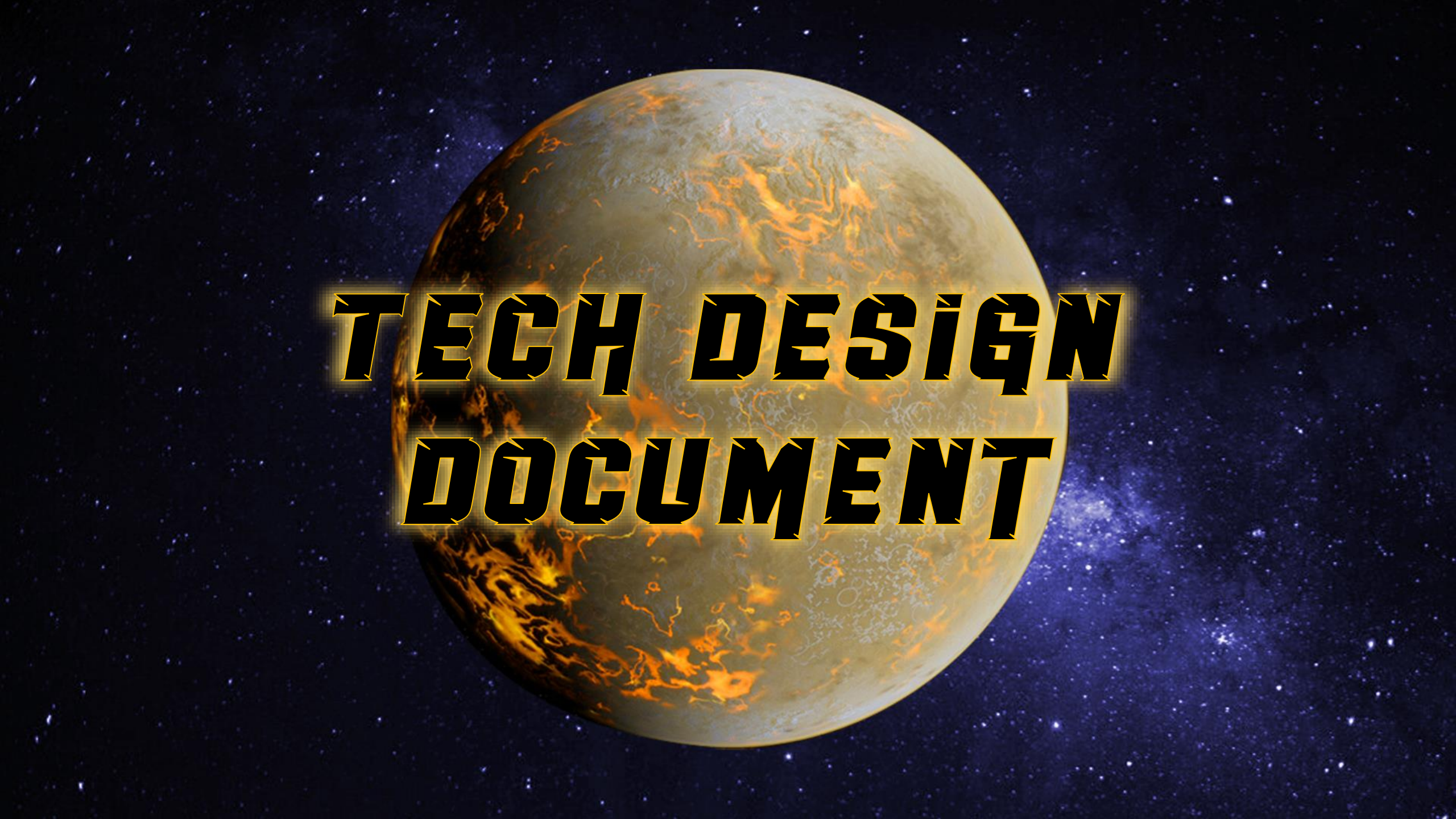

TECH DESIGN
DOCUMENT

SUMMARY


GAME MECHANICS
SELECTED

GAME MECHANICS SELECTED

Gameplay	%	Game Mechanics	Details	%
Exploration	50 %	Discover Map	Find way to discover	40 %
		Collect Resources	Minerals / Worker	40 %
		Identify Objectives	Place bomb / Extract Special Resources / Kill	20 %
Fight	30 %	Opponent Races	NPC & local fauna	60 %
		Spells	Power that consume mana	20 %
		Weapons	Weapon by class	20 %
Manage	20 %	Squad	Give order	70 %
		Base	Build constructions / Select Resources	30 %


COLLECT RESOURCES

COLLECT RESOURCES

▪ Random Spawn Resrources (Crystal)

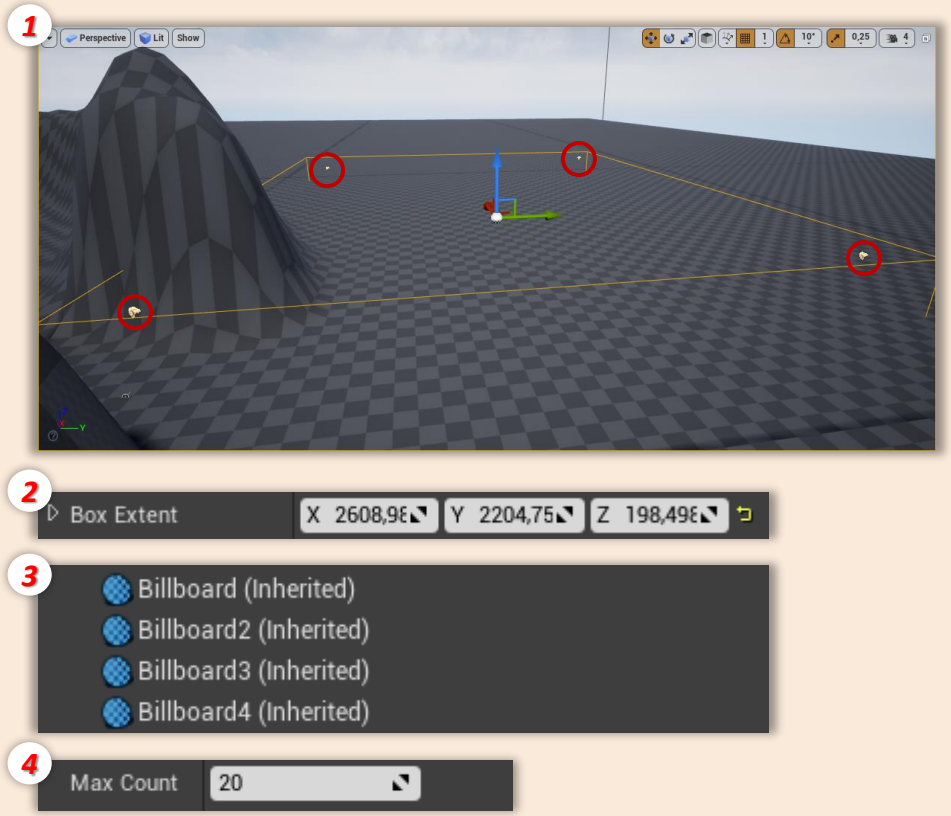
- **Editor**

1. ***Place*** *actor in the* ***scene***

2. ***Setup box extent*** *(max range of crystal spawn)*

3. ***Place billboards*** *in the scene*

4. ***Set max*** *number of crystal that* ***can spawn*** *in the box*



COLLECT RESOURCES

▪ Random Spawn Resrources (Crystal)

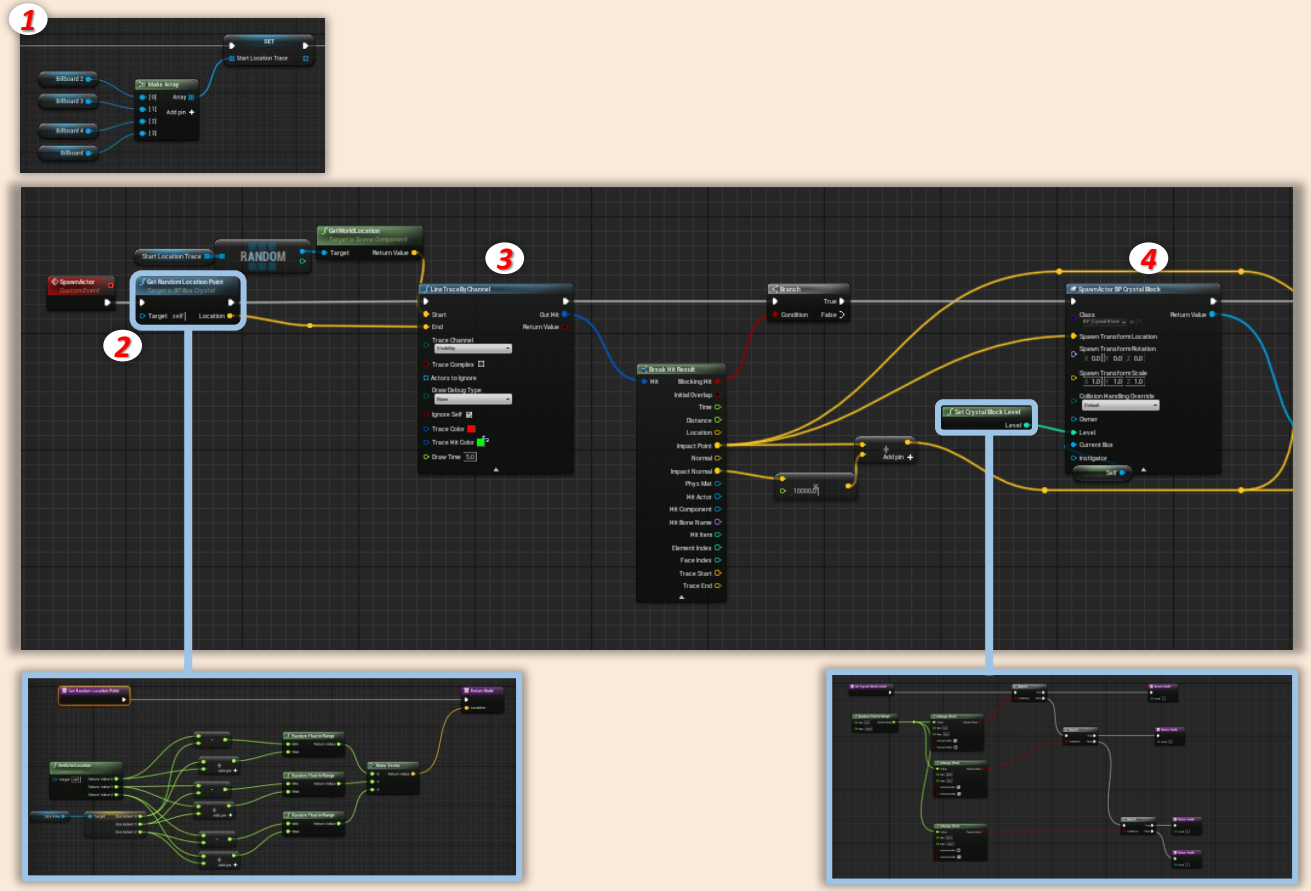
- **Spawn Crystal**

1. *All **billboards** are set in an **array** at the **start** of the game*

2. *A **random point** is taken in the box*

3. ***Trace** a **line between location** one of **billboards** selected **randomly** and the **point selected** before.*

