Goal Oriented Action Planning AI

PERSONAL PROGRAMMING PROJECT

FINAL PRESENTATION

BY JORDAN MARTIN

Objectives

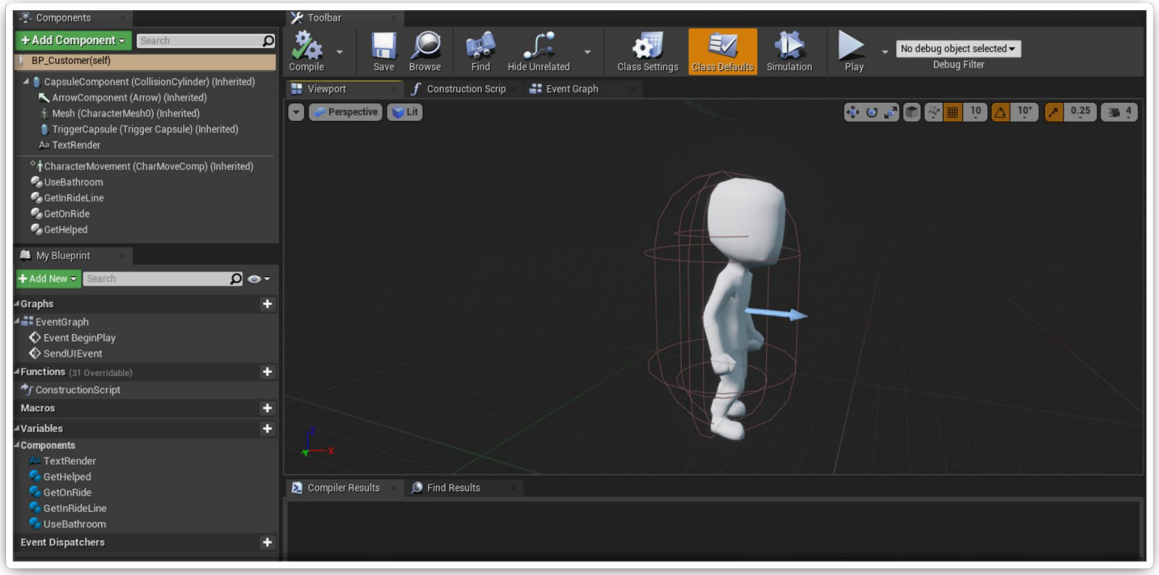
- Create a GOAP AI system in **Unreal**
- Integrate the GOAP AI system into an agent-based simulation
- Try applying the system to a theme park game demonstration

Stretch Goals:

- Incorporate game mechanics into the simulation

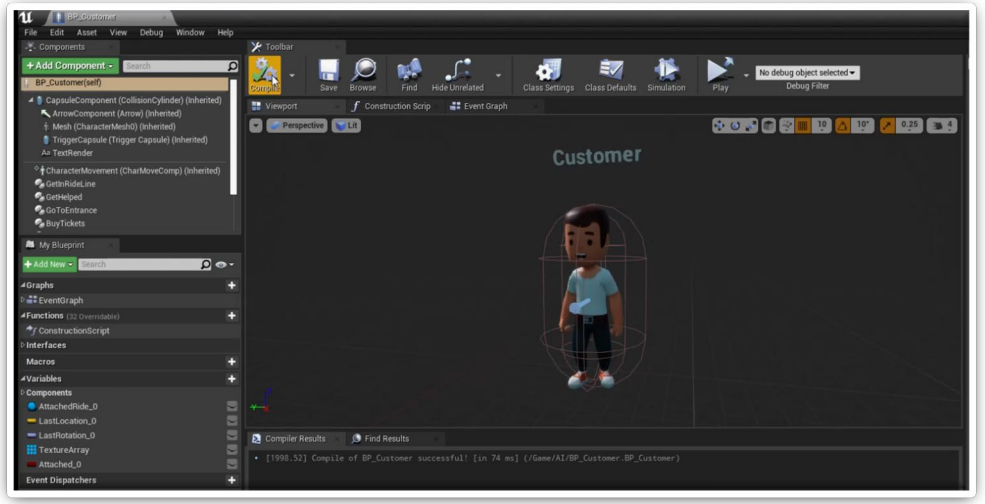
Mid-Semester Recap

- GOAP System created in C++ classes from scratch (no behavior trees)
- Multi-step plans and multiple agents
- Inventories
- Game elements: RTS Camera, UI Watch Window, Speed
- Unreal C++: Collision, Spawning, Data Structures, Delegates



