

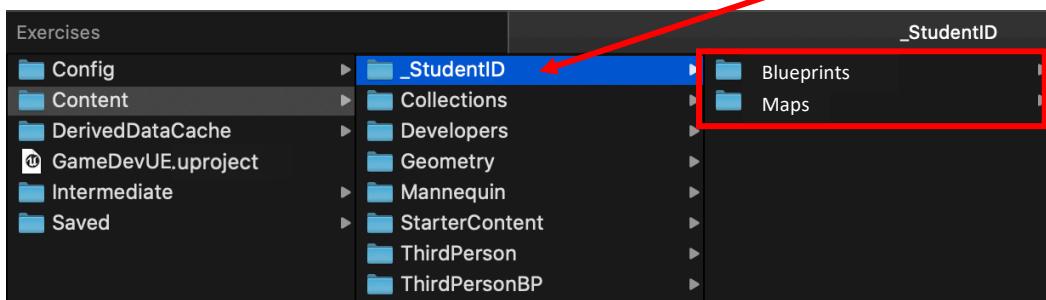
0) การเตรียม Project

0.1. ดาวน์โหลด GitHub Repository (ใช้ GitHub Desktop ได้) ซึ่งจะมีโครงสร้างดังรูป

<https://github.com/Game-Design-Dev-IT-KMITL/2020-GameDevWithUnreal-3.git>

0.2. สร้าง Branch

- สร้าง Branch (ใน GitHub Desktop) โดยใช้ชื่อ กลุ่มและรหัสนักศึกษา เช่น _Sec1_61xxxxxx
 - สร้างโฟลเดอร์ (ใน Unreal Engine) ใน Content สำหรับทำงานโดยใช้ชื่อเดียวกับ Branch และสร้าง sub-folders: Blueprints และ Levels ดังรูป



- งานทั้งหมดจะต้องทำอยู่ในไฟล์เดอร์ที่สร้างขึ้นใหม่เท่านั้น

0.3. การส่งงาน

- **Commit** และ **Push** ไปที่ Branch ที่นักศึกษาสร้างขึ้นเท่านั้น

<<< ห้ามส่งเข้า Main Branch >>>

ด้วย 50 คนนี้เป็น จ่าสัตว์ ห้าม congratulation

1) Custom Event, Macro, and Function

1.1. สร้าง Level, Blueprint และ Variable

1.1.1. สร้าง Level ใหม่ใน **Maps folder** เลือก Default และตั้งชื่อว่า **Map01**

1.1.2. สร้าง Blueprint ใหม่ใน **Blueprints folder** เลือก Parent Class เป็น **Actor** และตั้งชื่อว่า **BP_EV_MC_FN** → หมาย ให้นำมาใช้ 3 อย่างนี้ดูยังกัน

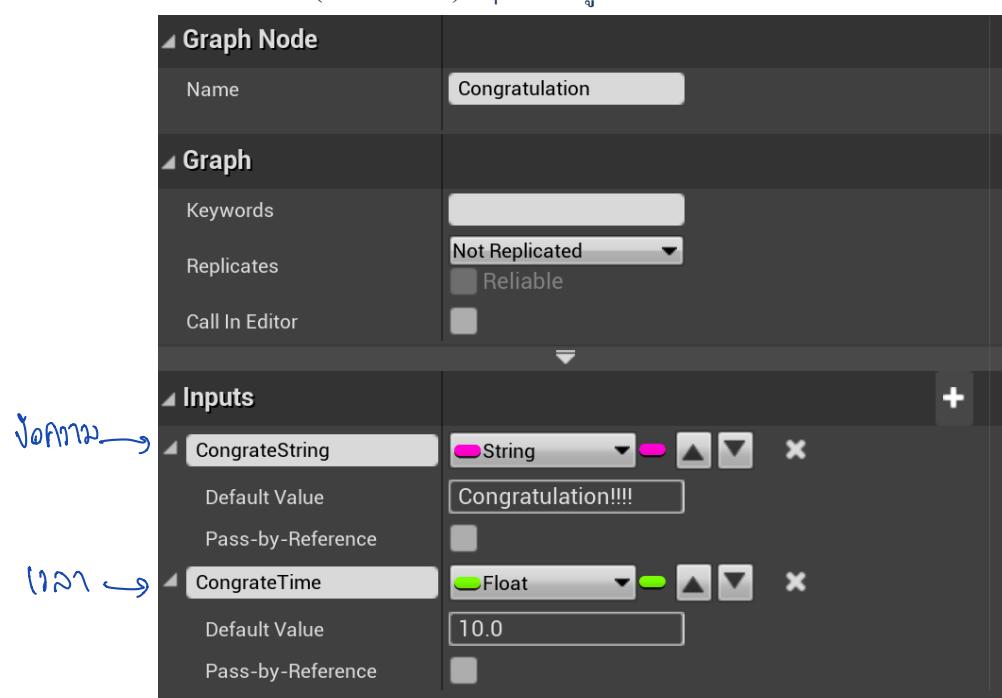
1.1.3. สร้าง Variables ดังนี้

- Grading ประเภท **Float** ค่า Default = 50
- Score ประเภท **Float** และ check ค่า **Instance Editable**

1.2. Custom Event

1.2.1. เปิด Blueprint **BP_EV_MC_FN**

1.2.2. สร้าง Custom Event ใหม่ (click ขวา เลือก Add Custom Event) ตั้งชื่อว่า **Congratulation**
แล้วทำการเพิ่ม (หน้า Detail) Inputs ดังรูป



ถ้าเราต้องการส่งค่า ที่เรียบร้อย

ให้เราสามารถสร้าง custom event เพื่อ กับบิจิก
โดยมีตัวไปร่วม กับไปร่วม
ที่เก็บข้อมูล str เท่า...
โดยกำหนด input ของ Default
อย่าง

ในการสร้าง custom
event ทุก str +
time เมื่อต้องการ

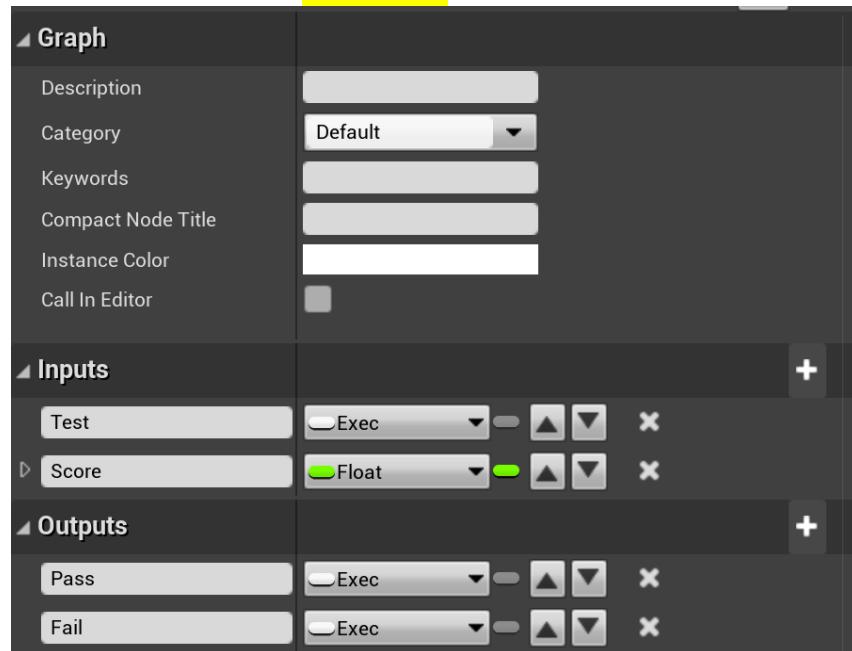
เวลาสร้าง event ต้องให้ต้อง ระบุว่า event นี้มี input อะไร ไม่ใน slide มอง มองว่า event ไม่มี input

คอมมอนของ input คือ แล้วเราจะใช้ event นี้มี input ให้เวลา มองว่า event นี้

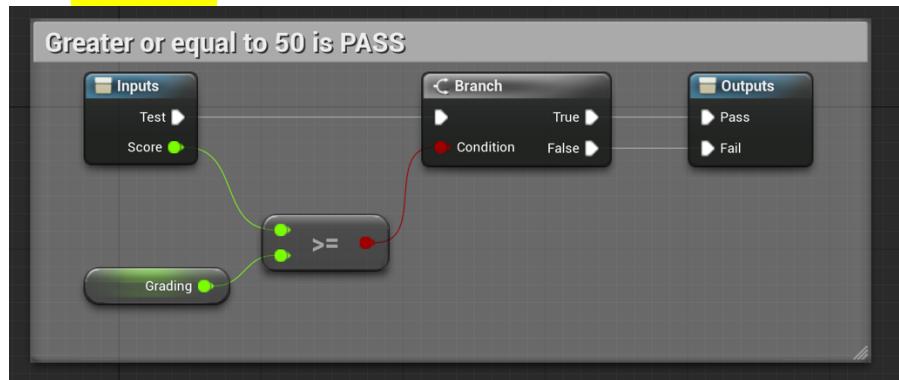
1.3. Macro

1.3.1. เปิด Blueprint BP_EV_MC_FN

1.3.2. สร้าง Macro ใหม่ ตั้งชื่อว่า Graduation และทำการเพิ่ม (หน้า Detail) Inputs/Outputs ดังรูป