4. ***Spawn crystal** at this **location** and set the **level randomly** (level has an **impact** of **number of resources** droped when actor is destroyed)*



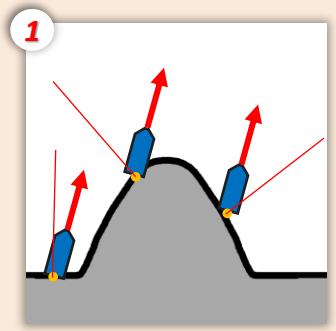
COLLECT RESOURCES

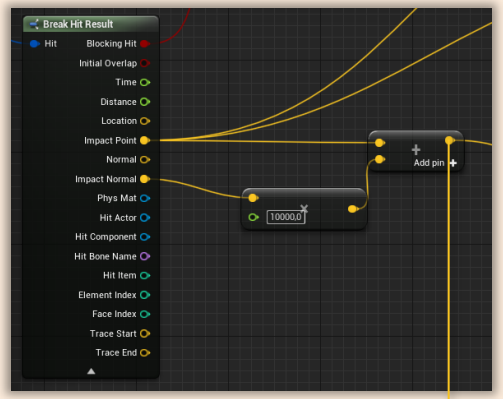
▪ Random Spawn Resrources (Crystal)

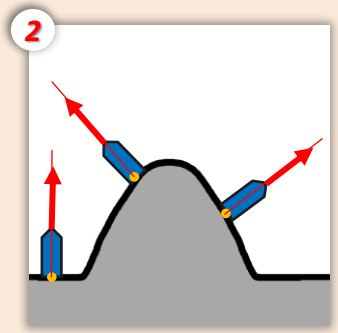
- **Set Crystal Rotation**

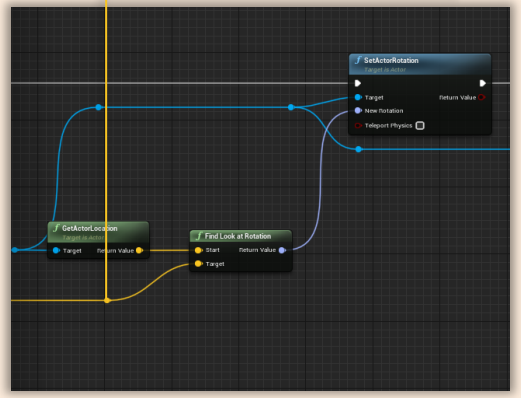
1. Get impact **normal vector** and **add** to **impact point**

2. **Set** crystal actor **rotation** by crystal **actor looking** this **vector** (normal vector + impact point)









COLLECT RESOURCES

▪ Worker & Meat

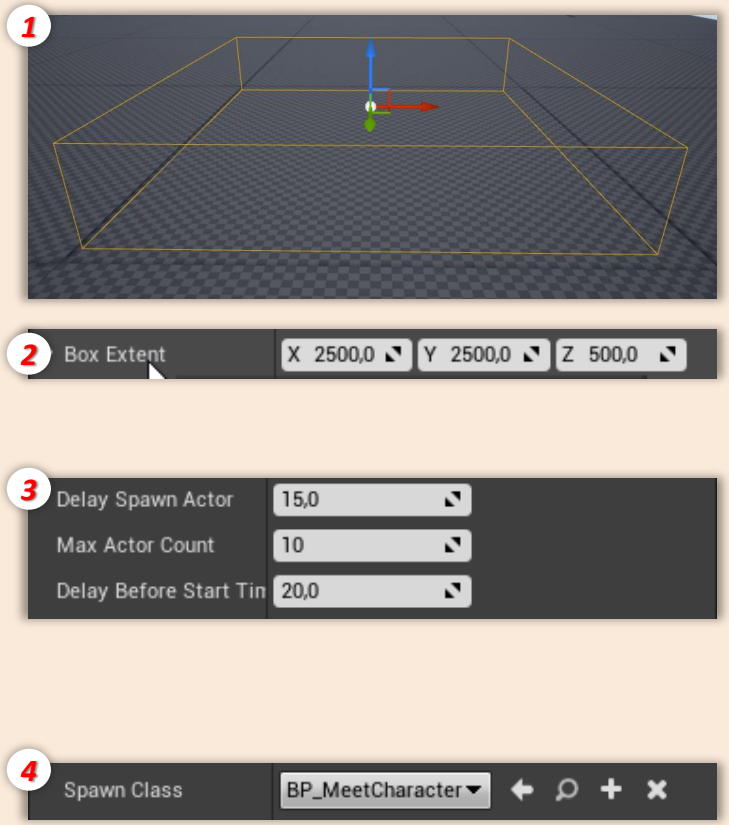
- **Editor**

1. ***Place*** *box in the scene*

2. ***Set box extend*** *(box extend determines* ***range*** *for the* ***spawn*** *of actor AND* ***range*** *for actors* ***movements*** *)*

3. *Set* ***max*** *number of* ***actor possible*** *in the box,* ***time between*** *each actor* ***spawn*** *and* ***delay before*** *first actor spawn*

4. *Select* ***type*** *of* ***actor***



COLLECT RESOURCES

▪ Worker & Meat

• Spawn Units

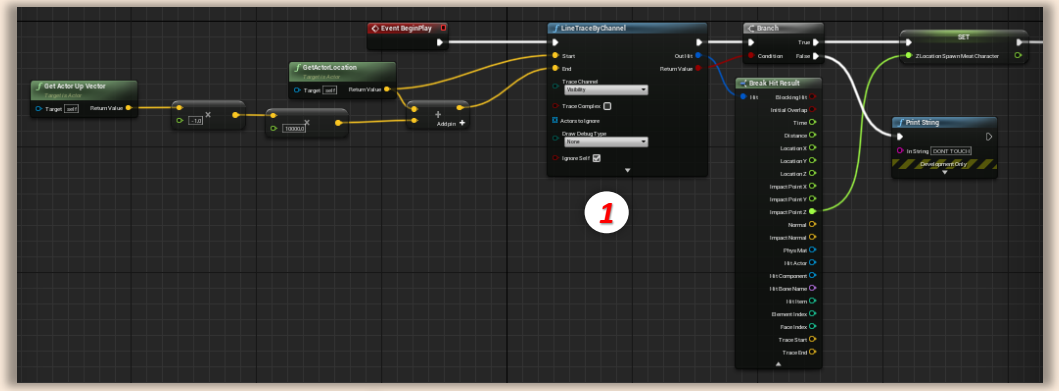
1. *At the **start** of the **game**, a **line** is traced **downward**. The **impact point** set the **location** of **Z** axis for actor's **spawn**.*

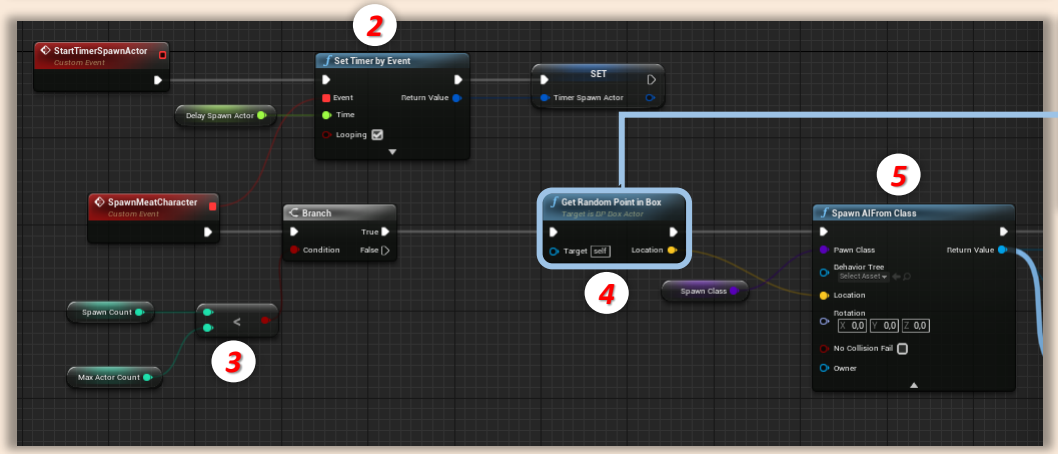
2. *A **timer** is **set** with the **delay** set in the **editor** and all the **X seconds** an **AI spawn**.*

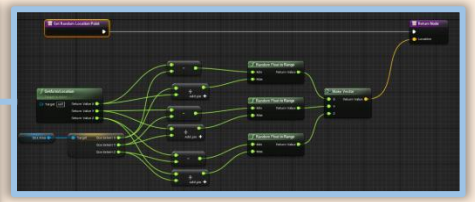
3. ***Check** if the **max actor** isn't reached.*

