

Functionality:

Clone Object Template File
fTemplateFile
fObjectFileName
fRetainOldObjectNames
fPrefix

Make Walls
fWallPaletteFile
fWallTGAFFile
fWallStyle
fCutAway

Make Wall Style
fID
fWallTGAFFile
fCutAway

Make Wall Pattern
fID
fWallBMPFile

Make Floor
fID
fWallBMPFile
fShadow
fPoolTile

Make Water Tile
fWallBMPFile
fWallTGAFFile
fID

Make Thick Walls
fWallBMPFile
fZoom

Make Cut Away Thick Walls
fWallBMPFile
fWallTGAFFile
fZoom

Make Draw Groups
fAttributesFile
fDGRPTGADirectory
fTotalSpec->wallStyleSpecs
fWallStylesToCompress

Make Sounds
fAttributesFile
fTotalSpec->soundSpecs

Make Slots
fAttributesFile
fTotalSpec->slotsSpecs

Make Headlines
fAttributesFile
fHeadlinePaletteFile
fHeadlineTGAFFile
fTotalSpec->headlineSpecs

Make Attributes

```
fAttributesFile
fTotalSpec->attributeSpecs
```

```
Make Skill Table
  fAttributesFile
  fTotalSpec->skillSpecs
```

```
Make Catalog
  fAttributesFile
  fTotalSpec->catalogSpecs
  fCatalogDirectory
```

```
Make Tree Tables
  fAttributesFile
  fTotalSpec->treeTableSpecs
```

```
Make BMPs
  fAttributesFile
  fTotalSpec->bmpSpecs
  fBMPDirectory
```

```
Make Careers
  career
  job
```

Pairing of symmetric tiles of multi tile objects.

Examples:

```
Dollhouse.omk, Table.omk
  Simple, one sprite, one draw group, no symmetry.
```

```
Global.omk
  thought balloons and other random sprites
  global routing slots
```

```
PersonGlobals.omk
  person attributes
```

```
Floors.omk
```

```
SocialObj.omk
  social object attributes
```

```
TestObject.omk
  test routing slots
```

```
Careers.omk
  career, job
```

```
Cursors.omk
  cursor sprites and draw groups
```

```
Destination.omk
  destination bitmap file
```

```
Duck.omk
  skill table, skill (obsolete?)
```

```
HelpSystem.omk
  help bitmaps and team bitmaps
```

Stuff that TDSB keeps track of, from TDSBSpecs.h:

```

FrameInfo
    StackString<16> size; // "small", "med", "large"
    Int row;
    Int column;

FrameInfoList : STD::vector<FrameInfo>

SpriteList : PtrList<SpriteSpec>

EmbeddedSprite
    Int dirOffset;
    SpriteSpec *original;

+ GenericSpec
    int destID;
    int language;
    ResourceName name;
    TargetSpec *objectTarget;

- TargetSpec : GenericSpec
    MHandle objd;
    iResFile *objectFile;
    ObjDefinition *def;

- CatalogSpec : GenericSpec
    FileName bmpFileName;
    FileName popupBmpFileName;
    StackString<2048> description;
    StackString<512> nameDescription;
    StackString<512> descriptionDescription;
    ObjDefinition *objDef;

- TreeTableSpec : GenericSpec
    TreeTable *treeTable;
    vector<int> indices;

- BMPResourceSpec : GenericSpec
    FileName bmpFileName;

+ StringSpec : GenericSpec
    StringSet strings;

+ SkillTableSpec
    StringSet animTables[4];

+ AttrStringSpec : StringSpec

+ SpriteSpec : GenericSpec
    int mPosition; // row number
    int mColumns; // number of columns per zoom
    int mFrames; // number of frames (rows)
    int mSquishType; // which side to shrink wrap from
    int mPairing;
    Boolean mReverse;
    int mRotation;
    RenderSpec *mSizes;
    int mSymmetry; // 0 for none, 1 for bi-lateral, 2 for radial, 3 for
radial-flip
    int mPaletteID;
    RectList myShrunkRects;
    FileName mFileName;
    FileName mMaskFile;
    FileType mFileType;