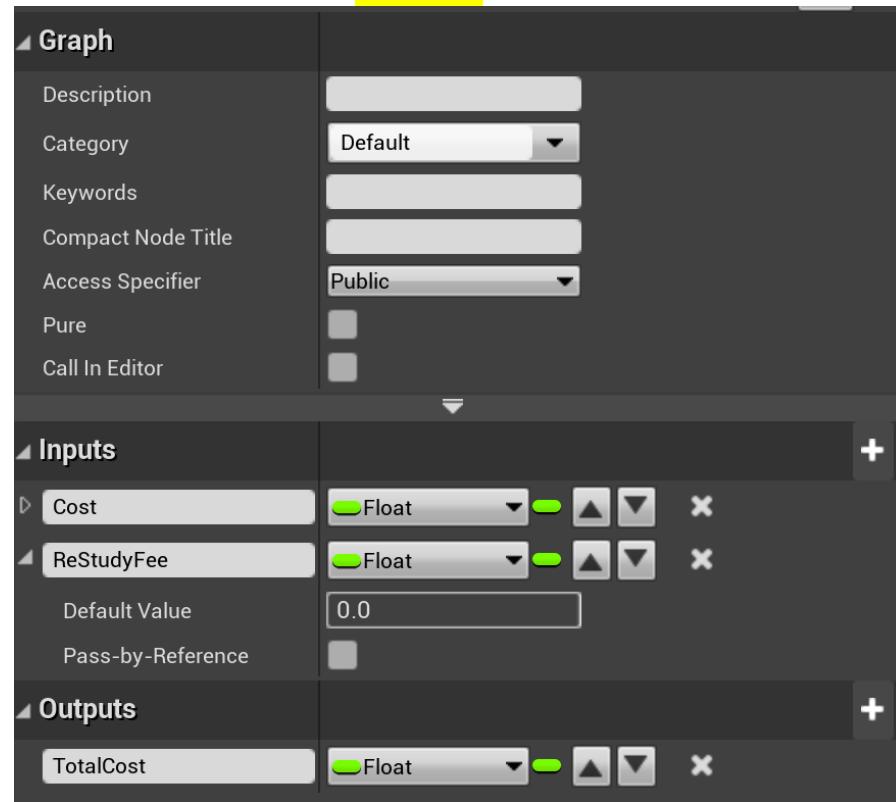
1.3.3. เปิด Graduation blueprint (double-click) และสร้าง Graph ดังรูป



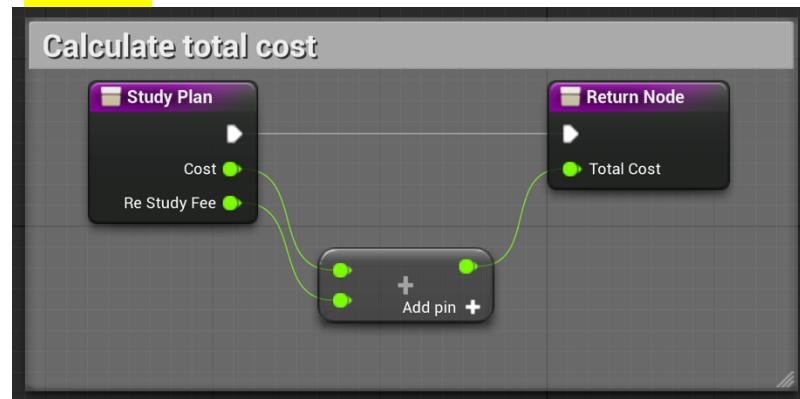
1.4. Function និង function ដែលបានគ្រប់គ្រង

1.4.1. เปิด Blueprint BP_EV_MC_FN

1.4.2. สร้าง Function ใหม่ តួន្យែថា StudyPlan แล้วทำการเพิ่ม (หน้า Detail) Inputs/Outputs

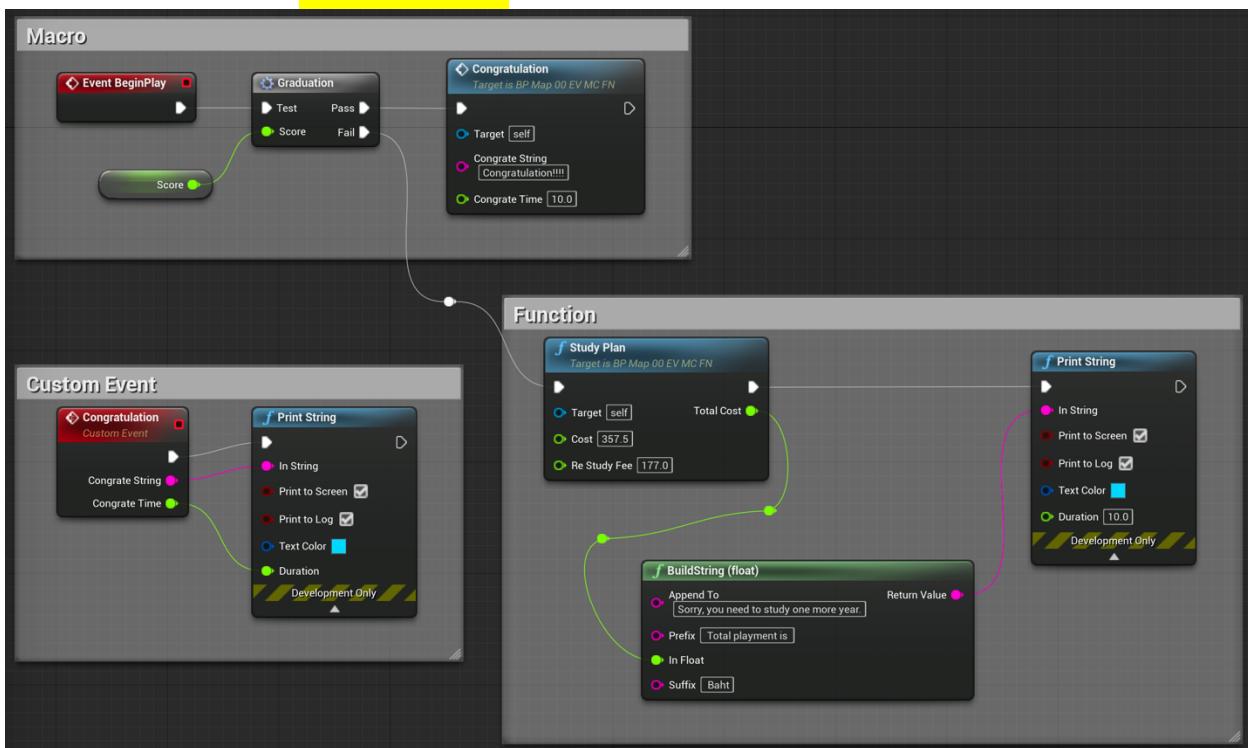


1.4.3. เปิด StudyPlan blueprint (double-click) และสร้าง Graph ចំណាំ



(๑) ๓ อย่าง ห้องนัก

1.5. เปิด Blueprint BP_EV_MC_FN และทำการสร้าง Graph ต่อไปนี้



1.6. ทำการ Play (Map01) เมื่อคำนวณค่า Score เพื่อทดสอบการทำงาน

- Score น้อยกว่า 50 จะแสดงข้อความ "Sorry, you have to study one more year. Total payment is 534.5 Baht."
- Score มากกว่า 50 จะแสดงข้อความ "Congratulation !!!!"

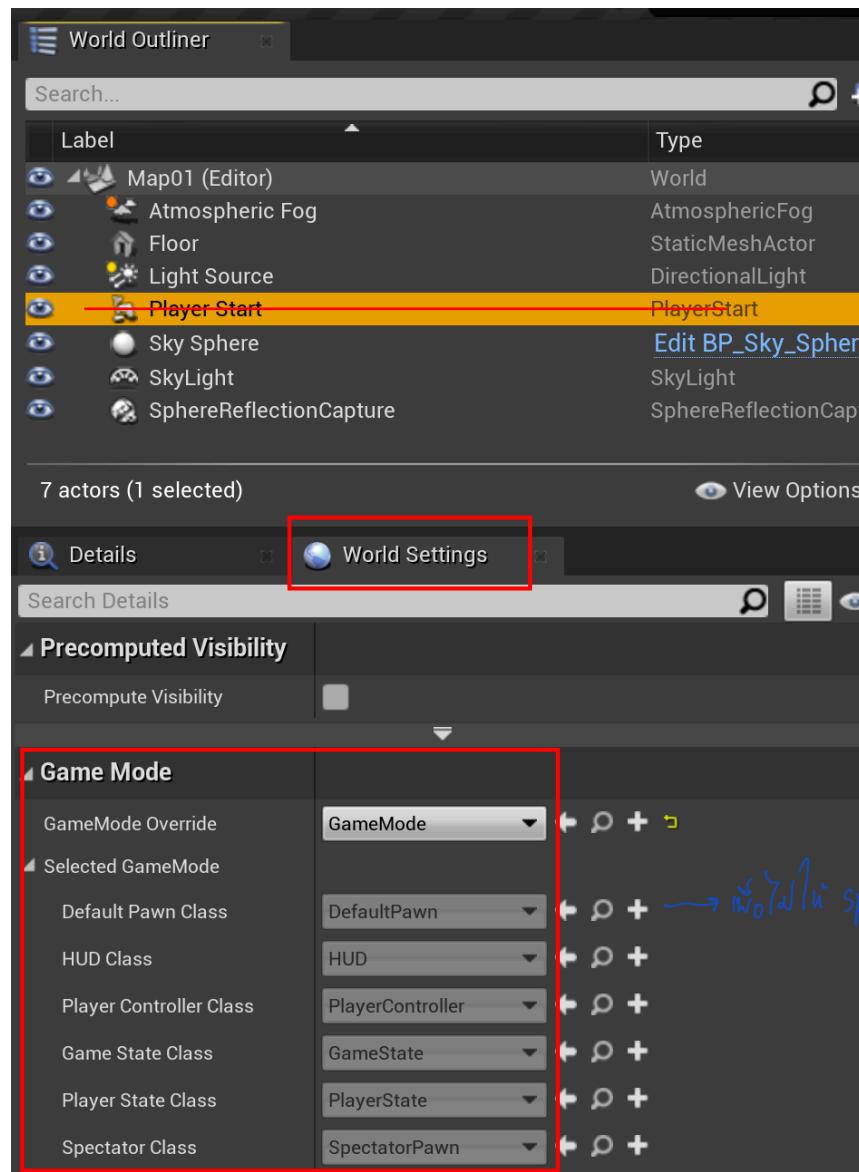
ที่ทุกๆ เทอมอยู่ 1 ลรร ไม่ผ่านกัน ไปตามรถ ใช้ร่วมกัน

