MULTI-AGENT BASED SECRET COMMUNICATION IN AUGMENTED VIRTUALITY—CRYPTO SHOOTER

COMP 768, Fall '14

Sarah J Andrabi & Sahil Narang

MOTIVATION

- Crowd Simulation
- Secret communication and message exchange
- No TCB on devices—no such known implementation so far
- Example applications:
  - Spy applications
  - Secret message exchange in games without direct player communication
- No known applications currently

PROJECT OVERVIEW

- Pursuit and Evade Crowd Simulation
  - Unity
  - RVO2
- Objective: Identify and Tag Target Agents in the Crowd
- Game Characters:
  - Main Player: User controlled agent
  - Secondary Agents: RVO Simulated Agents
    - Target agents
    - Non Target agents

PROJECT OVERVIEW

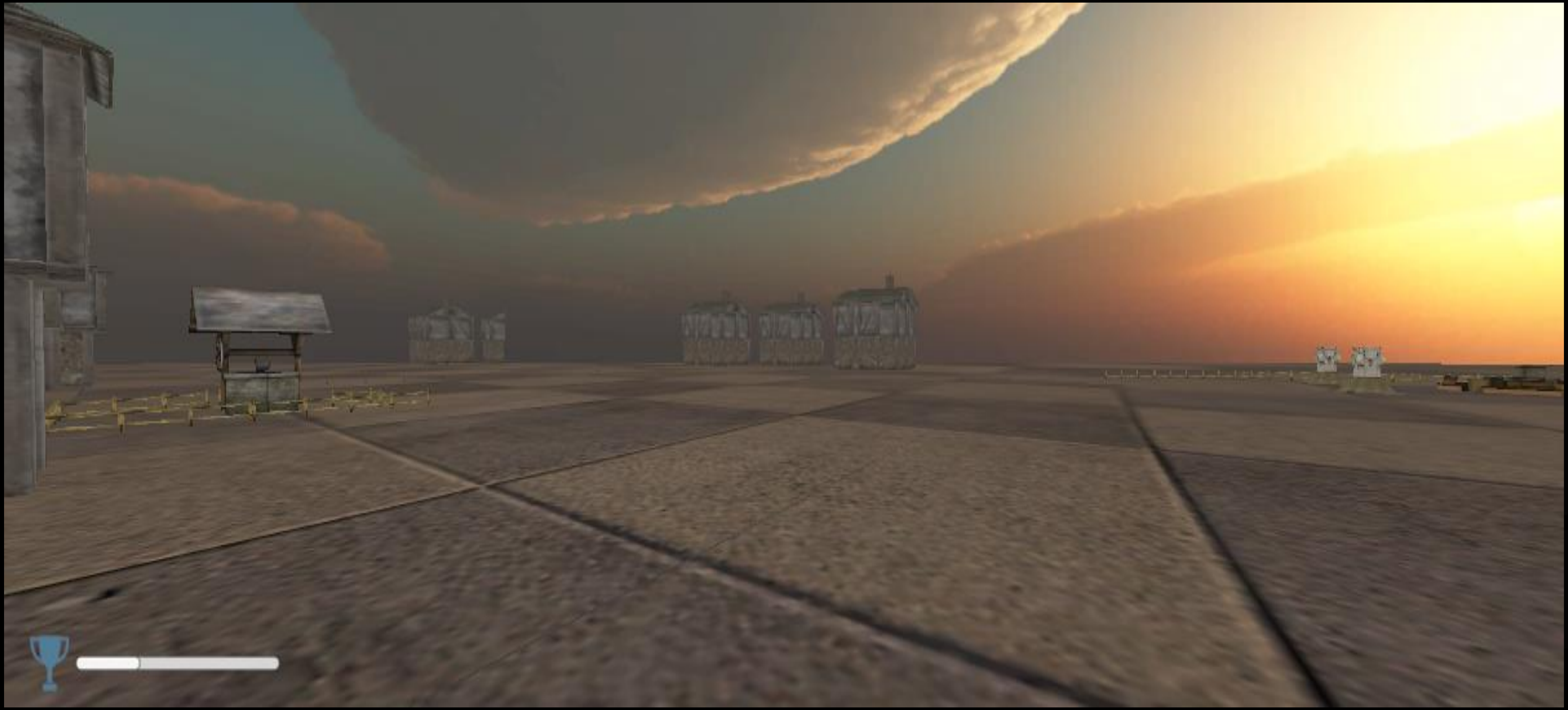
- Pursuit and Evade
  - Main player attempts to ‘catch’ simulated agents
  - Simulated agents evade
- Identify Target Agents
  - Isolate a simulated agent
  - Align main player’s visual share with secondary agent’s visual share
  - Decoding of aligned images left to the user
- Tagging Behavior
  - Initiate tagging motion and Collision Detection