```

```

- DGRPSpec : GenericSpec
  int ordering;
  iResFile* objectFile;
  Boolean overwriteOffsets;
  PtrList<EmbeddedSprite> embeddedSprites;
  DrawGroup theGroup;
  Boolean newDrawGroup;

- RenderSpec : GenericSpec
  int height[3];
  int width[3];
  int xorigin[3];
  int yorigin[3];
  int slopWidth[3];
  int borderWidth[3];

+ SoundSpec : GenericSpec

+ SlotsSpec : StringSpec
  SlotDescriptorList vec;

+ HeadlineSpec : GenericSpec
  int mPosition;
  int mFrames;

- WallStyleSpriteSpec:public : GenericSpec
  FileName mFileName;
  FileName mCutawayFileName;
  Boolean mGlobal;

- TotalSpec
  SpriteList sspecs;
  PtrList<DGRPSpec> dspecs;
  RenderSpec curSizes;
  PtrList<SoundSpec> soundSpecs;
  PtrList<SlotsSpec> slotsSpecs;
  PtrList<HeadlineSpec> headlineSpecs;
  PtrList<AttrStringSpec> attributeSpecs;
  PtrList<SkillTableSpec> skillSpecs;
  PtrList<TargetSpec> targetSpecs;
  PtrList<CatalogSpec> catalogSpecs;
  PtrList<TreeTableSpec> treeTableSpecs;
  PtrList<BMPResourceSpec> bmpSpecs;
  PtrList<WallStyleSpriteSpec> wallStyleSpecs;

```

Stuff that BuildObject keeps track of, from BuildObject.h:

```

BuildObjectArgs
  STD::string *fSyncMessage;
  CRITICAL_SECTION fMessageLock;
  TotalSpec *fTotalSpec;
  FileName fObjectFileName; // destination object file, required
  FILE *fErrFile; // error stream, required
  FileName fAttributesFile; // source omk file, if fMakeDGRPs

  // options
  Boolean fVerbose;
  Boolean fMakeNewObject;
  Boolean fMakeWalls;
  Boolean fMakeSounds;
  Boolean fMakeDGRPs;
  Boolean fMakeSlots;
  Boolean fMakeHeadlines;
  Boolean fMakeAttrs;
  Boolean fMakeSkillTable;

```

```

Boolean fPrintDGRPs;
Boolean fMakeCatalog;
Boolean fMakeTTabs;
Boolean fMakeBMPs;
Boolean fPrintHistory;
Boolean fMakeWallStyle;
Boolean fMakeWallPattern;
Boolean fMakeFloor;
Boolean fMakeThickWalls;
Boolean fMakeCutawayThickWalls;
bool fWaterTile;

// arguments
int fZoom; // if fMakeThickWalls, fMakeCutawayThickWalls
FileName fDGRPTGADirectory; // if fMakeDGRPs
FileName fBMPDirectory; // if fMakeBMPs
FileName fTemplateFile; // if fMakeNewObject (kinda)
Boolean fRetainOldObjectNames; // if fMakeNewObject (kinda)
ResourceName fPrefix; // if fMakeNewObject (kinda)
Boolean fCutAway; // if fMakeWalls, fMakeWallStyle
int fWallStyle; // if fMakeWalls
FileName fWallPaletteFile; // if fMakeWalls
FileName fWallTGAFFile; // if fMakeWalls, fMakeWallStyle, fMakeWaterTile,
fMakeCutawayThickWalls
    FileName fWallBMPFile; // if fMakeWallPattern, fMakeFloor, fMakeWaterTile,
fMakeThickWalls, fMakeCutawayThickWalls
    int fID; // if fMakeWallStyle, fMakeWallPattern, fMakeFloor,
fMakeWaterTile
bool fShadow; // if fMakeFloor
bool fPoolTile; // if fMakeFloor
FileName fHeadlinePaletteFile; // if fMakeHeadlines
FileName fHeadlineTGAFFile; // if fMakeHeadlines
FileName fCatalogDirectory; // if fMakeCatalog
vector<SInt16> fDrawGroupsToCompress; // if fMakeDGRPs
vector<SInt16> fWallStylesToCompress; // if fMakeDGRPs
FileName fHistoryOutput; // if fPrintHistory

```

FindOrCreateTarget (function used by several elements)

```

guid="123456"
object="name"

```

GenericSpec

Inherited by: TargetSpec, CatalogSpec, TreeTableSpec, BMPResourceSpec, StringSpec, AttrStringSpec, SlotsSpec, SpriteSpec, DGRPSpec, RenderSpec, SoundSpec, HeadlineSpec, WallStyleSpriteSpec.

```

<xxx
    id="0"
    name="name"
    language="0">

```

StringSpec

Inherited by: AttrStringSpec, SlotsSpec.

```

<xxx
    id="0"
    name="name"
    language="0">