2) Gameplay Framework Classes

2.1. Pawn's Possess

2.1.1. สร้าง Level ใหม่ใน Maps folder เลือก Default และตั้งชื่อว่า Map02

- ลบ Player Start จากหน้าต่าง World Outliner
- เปิดหน้าต่าง World Setting (ที่ Toolbar เลือก Setting → World Settings)
- เลือก Game Mode Override เป็น GameMode



- เมื่อทำการ Play จะไม่มี Player แสดงขึ้นมา เพราะ Default Pawn Class ถูก Override จาก Third Person Character เป็น Default Pawn

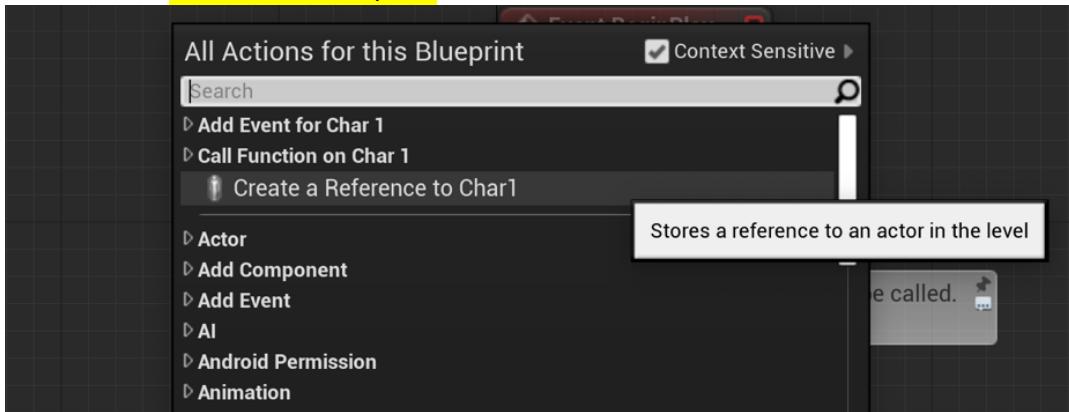
2.1.2. เพิ่ม Player จำนวน 2 ตัว ลงใน Level ดังรูป

- ThirdPersonCharacter จาก ThirdPersonBP\Blueprints folder ตั้งชื่อว่า Char1
- BP_ThirdP_Mannequin จาก Female_Mannequin\Blueprints folder ตั้งชื่อว่า Char2



2.1.3. เปิด Level Blueprint (ที่ Toolbar เลือก Blueprints → Open Level Blueprint)

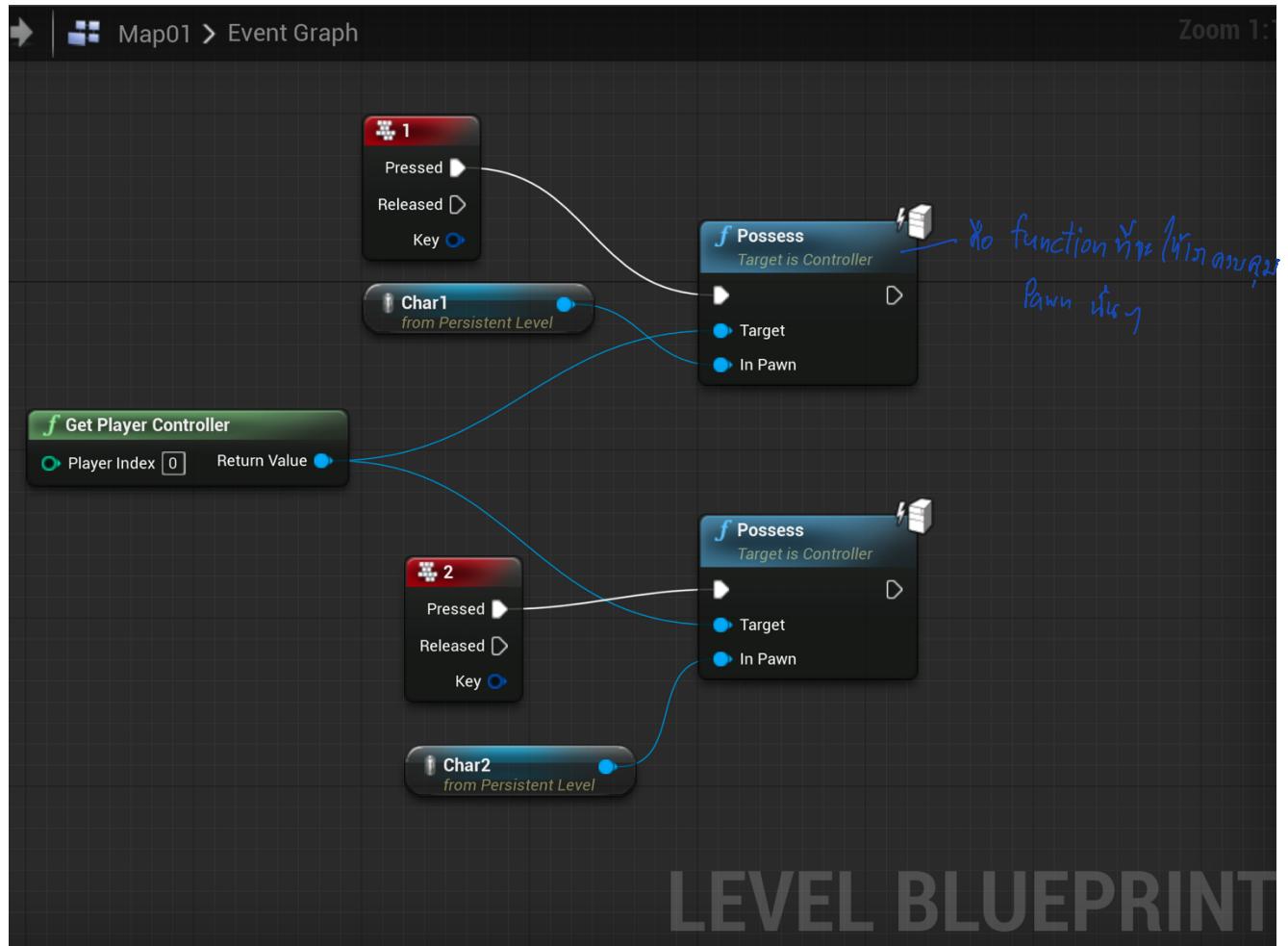
- ที่หน้า Level กดเลือก Char1 ในหน้าต่าง World Outliner
- ที่หน้า Level Blueprint กด click ขวา และเลือก Create a Reference to Char1 ดังรูป



- ทำข้อสำหรับ Char2
- เมื่อทำการ Play จะไม่สามารถควบคุม Player ได้ เพราะ Default Pawn Class ถูก Override จาก Third Person Character เป็น Default Pawn

2.1.4. ในหน้า Level Blueprint ทำการเพิ่ม Get Player Control และลาก Output เพื่อทำการเพิ่ม Possess function แล้วสร้าง Script ดังรูป (เพื่อเลือก Character ในการควบคุม)

