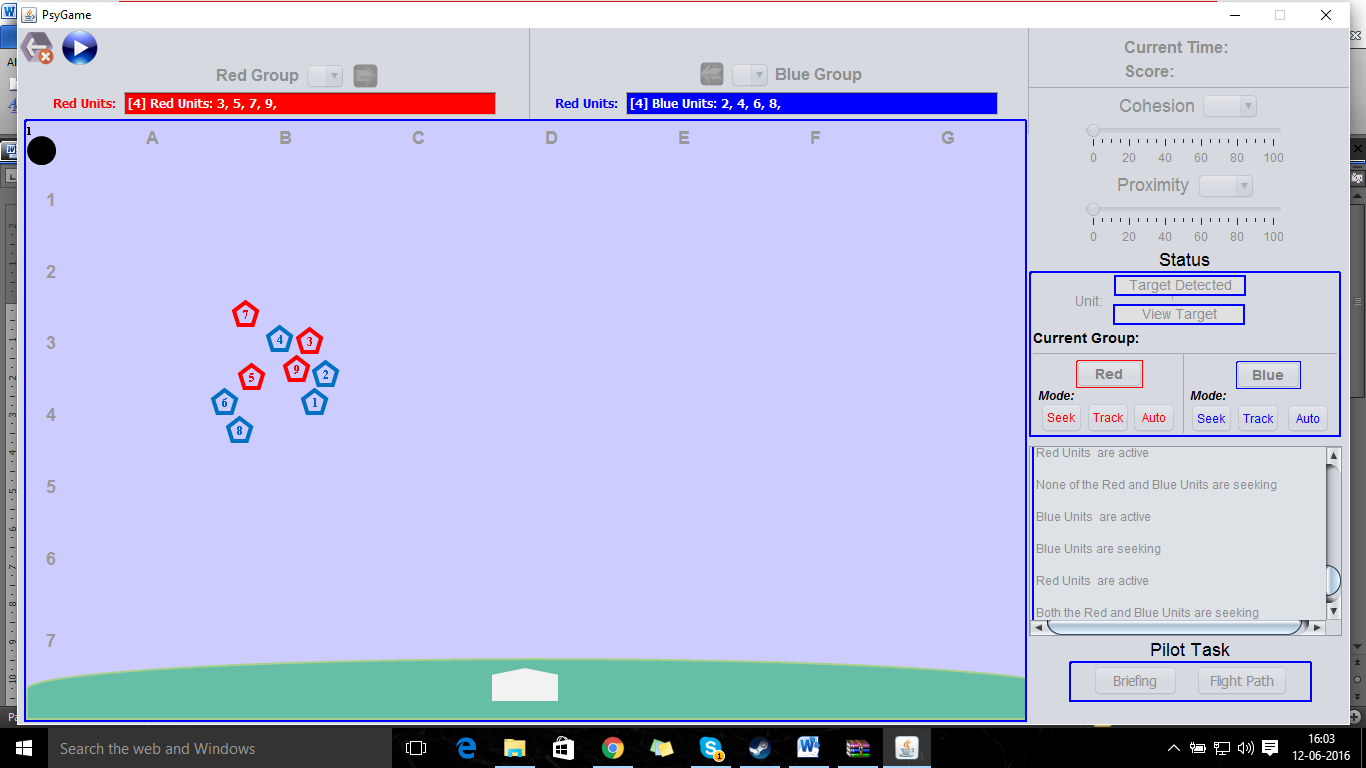
RESOURCE MANAGEMENT GAME

# Purpose:

This game is designed to measure the ability of the users to use their effective resources to their maximum potential. Users are provided with different scenarios to identify different types of targets. The ability of the users to identify and track different targets with the maximum utilisation of available resources under stipulated time is measured and it is tagged to a score. Each score is a reflection of user’s smart interactive thinking with the application.

# Game Rules:

Here is a small representation of the game in as a screenshot.

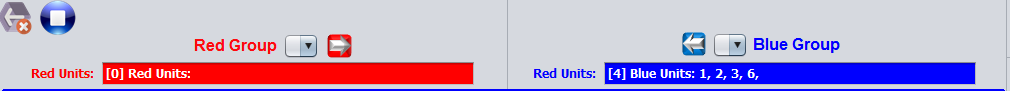


## What is a Group?

A Group is an object controlled by the user and contains a number of different units. The group defines the Search Area that units move within, the Tracking Mode of the units, and potentially other unit properties and behaviors.

