

Game Design & Development

Samart Moodleah, PhD. IT@KMITL, 2020

Artificial Intelligence: Behavior Trees

Week-12

Topics and Objectives

1. Behavior Trees (BT) Overview
2. Behavior Tree Editor — စောင့်ဆင်သွေးချေမှုများအတွက် AI

Reference: <https://docs.unrealengine.com/en-US/Engine/ArtificialIntelligence/index.html>

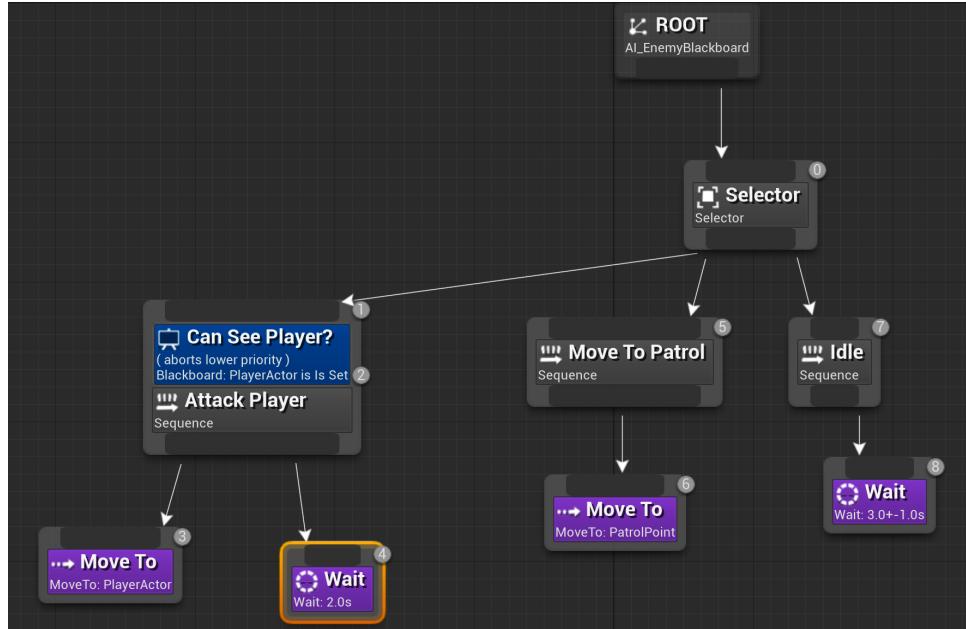
AI In UE

Artificial Intelligence systems in UE 4

1. Behavior Trees (BT) - โครงข่าย (ปั้นแบบ)
ตัวเลือกที่เกิดจากการตัดสินใจ หรือ การกระทำต่าง ๆ ในเกม
เพื่อให้บรรลุเป้าหมายที่ตั้งไว้
2. Environment Query System (EQS) (*Experimental*) (สำรวจ周遭)
การจัดการเกี่ยวกับข้อมูลของสภาพแวดล้อมต่าง ๆ ในเกม
3. AI Perception - ทรรศน์ของ AI ผ่าน Sensor
การรับรู้ข้อมูลผ่าน Sensor ด้านต่าง ๆ (เส้น ภาพ หรือ เสียง) ของ AI
4. AI Debugging
เครื่องมือในการวิเคราะห์ข้อมูลต่าง ๆ ที่ AI ใช้ในการทำงาน

Behavior Tree (BT) Overview

ເກມແກ່ວດຕົນໄສ້ມີ ວິຊາເລືອກທຸກຍົງ



Basics of Behavior Trees

- ใช้ Nodes แบบเดียวกับ Blueprints
 - Blueprints (BP) ใช้ Event Graph
 - Behavior Trees (BT) ใช้ Behavior Tree Graph
- BT จัดการเกี่ยวกับ Logic ต่าง ๆ
 - ในส่วนของข้อมูลเกี่ยวกับ BT จะถูกเก็บในไฟล์ Blackboard (Blackboard Keys)
 - และ BT จำเป็นต้องรู้ข้อมูลเหล่านี้ในการตัดสินใจต่าง ๆ