GAME COMPONENTS

- Scene design
- User controls for player
- Motion Models for player and agents
- Planning for agents
- Player-Agent Interaction
  - Suspend
  - Identify
  - Tag/Shoot
- Scoring
- GameOver

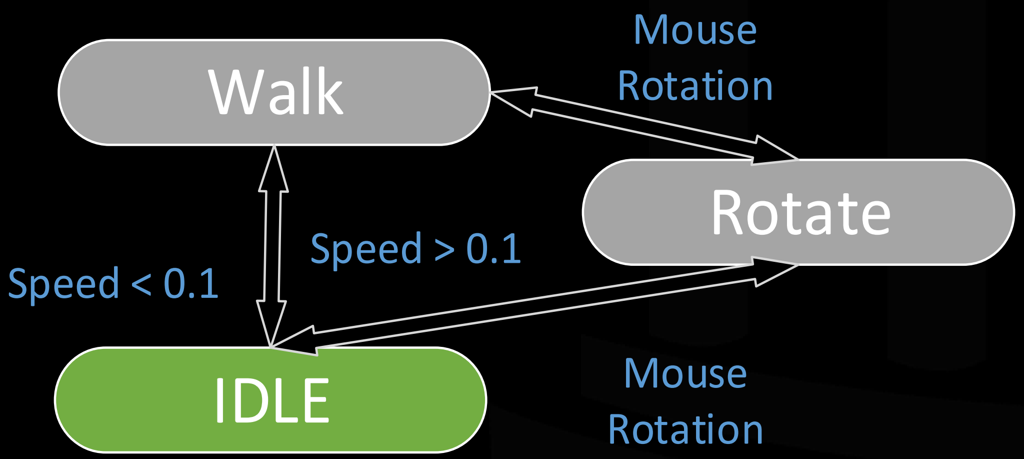
GAME SETUP

- Scene design



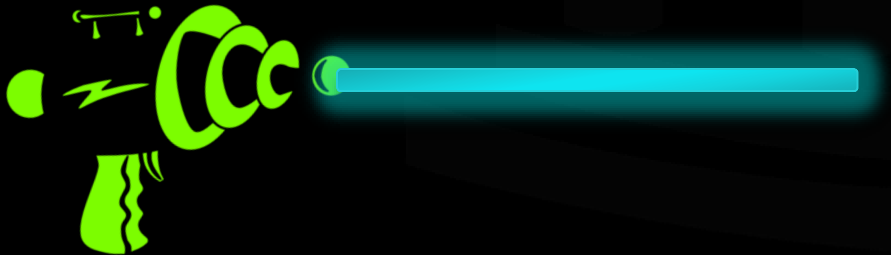
GAME SETUP

- Main Player
  - User controls
  - Animating player motion



GAME SETUP

- Main Player
  - Camera Control
    - Look at what player looks at
  - Player gets a visual share
  - Player shoots
    - Gun
    - Laser Bolt



