IGDA Unity SIG II

AI in Unity

Emil "AngryAnt" Johansen
Unity Technologies

AI in Unity

- Sensors
- Decision logic
- Navigation

Sensors

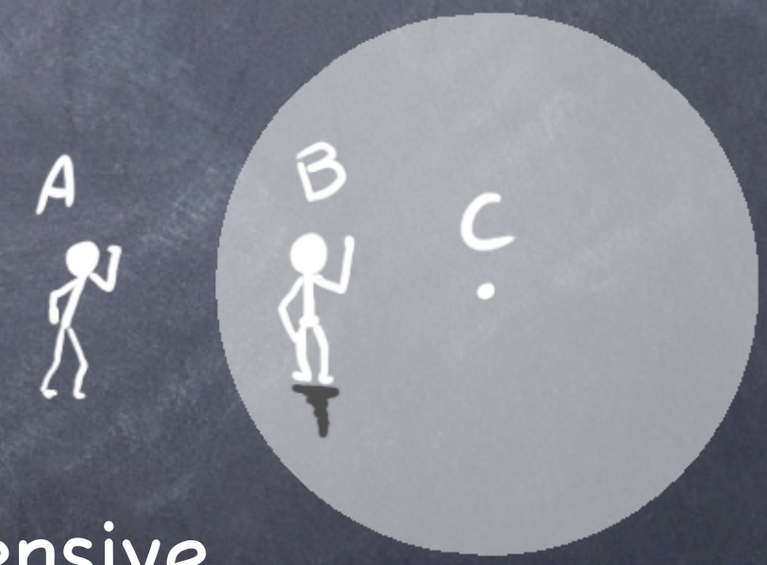
- Active
- "Passive"
- Message driven
- Second pass filtering
- Additional sensors or inclusion areas

Active

- Using Physics.OverlapSphere
- Iterating list of target objects

Using Physics.OverlapSphere

- Entities already represented in the physics simulation
- Filter by layer mask
- Cheap radius check
- Manual physics checks are expensive



Manual physics checks are expensive

- Do not check each frame
- Regular physics run at a fixed framerate of 20 fps per default
- Adapt polling frequency to when the data is needed. Setting up the sensor as a service for the behaviour logic to use is a good best practice

Iterating list of target objects

- Central registry
- Objects of interest register with a singleton OnEnable and unregister OnDisable
- Sensors filter through this list when needed
- Registry can do early sorting and grouping based on meta data
- Object.GetObjectsOfType

"Passive"

- Using physics triggers
- The physics simulation is running anyway, being clever about polling and prediction
- Requires either extra manual setup of the transform tree or initialization work
- You are served the data whether you need it or not
- Use layer filtering via the layer mask table or Physics.IgnoreCollision
- Most likely need to forward collision information from a child GO to the central sensor logic
- Radar structure

Message driven

- Central monitor
  - Polling and interpreting state changes on monitored subjects
  - Redirection of messages from clients
- Relative broadcast
  - Using active sensor to determine target audience
  - Transmitting messages to each member

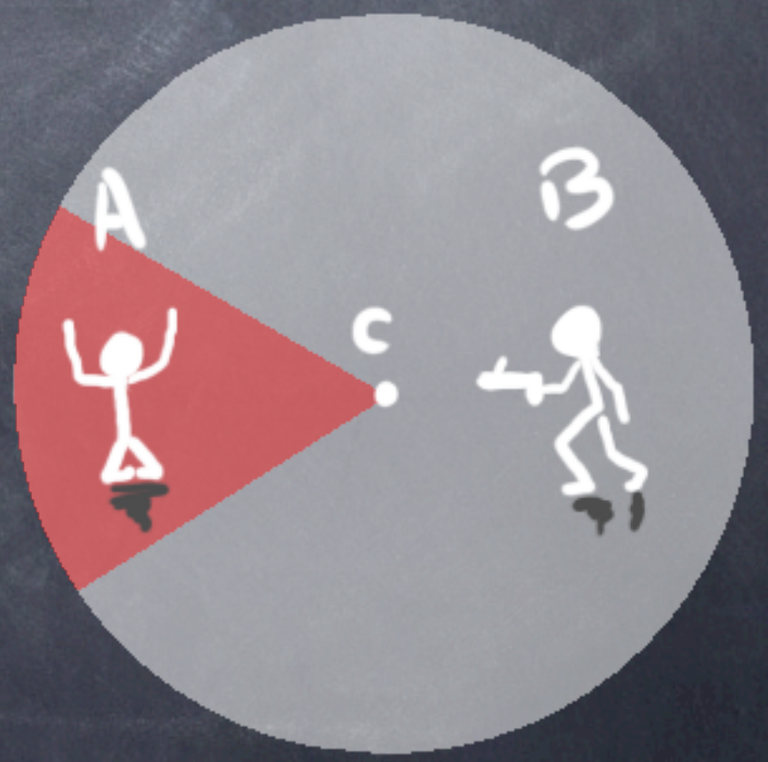
Second-pass filtering

- Limit object list based on meta data
  - Tag filtering - more expensive than layer mask, but operating on already filtered data set
  - Component based sorting - GetComponent

Second-pass filtering

- Visibility cone

```
Vector3.Angle (
      transform.forward,
      targetTransform.position - transform.position
) < coneAngle * 0.5f
```



Additional filters or inclusion areas

- Proximity overriding visibility cone
- "Audio" / event sensor
- Crowd influence - dispersal and relaxation

Decision logic

- Whatever is wrong with a bunch of nested if-statements?
- Useful state machine setups
- Available middleware

Whatever is wrong with a bunch of nested if-statements?

- It is just logic after all
- Systems handle abstraction and help keep you sane
- Does the term "spaghetti code" mean anything to you?

Useful state machine setups

- FSM combines state tracking and handling
- Simple to understand system
- Very slim implementation

Example:

enum + switch in AI update + function call

Example:

enum + delegate + dictionary in abstract class

Available middleware

- Why? Additional abstraction. Shortens the distance between idea and implementation
- PlayMaker [video]
- Behave [demo]

playMaker™
screen casts

visual scripting for Unity 3D

Getting Started:
PlayMaker Basics

hutong games

Demo: Behave

Navigation

- Setting up simple pathes
- Available middleware

Setting up simple pathes

- Linking transforms via linked list of node components
- Keep track of next node
- Adjust force or direction vector to fit - slerp
- Basic steering

Available middleware

- Aron Grandberg "A* Pathfinding"
- Path
- UnitySteer