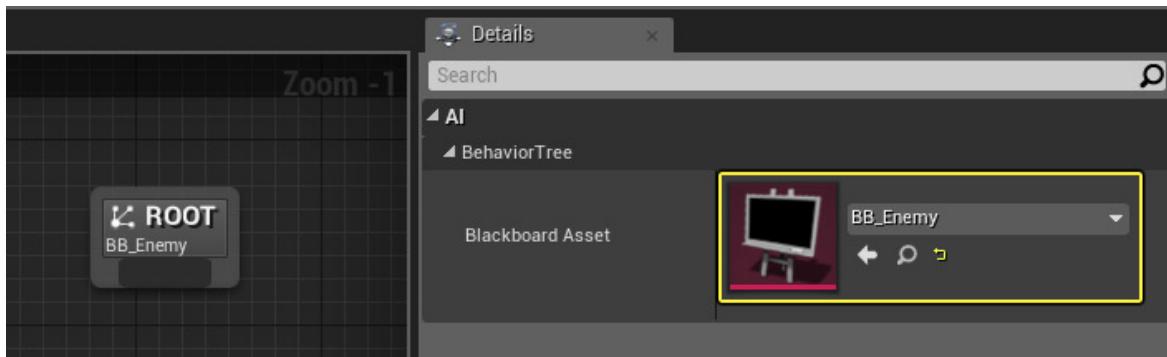
บล็อกเกอร์

BT กับ BB ทำงานร่วมกัน

กำหนดตัวแปร กำหนดตรรกะ
ทรัพยากร

Flow การทำงาน

- สร้างไฟล์ Blackboard
- สร้างข้อมูล Blackboard Keys
- สร้าง BT ซึ่งต้องใช้ Blackboard asset



ลำดับการทำงาน (Execution) ของ BT

- ซ้ายไปขวา และ บนลงล่าง

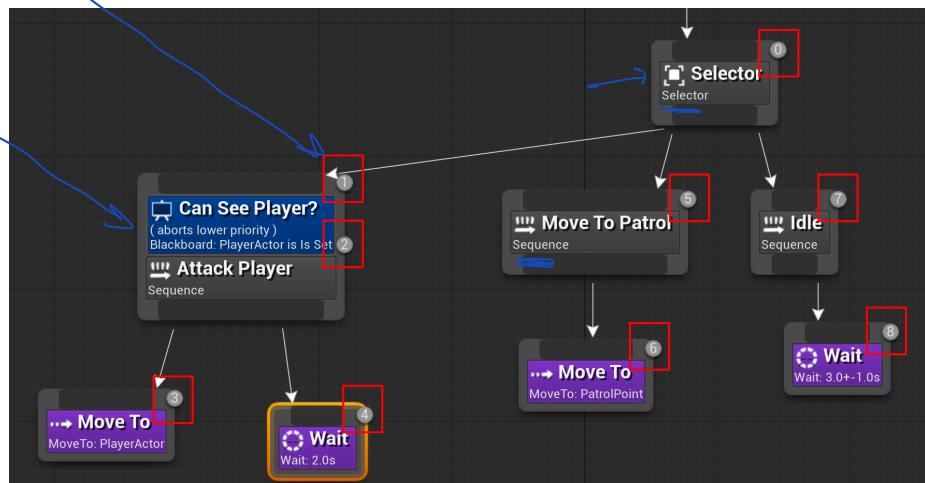
- มีตัวเลขกำกับเพื่อความชัดเจน

- Decorator

- Node สีน้ำเงิน

- เป็น Conditional ในระบบ BT

ที่ check อยู่กี่จุด



การ Run Behavior Tree ระหว่าง Gameplay

- ต้องทำงานผ่าน Pawn ที่เปรียบเสมือน AI Character (หรือ Object อื่น ๆ)
- Pawn นั้นต้องมี AI Controller สำหรับควบคุม Pawn ให้ทำ Action ต่าง ๆ เช่น การเรียกใช้งาน BT

เมื่อหัวหน้า Pawn เน้นว่า Body ก็อาจมุ่งหมาย ดังนี้

ประเภทการใช้งาน Behavior Trees ใน UE 4

- ▶ **เปรียบเทียบกับ Behavior Tree ทั่วไป (นอกเหนือจาก UE 4)**

- ▶ Event Driven

- ▶ Conditional Tasks (not leaf nodes) — Decorater

- ▶ Concurrent Behaviors → ทำหลายอย่างพร้อมกัน

leaf nodes (node ที่ไม่มีลูกเด็ก)