GAME SETUP

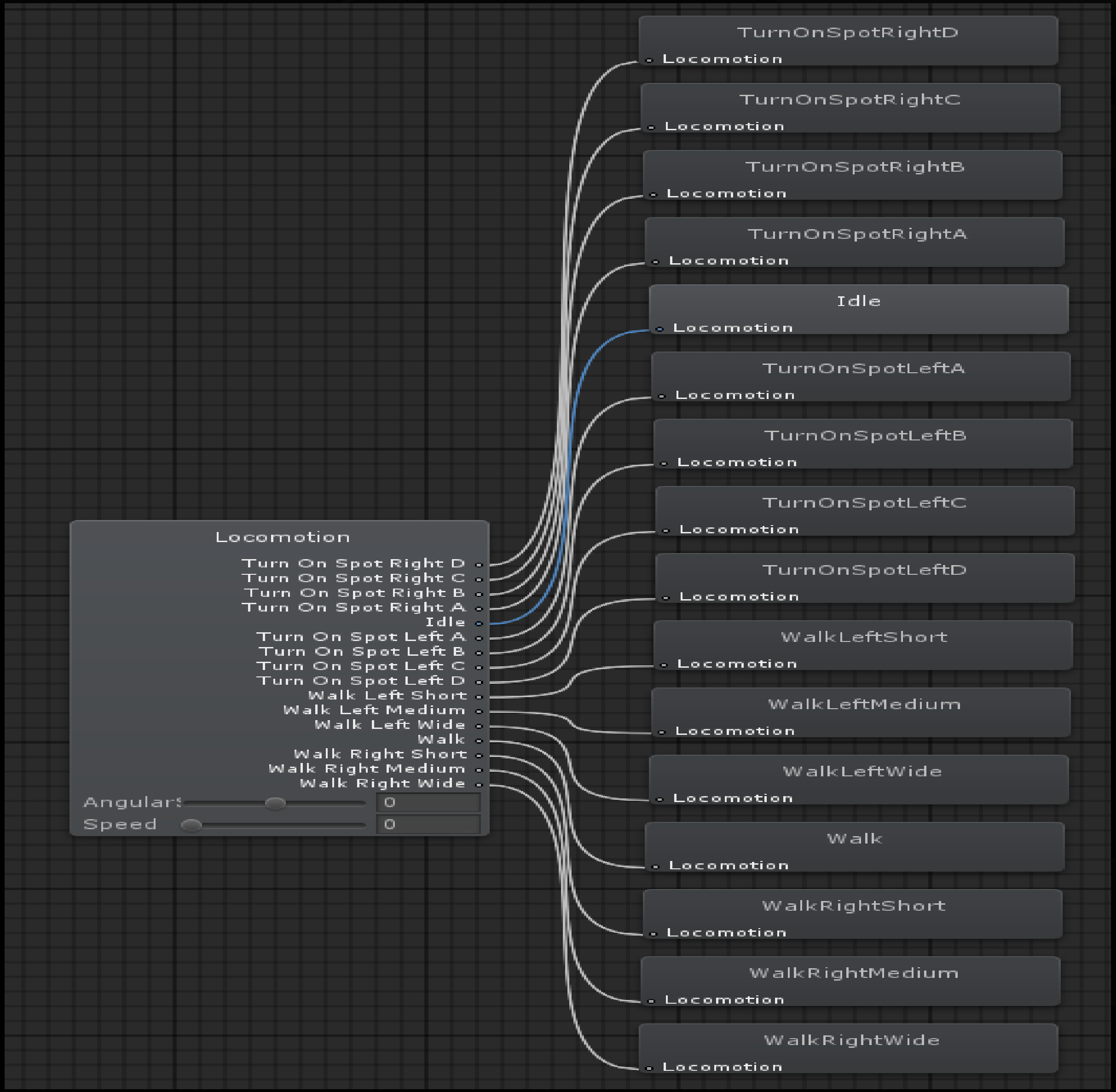
- Secondary Agents
  - Animating player motion
  - Randomly gets one of two visual shares

GAME SETUP

- Secondary Agents
  - Motion Models
    - Idle
    - Walking—Based on speed
      - Short/Medium/Long steps
    - Turning—Based on angular speed
      - Left and right turn