4. ***Get** a **random point** in the box.*

5. ***Spawn** an actor of the **class selected** in the **editor**.*





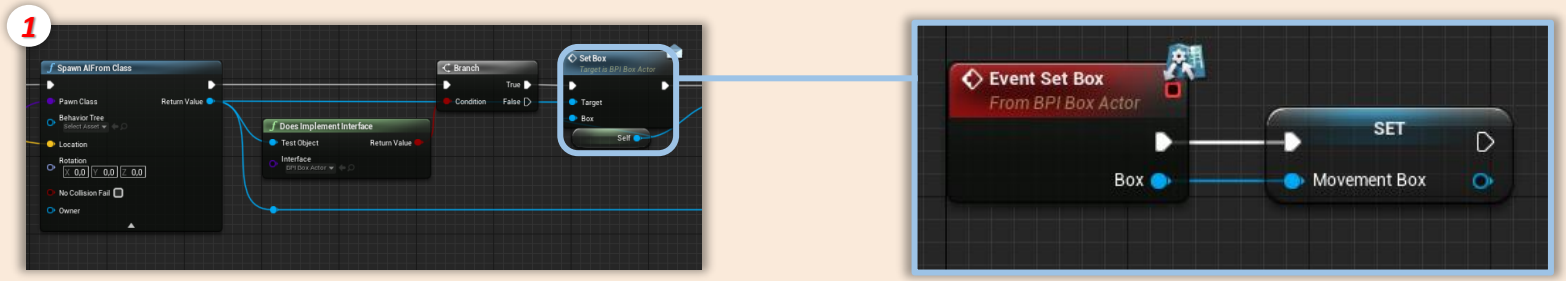


COLLECT RESOURCES

▪ *Worker & Meat*

• *Worker & Meat : Behaviour*

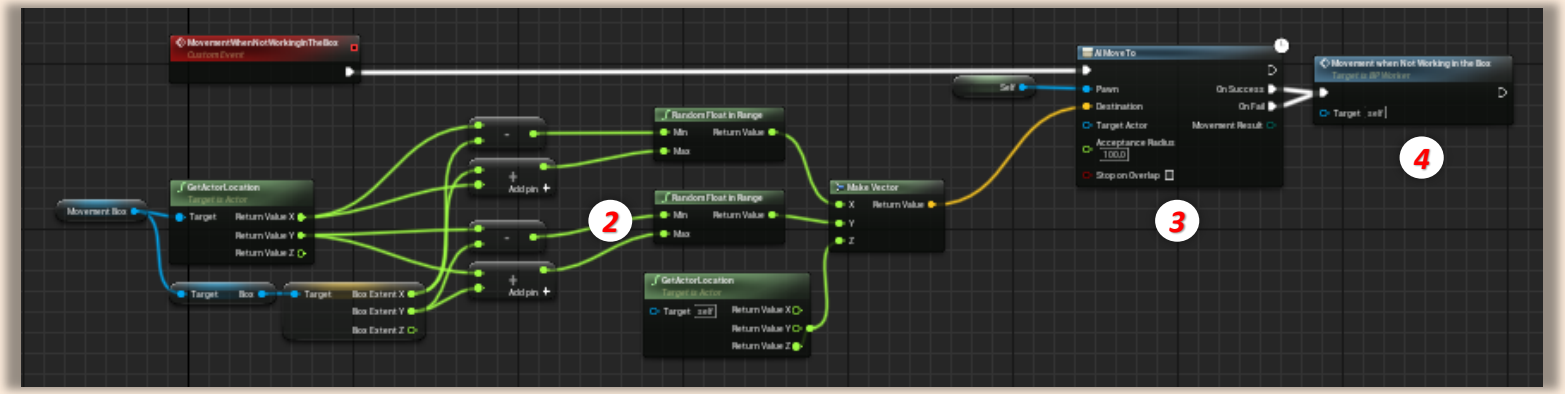
1. *After the spawn **AI** are **linked** to the **box**.*



2. *A **random point** is **taken** in the **box**.*

3. ***AI move** to this **location**.*

4. ***AI** can **reach** is point or **fail**, in the **two cases** it **starts again**.*



COLLECT RESOURCES

▪ Inventory System

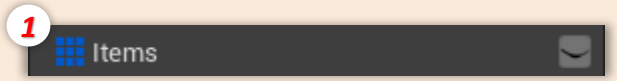
• Item structure

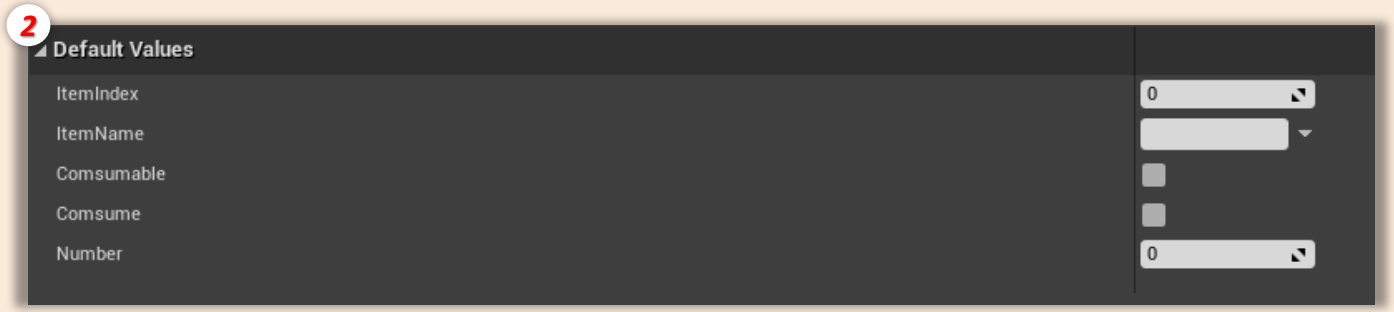
1. *Inventory is composed of a list of items*

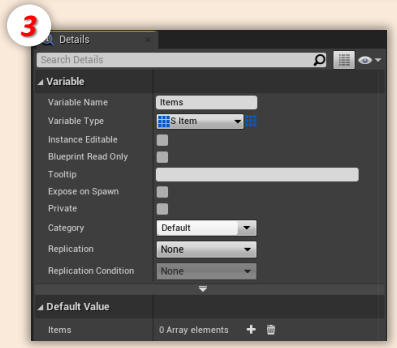
2. *An item has differents properties :*

- ***Item index :*** *it's a tag number unique to each item*
- ***Item Name***
- ***Consumable :*** *determine if the item can be used just one time*
- ***Consume :*** *if item is a consumable, determine if item is already used.*
- ***Number***

3. *Item structure list is a variable set on an actor*







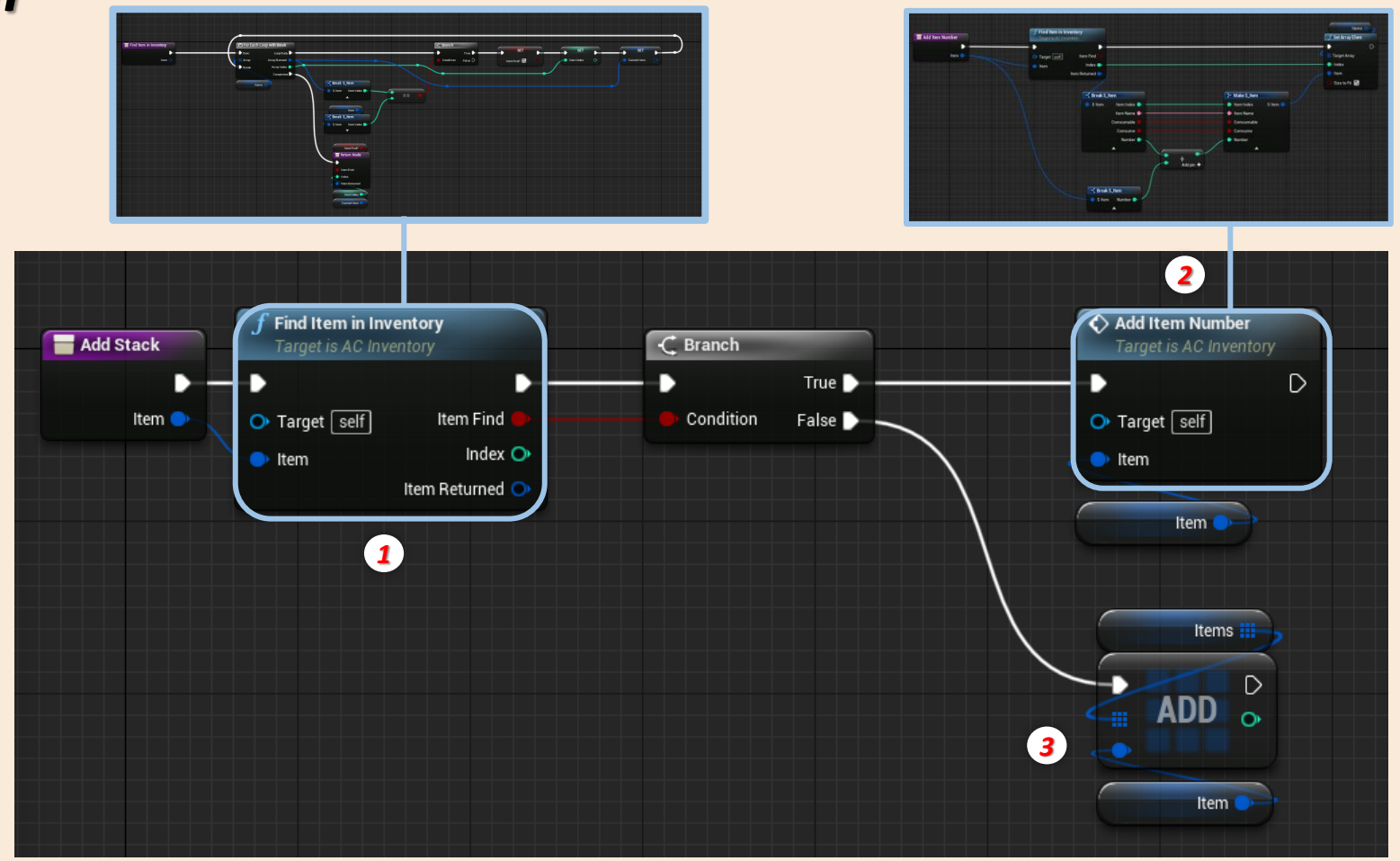
COLLECT RESOURCES

▪ Inventory System

• Add item to inventory

*When an actor possess an **item structure** and **interacts** with an actor that has an **inventory**, this **item** is **added** to **inventory** :*

1. ***Check** if item **doesn't aleardy exist** in the inventory (Find Item)*
2. *If it does, only the item **number** is **increased***
3. *If it doesn't a item stack is added*