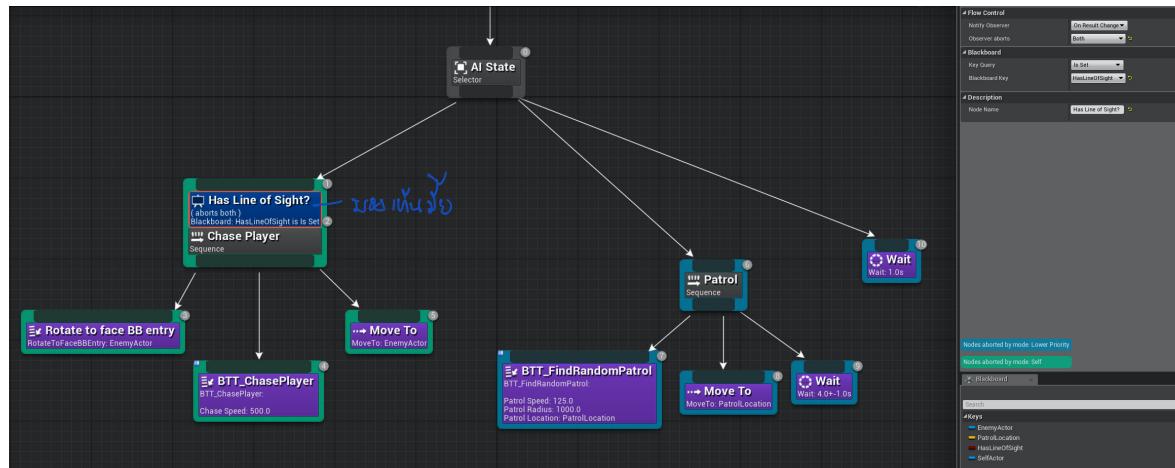
ไม่สามารถเรียกใช้ได้ใน event

ต้องรอผลของการ tasking

เราไม่ต้องตรวจสอบว่ามีเงื่อนไขใดๆ ให้เป็นจริงแล้ว เท่านั้นเอง เรายังสามารถตรวจสอบว่าสิ่งที่เราต้องการ เช่น check ว่าตัวละครต้องมีอาวุธอยู่ในกระเป๋า ก็สามารถทำได้โดย check ว่าตัวละครมีอาวุธอยู่ในกระเป๋า แล้วถ้ามี ให้ตั้งสิ่งนั้นเป็นตัวละครต่อไปแล้ว

Event Driven - ໃຫ້ ການ HF ຂັດໃຕ້ກ

- ໄມ່ໄດ້ທຳກຳໃນລັກຊະນະ checking event ຕລອດເວລາ
- Listen for events ຜຶ້ງທຳກຳທີ່ເໝົາມີຄົນເປັນ Trigger ໃນການເປົ້າຍັນຄ່າຕ່າງໆ ໃນ Tree

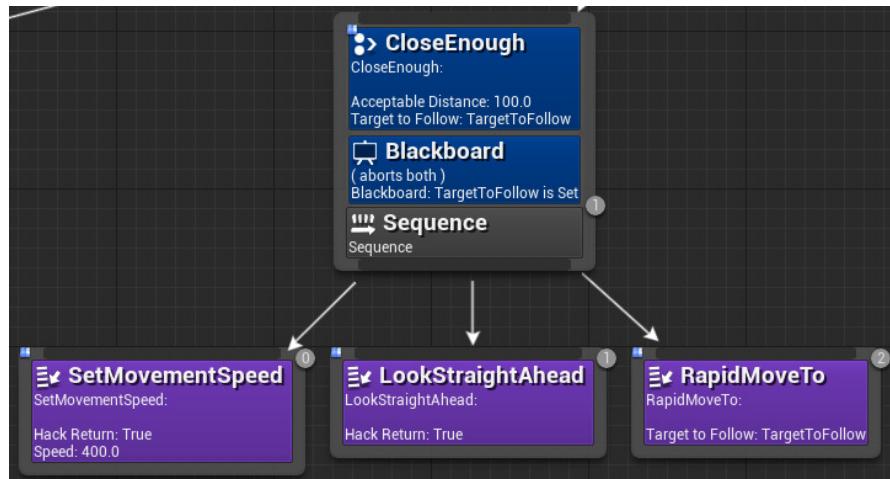


Conditional Tasks (not leaf nodes)

- BT ในระบบอื่น ๆ (นอกเหนือจาก UE4) ใช้ Conditional ในรูปแบบ leaf node
 - ส่งผลให้มีผลในรูปแบบ Succeed และ Fail
- BT ใน UE4 ใช้รูปแบบ Decorators
 - Conditional Decorators ช่วยให้ BT UI เข้าใจง่ายขึ้น
 - Leaf ทั้งหมดถูกพิจารณาเป็น Action ดังนั้น การจัดการลำดับในการทำงาน ทำได้สะดวกขึ้น
 - ระบบ BT อื่น ๆ Conditional จะถูกกำหนดอยู่ระหว่าง leaf ทำให้ยากในการทำความเข้าใจ

