7/15/2014

Parser class Modularization through dependency analysis and unit test. Discarding externally dependent members...

7/16/2014

Performing CRC analysis with class Level, MeshBuilder, TextureUV.

Level:

Level Manages prefab and instantiated game objects, rooms, actor, static objects. This class uses helper class

MeshBuilder, TextureUV e,t,c

MeshBuilder:

Generates unity mesh from tr2 mesh.

TextureUV:

An utility class for texture manipulation.

CRC using existing class:

When analyzing mesh builder member function input, main input should correspond to an existing tr2 mesh.

Function should be helper in nature that will serve an entity type, not an ID. ID is not finite in nature and context sensitive. Example:

CreateStaticObject(Mesh mesh, Vector3 position, Quaternion rotation, string name)

Here, Parameter should be Mesh type, not an ID type. ID type would restrict wide use of CreateStaticObject() ID is

bound to a specific context, such as an valid array of mesh. Here, CreateStaticObject() is not sensitive to such

array bound.

7/19/2014

Noticed object self collision while ray casing. Ignore attched collider to player.

TODO:

Handle out of index error while calculating animation clip count for a specific movable tr2 object.

7/21/2014

Attach AI with State Machine to NPC Object Tiger( Test purpose)

7/23/2014

Extending tr2 base class...

Attaching custom properties to cloned unity object. One way to do this is assigning a object containing custom

properties as a variable to MonoBehaviour script of that cloned object and then initialize the cloned object and it's component with that variable.

Example:

public class GameObjectEx: MonoBehaviour

{

public ClassData data; // data holding custom properties

void Start()

{

//perform runtime initialization like animation clip setup

//with data

}

}

8/04/2014:

An Note On CRC:

A class does not need to know about another class, unless itâ€™s task requires it. It will enforce strict

responsibility of a class. For example, an event system does not need to know how the event will be handled

and where.

An object based approach for a game engine

GameObject\* objects[]

GUIElement\* elements[]

Player\* player

Window\* window

while (true)

{

// read player input

if (key == DOWN) {

update player's position

}

if (joystick is moved) {

update player's position

}

// update each object belonging to the game objects

for (each game object) {

do physics simulation

do AI

}

// render each object belonging to the game objects

for (each game object in objects[]) {

Window->render(object)

}

// render each GUI element

for (each game element in elements[]) {

if (element has been updated) {

Window->render(element)

}

}

}

What we got is a God object, a well known anti-pattern. As we add more features to our game, maintaining such

all-knowing objects becomes a bottleneck of the project. It is also worth noting that when doing rendering,

the renderer needs to go through all the game objects to know which objects have been updated. The same applies

to the GUI elements. We don't use Observer pattern, so we can not notify the renderer about individual updates

as soon as they happen.

//more on http://en.wikipedia.org/wiki/God\_object

8/28/2014

Writting data on SDCard could be potential source of error!

Transparent/cutout/diffuse does not work for android?

9/1/2014

Developing custom TR2 level with TRLE.

Generating high frequency leave texture with ripple distortion and noise in PS. Texture highlight can be

enhanced by mid tone contrasting in PS.

9/3/2014

Handle SFX with animation

The highest level of these is the SoundMap[] array, which translates the internal sound index into an index into

SoundDetails[]. Each SoundDetails record contains such details as the sound intensity, how many samples to select

from, and an index into SampleIndices[]. This allows for selecting among multiple samples to produce variety;

that index is the index to the SampleIndices[] value of first of these, with the rest of them being having the

next indices in series of that array. Thus, if the number of samples is 4, then the TR engine looks in

Sample Indices[] locations Index, Index+1, Index+2, and Index+3. Finally, the SampleIndices[] array references

some arrays of sound samples. In TR1, these samples are embedded in the level files, and SampleIndices[] contains

the displacements of each one in bytes from the beginning of that embedded block.