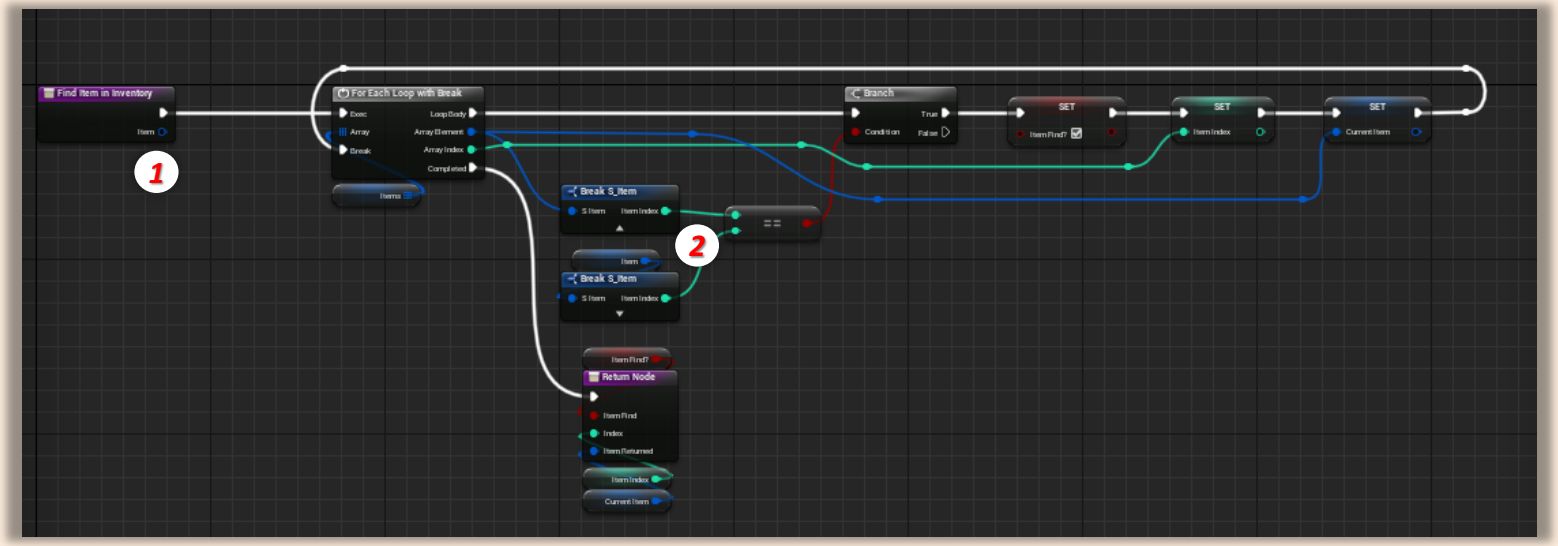
COLLECT RESOURCES

Inventory System

- **Find item in inventory**

1. *Enter an item to check*

2. *Get item list and for each element in the list, check the item index, if the index is the same than index of item to be found, list checking is stopped, item and is position in array are returned. If no item match with the item enter in the function the function returns false.*



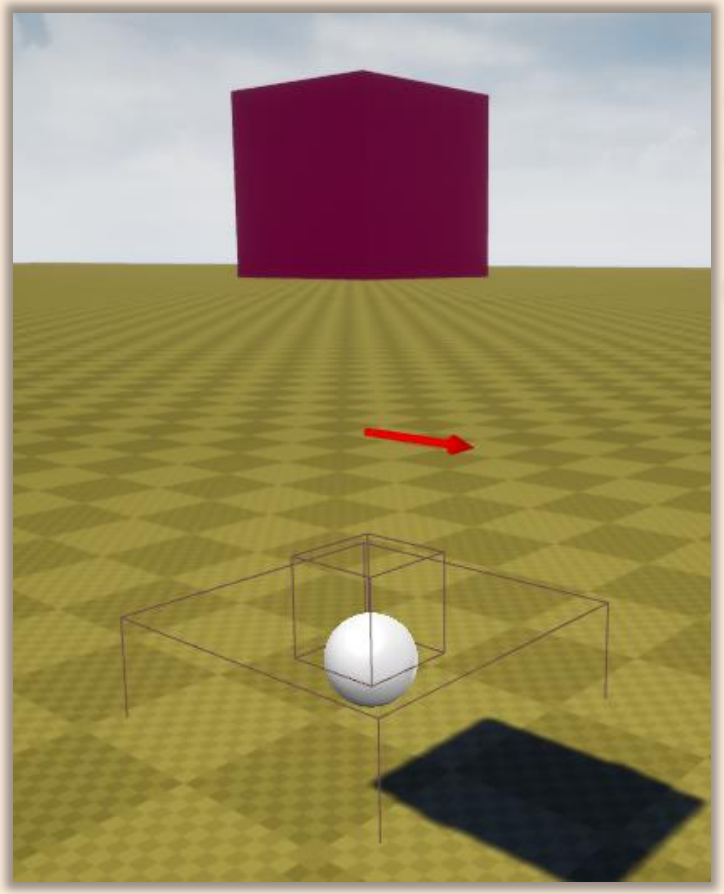

OPPONENT RACES

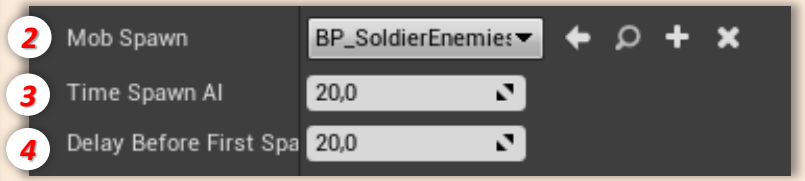
OPPONENT RACES

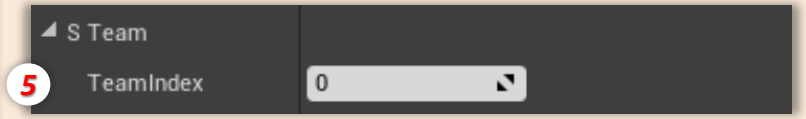
▪ NPC (Fyras)

• Mob generator (Editor)

1. *Fyras spawn by another actor class called BP_MobGenerator.*
2. ***Select class** of actor to spawn*
3. ***Set** the **delay** bewteen **each spawn** time*
4. ***Set** the **delay before** the **first spawn** time*
5. ***Set** team **index***





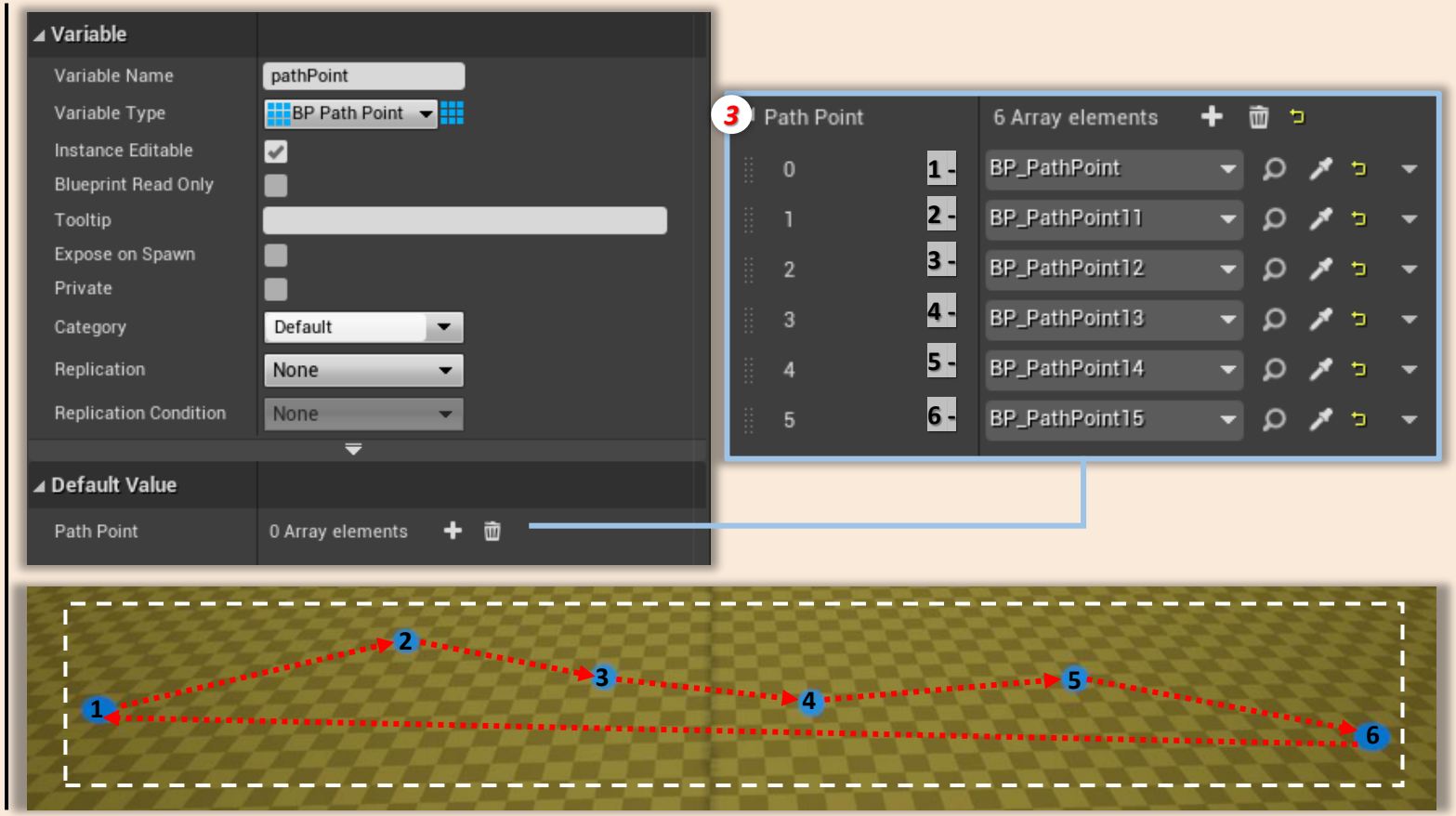


OPPONENT RACES

▪ NPC (Fyras)

