INDEPENDENT GAMES SUMMIT

Automating Data Implementation With IDs

Keir Miron
Programmer on Darkest Dungeon
@KeirMiron
business@keirmiron.com

Contents

Intro

- Darkest Dungeon
  - Turn-based RPG, Procedural
  - Windows/Mac/Linux/Ps4/Vita
  - 2-person programming team
  - C++ 11 custom engine and middleware

Syntax

- A **Class** object
  - Contains data that never changes
  - **Jester Hero Class**
    - All Jesters have a max hp of **35**
    - All Jesters can have a skill **"inspiring_tune"**
    - The Jester skill **"battle_ballad"** is ranged

Syntax

- String Format

  The code to print "string_named_test"

  - string.format("string_named_%s", "test")
  - print(string)

  will be shortened to

  - string_named_**[name]**
  - string_named_**test**

Generating Using IDs

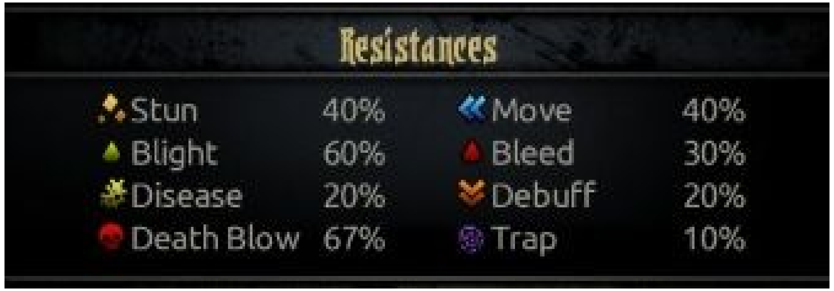
- When an ID is missing:
  - “FMOD: Could not find ID for event '**[audio_event_id]**'”
- When an object with an ID is referencing a missing ID:
  - “No effect of name **[effect_id]** so not adding to skill **[skill_id]**”

Generating Using IDs

- Using string formats, you can dynamically insert IDs into file paths
- When new IDs are created the file paths are automatically generated
- Identify missing files with error messages or temporary assets

Generating Using IDs

- File paths in the same folder
  - resistance\resistance_icon_**[resistance_id]**.png
  - resistance\resistance_icon_**bleed**.png
  - resistance\resistance_icon_**move**.png


Resistances

Stun	40%	Move	40%
Blight	60%	Bleed	30%
Disease	20%	Debuff	20%
Death Blow	67%	Trap	10%

Generating Using IDs

- File paths separated into different folders based on IDs:

[item_type]\inv_**[item_type]**+**[item_id]**.png

trinket \inv_**trinket**+**lucky_dice**.png

gem\inv_**gem**+**ruby**.png

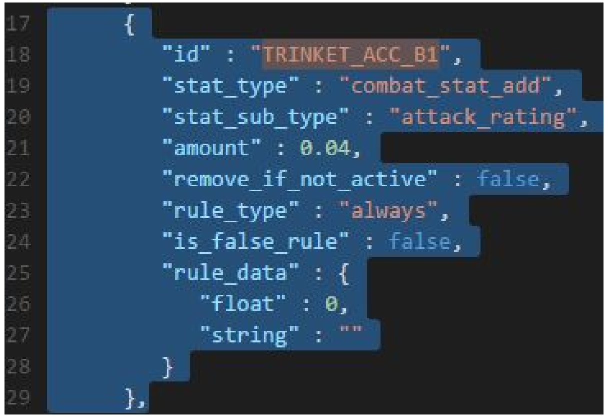
Generating Using IDs

- Multiple file paths with one ID
  - **[hero_class]\[hero_class]**.art.darkest and **[hero_class]\[hero_class]**.info.darkest
  - **leper\leper**.art.darkest and **leper\leper**.info.darkest
  - **jester\\jester**.art.darkest and **jester\\jester**.info.darkest

