Building Area Manual

Contents:

Object Follow Camera
Object Follow Camera is 3rd person camera, transition to target vector smoothly, move around when mouse is near the edges of screen, and rotate around target when middle moue button is pressed.

Properties:

Vector3 TargetObject	get, set	Camera will smoothly follow this vector when set it
Vector3 Direction	get	The look at direction (target - camera position)
Vector3 Target	get	Camera current target (or current transition target)
Vector3 Position	get	Camera current position (or current transition position)
float TransitionTime	field	Transition time in seconds
float Distance	field	Distance between camera target and camera position
float ScreenSides	field	Mouse sensitive area [0 … 0.5] 0 : mouse will never move camera 0.5 : mouse will always move the camera because all areas took half screen
Vector2 MoveSpeed	field	Camera maximum move speed when mouse is on the edge of screen
Vector2 RotateAroundSpeed	field	Camera rotate speed when middle mouse is pressed

Methods:

void SetDirection(Vector3 dir, bool invertIfNeeded)	Change the looking direction smoothly (orbit position around target) if (invert if needed) is true the direction may be inverted if its make transition distance less, false the direction will be set as it is

Draggable
Draggable allows game object to be dragged (if needed) when click and hold mouse button on it.

Properties:

bool Enabled	field	Enable or disable dragging
bool XEnabled	field	Enable or disable snapping on X-Axis
bool YEnabled	field	Enable or disable snapping on Y-Axis
bool ZEnabled	field	Enable or disable snapping on Z-Axis
int XSnapDistance	field	Snap distance for X direction
int YSnapDistance	field	Snap distance for Y direction
int ZSnapDistance	field	Snap distance for Z direction
bool IsDragging	get	Returns true if the object is currently in dragging mode

Events:

StartMoving	Called when the first click on object is down
Moving	Called when mouse is moving and after the StartMoving called
EndMoving	Called when mouse is up and after Moving or StartMoving called

Line ** this class suffering from weakness .. some properties need to be hidden
Line is a primitive unit of the plan, it holds wall data, the data representation is using index to vertex to highly reduce memory usage.

Enum LineType

Wall	Default value for all basic lines
Window	Automatically created when split wall line to walls and windows line when constructing 3D wall

Properties:

LineType LineType	field	The type of line, default is Wall
List<WallWindow> Windows	field	Windows inside this wall
Line ParentLine	field	Used to determine whole line when split wall line to walls and windows line when constructing 3D wall
float Height	field	Wall height
float LedgeHeight	field	The height of ledge used only for window sub-line
float WindowHeight	field	The height of window used only for window sub-line
Vector3 a	get, set	Gets or sets the first point of line, this will modify the source vertex list
Vector3 b	get, set	Gets or sets the second point of line, this will modify the source vertex list
int aID	get, set	Gets or sets the source index in vertex list of the first point
int bID	get, set	Gets or sets the source index in vertex list of the second point
Material InnerMaterial	field	Holds the inner face material (for indoor side of the wall)
Material OuterMaterial	field	Holds the outer face material (for outdoor side of the wall)
Material SideMaterial	field	Holds the door sides material
Material LineMaterial	get, set	Gets or sets the material of wireframe line
float Thickness	field	Holds the thickness of current wall
List<vector3> Vertices	get, set	Gets or sets the source vertex list
bool Enabled	get, set	Gets or sets a value indicating whether this Line is enabled.
Transform transform	get	Gets the transform of this object
Transform Parent	get, set	Gets or sets the parent

Methods:

void DetachA() void DetachB()	Detaches the desired vertex by make new vertex in the vertices list and assign desired vertex ID by new vertex Index
void Destroy()	Destroys the internal line renderer.

Static Methods:

bool RayRayIntersection(out Vector3 intersection, Vector3 line1Point1, Vector3 line1Point2, Vector3 line2Point1,Vector3 line2Point2)	Return true if ray1 intersects with ray2, false otherwise
List<Line> Split(List<Line> lines, Vector3 point)	Splits the line if point is between both vertices, this will create new line, new vertex, and replace one of the vertices ID of that line.
List<Vector3> WeldVertices(List<Line> lines)	Creates, and returns new vertices list less or equal the current vertices list by removing duplication, and assign lines vertex IDs
void Generate3DWallFacesFromLines(List<Line> _segments, Material WallWireframeMaterial, Material WallSelectedMaterial, out List<WallFace> outerWall, out List<WallFace> doorSides, out List<WallFace> innerWall, out GameObject upperWallFace)	Generates WallFaces, the algorithm work as follow, first extract line to sub-lines for windows, then offset each segment to both directions, then trim the line intersections, and finally extract topology of the generated contour and determine weather its inner or outer contour, extrude it and assign material
void OptimizePath(ref List<Line> lines)	Optimizes the path by remove zero-length lines, merge continuous lines, and split lines
void FillCap(List<Line> lines, out List<int> triangles, out List<Vector3> verts, out List<Vector2> uvs, out List<Vector3> normals)	Generates geometry that fills this path

WallFace
WallFace is a 3D quad defined by some parameters.

