)) <= (0))) it.deselect();
        }
        if (!beholder4_372_instances.has_selection()) goto event_901_6_end;
        if (!pick_random(beholder4_372_instances)) goto event_901_6_end;
        if (!((global_values->get(13)) == (0))) goto event_901_6_end;
        for (ObjectIterator it(beholder4_372_instances); !it.end(); ++it) {
            ((Active*)*it)->alterables->values.set(3, 0);
        }
        for (ObjectIterator it(beholder4_372_instances); !it.end(); ++it) {
            ((Active*)*it)->alterables->values.set(0, 0);
        }
        playermask_77_instances.clear_selection();
        for (ObjectIterator it(playermask_77_instances); !it.end(); ++it) {
            ((Active*)*it)->alterables->values.add(7, 1);
        }
        for (ObjectIterator it(beholder4_372_instances); !it.end(); ++it) {
            ((Active*)*it)->destroy();
        }
        for (ObjectIterator it(playermask_77_instances); !it.end(); ++it) {
            ((Active*)*it)->alterables->values.set(8, 20);
        }
        media->stop_sample("zapper 4");
        particlegenerator_83_instances.clear_selection();
        for (ObjectIterator it(particlegenerator_83_instances); !it.end(); ++it) {
            FrameObject * parent = ((Active*)get_single(beholder4_372_instances, it.current_index));
            if (parent == NULL) goto pos_end_901_6_0;
            (*it)->set_global_position(parent->get_x() + 0, parent->get_y() + 0);
            pos_end_901_6_0: ;
        }
        ((Active*)get_single(particlegenerator_83_instances))->alterables->values.set(0, 3);
        media->play_name("Explosion80", -1, 1);
    }
    event_901_6_end: ;
}
```

That is, all the events (which are normally interpreted in MMF2) are converted directly to C++. Even though the output is not very readable, the result is very efficient.

The important code related to the converter can be found here:

Extension converters
https://github.com/matpow2/anaconda/tree/master/Chowdren/chowdren/writers/extensions

Native event converters
https://github.com/matpow2/anaconda/blob/master/Chowdren/chowdren/writers/events/system.py

Native object converters
https://github.com/matpow2/anaconda/blob/master/Chowdren/chowdren/writers/objects/system.py

Converter core
https://github.com/matpow2/anaconda/blob/master/Chowdren/chowdren/converter.py

Runtime

The runtime is the base code which is compiled together with the converted C++ events, frame and object data.

On desktop platforms, it uses

- OpenGL 2.1
- GLSL 1.20
- OpenAL Soft
- SDL2

Currently, the status of each platform backend is

- Windows, Mac, Linux (complete)
- Wii U (complete)
- PS4 (complete)
- PS Vita (almost complete, audio remaining)
- 3DS (almost complete, audio and some shaders remaining)

The remaining work on the 3DS and PS Vita is expected to be complete in 2~3 months (and in that order).

The important code related to the runtime can be found here:

Extension runtime
https://github.com/matpow2/anaconda/tree/master/Chowdren/base/objects

Movement runtime
https://github.com/matpow2/anaconda/blob/master/Chowdren/base/movement.h

Native object & events runtime
https://github.com/matpow2/anaconda/blob/master/Chowdren/base/common.cpp