Reference IDs Inside of Files

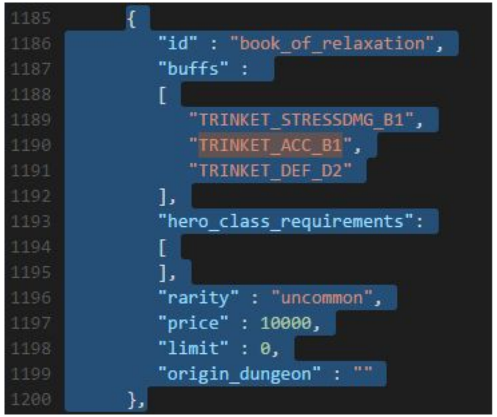
- Share logic between Classes

Buff Class

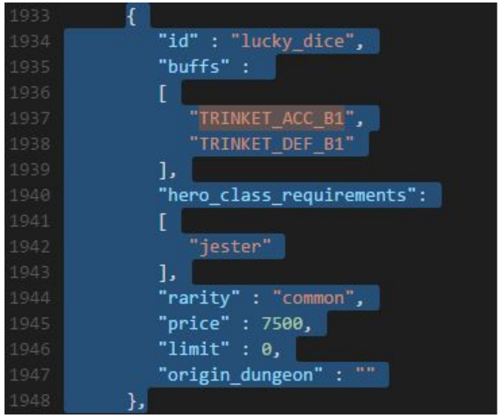


```
        {
            "id" : "TRINKET_ACC_B1",
            "stat_type" : "combat_stat_add",
            "stat_sub_type" : "attack_rating",
            "amount" : 0.04,
            "remove_if_not_active" : false,
            "rule_type" : "always",
            "is_false_rule" : false,
            "rule_data" : {
                "float" : 0,
                "string" : ""
            }
        },
```

Trinket Classes



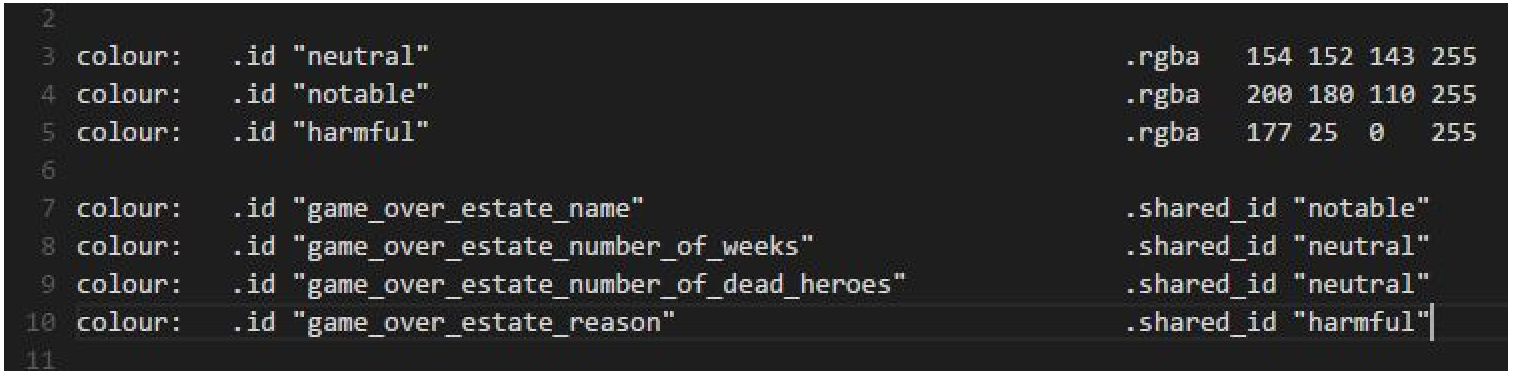
```
        {
            "id" : "book_of_relaxation",
            "buffs" :
            [
                "TRINKET_STRESSDMG_B1",
                "TRINKET_ACC_B1",
                "TRINKET_DEF_D2"
            ],
            "hero_class_requirements":
            [
            ],
            "rarity" : "uncommon",
            "price" : 10000,
            "limit" : 0,
            "origin_dungeon" : ""
        },
```



```
        {
            "id" : "lucky_dice",
            "buffs" :
            [
                "TRINKET_ACC_B1",
                "TRINKET_DEF_B1"
            ],
            "hero_class_requirements":
            [
                "jester"
            ],
            "rarity" : "common",
            "price" : 7500,
            "limit" : 0,
            "origin_dungeon" : ""
        },
```

