

Geoffrey Hendrikx
Research
UML Unity tools

DoelGroep

Designers en Developers

Google research (Wat bestaat er al?).

Het eerste wat ik deed was het opzoeken “UML Unity3d”. Hier vond ik een interessante forums die aan het discussiëren waren over een uml voor Unity3D games.

<https://forum.unity.com/threads/uml-in-unity.319225/>

Toen vond ik een forum post op de unity forums

<https://forum.unity.com/threads/generate-class-diagram-from-code.410257/>.

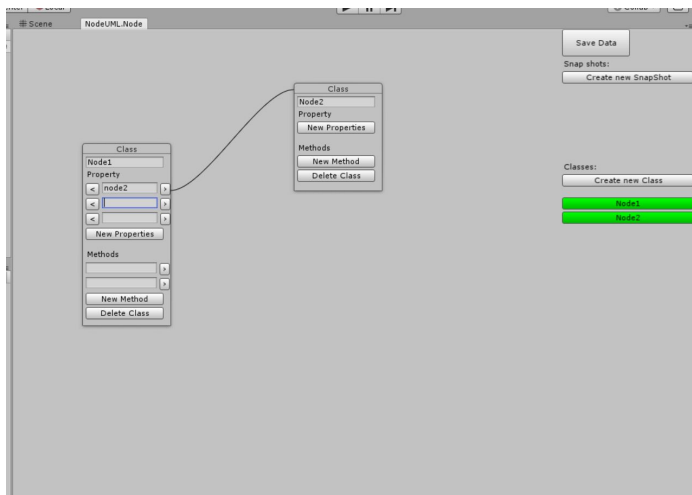
Helaas allemaal deprecated.

voor de rest vond ik een github genaamd NodeUML? <https://github.com/Zen12/NodeUML.git>

Deze uml tool is gemaakt in de unity versie 2017.2.0f3. Dit was te vinden in de

ProjectVersion.txt. die in de ProjectSettings map stond.

Deze tool zag er als volgende uit.



Wat ik erg goed vond aan deze tool de nodes paste zich aan aan hoeveel variables je had gemaakt. De lijntjes waren ook erg goed gemaakt. Je kon helaas niet de classes laten genereren of ophalen.

Code research.

```
2 references
public class NodeUML : EditorWindow
{
    private NodeController controller;

    private ClassHandler classHandler;

    [SerializeField]
    private UseCaseHandler useCaseHandler;
}
```

Wat mij erg opviel was dat de editor window niet heel veel instond. Hij heeft 3 variables waarbij hij de update aanroept. Voor de rest doet die helemaal niets in deze class.

```
[MenuItem("Window/UML Node")]
0 references
private static void ShowView()
{
    NodeUML editor = EditorWindow.GetWindow<NodeUML>();
    editor.Init();
}

1 reference
public void Init()
{
    controller = new NodeController();
    classHandler = new ClassHandler(this, controller);
    useCaseHandler = new UseCaseHandler(this);
}

0 references
void OnGUI()
{
    BeginWindows();
    classHandler.Update();
    //useCaseHandler.Update();
    EndWindows();
}
```

Dit is de hele editor window class er gebeuren hier niet heel veel interessante dingen.

```

1 reference
public ClassHandler(EditorWindow win, NodeController controller)
{
    this.controller = controller;
    this.win = win;
}

```

Waar ik nog nooit over nagedacht heb is dat je je hele editor window kan meegeven. dit is super handig en kan ik zeker gebruiken in mijn eigen project. Ik geef nu de hele tijd een rect mee waar mijn nodes binnen moeten blijven. Dus dit is echt een oplossing.

```

#if UML_NODE_DEBUG
//fill with some data
if (listNodes == null || listNodes.Count == 0)
{
    Debug.Log("Add some data");
    var n1 = new Node(new Rect(10, 10, NodeConsts.NodeWidth, NodeConsts.NodeHeight), "Win1", context);
    var n2 = new Node(new Rect(210, 210, NodeConsts.NodeWidth, NodeConsts.NodeHeight), "Win2", context);
    var n3 = new Node(new Rect(410, 310, NodeConsts.NodeWidth, NodeConsts.NodeHeight), "Win3", context);

    var f1 = new NodeInfoItem("1", n1, context);
    var f2 = new NodeInfoItem("2", n2, context);
    var f3 = new NodeInfoItem("3", n3, context);
    var f4 = new NodeInfoItem("4", n3, context);
    var f5 = new NodeInfoItem("5", n3, context);
    var f6 = new NodeInfoItem("6", n3, context);
    var f7 = new NodeInfoItem("7", n3, context);

    n1.AddField(f1);
    n2.AddField(f2);
    n3.AddField(f3);
    n3.AddField(f4);
    n3.AddField(f5);
    n3.AddField(f6);
    n3.AddField(f7);

    nodeRelation.AddRelation(n1.id, f1.ID, n2.id);
    nodeRelation.AddRelation(n3.id, f7.ID, n1.id);

    listNodes = new List<Node>()
    {
        n1, n2, n3
    };
}
#endif

```

what is the purpose of using #IF TEST instead of just if(TEST)?