```

AttrStringSpec

```

<attributes
    id="0"
    name="name"
    language="0">
    <attribute
        index="0"

```

```
label="label">
```

SkillTableSpec

```
<skill table>
  <skill
    index="0"
    name="name"
    type="a2a|a2o|c2c|c2o|a2c|c2a">
```

SlotsSpec

```
<slots
  name="name"
  id="0"
  language="0">
  <slot
    x="0"
    y="0"
    alt="0"
    type="contained object|headline|sprite|handle|routing|
    maximum size="0"
    surface="false"
    height="group|low table|end table|table|counter|non-
standard|sitting"
    standing="0"
    sitting="0"
    ground="0"
    target="0"
    min="0"
    max="0"
    optimal="0"
    gradient="0"
    resolution="0"
    north="false"
    south="false"
    east="false"
    west="false"
    northwest="false"
    northeast="false"
    southwest="false"
    southeast="false"
    no rotation snap="false"
    facing not required="false"
    opposite facing="false"
    ignore rooms="false"
    snap to direction="false"
    absolute="false"
    random="false"
    alt not required="false"
    average object location="false">
```

TargetSpec

```
<target
  id="0"
  name="name"
  language="0">
```

TreeTableSpec

```
<ttab
  id="0"
  name="name"
  language="0"
  guid="123456">
  <interaction
    index="0"
    name="name"
```

CatalogSpec

```

    <catalog
      id="2000"
      guid="123456"
      name="object name"
      bmpfile="name.bmp"
      thumbnail="name.bmp"
      popup bmpfile="name.bmp"
      cost="0"
      initial depreciation="0"
      daily depreciation="0"
      depreciation limit="0"
      standard depreciation="0"
      room sort="Bathroom|Bedroom|Dining Room|Family
Room|Kitchen|Study|Outside|Miscellaneous|"
      function
sort="Appliances|Decorative|Electronics|General|Plumbing|Seating|Surfaces|Lighting|
"
      hunger="0"
      comfort="0"
      hygiene="0"
      bladder="0"
      energy="0"
      fun="0"
      room="0"
      cook="0"
      mechanical="0"
      logic="0"
      body="0"
      creativity="0"
      charisma="0"
      study="0">

```

BMPResourceSpec

```

    <bmpfile
      id="0"
      name="name"
      language="0"
      filename="name.bmp">

```

SpriteSpec

```

    <sprite
      id="0"
      name="name"
      language="0"
      tgafile="name.tga"
      bmpfile="name.bmp"
      mask="name.bmp"
      position="0"
      frames="0"
      columns="1|2|4"
      shrink="true|top|false"
      reverse="false"
      paletteid="-1"
      symmetry="none|bi-lateral|radial|radial-flip"
      pairing="0"
      rotation="0"

```

HeadlineSpec

```

    <headline
      [{sprite atts} - {columns, shrink, reverse, symmerty}]
      position="0"
      frames="1">

```

SoundSpec

```
<sound
    name="name.wav">
```

DGRPSpec

```
<drawgroup
    id="0"
    name="name"
    language="0"
    ordering="fixed|obscured">
    <item
        sprite="true"
        alt="0"
        x="0"
        y="0"
        id="0"
        direction offset="0"
        luminous="false">
```

RenderSpec

```
<sizes
    id="0"
    name="name"
    language="0"
    slopwidth="0"
    uniform slopwidth="0"
    border="0"
    lwidth="0"
    mwidth="0"
    swidth="0"
    lheight="0"
    mheight="0"
    sheight="0"
    lxorigin="0"
    mxorigin="0"
    sxorigin="0"
    lyorigin="0"
    myorigin="0"
    syorigin="0">
```

WallStyleSpriteSpec

```
<"wall style"
    id="0"
    name="name"
    language="0"
    tgafile="name.tga"
    cutaway tgafile="name.tga"
    global="false">
```

<career>

```
<job>
    Not implemented.
```

Note on the storage of old sprites kClassicRLE:

The z and other planes appear as additional frames at the end of the sprite list.

```
if ((mVersion == cRenderer::kZBufferVersion ||
     mVersion == cRenderer::kPalettizedZVersion))
this->RepackageSpriteList( data, Memory::HGetSize(inSpr), shapes );

if ((mVersion == cRenderer::kZBufferVersion ||
     mVersion == cRenderer::kPalettizedZVersion))
{
    mCount /= 2;
```



```

mPixelRLEStreams = new cRenderer::Shape *[mCount];
mZRLEStreams = new cRenderer::Shape *[mCount];

for (int i=0;i<mCount;++i) {
    mPixelRLEStreams[i] = shapes[i];
    mZRLEStreams[i] = shapes[i+mCount];
}
} else if ( mVersion == cRenderer::kPalettizedZAlphaVersion ) {
    mCount /= 5;

    mPixelRLEStreams = new cRenderer::Shape *
[mCount];
    mZRLEStreams = new cRenderer::Shape *
[mCount];
    mPixelFringeRLEStreams = new cRenderer::Shape *[mCount];
    mZFringeRLEStreams = new cRenderer::Shape *
[mCount];
    mAlphaRLEStreams = new cRenderer::Shape *
[mCount];

    for (int i=0;i<mCount;++i) {
        mPixelRLEStreams[i] =
shapes[i];
        mZRLEStreams[i] =
shapes[i+mCount];
        mPixelFringeRLEStreams[i] =
shapes[i+2*mCount];
        mZFringeRLEStreams[i] =
shapes[i+3*mCount];
        mAlphaRLEStreams[i] =
shapes[i+4*mCount];
    }
} else {
    mPixelRLEStreams = new cRenderer::Shape *[mCount];

    for (int i=0;i<mCount;++i) {
        mPixelRLEStreams[i] = shapes[i];
    }
}
}

```

Need to be able to edit tree table entry name and description strings:
kTreeTableStringsResType

Sprite origin calculated thusly:

```

int cnt;
for (cnt=0; cnt<3; cnt++) {
    (tagvalue = attributes.FindValue(wstrs[cnt])) ||
ReqDie(wstrs[cnt]);
    sscanf(tagvalue, "%d", &width[cnt]);

    (tagvalue = attributes.FindValue(hstrs[cnt])) ||
ReqDie(hstrs[cnt]);
    sscanf(tagvalue, "%d", &height[cnt]);

    tagvalue = attributes.FindValue(xoriginstrs[cnt]);
    if (tagvalue && sscanf(tagvalue, "%d", &xorigin[cnt])==1)
        ;
    else
        xorigin[cnt] = width[cnt]/2;

    tagvalue = attributes.FindValue(yoriginstrs[cnt]);

```

```

        if (tagvalue && sscanf(tagvalue, "%d", &yorigin[cnt])==1)
            ;
        else
            yorigin[cnt] = height[cnt] - HTILEHI(3-cnt) -
(slopWidth[cnt]) - borderWidth[cnt];
    }

```

=====

had to define PROJECT_X and FIX_IN_PATCH=1 to catch bug fix in strset.cpp pertaining to setting other language strings.

Made dummy finalship.h file

=====

Done:

To Do:

```

+           puff up and shrink wrap sprites
+           shrink smaller zooms from largest zoom
+           palette management
+           edit object name and properties
+           export and import catalog pictures and icons
+           write out subdirectories for sprites and p/z/a
+           clean up exporter options (hide unnecessary options,
streamline)
           "exporter whizzer" radio buttons to select common scenarios
+           view multi tile objects
           Don't draw dynamic sprites. Call IsSpriteVisible().
+           splash screen, disclaimer
+           filter uninteresting objects:
           pedportal, visitorgenerator, helpsystem, npc
controller, phone line,
           comeandseeme,
           users, houses, options (done)
+           Figuring out invisible pixel.
           Complains if it can't find an unused transparent pixel,
when generating z buffer.
+           Use a reasonable z buffer value for far and near z buffers.
           Make a garbage can with a far z buffer.
           It can't be easily picked up or highlighted with the move
tool,
           because the z buffer is off the edge of the tile.
+           Need to remap guids in code we clone?
           Clone a moose head. View tree crashes in an infinite loop,
           because it's checking for an object of type "moose head
left", by the guid,
           which was changed.

```

TODO:

Now:

ez make importer check for importing into same file
work document

Later:

convex box, concave box fun make generic template objects, like rug, picture,
ez flesh out fancy object editor bit fields
fun drawgroup view magnifier, z-buffer x-ray (xyz-ray?)
fun run batch mode from command line
+ export, import and edit catalog text
+ import draw groups
+ import object definitions

Bugs:

+ When generating Z-Buffers for smaller scales,
the z values may have to be scaled around some z value.
Run a test case through the 3dsmax exporter to find out,
or measure some content that it created.

+ Need to handle draw group item floating point x y and z
offsets.