Reference IDs Inside of Files

- Reuse data from other definitions



```

colour:   .id "neutral"                                               .rgba   154 152 143 255
colour:   .id "notable"                                               .rgba   200 180 110 255
colour:   .id "harmful"                                               .rgba   177 25  0   255

colour:   .id "game_over_estate_name"                                 .shared_id "notable"
colour:   .id "game_over_estate_number_of_weeks"                      .shared_id "neutral"
colour:   .id "game_over_estate_number_of_dead_heroes"                .shared_id "neutral"
colour:   .id "game_over_estate_reason"                               .shared_id "harmful"

```

Reference IDs Inside of Files

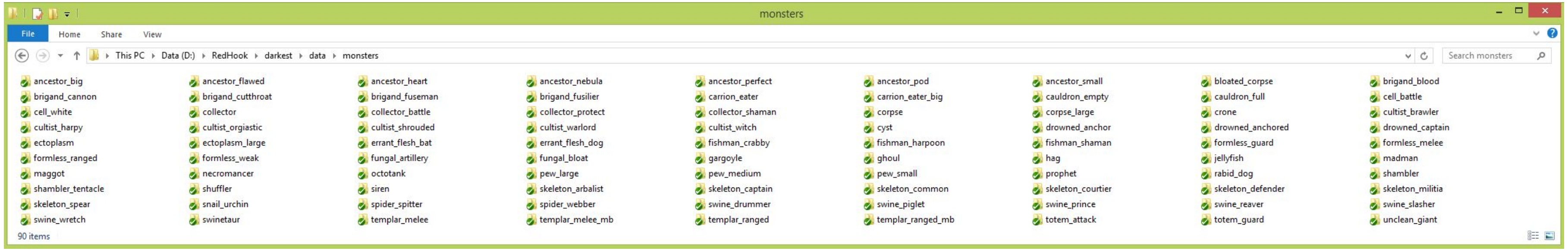
- Generate multiple types of classes
  - When parsing one Class you can generate another Class
    - Example:
    - All trinkets are items
    - For every trinket class we generate an item Class
    - Generated item Classes are of type of trinket and the same ID as the trinket class

Filesystem Hierarchy

- Filesystem hierarchy can be used to create your IDs when parsing your file system

Filesystem Hierarchy

- Example: Every folder in data/heroes is a hero class ID, and every folder in data/monsters is a monster Class ID



This PC › Data (D:) › RedHook › darkest › data › monsters

ancestor_big	ancestor_flawed	ancestor_heart	ancestor_nebula	ancestor_perfect	ancestor_pod	ancestor_small	bloated_corpse	brigand_blood
brigand_cannon	brigand_cutthroat	brigand_fuseman	brigand_fusilier	carrion_eater	carrion_eater_big	cauldron_empty	cauldron_full	cell_battle
cell_white	collector	collector_battle	collector_protect	collector_shaman	corpse	corpse_large	crone	cultist_brawler
cultist_harpy	cultist_orgiastic	cultist_shrouded	cultist_warlord	cultist_witch	cyst	drowned_anchor	drowned_anchored	drowned_captain
ectoplasm	ectoplasm_large	errant_flesh_bat	errant_flesh_dog	fishman_crabby	fishman_harpoon	fishman_shaman	formless_guard	formless_melee
formless_ranged	formless_weak	fungal_artillery	fungal_bloat	gargoyle	ghoul	hag	jellyfish	madman
maggot	necromancer	octotank	pew_large	pew_medium	pew_small	prophet	rabid_dog	shambler
shambler_tentacle	shuffler	siren	skeleton_arbalist	skeleton_captain	skeleton_common	skeleton_courtier	skeleton_defender	skeleton_militia
skeleton_spear	snail_urchin	spider_spitter	spider_webber	swine_drummer	swine_piglet	swine_prince	swine_reaver	swine_slasher
swine_wretch	swinetaur	templar_melee	templar_melee_mb	templar_ranged	templar_ranged_mb	totem_attack	totem_guard	unclean_giant

90 items

Filesystem Hierarchy

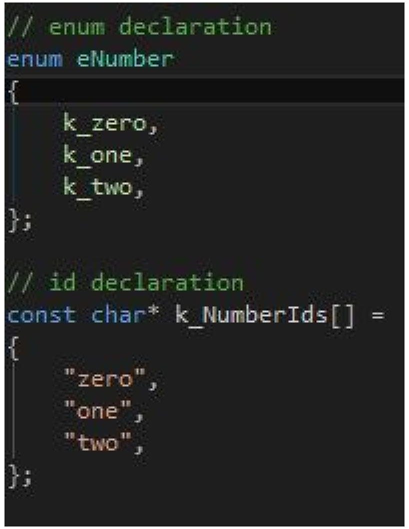
- We used regular expression-based file searches to get all files in a given folder
    - The folders:
        - data\monsters\bloated_corpse\
        - data\monsters\swinetaur\
        - data\monsters\unclean_giant\
    - Became monster IDs:
        - **bloated_corpse**, **swinetaur**, **unclean_giant**

