COSC 729 – Virtual Reality (Spring 2015)

SearchSwarm

A Virtual Reality Game

Weeam
Marvin Conn
Kenneth M'Balé

2015

Bowie State University

Project Information

SearchSwarm is a single-user memory game. Pairs of objects are scattered throughout the environment. The objective is to find as many matching objects as possible within a predefined time period. The single player controls one bot and competes against a swarm of bots with the same objective. The automatic bots uses various search strategies to find the pairs of objects. The project team members are Weeam, Marvin Conn and Kenneth M'Balé.

Contents

Rules of the Game

When the game starts, the user can specify the number of bots. A bot automatically begins to search the environment for matching shapes. Under automatic control, each bot follows its own search pattern across the environment. A bot can only see objects within its own field of view. The faster the bot moves, the narrower the field of view. Once sighted, a bot must move close to or collide with the object to touch it.

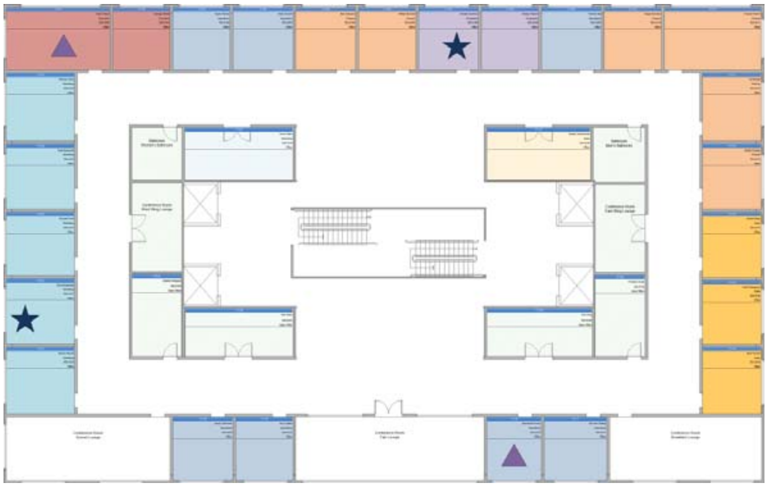
When a bot finds an object, it must touch it and then touch the matching object, resulting in a match. The touch is done with a mouse click or by colliding into the object. Each match results in the objects vanishing from the environment and a reward chime being sounded. The bot scores one point. When a bot touches an object, a timer begins that allows the bot or the player up to thirty (30) seconds to complete the match. If the timer expires, the game relocates both objects to a new location in the environment. The game ends when all objects are matched or the race timer expires. The winner is the bot with the most points.

A human user can take control of an avatar to participate in the game. Once an avatar is under human control, it is no longer operating automatically. The following keyboard commands are available:

Key	Action
W	Speed up
S	Slow down
⇐	Step left
⇒	Step right
⇓	Step back
⇑	Step forward
A	Look (Pan) left
D	Look (Pan) right
Mouse	Touch/Match the object (Optional or Collide)
Q	Full stop
X	Menu: Maze, Objects, Bots, No Timer, Reset, Quit

Top-View of the Environment

The following diagram is a depiction of an office space or class room space wireframe. In the game, the perspective will be three dimensional and the user will be immersed in the environment:



In this example, the game randomly placed two object pairs in the office; a star and a triangle. When a bot finds the star, for example, if it touches it, it has 30 seconds (based on the size of the environment) to find the matching star. A bot under automatic control does not touch the first occurrence of the object. It touches the second occurrence and backtracks to the previous object's location.

While we envision using an office environment, other types of environments can also work. The office provides several obstacles to navigation and to line of sight of the objects. As a result, the game is challenging to both automated and human-controlled bots.

Implementation Details

In this section, we address specific implementation requirements from the project specification in the syllabus.

Lights: Since the environment is an office space, rooms will have varying light sources. Some rooms will be dark while other will be normally light. The light will affect the visibility of the objects in the room.

Timers: The principal use of a timer in the game involves the matching interval. Once an avatar touches an object, a timer begins. The matching object must be touched before the timer expires in order to remove the object pair from the environment. If the timer expires, the game relocates the object pair. The second important use of the timer is the race. There is a fixed amount of time before the game ends and the scores are tallied.

Keyboard: The keyboard functionality enables the user to select and control the behavior of the avatars.

Audio: upon occurrence of the various events during the game, sound will be used to immerse the user. For example, when the user finds matching objects, a reward chime will sound. As the bots search the rooms, if they bump into walls or other avatar a collision sound will chime. When a bot wins the game, a winning chime will sound.

Sensors: The game relies on proximity sensor associated with the object. When the avatar is sufficiently close to the object, the object becomes visible. When the avatar is closer still, the object's change in behavior indicates that the avatar can touch it. An automatic avatar touches the object automatically. A manual avatar requires a keyboard command to touch the object.

Collision: The bot can collide into walls and furnishings. An automatic bot routes around the obstacle without user intervention. A manual bot stops.

Avatar Animation: We plan to use two animations. One animation will depict sadness when a match fails. Another animation will depict happiness when a match succeeds.

Stretch Goals

Our stretch goal is to allow the user to select the environment and the number of object pairs.