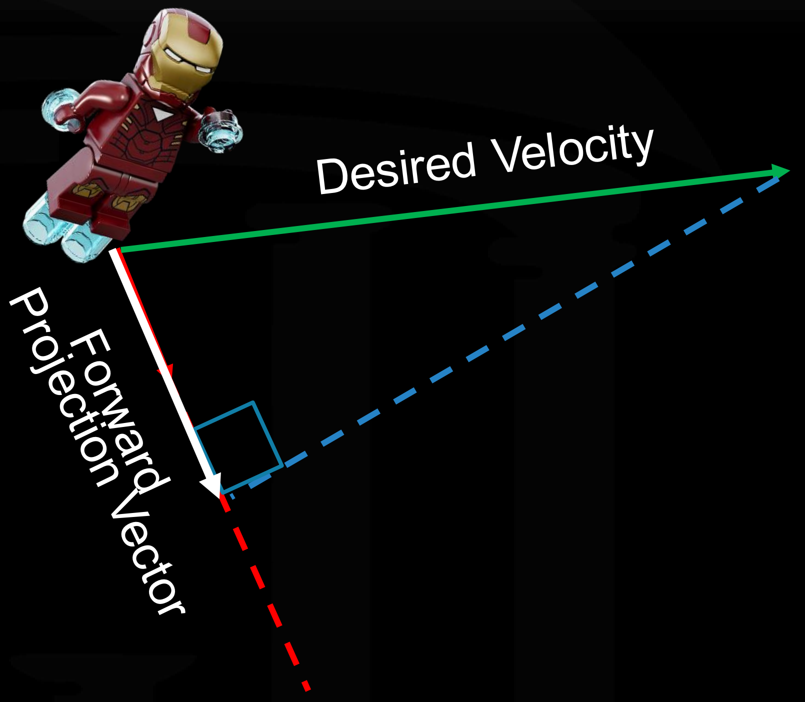
GAME SETUP

- Secondary Agents



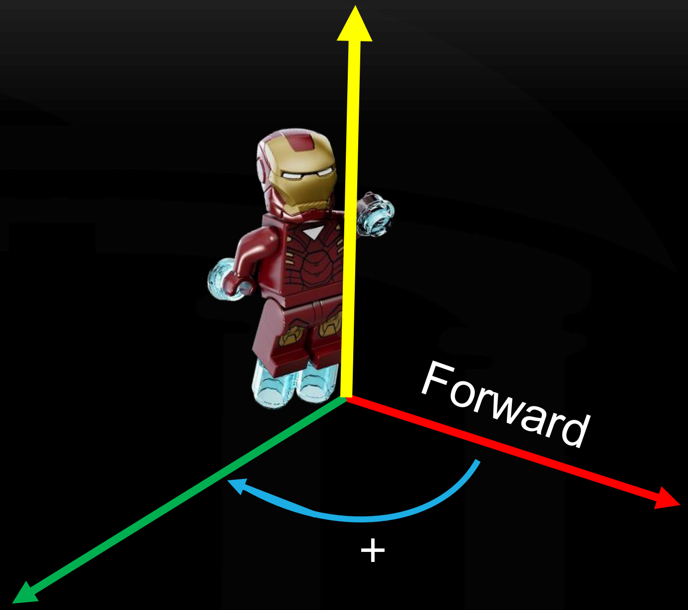
GAME SETUP

- Secondary Agents
  - Moving
    - Which direction to move in
    - Speed← Magnitude of Projection Vector



GAME SETUP

- Secondary Agents
  - Moving
    - Which direction to move in
    - Speed← Magnitude of Projection Vector
  - Turning
    - Wide turns
    - Short turns
    - Don't want snaky motion



GAME SETUP

- Synchronizing RVO and Unity
  - Agent Initialization
    - Add to RVO Simulator
    - Spawn in Scene
  - Obstacle Initialization
    - Add to Scene
    - Add and process in RVO Simulator
  - Scene Layout
    - Construct Roadmap
  - Assign Plans

SIMULATION LOOP

- Get main player’s position
- Update roadmap
- Set preferred velocities for simulated agents
- Get collision-free current velocity for each agent using RVO
- Animate each agent to move with its current velocity
- Reset roadmap