- กด 1 เพื่อควบคุม Char1
- กด 2 เพื่อควบคุม Char2

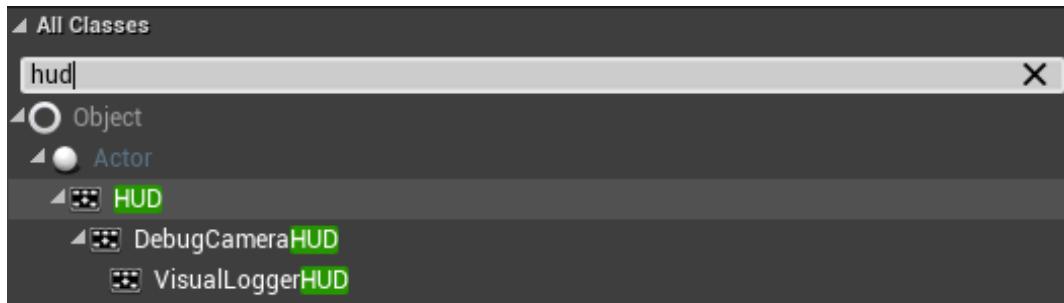


2.1.5. Play โดย แล้วทำการ กด 1 หรือ 2 เพื่อสลับการควบคุม Player แต่ละตัว

2.2. HUD Class → Tuples print String representations show how various terminals support them

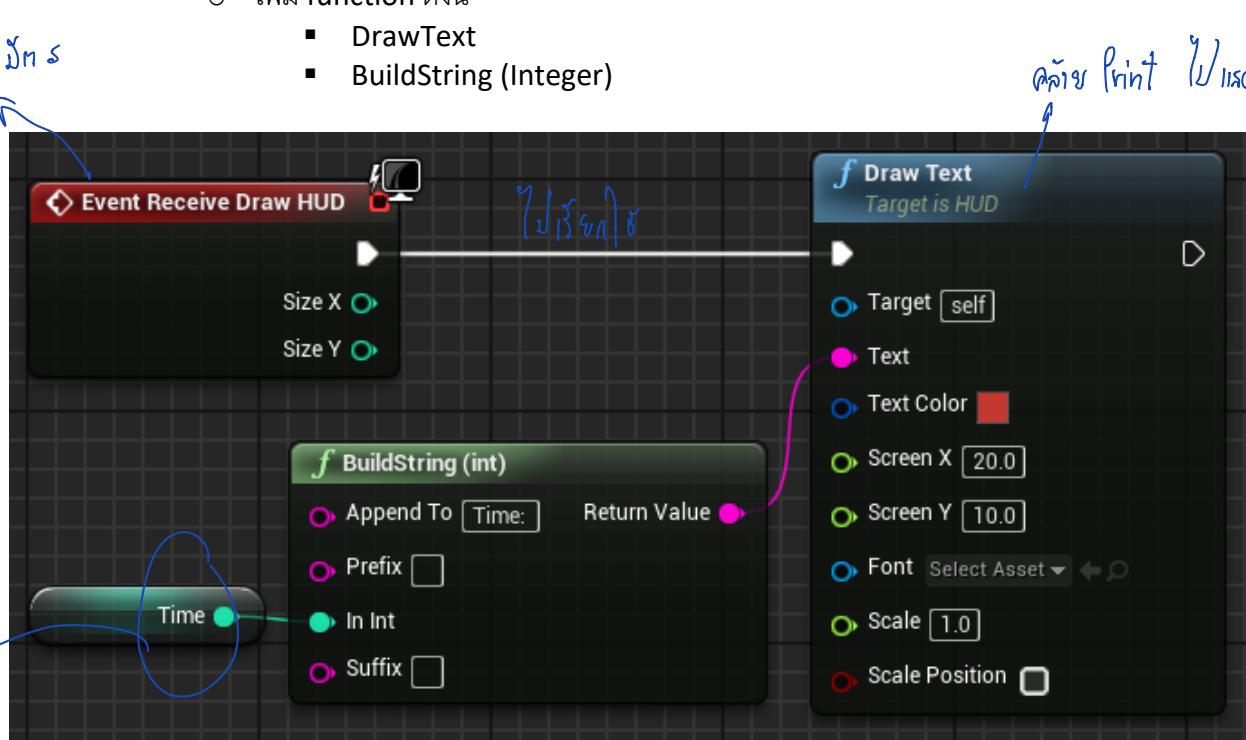
2.2.1. สร้าง Level ใหม่ใน Maps folder เลือก Default และตั้งชื่อว่า Map03

2.2.2. สร้าง Blueprint ใหม่ใน Blueprints folder เลือก Parent Class เป็น **HUD** (ขยายจาก All Classes) และตั้งชื่อว่า **BP_HUD_Timer**



2.2.3. เปิด BP_HUD_Time และสร้าง Graph ดังรูป

- สร้างตัวแปร Time ประเภท Integer
 - เพิ่ม Receive Draw HUD event
 - เพิ่ม function ดังนี้
 - DrawText
 - BuildString (Integer)



HUD

ໂຄງການທີ່ຕອນ ເວມຈາກ ຕົ້ນແປສ Time ແລ້ວຕົ້ນປັບປຸງໃຫຍ່ Time ເພີ່ມໃຫຍ່ທຳມະນີ

in f-set Timer by Event

T₁ ↗ man Count (Custom Event)

ស្ថិតិមាលា count នូវ រឹងក់ នៅក្នុងក្រុមហ៊ុន ឬ នៃ ការបន្ទាន់លើក

លេង ទោ ព័ត៌មាន set ពីរ ឬក្រោម time

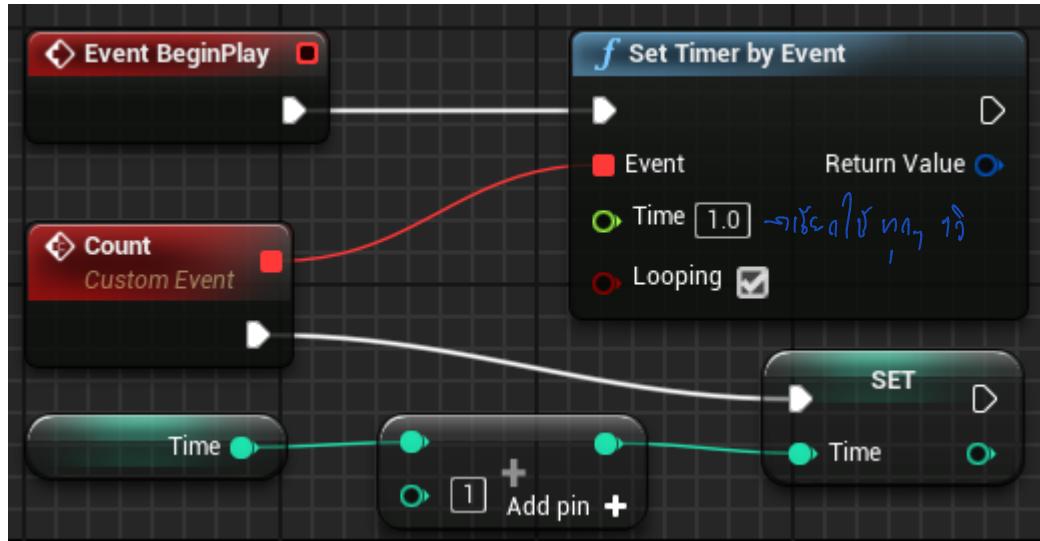
Week 09 (07 Oct' 20)

Game Design and Development

Room 205

2.2.4. ត្រូវការគុណកម្មនៃពេលវេលា ដំឡូង

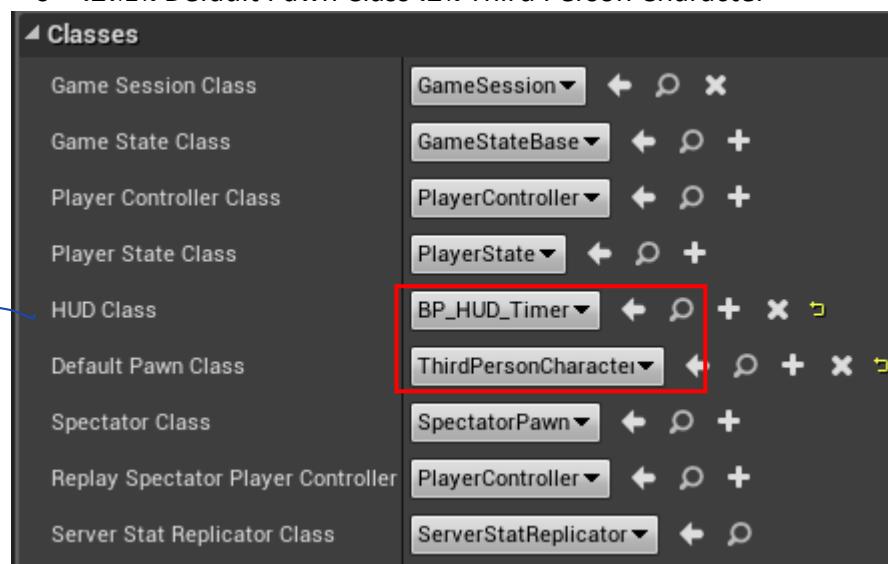
- ធំន Add Custom event តួចខ្លួនថា Count
- ធំន Set Timer by Event function



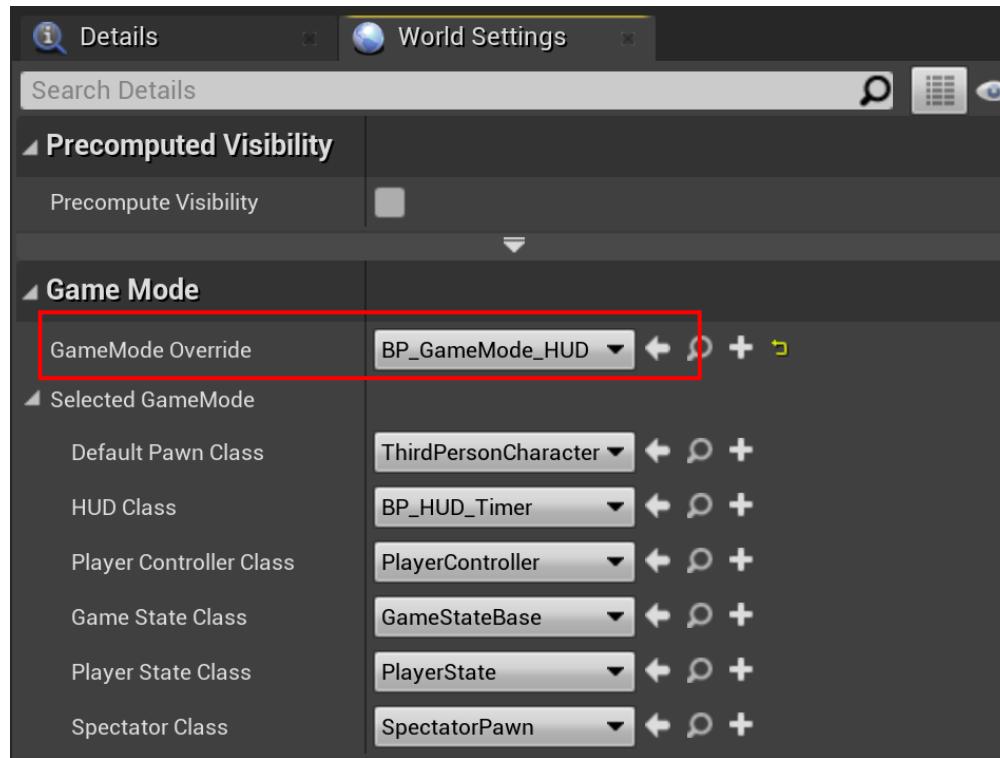
2.2.5. ត្រូវ Blueprint ឲ្យមែន **Blueprints folder** តើក Parent Class មែន **Game Mode Base** និង តួចខ្លួនថា **BP_GameMode_HUD**