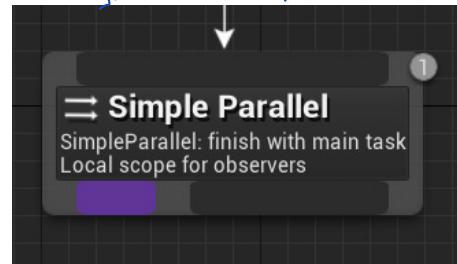
ตัวอย่าง Conditional Decorators ใน UE 4

- CloseEnough และ Blackboard คือ Conditional Decorators
 - อาจจะเรียกให้ Sequences node's children ทั้งหมดทำงาน หรือ ไม่ทำงานก็ได้



Concurrent Behaviors

- BT ทั่วไป ใช้ Parallel nodes ในการจัดการหลาย ๆ behaviors พิริอุ่ม ๆ กัน
- Parallel nodes มีความซับซ้อน และ บุ่งยากในการบริหารจัดการ — *ต้องดูงาน ท่าทางที่งานพัฒนาทำกัน
ในทางปฏิบัติ ไม่ผูกมุ่นกัน*
- ใน UE 4 ใช้วิธีการที่ง่ายกว่า โดยประกอบด้วย
 - Simple parallel มีแค่ 2 children nodes
 - เป็น single task 1 node และ sub-tree อีก 1 node
 - Special node ชื่อ Services
 - สำหรับการ Callback ในเวลาที่กำหนด หรือ การ update ต่าง ๆ ในเวลาที่กำหนด
 - Observer Aborts
 - ทำการตรวจสอบเงื่อนไขว่าสามารถ Abort ได้หรือไม่
- ข้อดีของการจัดการ Concurrent Behaviors แบบนี้คือ ความง่ายในการ Debug และ Optimize



ท่านนายอย่างพอดีกัน แล้วมานะเนี่ย มาดูยังกันด้วย ใจพูดไม่ถูก GAME DESIGN & DEVELOPMENT 14

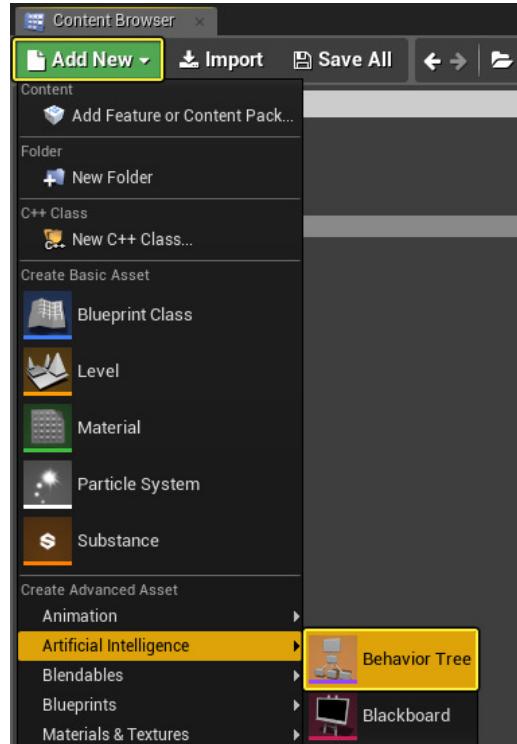
*แบบนี้มันจะ เผาไฟฟ้ากันบ่ย
เป็น Multitasking จริงๆ*