Linking Code Data to File Data

- Uses preprocessor macros and enums
- Links the declaration of enums to IDs
- Enums can be used in code
- IDs can be used for parsing data files and generating paths

Linking Code Data to File Data

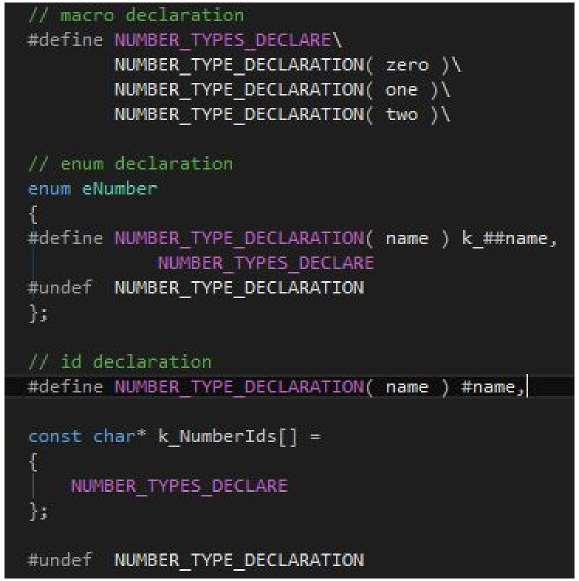
- Unlinked
  - eNumber and k_NumberIds have to be kept the same size
  - eNumber and k_NumberIds can have spelling inconsistencies
  - Spelling errors are not caught at compile time



```
// enum declaration
enum eNumber
{
    k_zero,
    k_one,
    k_two,
};

// id declaration
const char* k_NumberIds[] =
{
    "zero",
    "one",
    "two",
};
```

Linking Code Data to File Data

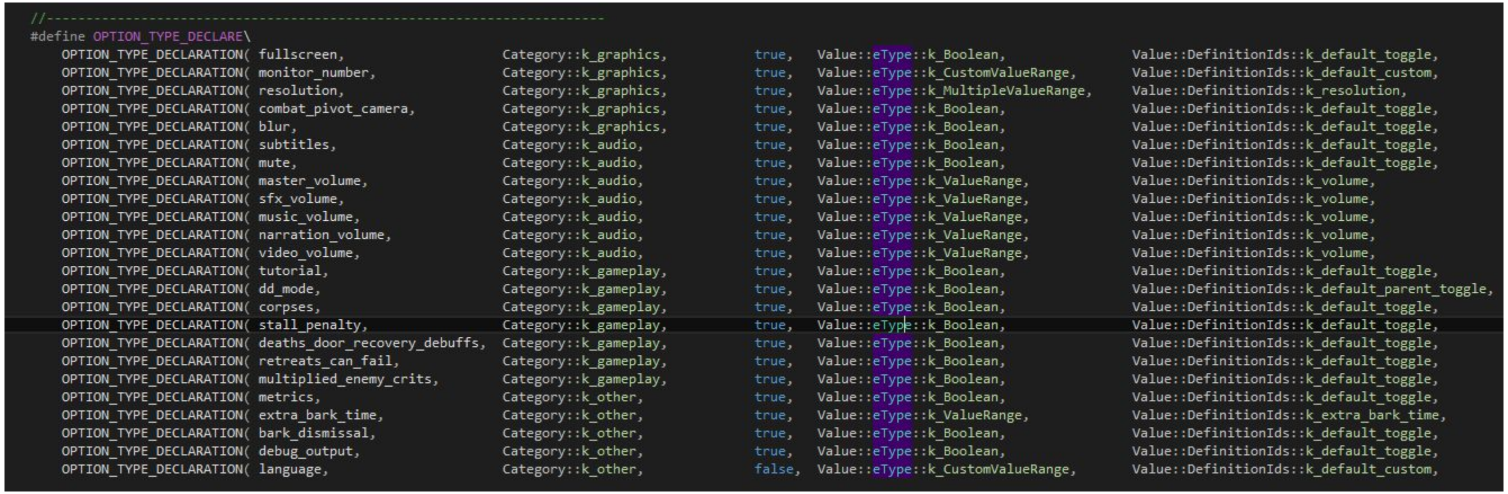
- Linked
  - When types are added to NUMBER_TYPES_DECLARE
    - new enum elements are created
    - new IDs are created
    - spelling is consistent between eNumber and k_NumberIds
    - Spelling errors are caught at compile time



```
// macro declaration
#define NUMBER_TYPES_DECLARE\
        NUMBER_TYPE_DECLARATION( zero )\
        NUMBER_TYPE_DECLARATION( one )\
        NUMBER_TYPE_DECLARATION( two )\

// enum declaration
enum eNumber
{
#define NUMBER_TYPE_DECLARATION( name ) k_##name,
            NUMBER_TYPES_DECLARE
#undef  NUMBER_TYPE_DECLARATION
};

// id declaration
#define NUMBER_TYPE_DECLARATION( name ) #name,

const char* k_NumberIds[] =
{
    NUMBER_TYPES_DECLARE
};

#undef  NUMBER_TYPE_DECLARATION
```