2.2.6. បើក **BP_GameMode_HUD** និងកត់តើក Class Defaults ពី Toolbar និងការការពារគ្នា ដំឡូង

- ផ្សេងៗ HUD Class មែន **BP_HUD_Time**
- ផ្សេងៗ Default Pawn Class មែន **Third Person Character**



- 2.2.7. ในหน้า Level กดเลือกหน้าต่าง World Settings แล้วทำการเลือก **BP_GameMode_HUD** ในช่อง **GameMode Override** ดังรูป



- 2.2.8. เมื่อทำการ Play จะปรากฏค่าเวลาเริ่มนับ ตั้งแต่ 1 และจะเพิ่มค่าขึ้นไปเรื่อยๆ ทุกๆ วินาที ที่มุ่งบนช้ายของ **หากตัวเลขมีขนาดเล็ก สามารถทำการเปลี่ยนค่า Scale หรือ เปลี่ยน Font ใน Draw Text function ได้ (ข้อ 2.2.3)**

3) Simple Game

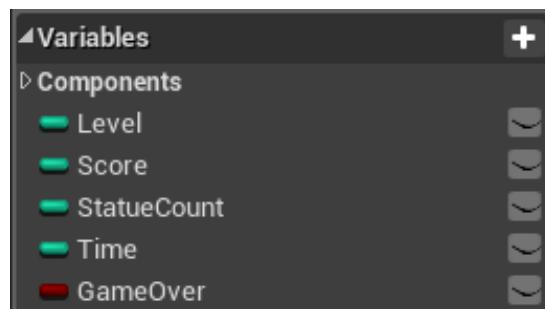
3.1. สร้าง Level ใหม่ ชื่อ MapSimpleGame

- เลือก Default
- เปลี่ยนขนาด (Scale) ของ Floor เป็น 2:2:1 (X:Y:Z)

3.2. สร้าง Blueprint ชื่อ BP_GameMode เลือก Parent Class เป็น Game Mode Base

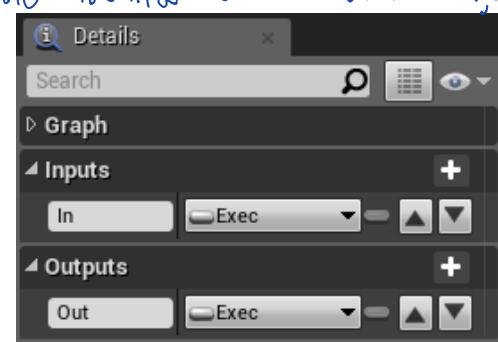
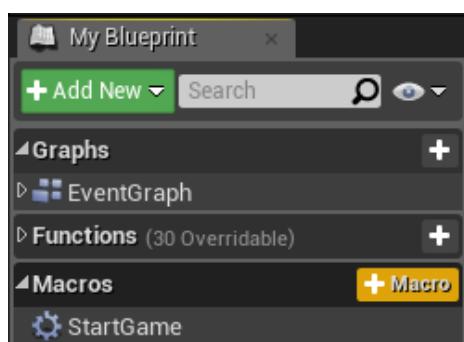
3.2.1. เปิด BP_GameMode แล้วสร้าง Variable ต่อไปนี้

- Level ประเภท Integer เก็บข้อมูล Current Level ของ Player
- Score ประเภท Integer เก็บข้อมูล Score ของ Player
- StatueCount ประเภท Integer เก็บข้อมูล จำนวนรูปปั้นที่เก็บได้
- GameOver ประเภท Boolean เก็บข้อมูลสถานะของการเล่นเกม



3.2.2. สร้าง Macro ชื่อ StartGame ดังรูป

StartGame จะทำหน้าที่ในการ Initializing ค่า Variables ต่าง ๆ ในเกมที่ใช้ในการควบคุม State ต่าง ๆ

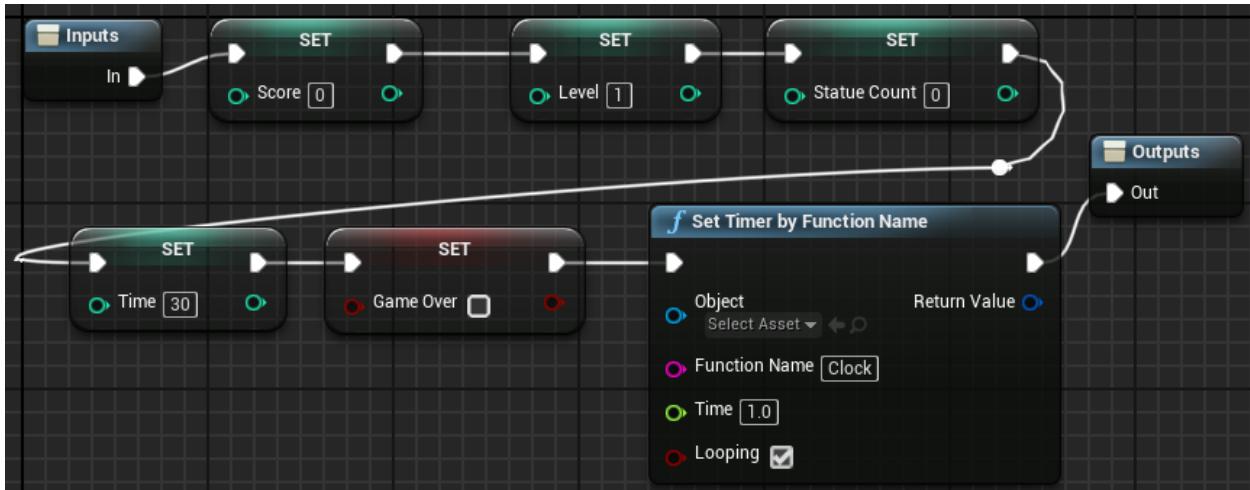


↑ ต้องเริ่มจาก set ตัวต่อไปนี้เป็นตุนๆ , set level 1
, set สีหน้าจอ
ให้ = 0
) set ใจ = 30
) set Game Over
) ลบ show function
ในหน้าจอ

ถ้าไม่กำหนดตัวต่อไปนี้ตั้งแต่แรก宏 Macro

- เปิด StartGame macro (double-click) และทำการสร้าง Graph ดังรูป

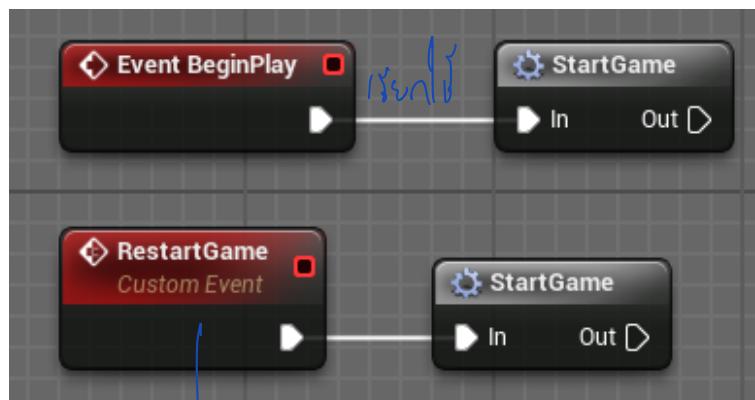
Set Timer คือ Function ที่สร้าง Timer Object ที่จะเรียกใช้ Custom Event ชื่อ Clock ทุก ๆ วินาที



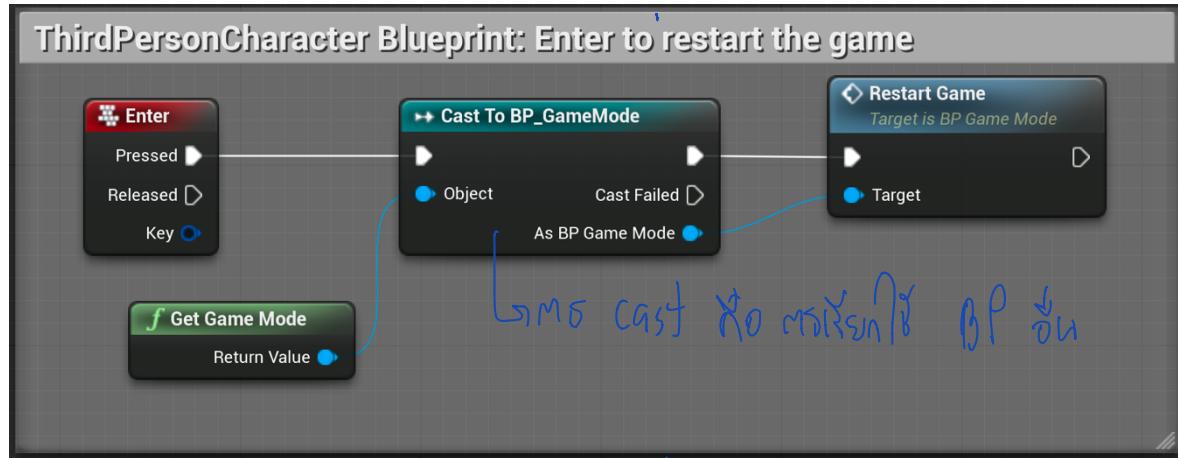
3.2.3. เรียกใช้งาน StartGame Macro

- ใน BP_GameMode blueprint ดังรูป

Macro (StartGame) จะถูกเรียกใช้ด้วย Event BeginPlay และ Custom Event ชื่อ RestartGame ซึ่งจะถูกเรียกใช้ใน ThirdPersonCharacter blueprint