- **Path manager (Editor)**

1. *Path manager contains a **list of path points***
2. *We can have paths **points as much as we want** in the list*
3. ***Place path points** in the scene and **add them to the list** (AI will follow the path points in list order)*



OPPONENT RACES

▪ NPC (Fyras)

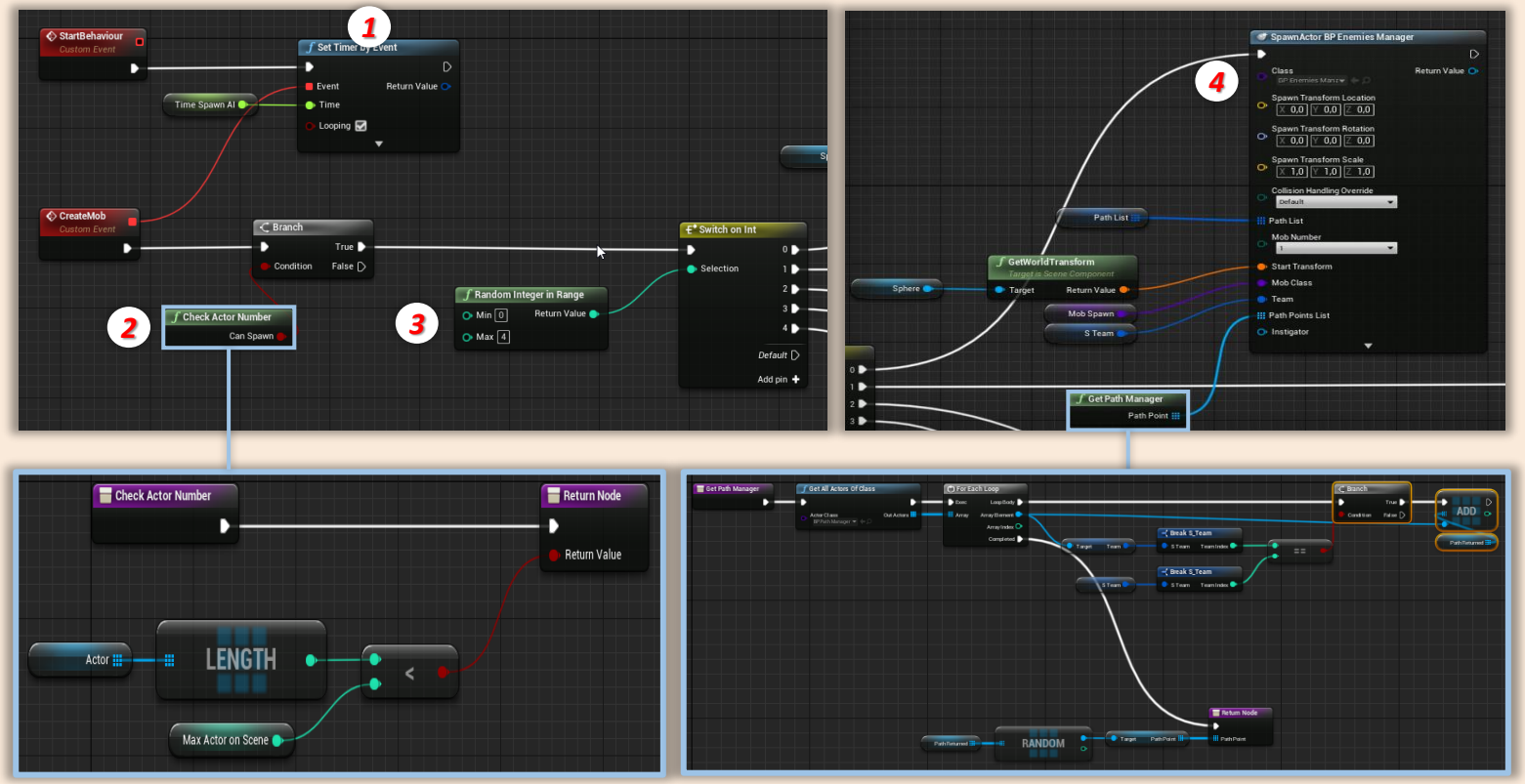
- **Mob generator (Spawn)**

1. *Each delay a group of mob is created*

2. *Check number of mobs on the map, if number is less than the max number set in the editor, a group can spawn*

3. *Group number is selected randomly*

4. *Spawn number of mobs and set the team link to them and path points AI follows (team is the same than mob generator)*

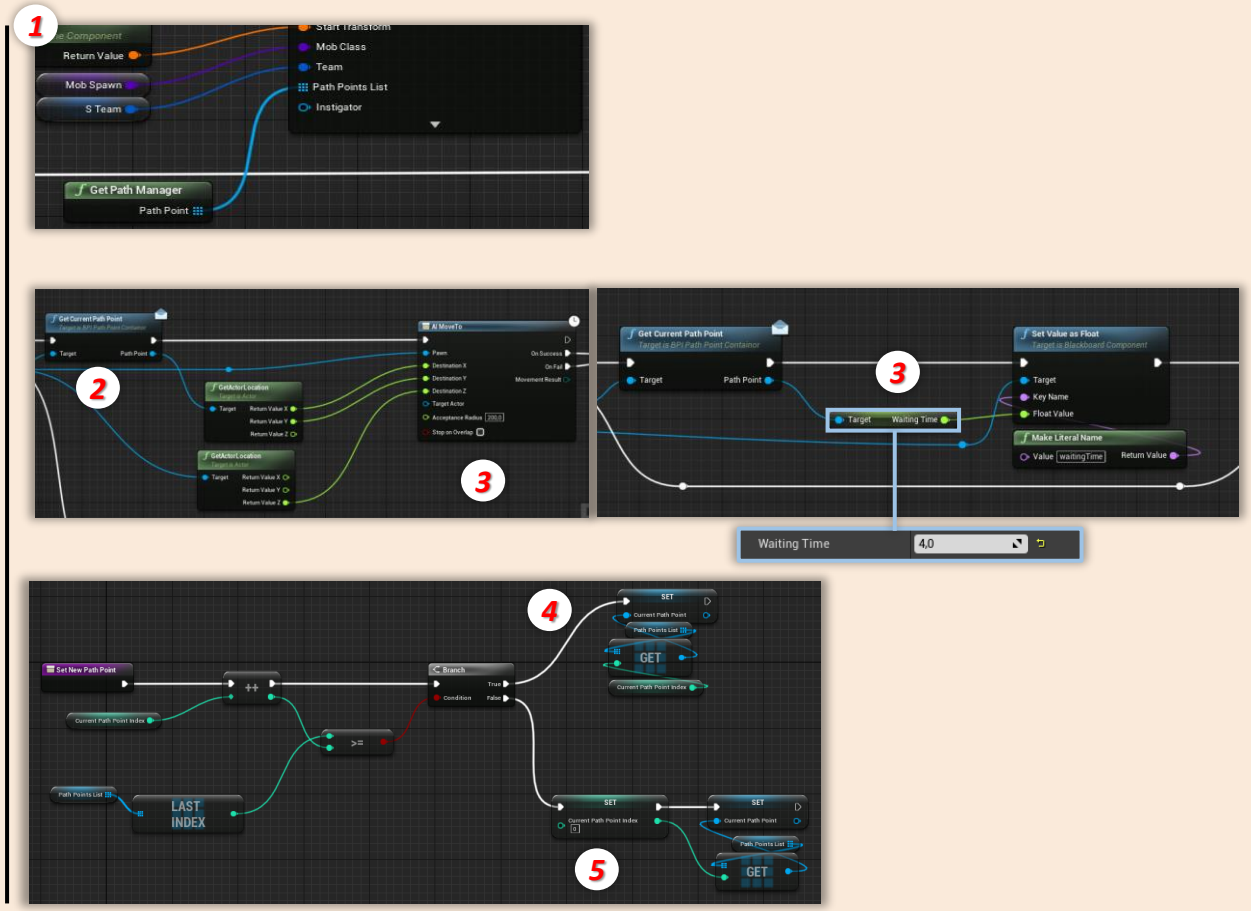


OPPONENT RACES

▪ NPC (Fyras)

- **Behaviour – Follow Path**

1. *NPC has a path points list set when it spawns*
2. *Get the first path point in the list*
3. *Move to this path point and wait time set in editor*
4. *Get next path points in the order of the list*
5. *When the last path point of the list is reached, restart with the first.*