Linking Code Data to File Data

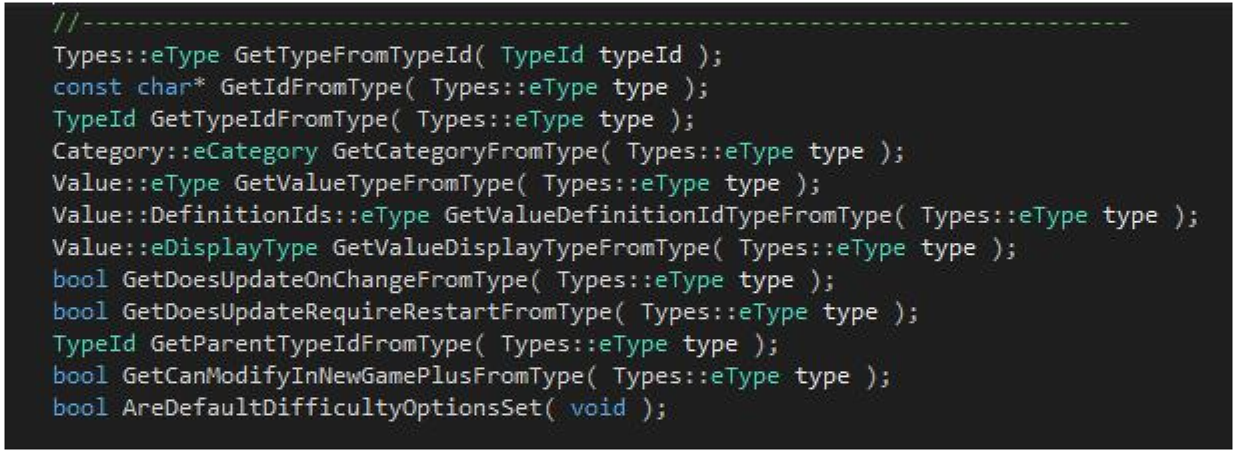
- Multiple constants can be linked to an enum



```
//-----------------------------------------------------------------------------
#define OPTION_TYPE_DECLARE\
    OPTION_TYPE_DECLARATION( fullscreen,                   Category::k_graphics,        true,   Value::eType::k_Boolean,              Value::DefinitionIds::k_default_toggle,
    OPTION_TYPE_DECLARATION( monitor_number,               Category::k_graphics,        true,   Value::eType::k_CustomValueRange,     Value::DefinitionIds::k_default_custom,
    OPTION_TYPE_DECLARATION( resolution,                   Category::k_graphics,        true,   Value::eType::k_MultipleValueRange,   Value::DefinitionIds::k_resolution,
    OPTION_TYPE_DECLARATION( combat_pivot_camera,          Category::k_graphics,        true,   Value::eType::k_Boolean,              Value::DefinitionIds::k_default_toggle,
    OPTION_TYPE_DECLARATION( blur,                         Category::k_graphics,        true,   Value::eType::k_Boolean,              Value::DefinitionIds::k_default_toggle,
    OPTION_TYPE_DECLARATION( subtitles,                    Category::k_audio,           true,   Value::eType::k_Boolean,              Value::DefinitionIds::k_default_toggle,
    OPTION_TYPE_DECLARATION( mute,                         Category::k_audio,           true,   Value::eType::k_Boolean,              Value::DefinitionIds::k_default_toggle,
    OPTION_TYPE_DECLARATION( master_volume,                Category::k_audio,           true,   Value::eType::k_ValueRange,           Value::DefinitionIds::k_volume,
    OPTION_TYPE_DECLARATION( sfx_volume,                   Category::k_audio,           true,   Value::eType::k_ValueRange,           Value::DefinitionIds::k_volume,
    OPTION_TYPE_DECLARATION( music_volume,                 Category::k_audio,           true,   Value::eType::k_ValueRange,           Value::DefinitionIds::k_volume,
    OPTION_TYPE_DECLARATION( narration_volume,             Category::k_audio,           true,   Value::eType::k_ValueRange,           Value::DefinitionIds::k_volume,