In TR2 and TR3, these samples are concatenated in the file "MAIN.SFX" with no additional information;

SampleIndices[] contains sequence numbers (0, 1, 2, 3, ...) in MAIN.SFX. Finally , the samples themselves are

all in Microsoft WAVE format.

typedef struct { // 2 bytes

bit16 Value;

} tr2\_anim\_command;

Here are all the AnimCommand opcodes and their operands:

// 1: 3 operands. Position reference: (x,y,z); found in grab and block-move animations

// 2: 2 operands. Position reference on surface for jumping: (x,z) for horizontal and (y,z) for vertical surfaces(?)

// 3: No operands. Not clear; occurs in animations that are "slaved" to other animations, such as Lara throwing switches or moving blocks.

// 4: No operands. Not clear; occurs in some death and settling-down animations, but not all.

// 5: 2 operands. The first one is a frame number, and the second one is the ID of the sound to play at that frame (internal sound index).

In TR2 and TR3, one of the sound indices two highest bits may be set; when they are, their meanings are

0x4000 -- play this sound when on dry land (example: footsteps)

0x8000 -- play this sound when in water (example: running through shallow water)

9.8.2014

Developing custom TR2 level with TRLE

Trapdoor not working! Help!

You have placed a trapdoor object covering a door to the room below. You have placed a switch to it, all works ok,

but in game when the trapdoor is closed it is not solid and Lara go through it.

For trapdoors and similar objects you have to do an additional step to make them solid stand-able, you have to

put in the solid sector below the trapdoor object a TRAPDOOR trigger with the command Item(x) where xÂ is the

trapdoor object you are making solid.

9.16.2014

Realize engine architecture.

Elegant way to using Game Engine is to use its query mechanism. Engine maintains a list of Objects with component.

Why not let engine handle them. Push Object to engine and If anything to know ask it for that. Letâ€™s extend room

objects with MonoBehabiour component holding important room info. Letâ€™s convert RoomEx in to Monobehabiour. Letâ€™s

limit Responsibility of Level class as component object factory and passing them to engine.

Learned: It takes time to make an update which has measurable cost!

Define Player Events:

OnPositionChanged()

OnSectorChanged()

OnRoomChanged()

CanHandlePrimaryAction()

Smartly determine possibility(Grab Ledge, Pull self up, Pull Push block e.g.) analyzing surface geometry

9.20.2014

Physics Generated events:

OnCollision() //Handled case collision generated during flying, jumping, landing time collision //General handled

case is going to idle state;

9.21.2014

Updating AnimationStatePlayer! Mostly cleanup tasks! All Physics handlers are moved to Player Class. Now

AnimationStatePlayer is just a view of Player class. Added Event for Animation states that Player class will handle. For example, OnJump event. Collision checks are mostly done on movement events. Thatâ€™s a great saving against continuous collision check!

FIXED: Lara is initially in negative height position in some level. but mistakenly her initial setup height

position was 0. So while landing (without ground collision check!) Lara was trying to reach 0 position

her assumed landing position!). Which is higher than her initial jump position? This was giving floating on

air error.

FIXED: Every transform that gets hit in room by Lara is not room transform.

9.22.2014

Pull Up Movement State Should be generated with DPAD + Action Key combination

9.23.2014

Write algorithm to detect ladder step (platform) and find highest step from triangle network.

First select a triangle from triangle network

Then find a edge that is normal to ground.

If normal direction going upward than this triangle does

not have platform edge.

Look for diagonal edge.

Select another triangle sharing this diagonal edge

Where selected triangle does not share any other vertex

from base triangle except vertex on diagonal edge.

else Select third vertex of triangle that does not lie

on normal edge.

Select highest vertex along normal edge.

Connect it with third vertex to create platform edge.

Select another triangle sharing this platform edge

where selected tri does not share any other vertex

from base triangle execpt vertex on platform edge.

Calculate curvature along edge. If it is flat then run

the whole procedure again

9.27.2014

FIXED: Get back to free position bug. Actually last freeposition must recorded unconditionally after every

movement processing.

weather collision happens or not. It must be unconditional. freeposition must also be checked in idle state and

start game.

10.1.2014

FIXED: Ladder Step Detection algorithm

First, select a face from face network

Then, find a edge that is normal to ground.

If normal going upward and found next edge diagonal - > go to mirrored face.

If normal going upward and found next edge horizontal - > Next edge ladder step

-> if ladder step found check it is topmost go to mirror face.

else

If normal going downward and found next edge horizontal - > go to mirror face.