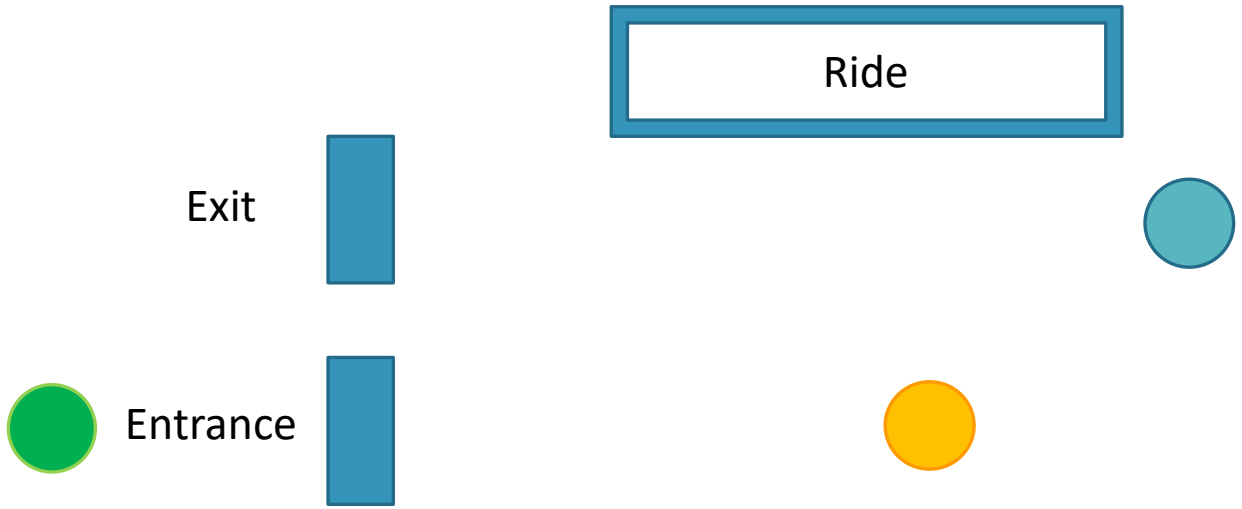
Final Progress

- Expanded on GOAP system: DuringPerform() and Animations
- Smart Objects added to Inventories
- Ride animations/socketing and Queueing system for agents
- Additional actions: Use Restroom, Take Break, Leave Park, etc.
- UI Watch Window Additions
- Unreal C++: Collision, Animations/State Machine, Socketing