PLANNING FOR SIMULATED AGENTS

- Do Nothing
  - If suspended OR tagged
- Set Preferred Velocity
  - If player is visible AND within range AND heading towards the agent
    - direction = player's orientation
    - Start node = closest visible node in direction
    - Goal node = farthest visible node in direction
    - Path = getPath(roadmap, start node, Goal node)
  - Else if not at Goal node
  - Else
    - Start node = Goal node
    - Goal node = random goal
    - Path = getPath(roadmap, start node, Goal node)
- Get collision-free velocity from RVO Simulator

MOTION MODELS

- Agents
  - Desired Velocities provided by RVO
  - Figure out speed from them
  - Feed them to the Unity Mecanim
  - Get appropriate walking/turning/idle animations

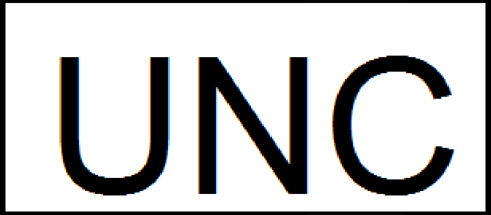
PLAYER-AGENT INTERACTION: OVERVIEW

- Suspend and Stop agents
- Identify
- Tag/Shoot

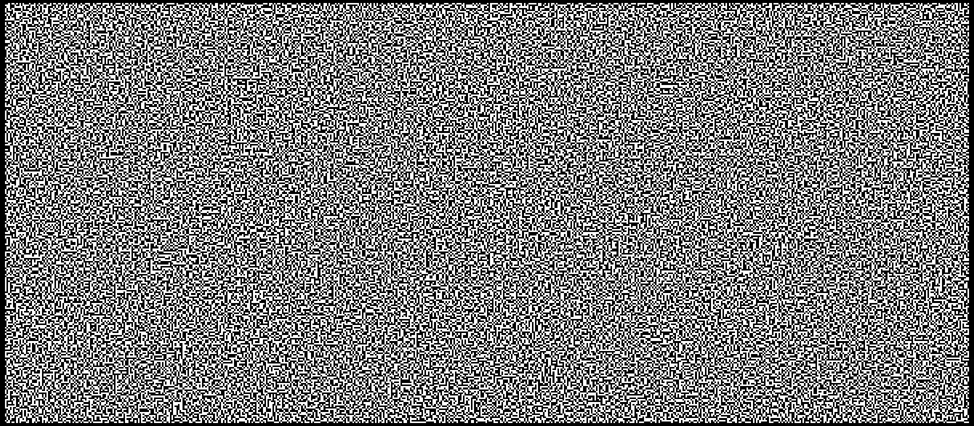
PLAYER-AGENT INTERACTION: SUSPEND

- If agent enters player’s ‘collider’
  - Set state to “suspend”
- While agent state == suspend
  - Skip RVO Planning
- If agent exits player’s collider AND is alive
  - Reset state