    OPTION_TYPE_DECLARATION( video_volume,                 Category::k_audio,           true,   Value::eType::k_ValueRange,           Value::DefinitionIds::k_volume,
    OPTION_TYPE_DECLARATION( tutorial,                     Category::k_gameplay,        true,   Value::eType::k_Boolean,              Value::DefinitionIds::k_default_toggle,
    OPTION_TYPE_DECLARATION( dd_mode,                      Category::k_gameplay,        true,   Value::eType::k_Boolean,              Value::DefinitionIds::k_default_parent_toggle,
    OPTION_TYPE_DECLARATION( corpses,                      Category::k_gameplay,        true,   Value::eType::k_Boolean,              Value::DefinitionIds::k_default_toggle,
    OPTION_TYPE_DECLARATION( stall_penalty,                Category::k_gameplay,        true,   Value::eType::k_Boolean,              Value::DefinitionIds::k_default_toggle,
    OPTION_TYPE_DECLARATION( deaths_door_recovery_debuffs, Category::k_gameplay,        true,   Value::eType::k_Boolean,              Value::DefinitionIds::k_default_toggle,
    OPTION_TYPE_DECLARATION( retreats_can_fail,            Category::k_gameplay,        true,   Value::eType::k_Boolean,              Value::DefinitionIds::k_default_toggle,
    OPTION_TYPE_DECLARATION( multiplied_enemy_crits,       Category::k_gameplay,        true,   Value::eType::k_Boolean,              Value::DefinitionIds::k_default_toggle,
    OPTION_TYPE_DECLARATION( metrics,                      Category::k_other,           true,   Value::eType::k_Boolean,              Value::DefinitionIds::k_default_toggle,
    OPTION_TYPE_DECLARATION( extra_bark_time,              Category::k_other,           true,   Value::eType::k_ValueRange,           Value::DefinitionIds::k_extra_bark_time,
    OPTION_TYPE_DECLARATION( bark_dismissal,               Category::k_other,           true,   Value::eType::k_Boolean,              Value::DefinitionIds::k_default_toggle,
    OPTION_TYPE_DECLARATION( debug_output,                 Category::k_other,           true,   Value::eType::k_Boolean,              Value::DefinitionIds::k_default_toggle,
    OPTION_TYPE_DECLARATION( language,                     Category::k_other,           false,  Value::eType::k_CustomValueRange,     Value::DefinitionIds::k_default_custom,
```