Enum WallFaceType

Outer	The outdoor side of the wall
Inner	The indoor side of the wall
DoorSide	The door side of the wall

Properties:

WallFaceType WallFaceType	field	Holds the type of the wall face
Line RelatedLine	field	Holds the related line of this wall face
Material SelectedMaterial	get, set	Gets ors sets the material of the lines when this wall face is selected
Material WireframeMaterial	get, set	Gets ors sets the material of the lines when this wall face is in wireframe mode
Material WallMaterial	get, set	Gets or sets the material of the wall face it self
Vector3 a	get, set	Gets or sets the first vertex
Vector3 b	get, set	Gets or sets the second vertex
float Height	get, set	Gets or sets the height of the wall face
float UpOffset	get, set	Gets or sets the up offset of the wall face (height from ground)
bool Selected	get, set	Gets or sets a value indicating whether this WallFace is selected.
bool Solid	get, set	Gets or sets a value indicating whether this WallFace is solid.
bool Wireframe	get, set	Gets or sets a value indicating whether this WallFace is wireframe.
Transform Parent	get, set	Gets or sets the parent
bool IsFacingCamera	get	Returns true if this wall face is in the direction that facing camera, otherwise false

Methods:

void Destroy()	Destroys the components of this object.
bool IsMouseOver()	Returns true if the mouse is over this wall face, otherwise false
bool IsMouseOver(out float dst)	Returns true if the mouse is over this wall face, otherwise false and assigns dst to distance from camera to 3D point on the wall face

Wall Material
Container class, holds the values used in menu and editor and its related data
Fields

Sprite MaterialImage	field	Holds the icon of this item
Material InnerFaceMaterial	field	Holds the inner face material
Material OuterFaceMaterial	field	Holds the outer face material
Material SideFaceMaterial	field	Holds the side face material

Window Material
Container class, holds the values used in menu and editor and its related data

Sprite WindowMaterialImage	field	Holds the icon of this item
GameObject Model	field	Holds the shape of the window

Wall Window

Wall Window is a window controller contains info about location of the window and model of the window

Properties

Line Line	get, set	Gets or sets line which contains this window
Vector2 Position	get, set	Gets or sets the position of the window relative to first point of the line, in real world units
float WindowHeight	get, set	Gets or sets the height of this window
float WindowWidth	get, set	Gets or sets the width of this window
GameObject Window	get, set	Gets or sets the model of this window

Methods:

void Update()	Updates rotation of the window object to match the orientation of the line, and place it to correct position and scale

Materials Panel

Properties

BuildingArea BuildingArea	field	A reference to editor
WallMaterial[] WallMaterials	field	A set of wall materials to display on the panel

Methods

void Start()	Initialize items of this panel with icons and events
void materialClicked(int i)	Callback, when item is clicked this will call SetSelectedWallFaceMaterials(..) from the editor

Window Materials Panel

Properties

BuildingArea BuildingArea	field	A reference to editor
WindowMaterial[] WindowMaterials	field	A set of window materials to display on the panel

Methods

void Start()	Initialize items of this panel with icons and events
void materialClicked(int i)	Callback, when item is clicked this will call SetWindowMaterials(..) from the editor

Building Area
Building area controls everything.

Enum BuildingEditMode

None	No objects where selected, the user can select any object
Drawing	The user is currently drawing a line
WallFaceSelected	The user clicked on wall face, the transforming handles shown above the object
WallFaceMoving	The user is moving the wall.
WallVertexMoving	The user is stretching wall by moving the vertex from one of the sides.

Properties:

BuildingEditMode Mode	get, set	Gets or sets current mode, if mode is WallFaceSelected, the wall materials panel and window materials panel will be active
int snapGridDistance	field	The distance between grid points, default value is 1
GameObject VertexHandle	field	A game object will be placed above vertex as a sign to make user able to transform it.
GameObject snapObject	field	A game object will be placed on the grid points only while moving mouse over it
Material DraggedLineMaterial	field	After selecting first point a line appear following the mouse before placing second point, this is the material of that line
Material LineMaterial	field	The material of the line after create it
Material WallSelectedMaterial	field	The default material of selected wall face
Material WallWireframeMaterial	field	The default material of wireframe wall face when enable wireframe mode
GameObject MaterialsPanel	field	Reference to panel to enable or disable it when no wall face is selected or nothing is selected
GameObject WindowMaterialsPanel	field	Reference to panel to enable or disable it when no wall face is selected or nothing is selected
Material DefaultOuterWallMaterial	field	Default outer wall face material
Material DefaultInnerWallMaterial	field	Default inner wall face material
Material DefaultSideMaterial	field	Default side face material
float DoubleClickCatchTime	field	In seconds, the max time between first click and second click to detect if its double click default value is 0.25
WallFace selectedWallFace	get, set	Gets or sets the selected wall face this wall mark it as selected, and shows transformation controls above the Related line
List<WallFace> wallFaces	field	The list containing all active wall faces this list is regenerated after any edit