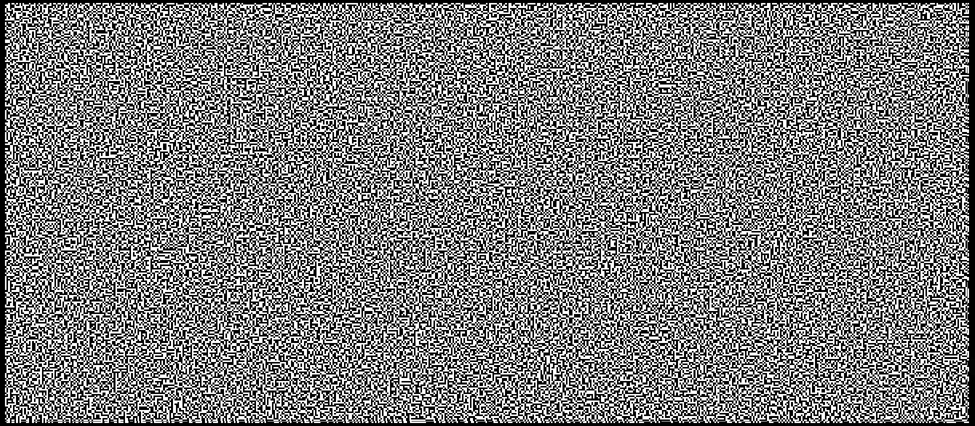
VISUAL CRYPTOGRAPHY



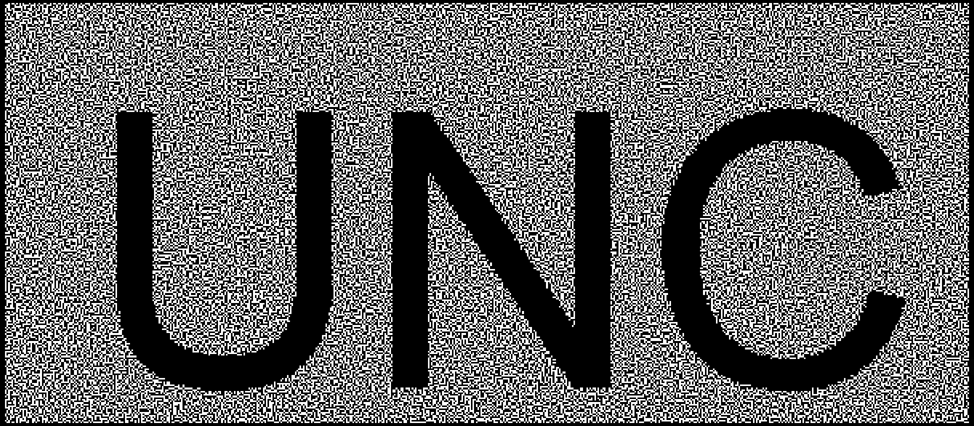
VISUAL CRYPTOGRAPHY



VISUAL CRYPTOGRAPHY

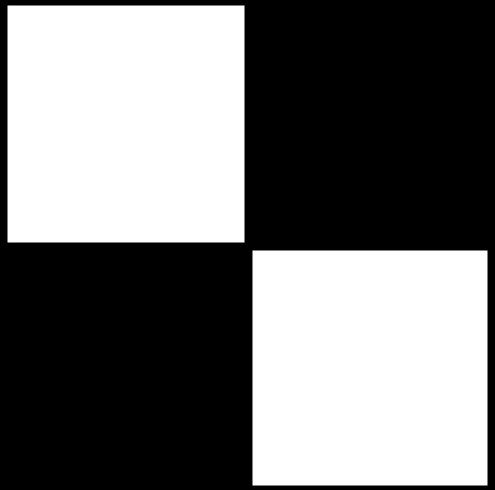


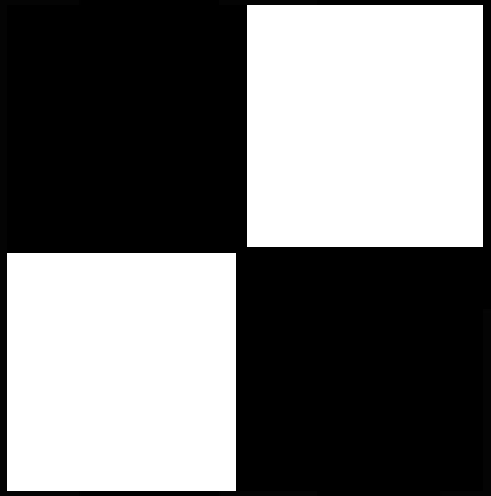
VISUAL CRYPTOGRAPHY



PLAYER-AGENT INTERACTION: IDENTIFY

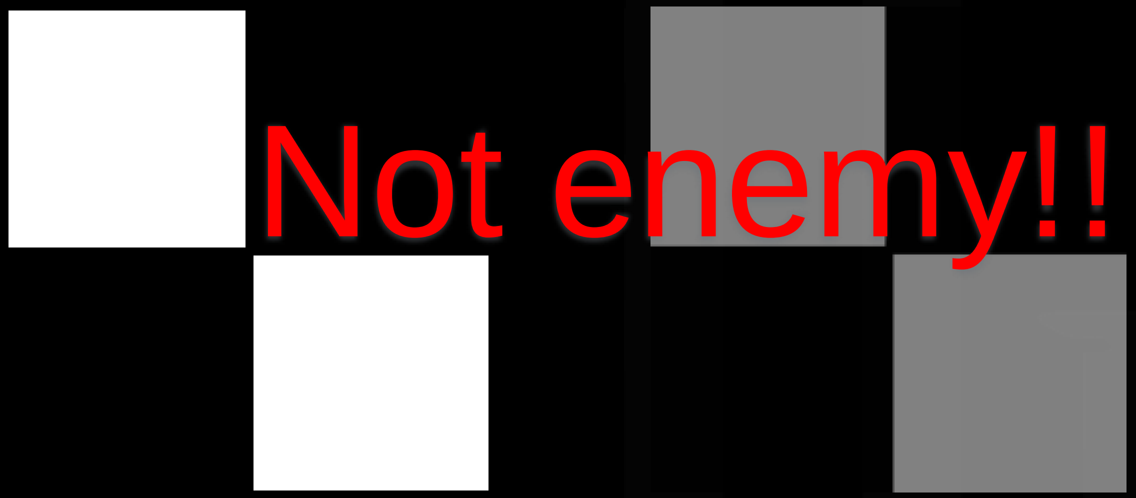
- Visual Shares


Share 1


Share 2