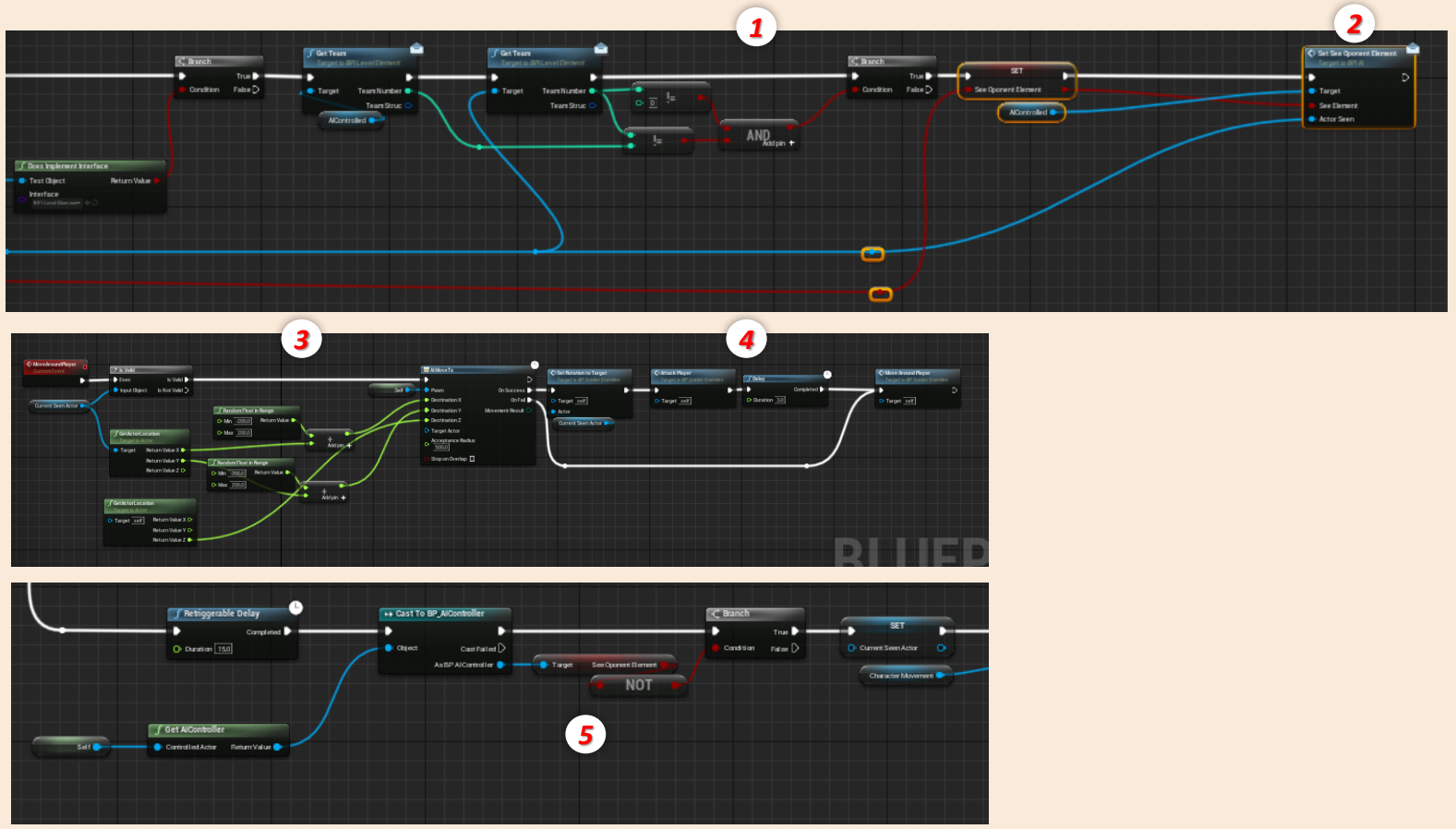
OPPONENT RACES

▪ NPC (Fyras)

• Behaviour – Focus target

*When an actor **enter** in the AI **field of view** :*

1. *Check if the **actor** seen is **not neutral** or **friendly** with the **AI** (check **team number**)*
2. *If the **actor** is **hostile** to AI switch state and set is **actor** as a **target***
3. *Get a **random location around** this target and **move** to this **location**.*
4. ***Focus** the **target** and **attack** it, **repeat** the behaviour **after a delay**.*
5. *When **target leave** the AI **field of view**, start a **delay** and after end of it, if actor is **still out**, **return** to Follow Path Behavior*




WEAPON /GEAR
SYSTEM

WEAPON / GEAR SYSTEM

Weapon and gear structure

*All differents **weapons** and **gears** derive from a **big parent class** that has **properties**, the **main** are :*

1. ***maxAmmoNumber***
2. ***projectileClass** : **class** of projectile **instantiate** when the weapon is **firing**, if **class isn't selected,** weapon is set as an **hitscan weapon***
3. ***currentTimePerShot** : delay between **each shot***
4. ***canMaintainFire?** : if the player can **maitain** the fire **input pressed or** he needs to **press again** after a shot*
5. ***characterHandler** : character who **possess this weapon***

WEAPON / GEAR SYSTEM

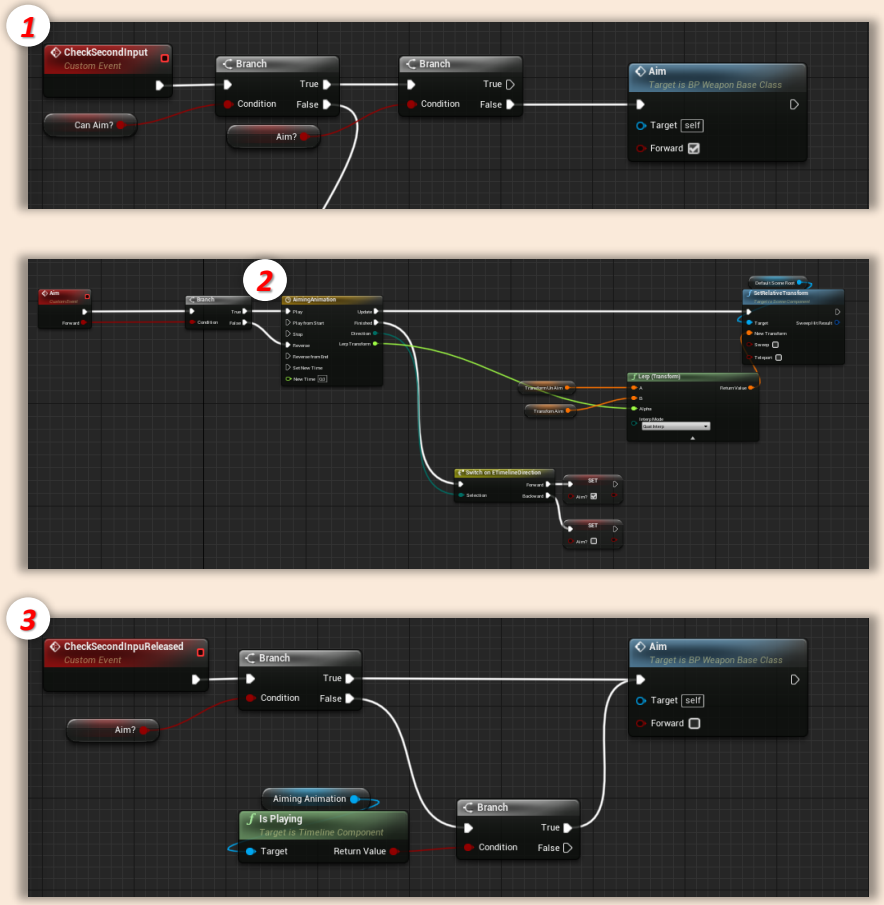
Weapon behaviour

- **Aim**

1. When **input is received**, **check** if character **can aim** with this weapon

2. If **yes**, **play aim animation** and at the end of it and **set aim state**

3. When **input is released** play **reverse** animation and **reset aim state**



WEAPON / GEAR SYSTEM

Weapon behaviour

Fire (Hit Scan)

1. When **input** is **received**, **check** if is **weapon** is in **aim mode** (for **some weapon**, **configuration** for fire is **different** between **aim** and **unaim**)

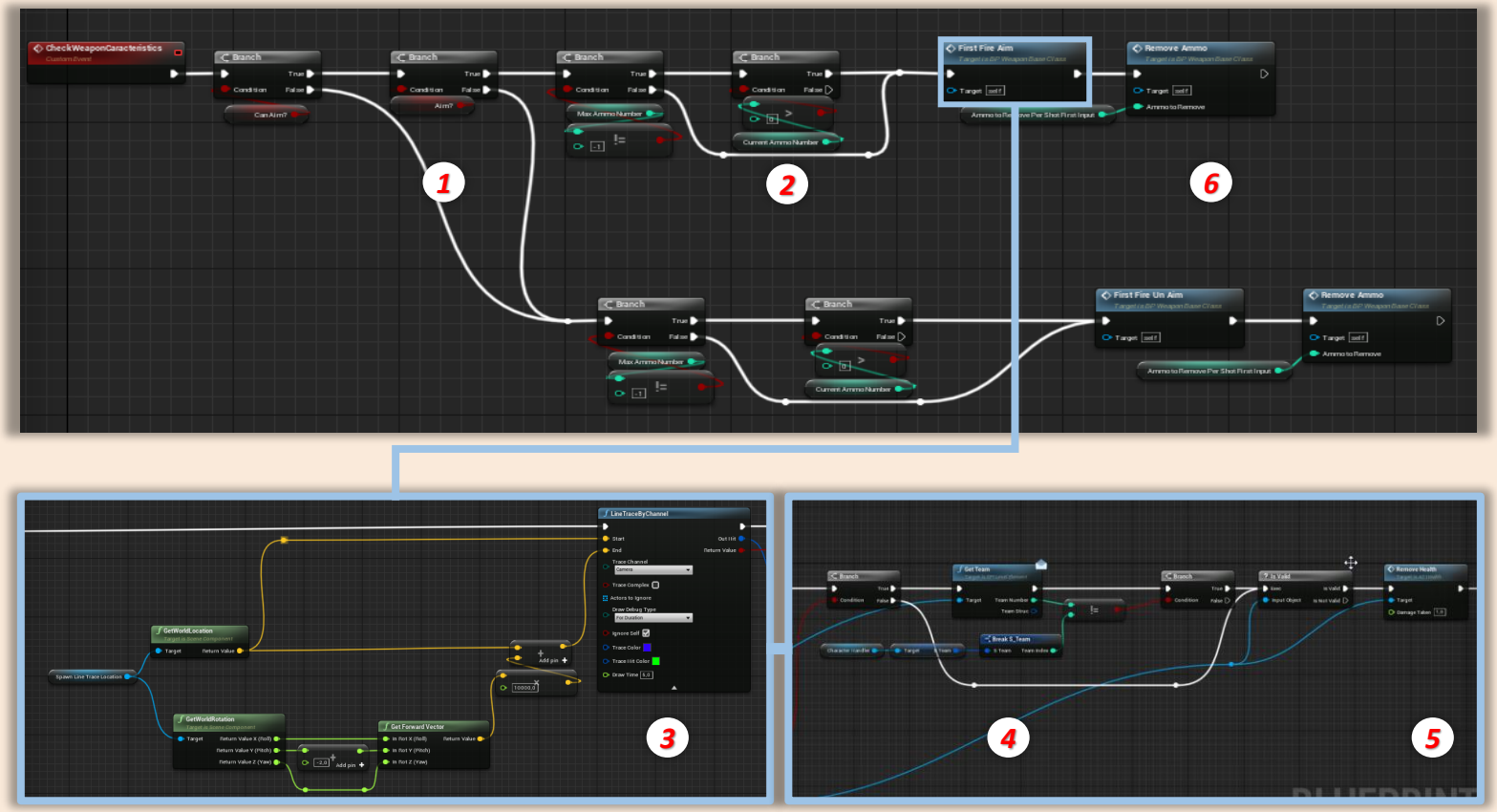
2. **Check** if **ammo** number is **higher** than **0**