List<Vector3> lineVertices	field	The current vertices list used in every line in List<Line> lines
List<Line> lines	field	Contains the whole structure that needed to build 3D walls, windows, doors, and items
List<int> verticesSelected	field	used when right click is transforming vertex in planning mode .. it will never be more than 1 element !
bool PointASelected	field	flag to check if first point is selected
bool snapEnabled	field	if true any vertex will be created will be aligned to grid
Vector3 pointA	field	Saved location of the first point clicked by user
Vector3 MouseStartPosition	field	3D location where the mouse clicked
float MouseStartDistance	field	The distance from camera to where the mouse clicked first time (this will help on moving object on this distance when 3D mouse points to another place, while it actually most point to place near MouseStartPosition)
Draggable wallFaceHandleDraggable	field	The Draggable component of the wall handle
GameObject wallFaceHandleObject	field	A game object will be placed above the wall as a sign to make user able to transform it this is instance of VertexHandle
Draggable vertexAHandleDraggable	field	The Draggable component of the first vertex handle
GameObject vertexAHandleObject	field	A game object will be placed above first vertex as a sign to make user able to transform it this is instance of VertexHandle
Draggable vertexBHandleDraggable	field	The Draggable component of the second vertex handle
GameObject vertexBHandleObject	field	A game object will be placed above second vertex as a sign to make user able to transform it this is instance of VertexHandle
Button DetachButton	field	A reference to Detach button
Button DeleteButton	field	A reference to Delete button
GameObject upperWallFace	field	The generated upper mesh of the walls
ObjectFollowCamera gameCamera	field	A reference to ObjectFollowCamera component
Vector3 cameraTarget	field	Set on the first click or second click and used to set camera target when double click occurred
bool planningMode	field	true for Planning mode or false 3D walls mode
Line DraggedLine	field	DraggedLine containing the line from first point to mouse position (on Drawing Mode)
float lastClickTime	field	Contains the time when last click occurs

Methods:

void WallFaceHandleDraggable_StartMoving (GameObject sender, Vector3 oldPosition, Vector3 newPosition)	Callback, disable draggable component on other vertices and change mode to BuildingEditMode.WallFaceMoving
void WallFaceHandleDraggable_Moving (GameObject sender, Vector3 oldPosition, Vector3 newPosition)	Callback, move vertices of this line to new position and regenerate path
void WallFaceHandleDraggable_EndMoving (GameObject sender, Vector3 oldPosition, Vector3 newPosition)	Callback, enable draggable component on other vertices and change mode to BuildingEditMode.WallFaceSelected
void vertexAHandleDraggable_StartMoving (GameObject sender, Vector3 oldPosition, Vector3 newPosition)	Callback, disable draggable component on other vertices and change mode to BuildingEditMode.WallVertexMoving
void vertexAHandleDraggable_Moving (GameObject sender, Vector3 oldPosition, Vector3 newPosition)	Callback, move first vertex of this line to new position and regenerate path
void vertexAHandleDraggable_EndMoving (GameObject sender, Vector3 oldPosition, Vector3 newPosition)	Callback, enable draggable component on other vertices and change mode to BuildingEditMode.WallFaceSelected
void vertexBHandleDraggable_StartMoving (GameObject sender, Vector3 oldPosition, Vector3 newPosition)	Callback, disable draggable component on other vertices and change mode to BuildingEditMode.WallVertexMoving
void vertexBHandleDraggable_Moving (GameObject sender, Vector3 oldPosition, Vector3 newPosition)	Callback, move second vertex of this line to new position and regenerate path
void vertexBHandleDraggable_EndMoving (GameObject sender, Vector3 oldPosition, Vector3 newPosition)	Callback, enable draggable component on other vertices and change mode to BuildingEditMode.WallFaceSelected
void DetachSelectedWall()	Callback, detach both vertices of the current selected line
void DeleteSelectedWall()	Callback, delete selected line
void SetSelectedWallFaceMaterials(Material innerMaterial, Material outerMaterial, Material sideMaterial)	Sets the selected wall and all connected walls materials MaterialsPanel must call this to apply material change ******
void SetWindowMaterials(GameObject obj)	Sets the selected wall window material (shape) *****

void SetSelectedWallFaceWindowCount(string count) void SetSelectedWallFaceWindowCount(int count)	Splits the selected wall to number of windows *****
void PlanningMode(bool x)	enable or disable planning mode and apply changes
void Start()	Initialize fields values, assign events and get components
void Update()	Enable and disable walls and wireframe according to camera position, handle input for planning and 3D mode
WallFace getSelectedWallFace()	Returns nearest wall face from mouse or null if mouse is not point to wall face
bool snap (Vector3 pos, float maxlength, out Vector3 nearest)	snaps pos to nearest vertex of all lines around, maxlength limits search area, and returns true if found otherwise false
Vector3 snapToGrid(Vector3 pos)	snaps pos to grid defined by snapGridDistance
void regeneratePath(bool optimize)	regenerate 3D wall faces and upper face, if optimize is true lines will be optimized by removeng zero-length lines, and split when required