PLAYER-AGENT INTERACTION: IDENTIFY

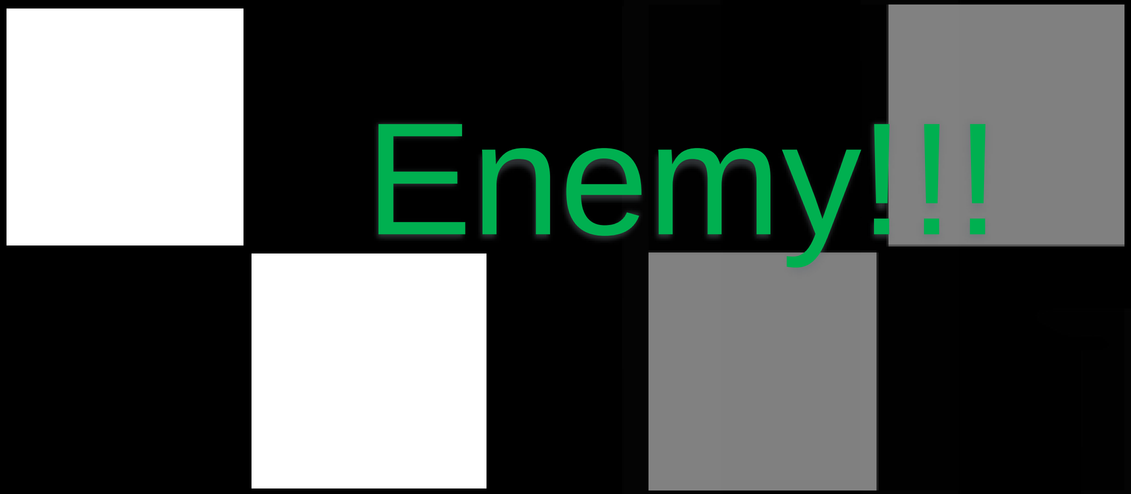
- Visual Shares



Share 1 + Share 2

PLAYER-AGENT INTERACTION: IDENTIFY

- Visual Shares



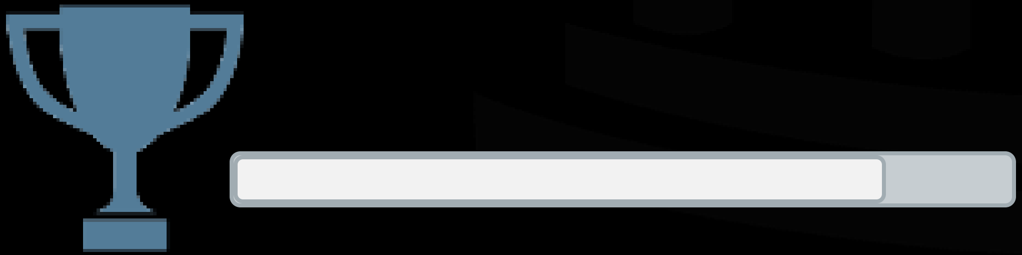
Share 1 + Share 2

PLAYER-AGENT INTERACTION: TAG/SHOOT

- Once identified the enemy
  - SHOOT!!
  - Laser Ray Casting
- Enemy's dying animation
- Let Unity and RVO know agent has died