Linking Code Data to File Data

- Constants can be accessed by enum input functions

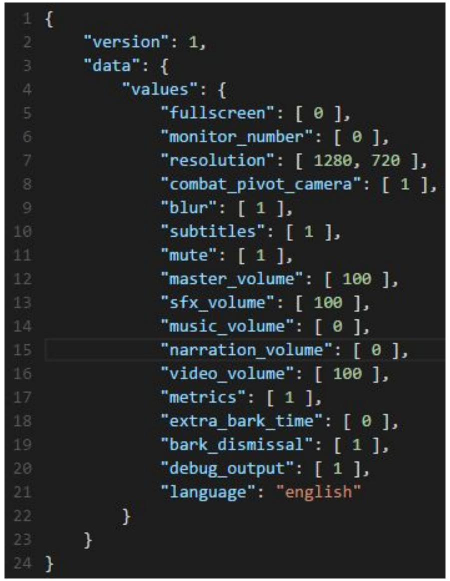


```
//--------------------------------------------------------------------------------
Types::eType GetTypeFromTypeId( TypeId typeId );
const char* GetIdFromType( Types::eType type );
TypeId GetTypeIdFromType( Types::eType type );
Category::eCategory GetCategoryFromType( Types::eType type );
Value::eType GetValueTypeFromType( Types::eType type );
Value::DefinitionIds::eType GetValueDefinitionIdTypeFromType( Types::eType type );
Value::eDisplayType GetValueDisplayTypeFromType( Types::eType type );
bool GetDoesUpdateOnChangeFromType( Types::eType type );
bool GetDoesUpdateRequireRestartFromType( Types::eType type );
TypeId GetParentTypeIdFromType( Types::eType type );
bool GetCanModifyInNewGamePlusFromType( Types::eType type );
bool AreDefaultDifficultyOptionsSet( void );
```

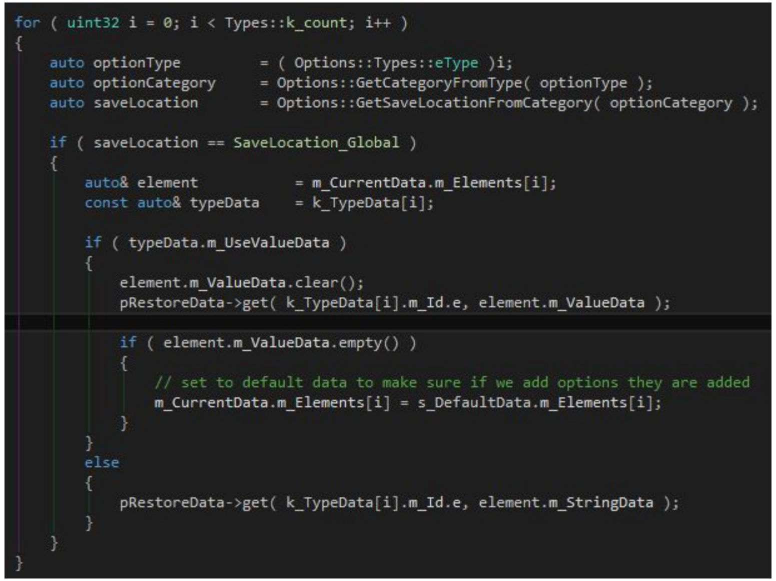
Automatically Generating Data

- Save/Load is JSON based
- Analytics data is JSON based
- Parsing consists of going through an enum and using the linked IDs as keys in JSON dictionaries

Automatically Generating Data

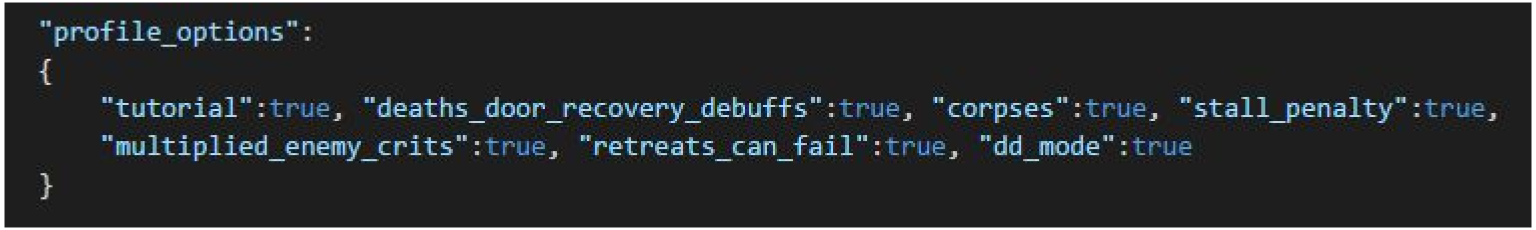


```json
{
    "version": 1,
    "data": {
        "values": {
            "fullscreen": [ 0 ],
            "monitor_number": [ 0 ],
            "resolution": [ 1280, 720 ],
            "combat_pivot_camera": [ 1 ],
            "blur": [ 1 ],
            "subtitles": [ 1 ],
            "mute": [ 1 ],
            "master_volume": [ 100 ],
            "sfx_volume": [ 100 ],
            "music_volume": [ 0 ],
            "narration_volume": [ 0 ],
            "video_volume": [ 100 ],
            "metrics": [ 1 ],
            "extra_bark_time": [ 0 ],
            "bark_dismissal": [ 1 ],
            "debug_output": [ 1 ],
            "language": "english"
        }
    }
}
```