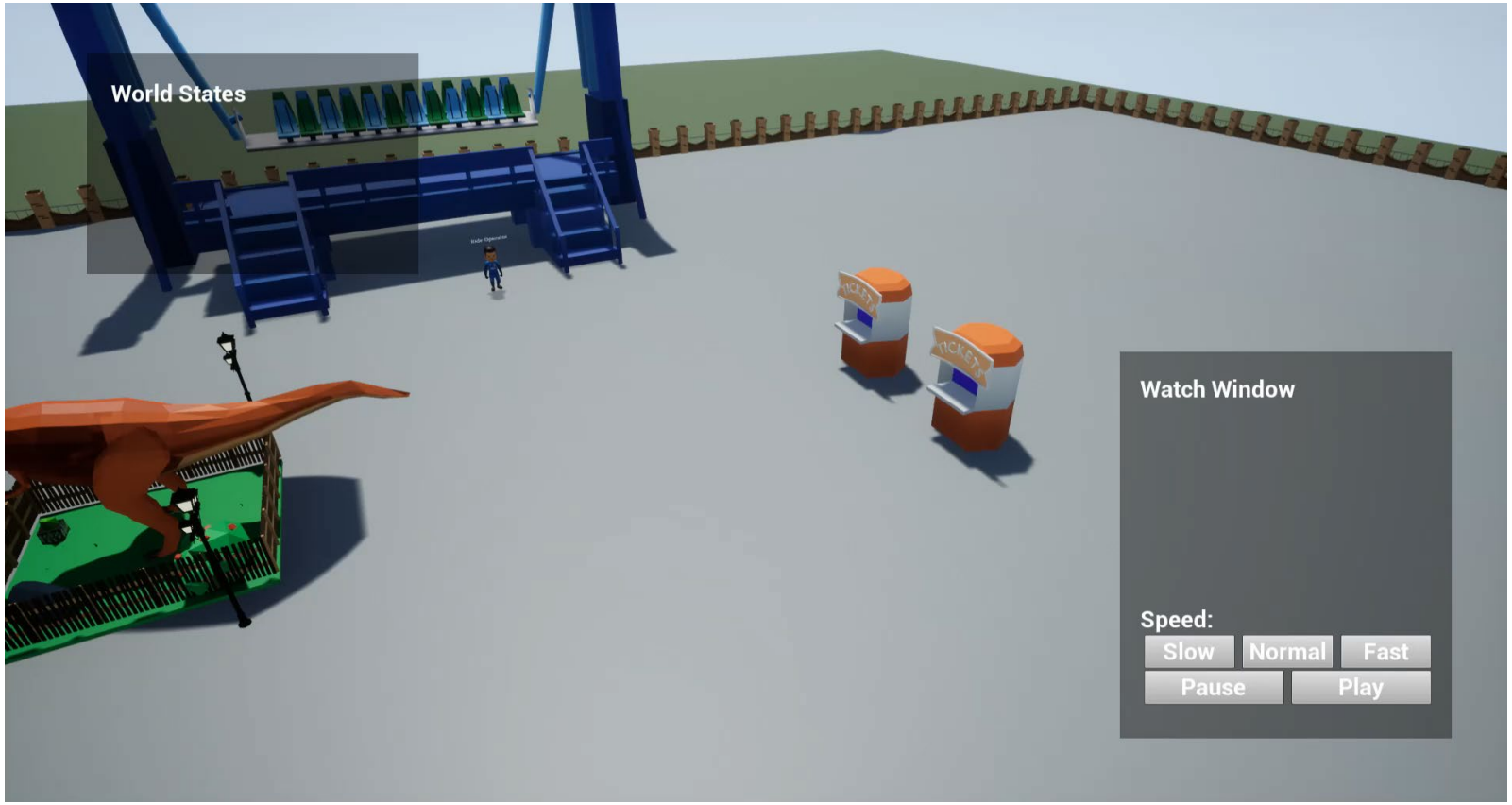
Home

Theme Park Example



Live Demo

Demo Video

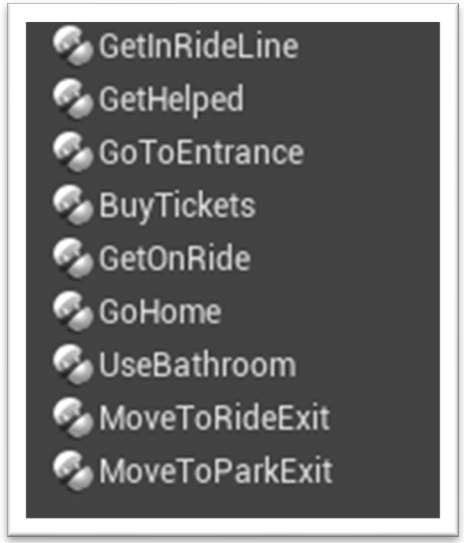

World States
Watch Window
Speed:
Slow
Normal
Fast
Pause
Play

Schedule

Goal	Week	Previous Schedule	Actual Schedule
Integrating GOAP System into Agent-Based Simulation	Week 6	**Update Presentation** • Complete multi-step plans • Flesh out UI Window • Animate State	**Update Presentation** • Complete multi-step plans • Flesh out UI Window • Animate State
	Week 7	Adding “Smart Objects” to the world • Add objects that can be used to fulfill goals	Adding “Smart Objects” to the world • Add objects that can be used to fulfill goals • Agent animations
	Week 8	Revalidation of plans • Add changes to the world state that require plan changes	Revalidation of plans • Add changes to the world state that require plan changes • Ride animations
Applications towards Games and Simulation	Week 9	Priority of Goals for agents • Execute plans while having competing priorities	Priority of Goals for agents • Execute plans while having competing priorities • Queueing multiple agents onto rides
	Week 10	Create a Theme Park Simulation • Implement all agents and actions • Integrate GOAP system into Demo Scene • Time for fixes and polish	Create a Theme Park Simulation • Implement all agents and actions • Integrate GOAP system into Demo Scene • Time for fixes and polish
	Week 11	**Final Presentations**	**Final Presentations**

Positive Outcomes

- Understanding of GOAP systems
- System benefits realized:
  - Modular actions
  - Competing priorities
  - Inventory system for both resources and items
- More experience in Unreal Engine / C++



Problems Faced

- GOAP Difficulties:
  - Difficult to debug when actions are hidden behind planner
    - Creating debug tools early on is critical
- Unreal Engine difficulties:
  - Collision, animations, socketing, communicating C++ with BPs
  - Standard library containers, Smart Pointers
- Performance issues
  - Agent continuously accesses the planner on tick if preconditions are not met

Next Steps

- Expand the system to work with shops and rides
  - Eat/drink food
  - Additional rides
- Add more of the planned characters
  - Mechanics and Entertainers
- UI Improvements
  - Inventory of each agent in the menu

Thank you!