

机器智能实验

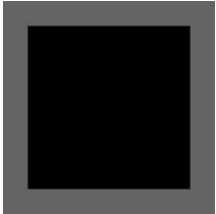
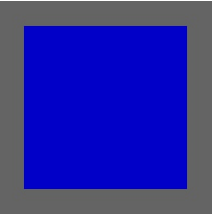
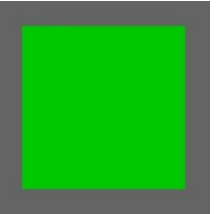
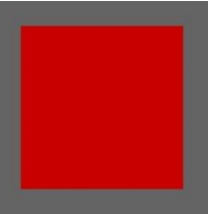
智能 Agent 感知-推理-行为实验报告

实验 1: 实现智能设备, 根据环境状态自动切换状态。实验代码为 `sense_and_change.txt`。

实验步骤:

1. 创建智能设备的三维形状和纹理

设备的三维形状设定为正方体, 其中五面为灰色, 剩余一面插入纹理, 设备共有四种状态, 对应四种纹理。

纹理				
状态名	off	default	happy	angry

2. 用 `llSensor` 等函数设计感知对象和事件, 根据感知对象和时间的属性改变自身的纹理属性特征。每个状态的 `state_entry()` 函数中均有改变为相应纹理的操作。

- 在 `default` 状态中使用 `llSensor` 感知是否有 AGENT 出现在距离自身 3m 以内, 若有则进入 `happy` 状态。
- 在 `happy` 状态中使用 `collision` 感知碰撞, `touch_start` 感知交互, 若碰撞则进入 `angry` 状态, 若交互则进入 `off` 状态。
- 在 `angry` 状态中用 `llSetTimerEvent` 和 `timer` 感知时间, 处于 `angry` 状态 10s 后进入 `off` 状态。
- 在 `off` 状态中用 `touch_start` 感知交互, 若进行交互则进入 `default` 状态。

3. 在 `Second Life` 中进行测试。下图展示了 AGENT 靠近时进入 `happy` 状态的功能。



实验 2：实现基于规则的简单反射性 Agent。实验代码为 simple_reflex_agenta.txt 和 simple_reflex_agentb.txt。

实验步骤：

1. 创建 Agent 的三维物体 Prim，用白色、黑色的正方体分别代表两类 Agent，名为 AgentA 和 AgentB。

2. 根据以下规则编写状态

- 在 Sensing 状态中有一个 Sensor，将 Sensor 感知到的所有 AgentA 和 AgentB 用一个 List 储存。
- 在 Reasoning 状态中，利用 Sensing 状态中所得的 List，计算当前 Agent 与其感知到的所有 Agent 的相对位置，并计算它们之间的“力”，同类之间为“引力”，异类之间为“斥力”，这样可以使得 Agent 向同类移动而远离异类。“力”的大小按照距离反比计算，将计算出来的各个“力”进行向量求和，从而可以支持“远近”“多少”等模糊规则。
- 在 Action 状态中，使用 llMoveToTarget 函数进行 Agent 的移动。

3. 在 Second Life 中进行测试，拷贝生成多个物体对象，检查其运动规律。以下两幅图分别是 Agent 的初始位置与运动终止后的位置。可以看出，确实符合“识别同类为目标并移动，同时远离非目标对象”的运动规律。

