

Software Studio

軟體設計與實驗

Slime AI

Hung-Kuo Chu

Department of Computer Science
National Tsing Hua University



Important

- We use Cocos Creator **2.4.8** in this course.
- Please make sure you use the same version, otherwise your program may crash when we test it.



Goal

- Fork repo Lab8-Slime AI to your repository.
- Open scene **level1-1**.
- Open the project in your code editor and search for **TODOs**, then complete the requirements.
- **For this lab, you will only need to modify scripts under the “scripts/ai” folder!**



Goal (Cont'd)

- You can see TA's demo here: <https://ab-4a13f.web.app/>



Controls

- Use W A S D to move the main character.



TODO 1

- Files: **Navigator.ts (1.1), NavWanderer.ts (1.2), NavWanderAgent.ts (1.3)**
- Follow the hints in the files to implement yellow slime's behavior.
- **End result: The yellow slime should be changing directions on the four waypoints of Waypoint Graph 1.**



TODO 2

- Files: **WaypointGraph.ts (2.1), NavChaser.ts (2.2)**
- Follow the hints in the files to implement red slime's behavior.
- **End result: The red slime should be able to track down the player when the player is in the right side of the map.**



TODO 3

- File: **TODO5.ts (3)**
- Turn the slimes in the scene into prefabs (you can put them under the **Prefab** folder) and instantiate them in the scene according to TODO3.ts's instructions.



Some important concepts

```
onLoad() {  
    // #region [YOUR IMPLEMENTATION HERE]  
  
    // #endregion  
}  
  
start() {  
    // #region [YOUR IMPLEMENTATION HERE]  
  
    // #endregion  
}
```

- **onLoad():** Called when the node the component is attached to is loaded.
- **start():** Called after onLoad().



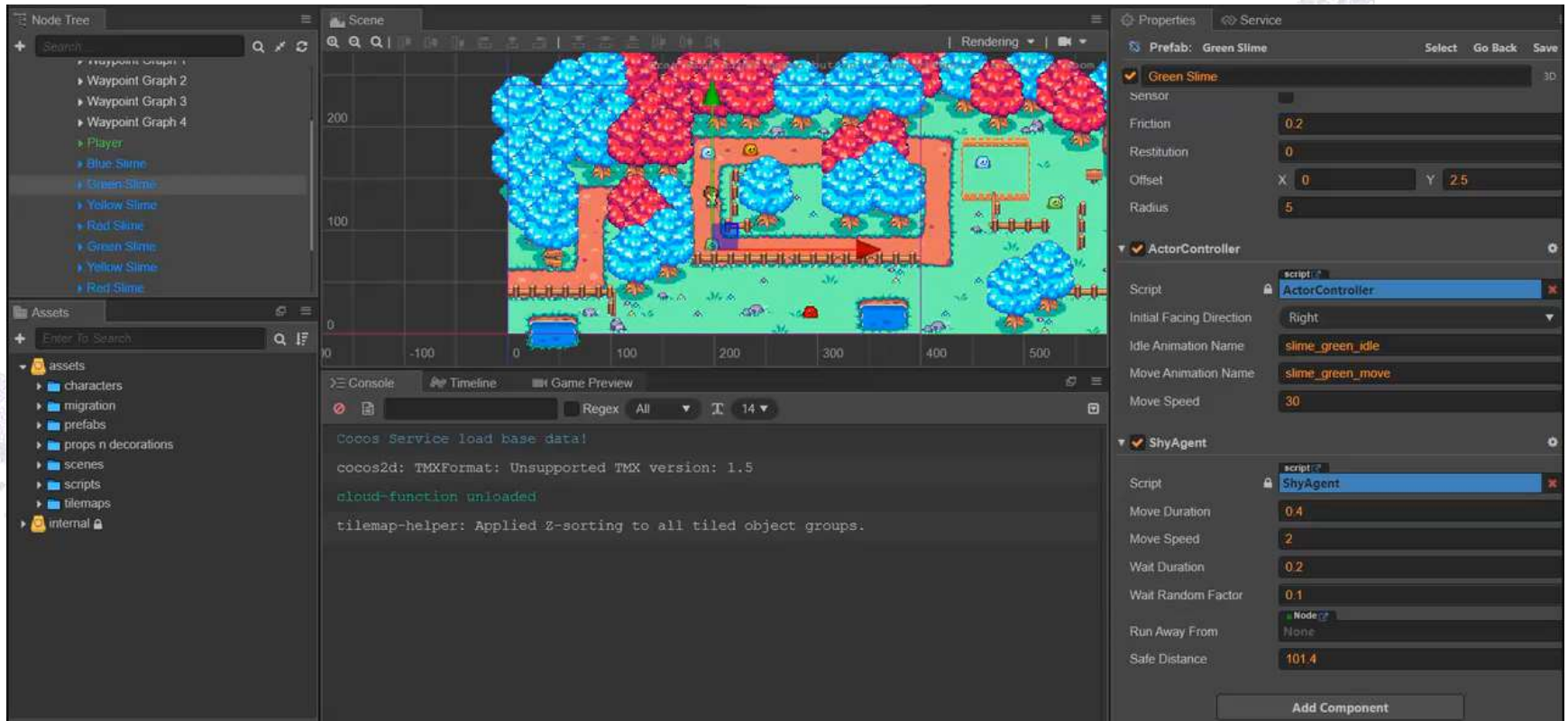
Some important concepts

- **update(dt: number):** Called every frame in the game. **dt** is time passed since last frame.

```
protected update(dt: number) {  
    this.agentUpdate(dt);  
}
```



Some important concepts



You can drag a node that has a certain component into the inspector to assign a **reference** to that component.



Hints

- You can complete every item **without** checking the Cocos Creator API.
 - Use the methods we have prepared for you!
- You'll mainly need knowledge on **TypeScript OOP**.
 - Review past lecture slides or search online if you aren't sure about the syntax!
- Read the **comments** to understand what each class property and method is for!



Grading Policy

Content	Score
Yellow slime uses the NavWanderer strategy properly.	40%
Red slime uses the NavChaser strategy properly.	20%
All slimes are turned to prefabs and instantiated in the scene according to TODO5.ts .*	40%

Deadline: 2024/04/23 17:20



Grading Policy (Cont'd)

- *For TODO3, you will receive score if the slimes **you have implemented** are behaving properly.
 - Ex: If you failed to implement red slime, we will only verify blue, green, yellow slimes' prefabs.
- **If none of these slimes are behaving properly, you won't receive any marks.**



thank
you!

Question