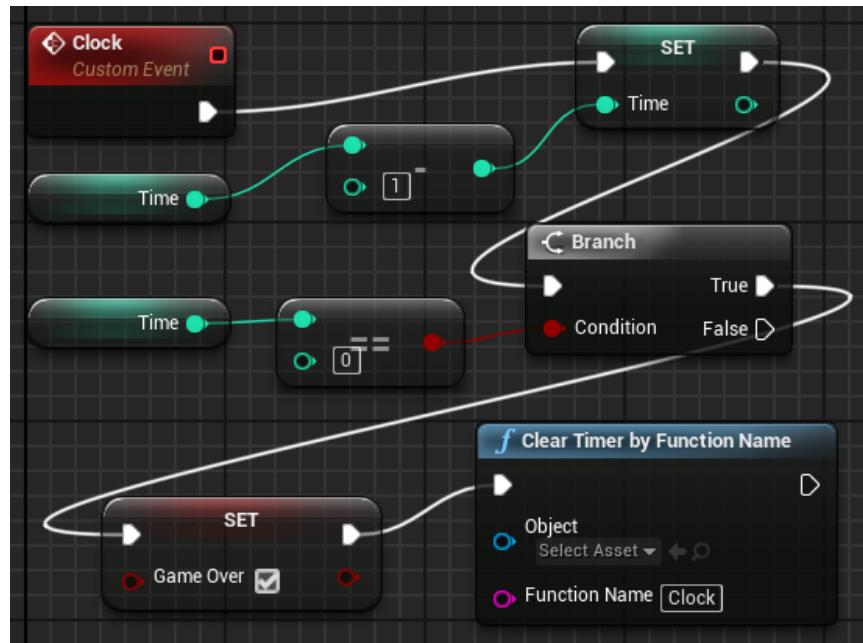


- ใน ThirdPersonCharacter blueprint ดังรูป



3.2.4. สร้าง Custom Event ชื่อ Clock ดังรูป → ไม่ต้อง เก็บ ค่า Time

Clock คือ Custom Event ที่จะถูกเรียกใช้งาน โดยจะถูกลดค่าลง ทีละ 1 หน่วย (วินาที) และเก็บค่าไว้ในตัวแปร Time
เมื่อ Time มีค่าเป็น 0 ค่า GameOver (Boolean) จะถูกกำหนดเป็น True และจะทำการ Clear Timer

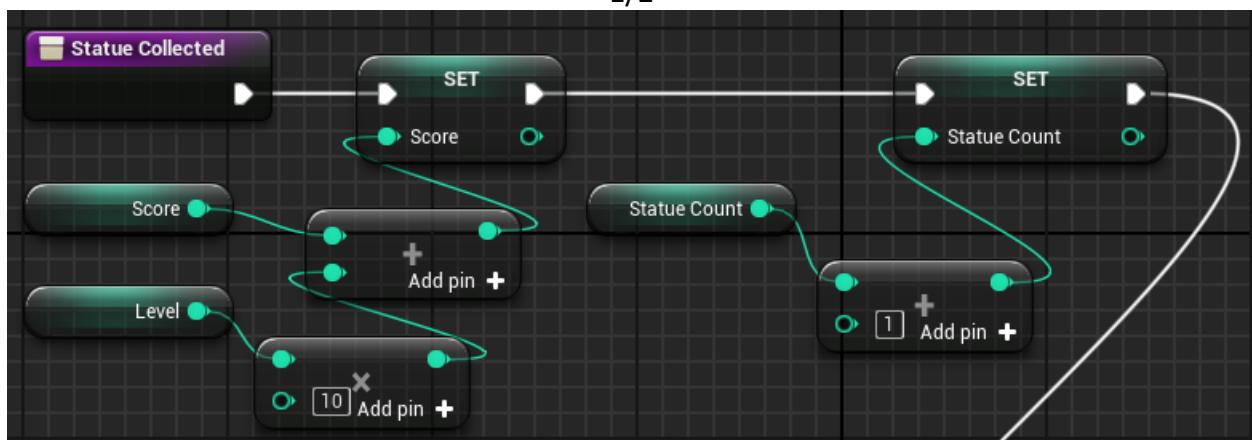


3.2.5. สร้าง Function ชื่อ StatueCollected ดังรูป $\rightarrow f_{\text{รูปแบบที่เราต้อง}}$

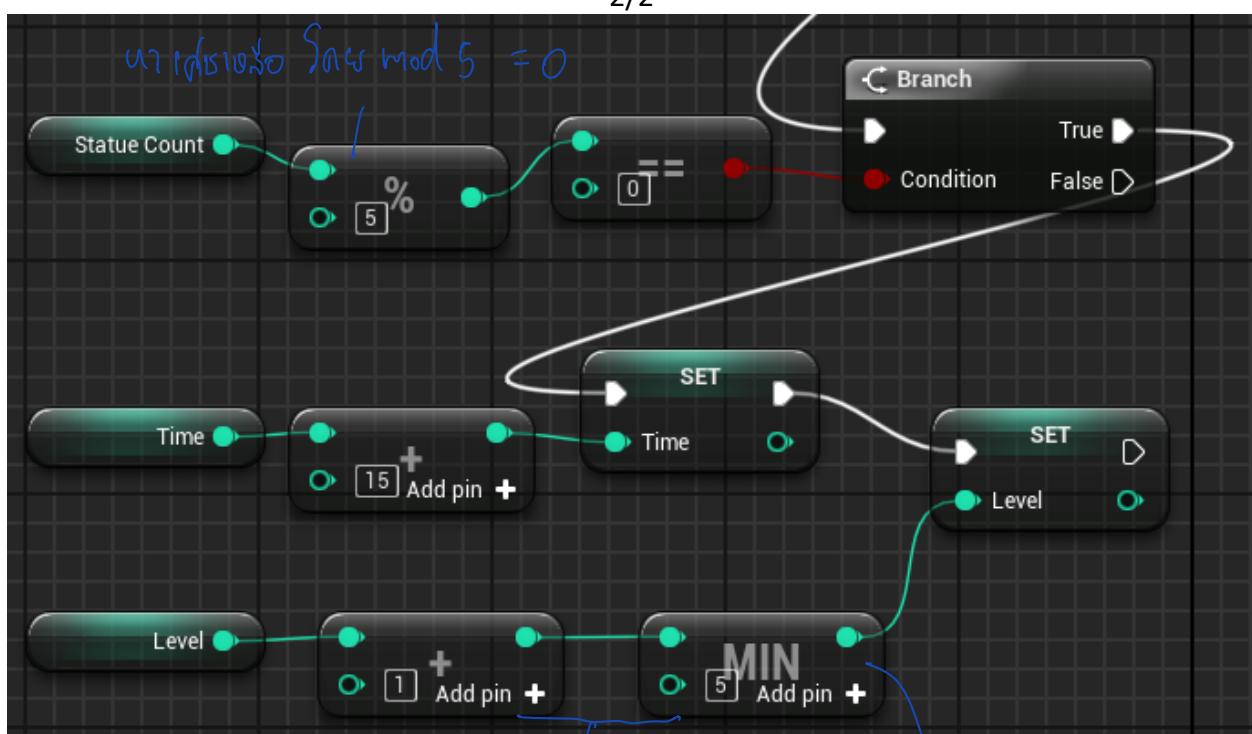
StatueCollected คือ Function ที่จะถูกเรียกให้เมื่อ Player ทำการเก็บ Statue ได้

- เมื่อเก็บ Statue ได้ 5 ตัว จะผ่านด่าน
- เพิ่มค่า Score เป็น $10 \times \text{Level}$ ต่อ Statue จำนวน 1 ตัว
- เพิ่มค่า Statue ที่ลํา 1
- เพิ่ม Time จากค่าปัจจุบัน 15 วินาที เมื่อเล่นผ่านด่าน
- เพิ่ม Level ที่ลํา 1 สูงสุด ไม่เกิน level 5

1/2



2/2



check มากไป 5
ถ้า min มากกว่า 5
Level 1 $\rightarrow 1$
 $6 \rightarrow 5$ (อย่างต่ำ 5)