3. For hit **scan weapon**, a **line** is **traced from** the **canon** to the **max fire distance** of the **weapon**

4. If the line **hits** an **actor, check** if actor is **hostile** to **player** (to **avoid friendly fire**)

5. Check if **actor** hit **can receive damage**, if yes **apply damage** and **revome health**

6. **Remove one unit** to ammo **number**

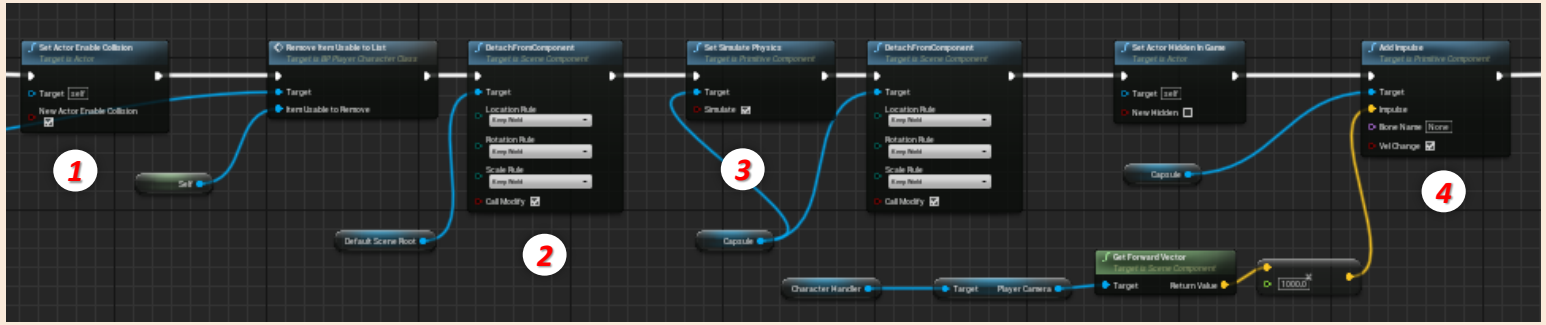


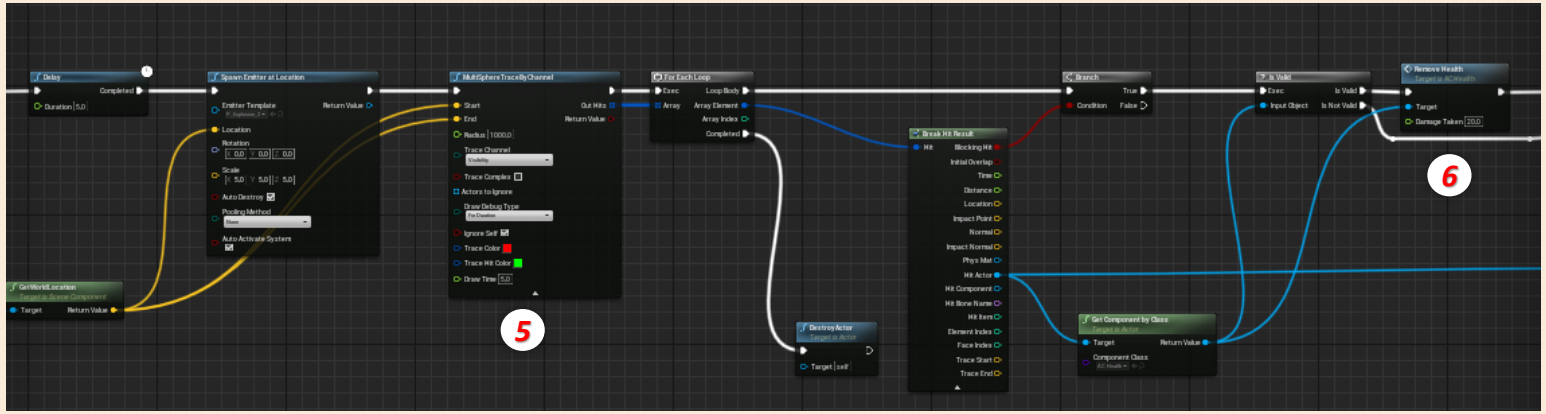
WEAPON / GEAR SYSTEM

Gear behaviour

- **Use gear (Grenade)**

1. When **input** is **received**, **collision is activated**
2. **Grenade is detached from its owner**
3. **Physics** of grenade collider is **activated**
4. An **impulse** is **added** to actor in the **direction** where **player aim**
5. For the **grenade**, when it is **dropped**, a **delay** is **set** and after that, a **sphere** is traced from is **location**
6. All **actors hit** are **checked** and if they **can receive damages**, **apply** damage





WEAPON / GEAR SYSTEM

▪ Weapon / Gear Management

- **Set weapons and gears at start**

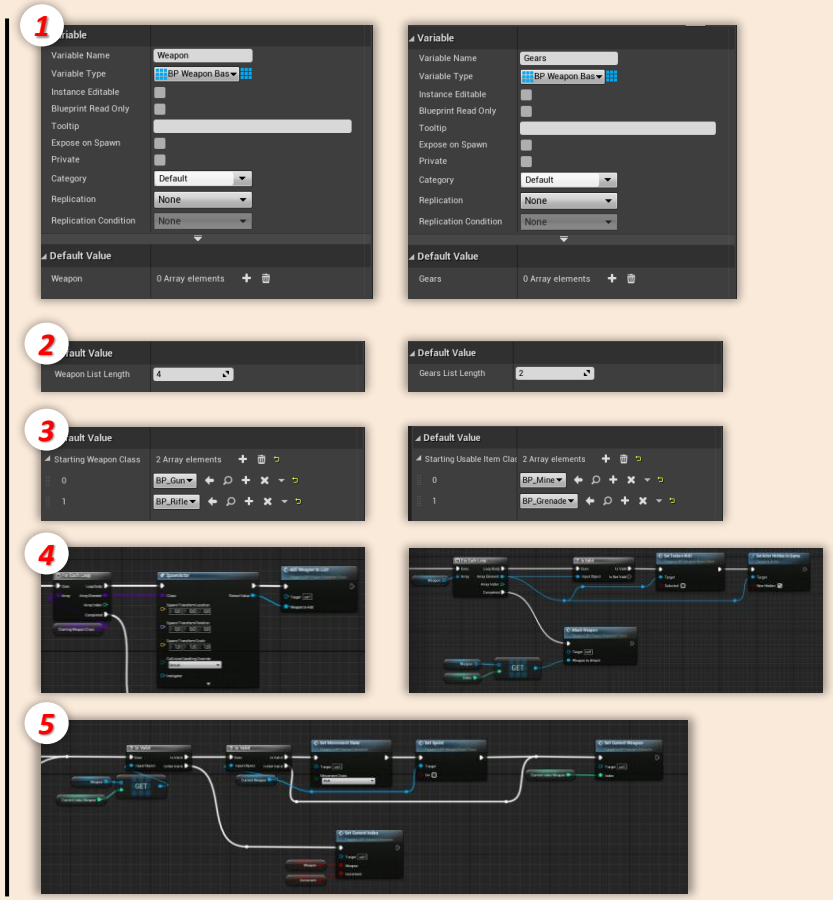
1. *Weapon and gear are the same actor type but they are set in two differents array, they are contained in player*

2. *Set the length of weapon and gear list*

3. *Select class disponible at the start of the game*

4. *For each list spawn actor of class selected and attach actor to player character and set a default weapon selected*

5. *To switch weapon an index is increased or decreased and for each changes get the weapon of index emplacement in array and set it as current weapon.*



Back to summary

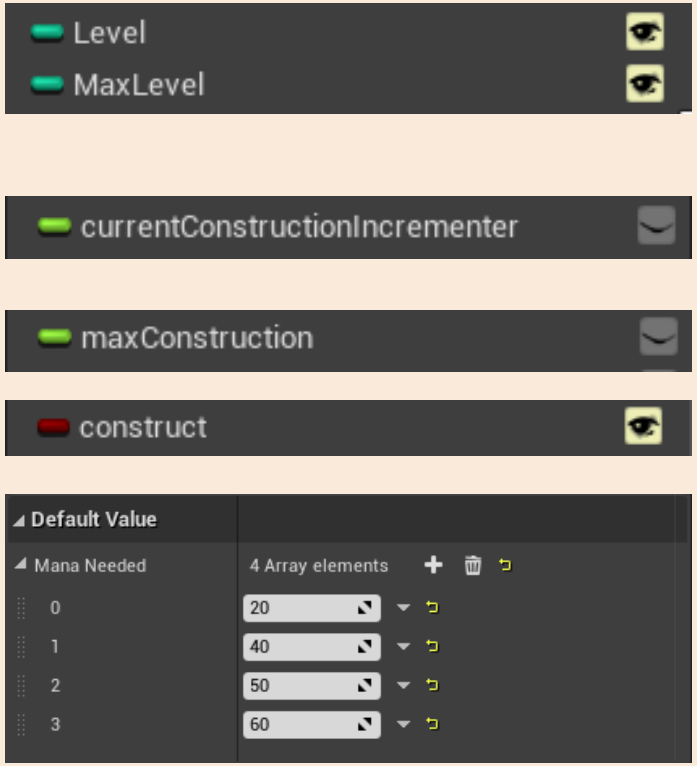

BUILDING SYSTEM

BUILDING SYSTEM

Building structure

All differents buildings derive from a big parent class that has properties, the main are :

1. ***Level :*** *current level of building, at the start of the game this level is 0*
2. ***Max level :*** *the maximum level can be achieve by building*
3. ***currentConstructionIncrementer :*** *current unit to achieve next level*
4. ***maxConstruction :*** *unit needed to achieve next level*
5. ***Construct :*** *return if building is contructed or not*
6. ***ManaNeeded :*** *list of mana necessary for each level*