Behavior Tree User Guide

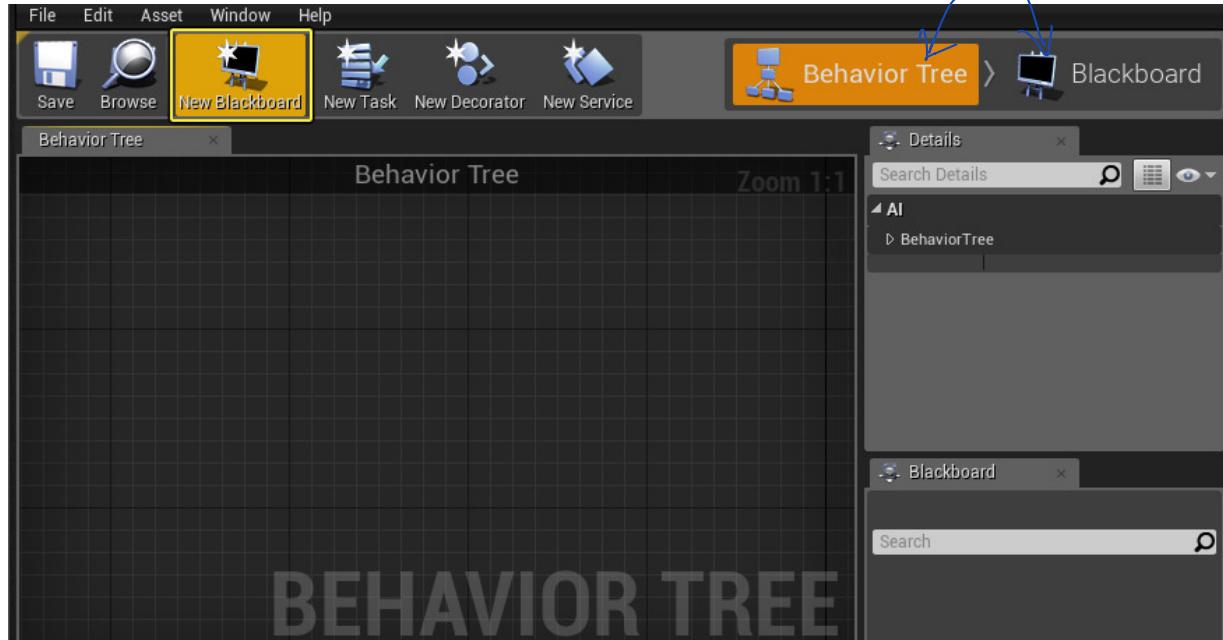
- สร้าง Behavior Tree
- สร้าง Blackboard
- Node: Selector, Sequence และ Decorator

สร้าง Behavior Tree และ Blackboard

- BT เปรียบเสมือน AI Processor
 - ตัดสินใจและควบคุมการทำงานต่าง ๆ จากผลตัดสินใจที่เกิดขึ้น
- Blackboard เปรียบเสมือนสมองสำหรับ AI
 - เก็บข้อมูล (Key values) สำหรับ BT ใช้ในการตัดสินใจ
- การสร้าง จะอยู่ใน Artificial Intelligence เมนู ดังรูป



Behavior Tree Editor



Selector vs. Sequence

ມີ node ທີ່ເກີດໄວ້ ໂດຍນີ້ Selector ນາຍ Node

- Selectors: Go through children from left to right
ມີກັບອຳນວຍທີ່ເກີດໄວ້ນີ້
ຮັງປັບຖາງ node ຖາຍໃຈ

looking for a successful node. If a node fails, it tries
ເລື່ອຕັ້ງທີ່ເກີດໄວ້ນີ້
the next one. When successful, the node is
ກໍາໄປສິໄງສັນຍົງ ຫຼັງປັບຖາງ ຄາສີຈາກນີ້ໃນ root node ໃຫ້
completed, and we can go back up the tree.

- Sequence: Execute from left to right, until a node
fails. If the node is successful, do the next one. If
the node fails, go back up the tree.

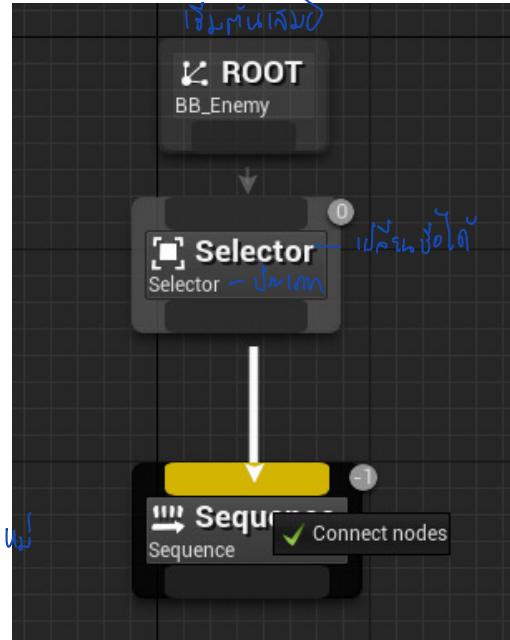
selector ນາຍຂັ້ນໄວ້ ຕ້າອັນເກດສັງເຮັດ — Selector ຈະລັບໃນ root ໃຫ້

— Sequence ຈະລັບກັນ

ອັນທີ 1 ສັງເຮັດ ທີ່ບໍ່ສັງເຮັດ

2 ທີ່ບໍ່ສັງເຮັດ

ຢູ່ປັບຖາງຂອງນີ້ ນອດນັກການໃໝ່ກຳເຊີງ ເພື່ອນັດລົມເຫັນກໍ
ກັນໄປ Root ໃຫ້



สร้าง Node Decorator