3.3. สร้าง Blueprint ชื่อ BP_HUD เลือก Parent Class เป็น **HUD**

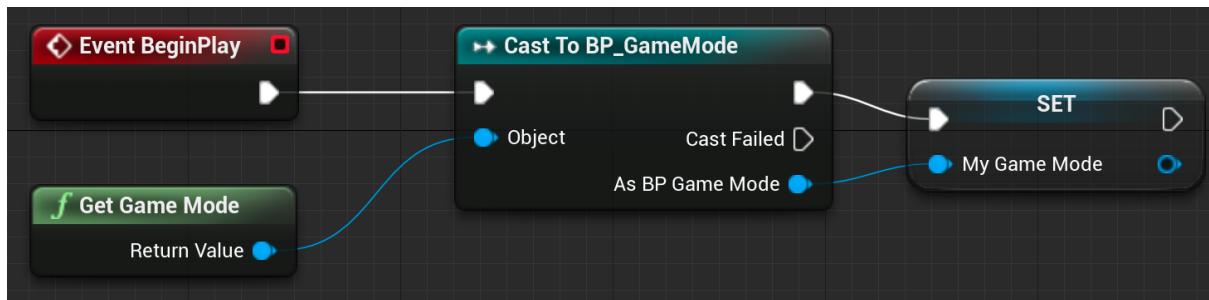
3.3.1. เปิด BP_HUD blueprint

3.3.2. สร้าง Variable ดังนี้

- My_GameMode ประเภท BP_GameMode ชนิด Object Reference

3.3.3. สร้าง Graph ดังรูป

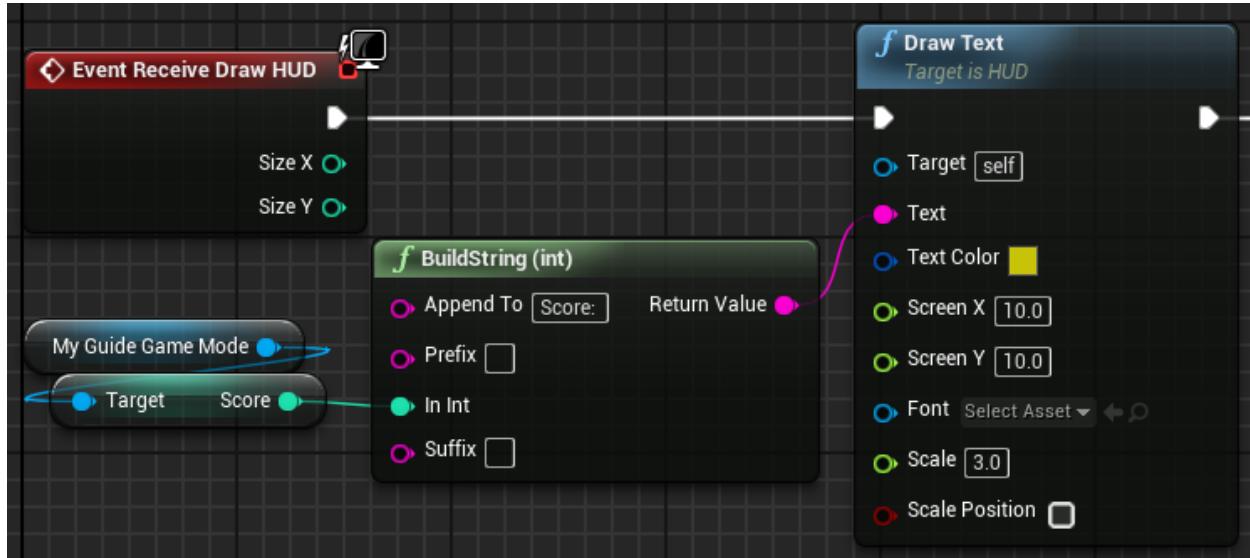
Casting คือการเรียกใช้ Blueprint อื่นจาก BP_HUD แล้วเก็บค่า Reference ในตัวแปร My_GameMode



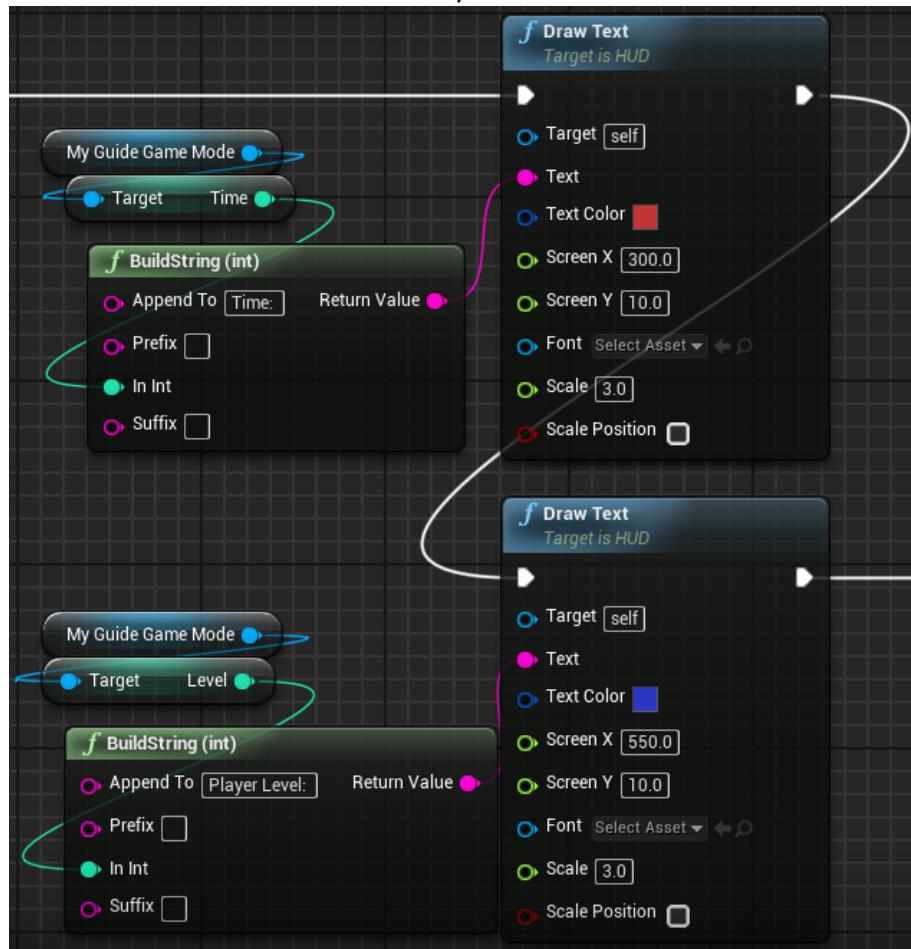
3.3.4. สร้าง Graph ต่อไปนี้ (ใน BP_HUD) เพื่อการแสดงผลบนหน้าจอ

การแสดงผล HUD (Score, Time, Level) ของเกมในตำแหน่งที่กำหนดไว้

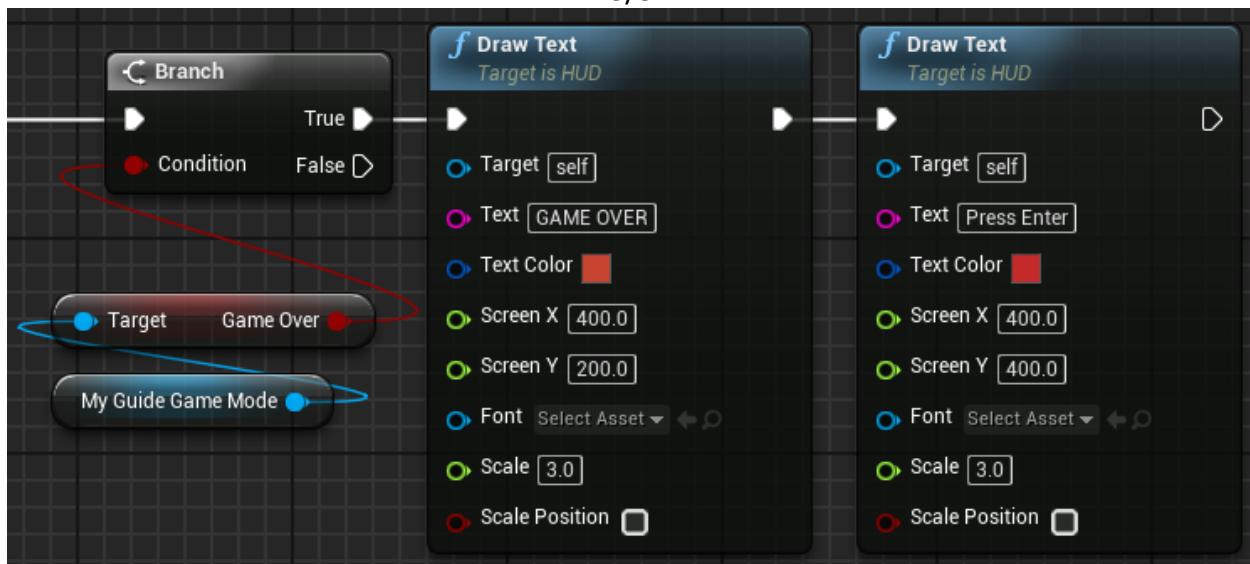
1/3



2/3



3/3



3.4. สร้าง Blueprint ชื่อ BP_Status เลือก Parent Class เป็น Actor

→ โครงสร้างปูนหิน

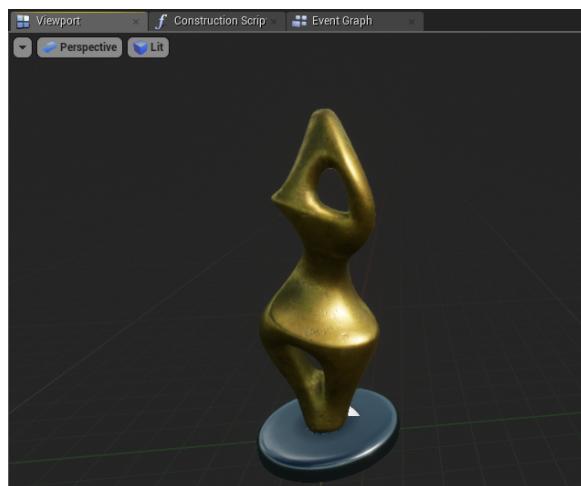
3.4.1. เปิด BP_Status blueprint

3.4.2. สร้าง Variable ดังนี้

- My_GameMode ประเภท BP_GameMode ชนิด Object Reference

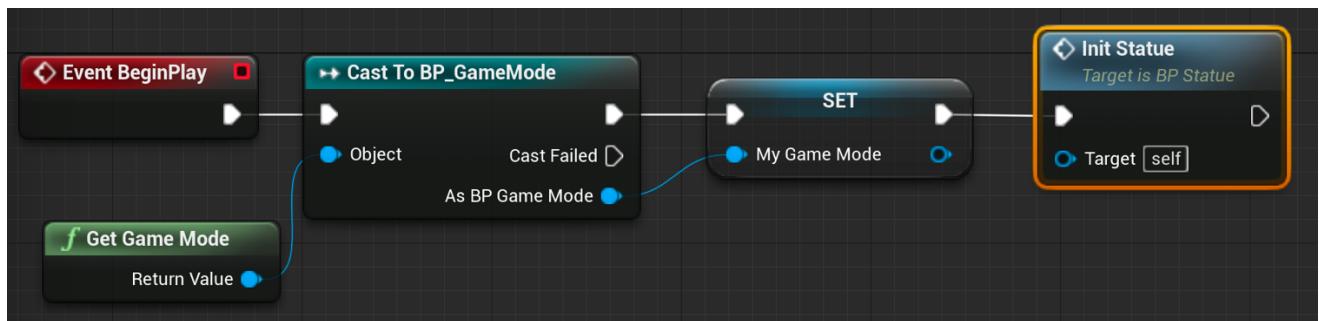
3.4.3. เพิ่ม StaticMesh เลือก SM_Status (หรืออื่นๆ ที่ชอบจาก Starter Content)