If normal going downward and found next edge diagonal - > third edge is ladder step

-> if lader step found check it is topmost go to mirror face

10.3.2014

FIXED: Stand position after pulling self up on a Block. Unfortunately, I added Block height with last hanging

height (of foot). In fact it will be hanging height (of foot) + body height, so that stand position is exactly

on top of block.

Note: Pivot point is Lara's foot.

New Bug: Flickering after height adjustment!!

When pull up animation going there is difference between foot position

and actual pivot position. Difference remains at the end of animation. When we try to reposition pivot to expected

foot position, model body is translated to new height. When animation is reset or new animation is applied, pivot

gets back to actual position. This back and forth translation causes Flickering.

10/22/2014

static event in DayNight system still refferencing invalid event handler! should use singleton type DayNight

system for easy access.

10/24/2014

TODO: Implement ladder step alorithm for boulder block. Perform raycast on boulder mesh. Build manifold mesh.

Get shared triangle.Calculate vertex triangle network for boulder mesh.

11/6/2014

Added Event managment system OVRMessenger.cs to handle event listner and refference. Some time event listner or

call back handler gets invalid because of delegator object distruction. This commonly happens for static events.

11/8/2014

Checked for valid tr2 file type on level load by checking file version! Otherwise returned to Level Load Screen

with massasge.Added Last Loaded Level path to Player Pref with last successfull level load! Checked if directory

exsists or empty.

FIXED: SetRoomForPlayer() in class level. Downward raycast was unable to hit room plane. Because raycast origin

was started on room plane. Setting raycast origin slight above room plane solved the problem.Same error could

happen in PositionChanged() in Player class.

Thinking to create custom wrapper class as raycast utility handle ground hit ditection by player.

1/5/2015

Disabled room shadow casting on interior objects

1/27/2015

m\_Room.DebugRoomSurface(facing\_edeges) causing null refference exception when playing level generated in edit mode.

Disabling Lara stops this exception.

2/7/2015

Changed Class name AnimationStatePlayer to LaraStatePlayer. It was handling specially Lara's animation state changes.

So I though it would be right naming and intutive also :)

Most of the AI should inherit a common visibility function. May use ray cast to determine proximity of player.

2/7/2015

Some thoughts that made me sleepless :)

Software should mean softlywared, meaning loosely coupled and detachable when required by other.Because people may not be intersted to system as a whole, they could be interested some donatable self sufficient part of system.

Thats the perspective of object orientation should be I think.

Best practice of objec orientation could be acheived by reusibility improvment, instead of practicing differentited implementation (or reinventing the wheel) all the time. For example, here I used totally

different camera for door opening, where funtionality like smooth movement that I implemented in main camera is missing. Instead of implementing whole functionality for scratch, why not reuse main camera handler.

Now question of improvment practice come here. Develope main camera handler such that it could be open for reusibility and extention.

Moved BuildAnimationStateMap() utility function LaraStatePlayer class to its own AnimationStateMapper class.

This is more meaningful.

2/8/2015

Noticed unstable behaviour when LaraStatePlayer is enabled before component Player. Trying to look for dependency here. Turned out, key mapper should be initialised first.Animation state codes are coupled with keycodes. Initialise Keymapper first.

Required instaniation:

ThirdPersonCam (Prefav camera)

KeyMapper (Prefav KeyMapper)

Mouse (Prefav Mouse)

Gui

DayNightSystem

Ambient light

TODO: Generate prefab for Lara with minimalistic requirement

2/13/2015

FIXED: Free fall movement error by allowing free fall if not already free falling or jumping in Player.cs

if(!m\_bFreeFall && !m\_bJumping)

{

if(hit.point.y != m\_GroundHeight)//try free fall

{

m\_bFreeFall = (transform.position.y - hit.point.y > m\_FreeFallStartHeight) ;

physics.StartFreeFall(transform.position);

m\_FreeFallStartTime = Time.time;

Debug.Log("Start Free Fall");

}

}

2/25/2015

Is there any way to set Time.fixedTime from script? Otherwise random fixed time setting in projec t can effect animation keyframe timing.