CCP: Configurable Crowd Profiles

Andreas Panayiotou, Theodoros Kyriakou, Marilena Lemonari, Yiorgos Chrysanthou, Panayiotis Charalambous

In Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings (SIGGRAPH '22 Conference Proceedings)

Source Code Documentation

Run Trained Model

Open Project and run a simulation

1. Unity Version: 2020.3.26f1
2. Navigate to Assets/Scenes/Demo/
3. There 8 demo scenes (simulation environments) available:
    - **Hall:** Hallway_Demo.unity
    - **4-Way Crossing**: Cross_Demo.unity
    - **Circluar**: Circular_Demo.unity
    - **Museum:** MultiArea_Lage_Demo.unity
    - **Hall with 6 agents:** Hallway_Small_Demo.unity
    - **4-Way Crossing with 8 agents:** Cross_Small_Demo.unity
    - **Hall with not obstacles:** Hallway_Demo_NoObstacles.unity
    - **4-Way Crossing with Baseline Collision Avoidance model**: Cross_OnlyGoal_Demo.unity
4. You can control the behaviors of the agents in real-time using the four slider that appear on the screen.
5. Parameters including number of agents, number of neighbors, spawn type etc., can be altered and they are described in the table at the end of this document.

Training

Installation

1. Install Python
2. Create Virtual Environment
    - Navigate to project directory
    - Create environment name “venv”: *python -m venv venv*
    - Activate Environment: *venv\Scripts\activate*
    - Upgrade pip: *python -m pip install --upgrade pip*
    - Install dependencies from file: *pip install -r venv_requirements.txt*

Run a Training

1. Activate Environment (From project folder)
    - venv\Scripts\activate
2. Run simulation
    - **Manage Config file:** File config/WeightAgent_Training.yaml is the default file used in the original training. You can alter every hyperparameter to suit your needs.
    - **Start training:**
        - **Run inside the virtual environmnet:** *mlagents-learn config/WeightAgent_Training.yaml --run-id=WeightAgent_Training*
        - **Start Scene in Unity**: Scenes/Training/Square_Training
    - **Resume simulation:**
        - **Run inside the virtual environmnet:** *mlagents-learn config/WeightAgent_Training.yaml --run-id=WeightAgent_Training –resume*
        - **Start Scene in Unity**: Scenes/Training/Square_Training
    - **Use the trained model to run a simulation:**
        - **Brain can be found at:** results/WeightAgent_Training/WeightAgent_Training.onnx
        - **Insert brain to Unity**
        - **Navigate to the agent prefab:** Assets/My Assets/Prefabs/Demo/ WalkAgent_Weights_Demo.prefab
        - **Attach brain to the agent:** Insert brain to agent’s Behavior Parameters in the “Model” field**.**
    - **Tips:**
        - You can follow the training strategy that is described in Section 3.4 of the provided paper.
        - When resuming a training be careful to add --*resume*, if not, old training will get overwritten and start from 0.
        - Increase Timescale in Unity(e.x. 10-20) for faster training, if your system is capable.

3. View Training Evolution in Tensorboard
    - **Run:** tensorboard --logdir results --port 6006
    - **Browser:** http://localhost:6006/

Environment Parameters

- These parameters are used during simulation and training. They can be found in Monitor_Training script which is attached to "Environment" GameObject located in the scene hierarchy window.

Environment		
float	timeScale	Simulation time scale
int	numOfAgents	Number of Agents [2,n]
bool	spawnGradually	Spawn agents gradually instead of all together. If enabled, disable GridSpawn.
bool	spawnGraduallyNumber	Number of agents to spawn each time [1,10]
int	spawnEqually	Spread agents to spawn areas equally
bool	gridSpawn	Spawn agents in a grid pattern
bool	setAllInteraction	Set all the objects in scene as interaction
float	timeToInheritWeights	Seconds to inherit weights by the time agent enter weights zone [0,n]
float	maxDistance	Max distance can happen in the environment (0,n], used of distance normalization
bool	keepInheritDistance	If enabled, agents keep inherited weights when outside weights zone
float	inheritWeightsDistance	Radius of weights area (0,n]
Phase		
int	phase	Current phase in training
float	changePhaseInteval	Seconds to change phase [1,n]
Agents		
bool	coloredWeights	Color agents based on the weights
bool	oneEpisodeOnly	Run only one episode
bool	unlimitedEpisodeSteps	Unlimited Episode Steps
bool	disappearOnGoal	Destroy agent object on goal and recreate
Weights		
bool	extremeWeights	If enabled, you can use weights outside the limits by adjusting the min/max too.
float	goalWeight	Current goal weight
float	collisionWeight	Current collision avoidance weight
float	interactWeight	Current interaction weight
float	groupWeight	Curent group weight
float	goalMin	Goal weight minimum value
float	goalMax	Goal weight maximum value
float	collMin	Collision avoidance weight minimum value
Float	collMax	Collision avoidance weight maximum value
float	interMin	Interaction weight minimum value
float	interMax	Interaction weight maximum value
float	groupMin	Grouping weight minimum value
float	groupMax	Grouping weight maximum value

Reward Thresholds		
float	goalDistanceThreshold	Distance to set goal arrival [1,n]
float	groupDistanceThreshold	Distance to assume grouping [1,n]
float	interactionDistanceThreshold	Distance to assume interaction [1,n]
int	maxNeighbours	Max neighbors to enable group and interaction. [2,n]
Hall/Cross Environment Only		
bool	oppositeGoal	Set goal point on the opposite spawn/goal area instead of random
Circular Environment Only		
bool	circularSpawn	Spawn agents in a circular pattern in radius
ffloat	circularSpawnRadius	Radius for circular spawn (0,n]
Museum Environment Only		
bool	demoScenes	If selected start agents with goal behavior until enter weights area.
bool	multiBehaviors	Use many behaviors simultaneously
float	goalPercentage	Percentage of the agents have goal behavior [0,100], all behavior's percentages must add to 100
float	groupPercentage	Percentage of the agents have group behavior [0,100], all behavior's percentages must add to 100
float	interactionPercentage	Percentage of the agents have interaction behavior [0,100], all behavior's percentages must add to 100
Save Routes		
bool	saveRoutes	Enable BEFORE RUN to save routes
bool	stopSaving	Enable DURING RUNTIME to stop recording and save files
float	timestepInterval	Seconds between recorded points, e.g. 0.04
string	directoryPath	Save directory

For Questions/More info contact:

- a.panayiotou@cyens.org.cy