- กำหนดค่า Element 0 ใน Material เป็น M_Metal_Gold (หรืออื่นๆ ที่ชอบ)
- กำหนดค่า Collision Presets เป็น OverlapAllDynamic
- เพิ่ม BP_Status จำนวน 3 ชิ้น ลงใน Level (วางในตำแหน่งที่ต่างกัน)



3.4.4. สร้าง Graph ดังรูป

ทำการสร้าง Reference ไปยัง BP_GameMode blueprint และเก็บในตัวแปร My Game Mode และทำการ execute Init Statue function



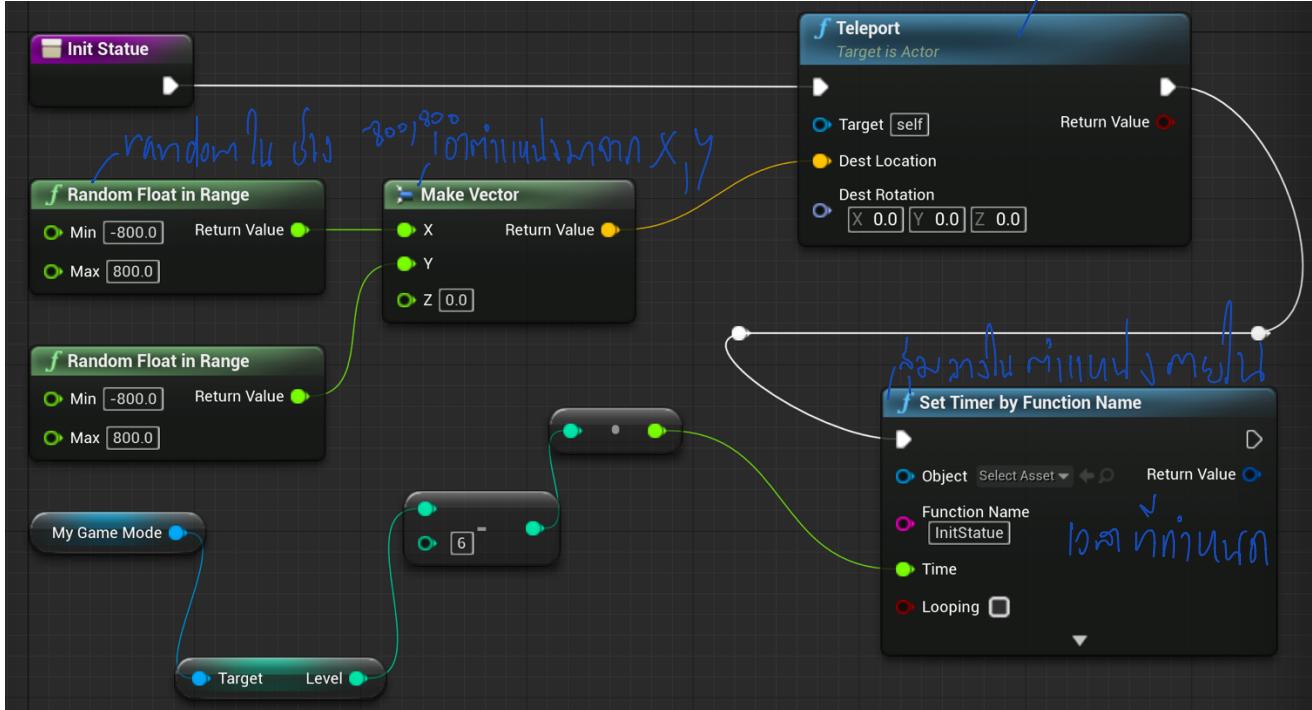
3.4.5. สร้าง Function ชื่อ Init Statue

→ ស្ថិតិ នូវ រូបរាង

- Teleport គឺការ location ใหม่ໃຫ្យក្នុង Statue

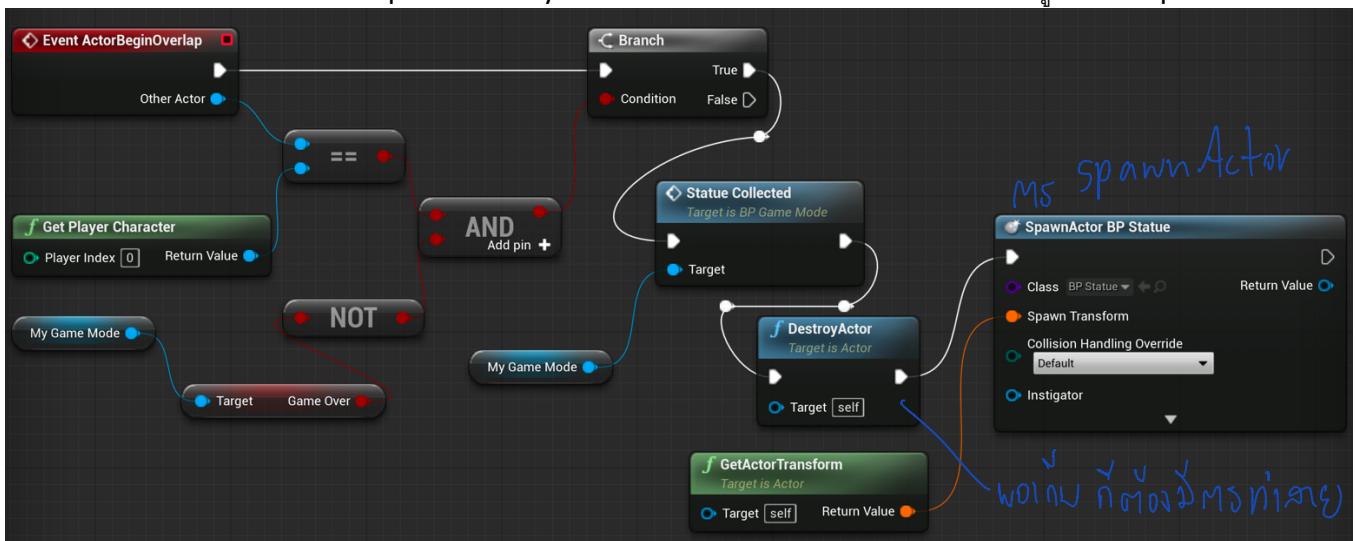
- Set Time by Function Name គឺ function ទាំងអស់ដែលត្រូវបានការពារសម្រាប់បញ្ចប់
ឱ្យលើក្នុង Statue ដើម្បីការពេញលេខការការពារជាប្រភព 6-level

ត្រូវបានការពារសម្រាប់បញ្ចប់
ឱ្យលើក្នុង Statue ដើម្បីការពេញលេខការការពារជាប្រភព 6-level

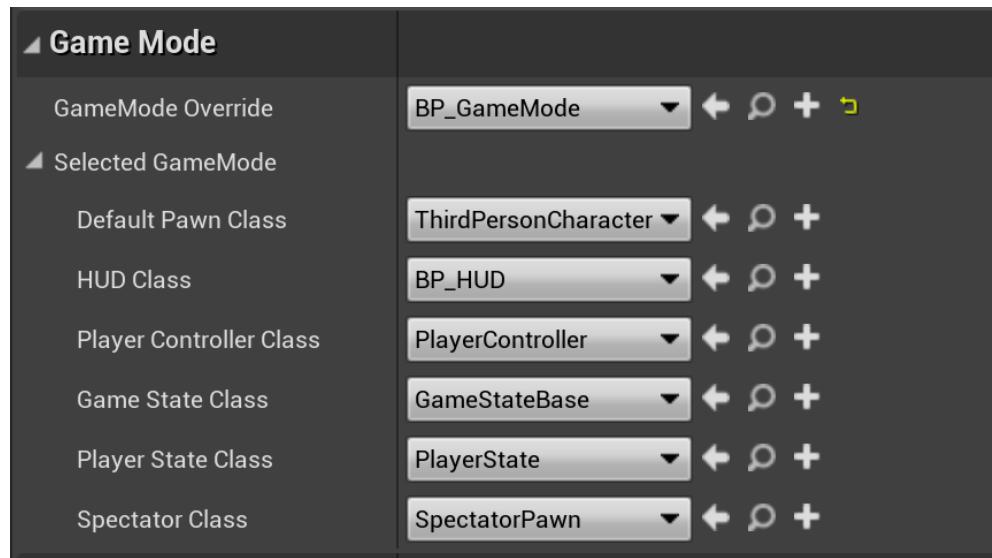


3.4.6. สร้าง Graph ដំឡើង (នៃ BP_Status blueprint)

ដើម្បីការ Overlap រវាង Player ก្នុង Statue ក្នុងលទ្ធផលនៃ Overlap នឹង



3.5. ตั้งค่า Game Mode เพื่อทำการ Override Project Setting สำหรับใช้ค่าจาก BP_GameMode



3.6. Play แล้วทดสอบการเล่น