TEST is not a variable, nor a constant. It doesn't even exist at run time. It is a flag passed to the compiler so it can decide on compiling some code (i.e putting it into the executable)

Geen idee wat hij hierboven doet dus ik ging dit onderzoeken. Ik weet wat preprocessor is, maar ik weet niet precies wat die hier doet aangezien ik het alleen ken zoals: #if UNITY_EDITOR.

Na wat onderzoek.

Het blijkt dat dit via een global define werkt. deze define zet je tussen cmcsc.bat. Voor de rest had ik een define manager editor tool gevonden maar helaas 404.

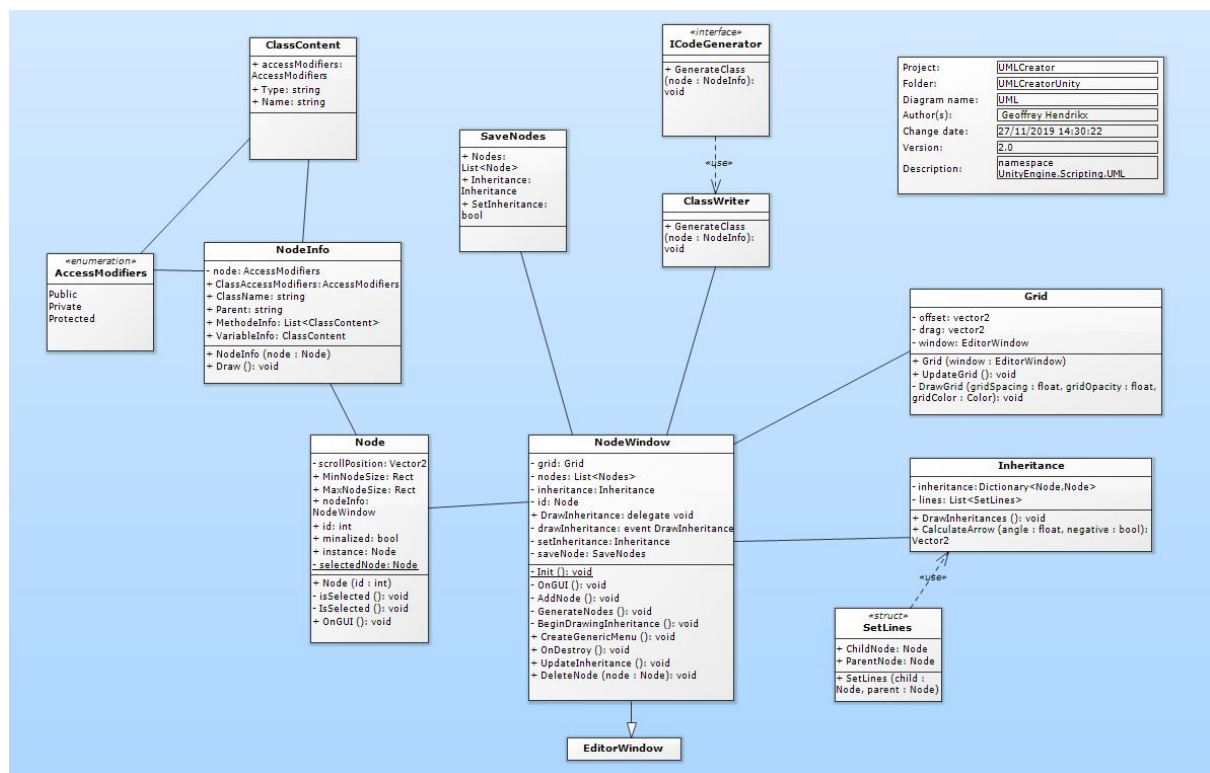
Nu ik weet wat de #if betekent ga ik door met mijn onderzoek.

Wat maakt dit minder goed of juist goed?