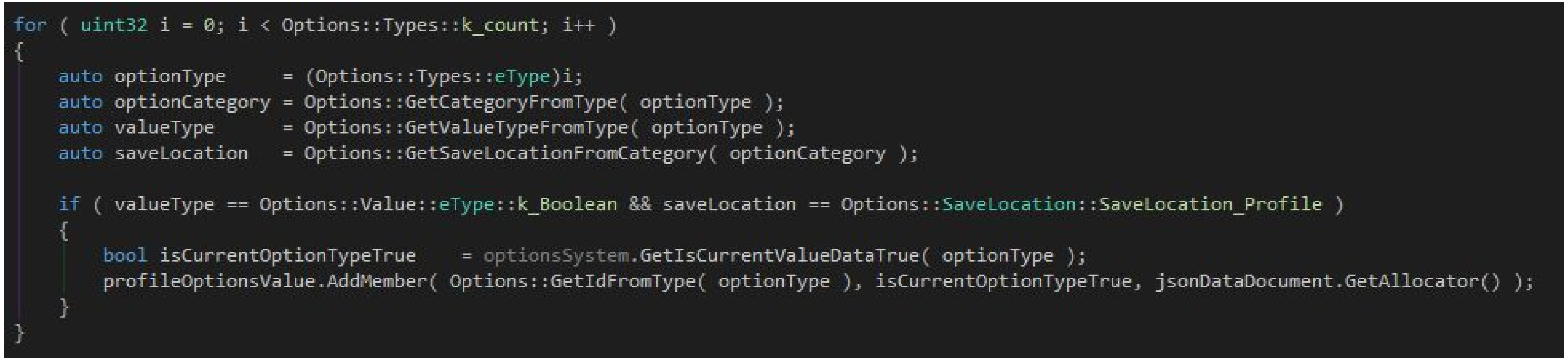


```cpp
for ( uint32 i = 0; i < Types::k_count; i++ )
{
    auto optionType         = ( Options::Types::eType )i;
    auto optionCategory     = Options::GetCategoryFromType( optionType );
    auto saveLocation       = Options::GetSaveLocationFromCategory( optionCategory );

    if ( saveLocation == SaveLocation_Global )
    {
        auto& element            = m_CurrentData.m_Elements[i];
        const auto& typeData     = k_TypeData[i];

        if ( typeData.m_UseValueData )
        {
            element.m_ValueData.clear();
            pRestoreData->get( k_TypeData[i].m_Id.e, element.m_ValueData );

            if ( element.m_ValueData.empty() )
            {
                // set to default data to make sure if we add options they are added
                m_CurrentData.m_Elements[i] = s_DefaultData.m_Elements[i];
            }
        }
        else
        {
            pRestoreData->get( k_TypeData[i].m_Id.e, element.m_StringData );
        }
    }
}
```

Automatically Generating Data



```
"profile_options":
{
    "tutorial":true, "deaths_door_recovery_debuffs":true, "corpses":true, "stall_penalty":true,
    "multiplied_enemy_crits":true, "retreats_can_fail":true, "dd_mode":true
}
```



```
for ( uint32 i = 0; i < Options::Types::k_count; i++ )
{
    auto optionType     = (Options::Types::eType)i;
    auto optionCategory = Options::GetCategoryFromType( optionType );
    auto valueType      = Options::GetValueTypeFromType( optionType );
    auto saveLocation   = Options::GetSaveLocationFromCategory( optionCategory );

    if ( valueType == Options::Value::eType::k_Boolean && saveLocation == Options::SaveLocation::SaveLocation_Profile )
    {
        bool isCurrentOptionTypeTrue    = optionsSystem.GetIsCurrentValueDataTrue( optionType );
        profileOptionsValue.AddMember( Options::GetIdFromType( optionType ), isCurrentOptionTypeTrue, jsonDataDocument.GetAllocator() );
    }
}
```

Outro

- Thanks to
    - Red Hook Studios
    - Power Up Audio
    - Pierre Tardif and Kelvin McDowell
- Contact
    - @KeirMiron
    - business@keirmiron.com