BUILDING SYSTEM

Check buildings in the base

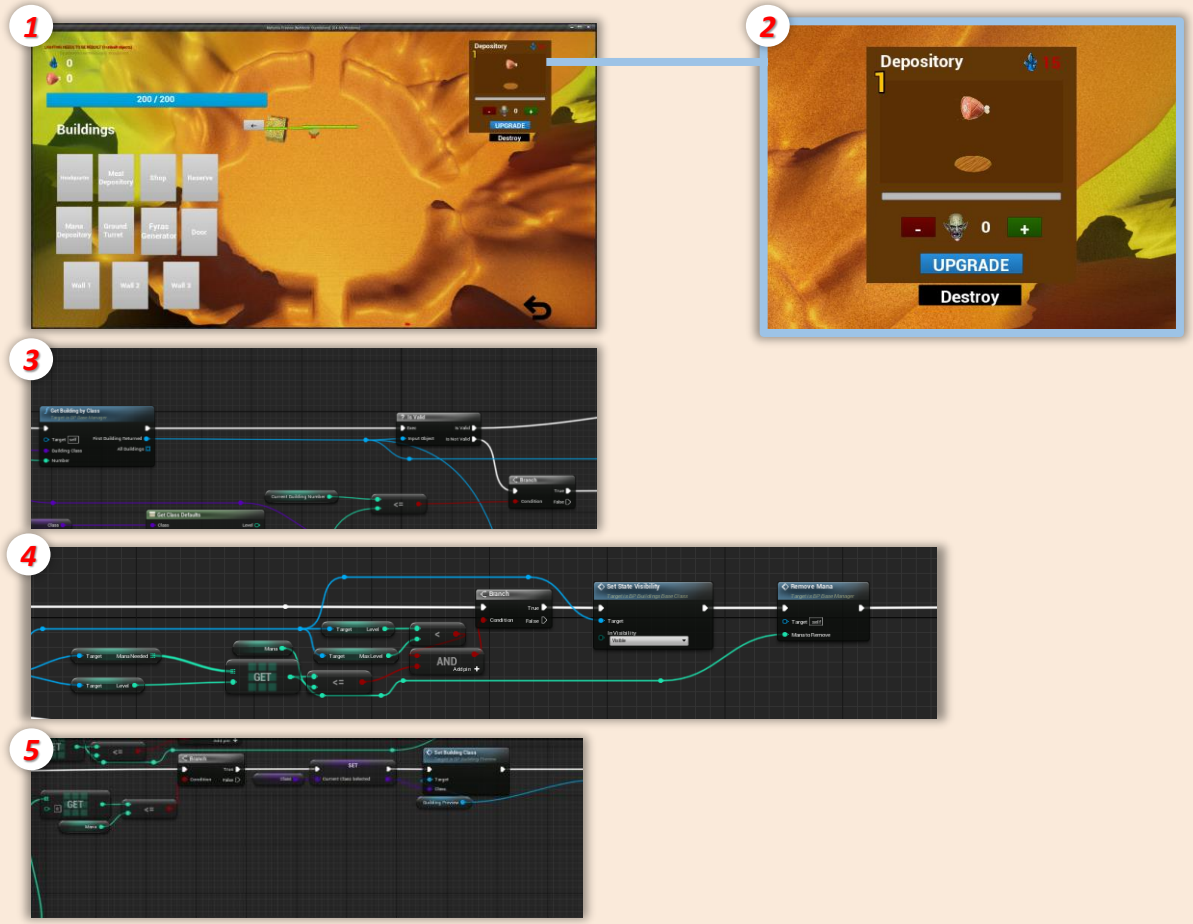
1. *When player is in **managment mode**, a **top down view** is set and **all buildings disponible** in the base are **visible** (even those already constructed)*

2. *When a **building** is **selected** informations appear at top **right of the screen**.*

3. *When Construct/ Update **button is pressed**, **check** if building is **already contructed***

4. *If **yes** check the **compare level** to **max level** that can be achieved, if they are **equal nothing happens**, if **yes building** enter in **upgrade mode***

5. *If building is **not already constructed**, building enter in **preview mode***



BUILDING SYSTEM

Preview Mode

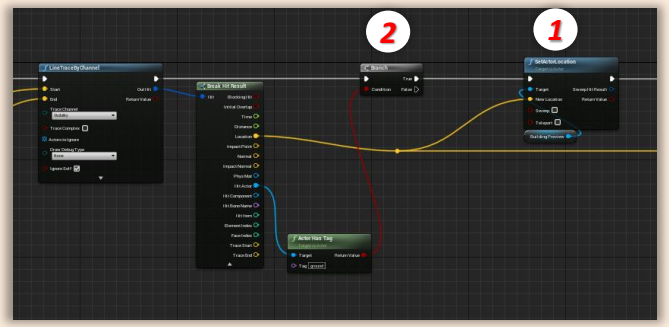
1. *When building is in **preview mode**, this **location** is set to **mouse cursor location** in the **world***

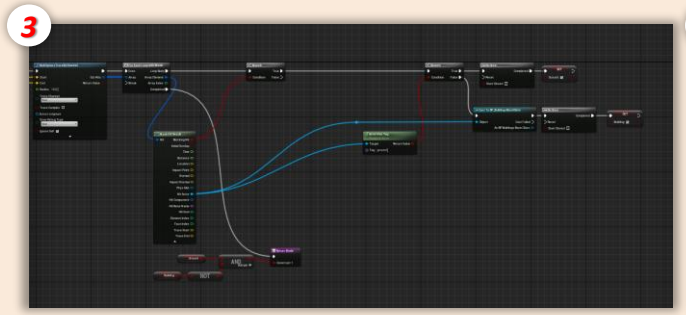
2. ***Check** if location is **on the ground** by checking **actor tag***

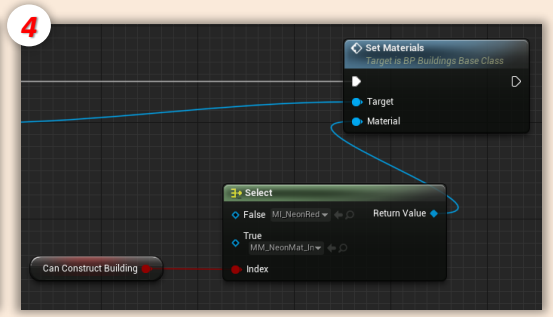
3. *Check if location is **in base area**(diffined by a **big sphere**), **trace a sphere** at the **impact location**, if the **trace overlaps** the **big sphere** and **not overlaps** a **building already constructed,** player **can constuct** his building at this location*

4. ***Material** is **set** to give the **information** to player*

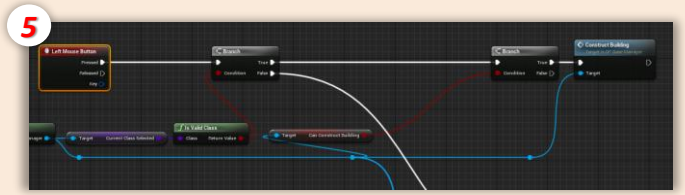
5. *If player **can constuct** and input **left click** is **pressed, building spawn** at **this location** and enter in **construction mode***







3



BUILDING SYSTEM

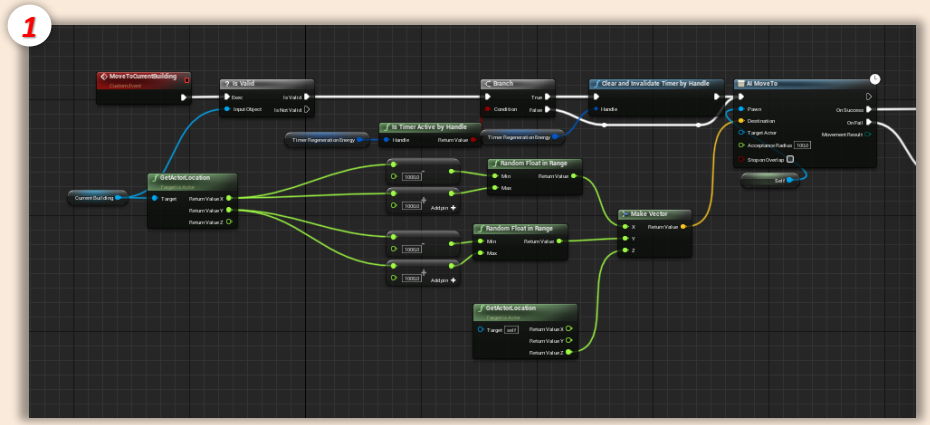
Construction Mode

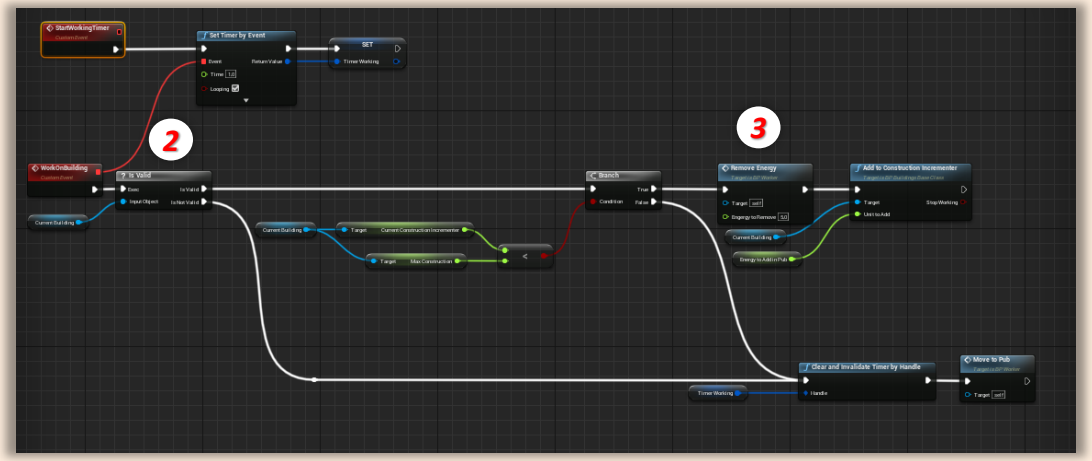
*When building is in **construction mode**, player can **add workers** present in base to this **task***

1. *When a worker is **assigned** to a build, he **moves to building location***

2. *When location is **achieved**, a **timer is set** and **each delay** the variable **« currentConstructionIncrementer »** is **increased** and **check** if is **equal** or **superior** to variable **« maxConstruction »***

3. *At the same time **energy of worker** is **decreased** and if this energy is **less** or **equal** to **0 timer is cleared** and **worker goes** to the **reserve**.*





BUILDING SYSTEM

Level Up

1. When « maxConstruction » value is achieved, building can level up

2. Check if current level is not superior to max level and is not set current level properties (health, firerate…, depend of building class), variable « construct » is set to true

3. Building returns to normal state, all workers are removed from this building

To upgrade building, its the same prossess than construction (the step « Preview » is skipped)