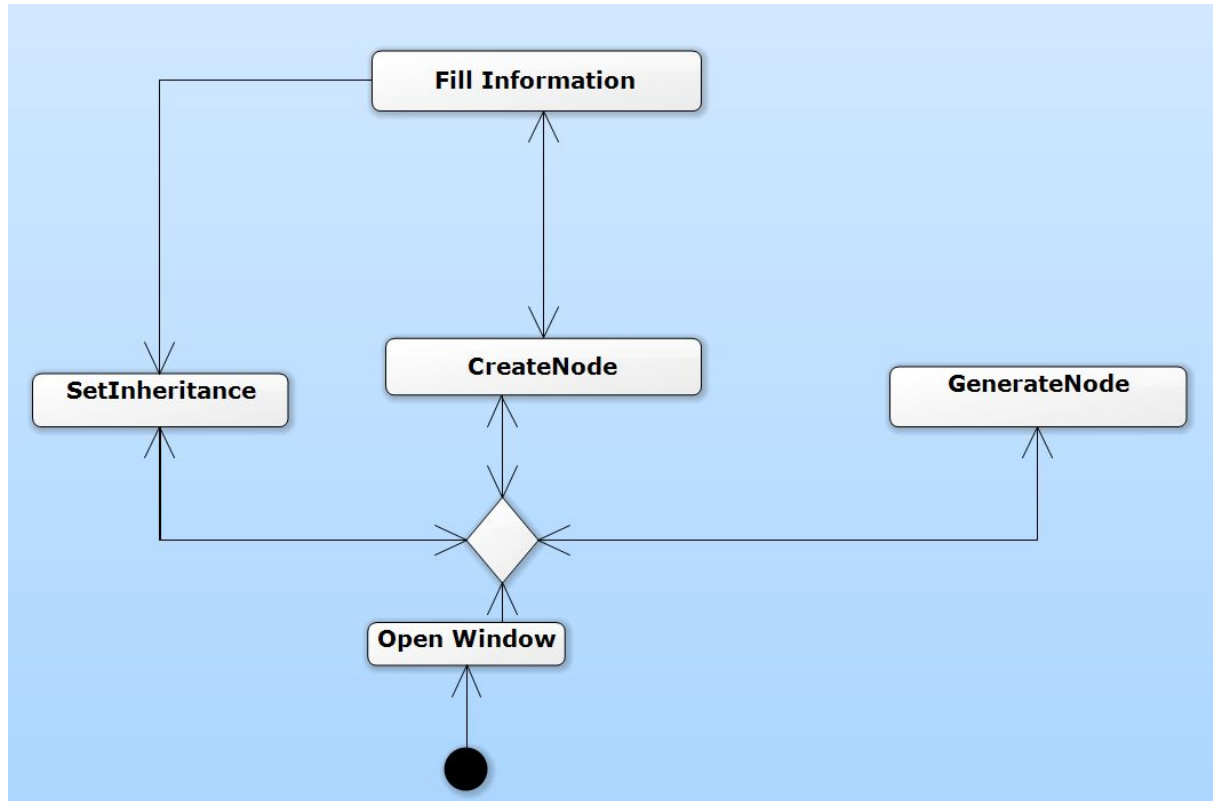
Deze tool is om een basis idee te krijgen van je code, maar het kan geen code voor je genereren wat ik met mijn tool wel wil bereiken. Deze tool kan wel save van classes en laden via json. dit wil ik eventueel ook nog bereiken.

UML

Class Diagram



Activity Diagram



Toelichting Ontwerp

Uml Generator is een tool die je in unity een UML laat maken. Het maakt het automatiseren van UML super makkelijk en het Uml kan je vervolgens laten genereren. Je kan het gelijk je script uitbreiden.

Tests

Wat ik mij het eerste afvroeg is of mensen erachter kwamen dat je een node moest maken met rechter muisknop. gelukkig wist iedereen dit erg snel te ontdekken en heb ik hier ook niet verder aandacht aan besteed. Mensen wisten op het begin de tool erg makkelijk te breken door te zeggen: Ik heb een inheritance en degene waarbij die erft verwijder ik.

Ik heb alle tests die iets lieten crashen opgelost. Op het laatst had ik een testpersoon die speciale tekens in het script had ingevuld. Dit handelde de compiler goed af door een error te geven dat dit niet in een script hoort.

Marciano Sordam
Job Elfrink,
Nathan Nieuwehuizen
Bas de Koningh

Reflectie

1. Handelen en ervaring:

Wat wil ik bereiken?

Ik wil bereiken dat je geen externe tool meer nodig hebt om een UML te maken.

Waar wil ik op letten?

Ik wil letten op het goed structureren van mijn code. Het splitsen van Window GUI en de functionaliteit is hierbij erg belangrijk.

2. Terugblikken:

Wat is er gebeurd?

Ik was niet tevreden met de vormgeving van mijn tool dus ben ik helemaal opnieuw begonnen. Oude code die nog wel bruikbaar was had ik nog altijd achter de hand.

3. **Bewustwording van essentiële aspecten:**

Wat vond ik in fase 1 en 2 belangrijk?

Het belangrijkste van fase 1 en 2 was dat ik in fase 1 de code niet goed had gesplitst.

De GUI code en de functionaliteit liepen door elkaar. Dit is de reden dat ik helemaal opnieuw ben begonnen met de tool in fase 2. Dit is het beste wat ik heb kunnen doen.

4. **Alternatieven ontwikkelen en daaruit kiezen:**

Tot welke voornemens en leerwensen leidt dat?

De volgende keer zorg ik dat ik de GUI code en functionaliteit bij een tool gelijk heb gesplitst.