**Prefabs (instantiate)**

Prefabs คือ อ็อบเจ็กต์เกมที่กำหนดค่าไว้ล่วงหน้าซึ่งสามารถใช้เป็นเทมเพลต เมื่อเราสร้าง Node ลากและวาง prefab จากแผง**เนื้อหา**ไปยัง **Node Tree** หรือแผง **Scene** เพื่อสร้าง **Prefab Instance** ในฉาก Node Prefab แสดงในสีฟ้าใน Node ต้นไม้หากคุณสมบัติของรูปแบบสำเร็จรูปในฉากถูกแก้ไข จากนั้นฉากจะถูกบันทึก ข้อมูลที่แก้ไขจะถูกเก็บไว้ในรูปแบบสำเร็จรูปนั้น

**Animation (Tween,Tools)**

Animation ระบบที่มาพร้อมกับ Cocos Creator การเคลื่อนไหวมาตรฐาน การหมุน มาตราส่วน และสไปรท์ เพิ่มสิ่งนี้โดยให้การสนับสนุนการเปลี่ยนแปลงตามเวลาของส่วนประกอบที่กำหนดโดยผู้ใช้และคุณสมบัติที่กำหนดเอง

**Tween**

เมธอดที่สร้างขึ้นโดยลูกโซ่ ซึ่งสามารถจัดการกับ อ็อบเจกต์ ใดๆก็ได้ และทำให้คุณสมบัติของ อ็อบเจ็กต์ลดลง

**Load Resource (image)**

When loading resources dynamically, pay attention to two points. All resources that require dynamically loaded via script, must be placed under resources folder or its subfolders, resources need to be manually created in the assets folder and must be located in the root directory of the assets

เนื้อหาในresourcesโฟลเดอร์สามารถอ้างถึงเนื้อหาอื่นนอกโฟลเดอร์ และยังสามารถอ้างอิงได้จากฉากหรือเนื้อหาภายนอก เมื่อสร้างโปรเจ็กต์แล้ว ทรัพย์สินทั้งหมดในresourcesโฟลเดอร์พร้อมกับเนื้อหาภายนอกresourcesโฟลเดอร์ที่เชื่อมโยงจะถูกส่งออก นอกเหนือจากฉากที่ได้รับการตรวจสอบในแผงการสร้าง

**Life cycle**

ฟังก์ชันการเรียกกลับของวงจรชีวิตสำหรับสคริปต์ส่วนประกอบ ตราบใดที่ผู้ใช้กำหนดฟังก์ชันการเรียกกลับเฉพาะ ผู้สร้างจะเรียกใช้สคริปต์ในช่วงเวลาที่กำหนด ผู้ใช้ไม่จำเป็นต้องเรียกใช้ด้วยตนเอง

**onLoad**

ในขั้นตอนการเริ่มต้นของสคริปต์ส่วนประกอบ เรามี onLoad ฟังก์ชันเรียกกลับ onLoad การเรียกกลับจะถูกทริกเกอร์เมื่อเปิดใช้งาน Node ครั้งแรก เช่น เมื่อโหลดฉากหรือเมื่อเปิดใช้งาน Node

**Start**

startฟังก์ชันการเรียกกลับจะถูกเรียกก่อนที่จะเปิดใช้งานครั้งแรกขององค์ประกอบซึ่งเป็นก่อนที่จะดำเนินupdate เป็นครั้งแรก start มักใช้เพื่อเริ่มต้นข้อมูลที่ต้องการแก้ไขบ่อยครั้ง ซึ่งอาจมีการเปลี่ยนแปลงระหว่างupdate

**update**

ประเด็นสำคัญประการหนึ่งสำหรับการพัฒนาเกมคือการอัพเดท พฤติกรรม สถานะ และทิศทางของวัตถุก่อนเรนเดอร์ทุกเฟรม โดยปกติการดำเนินการ update เหล่านี้จะใส่ในการ update callback

**Late Update**

Update จะดำเนินการก่อนการ Update ของ Animation ทั้งหมด แต่ถ้าเราจะทำงานพิเศษบางอย่างหลังจาก Update Effect (เช่น Animation อนุภาค ฟิสิกส์ ฯลฯ) หรือต้องการดำเนินการอื่นหลังจาก updateส่วนประกอบทั้งหมดเสร็จสิ้น จากนั้นเราจะต้อง lateUpdate

ลำดับการทำงานของวงจรชีวิตไปตลอดชีวิตที่สมบูรณ์ของส่วนประกอบจากการเริ่มต้นที่จะเปิดใช้งานและการทำลายสุดท้ายคือ: onLoad -> onEnable -> start -> update -> lateUpdate -> onDisable -> onDestroy.