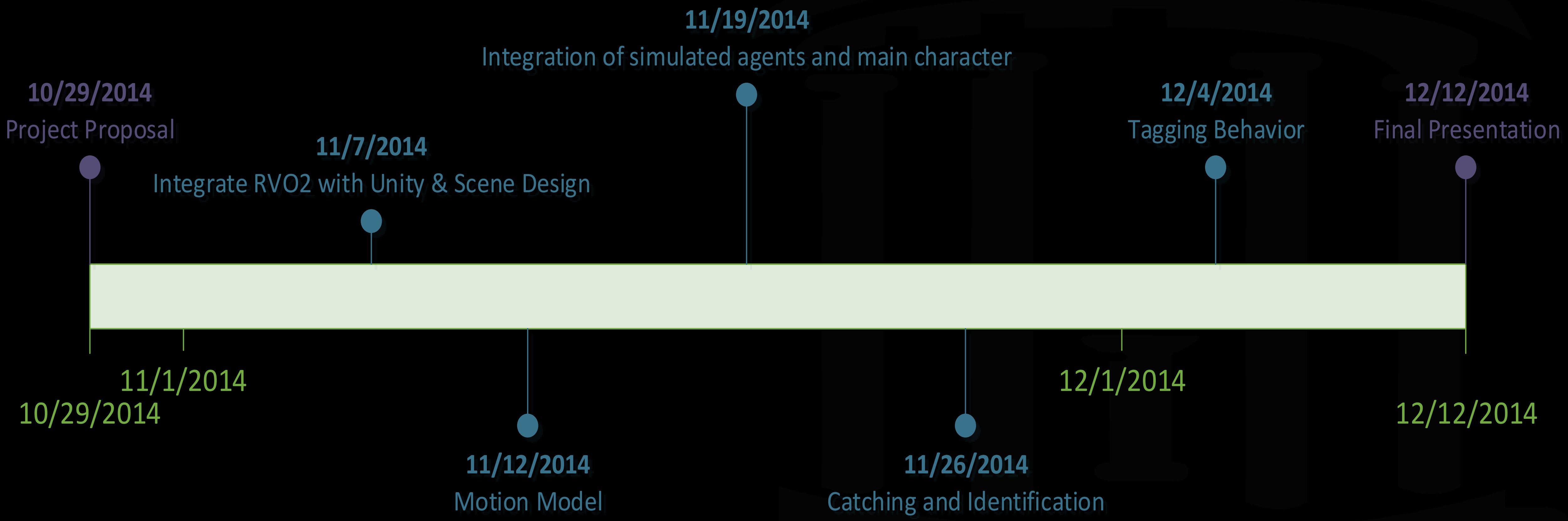
Scoring/GameOver

- Plus Points for shooting the correct enemy
- Minus for the incorrect one
- Game indicates whether shoot correct or incorrect enemy
  - Screen flashes
- Game ends when
  - Max Score—You Win!!! ☺
  - Min Score—You Lose!! ☹



DEMO

TIMELINE


11/19/2014
Integration of simulated agents and main character
10/29/2014
Project Proposal
11/7/2014
Integrate RVO2 with Unity & Scene Design
12/4/2014
Tagging Behavior
12/12/2014
Final Presentation
11/1/2014
10/29/2014
12/1/2014
12/12/2014
11/12/2014
Motion Model
11/26/2014
Catching and Identification

FUTURE WORK

- Make Visual shares more complex
- Integration with Oculus
- Inclusion of complex environment & mapping
- Better Global planner
- More polishing

QUESTIONS??