The Game is divided into two groups:

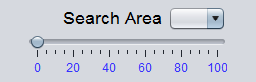
1. Red Group
2. Blue Group



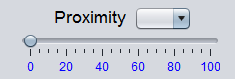
The user has control over both the groups. Each group may contain a minimum of one unit to a maximum of ‘n’ units, where the limit ‘n’ is set by the administrator. Each group may contain a different number of units. From the fig above, we can see the red group having 4 different units and the blue group having 5 different units.

Here are some characteristics for a group:

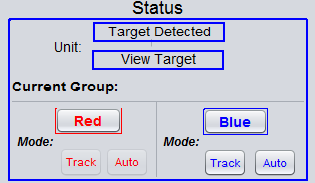
1. Search Area: Defines how large of a circle (the search area) the units may move within. Admins are able to turn off this feature.



1. Group Proximity: Defines how closely units will try to stick together while moving. A high proximity group will result in units moving together with other units in the group. A low proximity group will result in units moving more independently. Admins are able to turn off this feature.



1. Group unit count: Defined as the number of units within the group. All units within the group follow the same defined group behavior rules. Admins are able to turn off this feature by setting the number of units.
2. Group location: Defined by the center of the search area.
3. Tracking Mode: Defined by how the units interact with any targets that they detect.
   1. Seek mode: Default tracking mode. Units move within search area and do not change behavior after detecting a target.
   2. Track mode: Units will move to follow a target that has been detected, ignoring the search area boundaries. Requires user input to activate
   3. Auto mode: Automatically switches the group from seek mode to track mode once a target is detected.

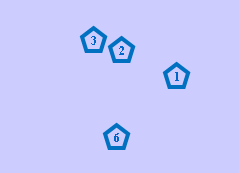


## What is a Unit?

A Unit is an object responsible to detect a target. Each Unit has different characteristics defined by the administrator.

Here are some of different characteristics for a unit.

1. Detection Range: It is defined by the area surrounding each unit within which it can detect a target.
   1. During relocation, detection range has a separate parameter.
2. Mobility: It is defined by maximum speed of the units to move inside a search area.
   1. Speed is now divided into a search speed and a relocation speed
3. Reliability: It is defined by how likely a target that is within the detection range will be detected.



Units with 100% reliability, very fast mobility and large detection ranges can easily predict the targets. In usual scenarios, such units are not provided.

The game is loaded with a standard set of units which are set by the administrator. The game provides different scenarios where each scenario may be different by the characteristic of number of search areas and the units which the user is provided with.

The user can toggle the units from one group to another. The user will be able to increase or decrease the proximity of the units inside a given search area (see Groups above). They can either fly very close or very far among themselves depending upon the position of the target. At any given point of time, the user should be able to see the search area boundaries to get an idea of how far the units can fly within the given search area (using the overlay feature below). The boundaries will be displayed only upon users request by right clicking and it will be for a brief period for the user to analyze the boundaries and play accordingly.

Once a unit, detects a target, the user can choose to track the target again and again in order to gain more points. The scoring criteria for the game are provided below.

## Additional Unit Adding Limits

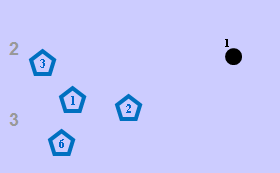
Number of units is defined by the administrator and the user usually has no control over this. However, in some scenarios you may be able to add units by spending Resources from the Resource Pool. The number of units would be defined by the following limits. There are two different limits for the total number of units in a given scenario.

1. Soft Limit: The user can choose to exceed the number of units set by the administrator, but he/she has to spend his available resources to buy subsequent units.
2. Hard Limit: The maximum number of units set by the administrator.

In most cases, the hard limit and the soft limit will be the same. More the number of units, more will be the probability of target detection which is directly proportional to increase the score.

## What is a Target?

A Target is an mobile object which flies around the entire available space. It is of primary concern to the user. The targets will only be visible if it comes inside the detection range of an unit. Once the user detects the visibility of the target, the user has to view the target and mark its identity.



Each game is provided with at least one target to a maximum target limit set by the administrator. Generally, user will be set with multiple targets to identify and track. There are certain characteristics which differentiates one target from another.

Here are some characteristics for the target:

* 1. Target identity
     1. True identity - It represents the levels of priority for each target.
     2. Marked status - It is the user’s perspective of marking the true identity. If the marking status and the true identity matches, the user gets the marking bonus points. Marking a target may have additional graphical changes of the target on the map.
     3. Image - Each Target is represented by an image.
     4. Text - It gives a brief description about the target
     5. Graphical representation - the image that represents the target on the map
  2. Target mobility
     1. Speed- It defines the measure of mobility for each target. Each target could travel at different speeds and through different itinerary.
     2. Behavior - It is about the trajectory of the target. For example, one target might move in a hexagonal direction and the other might move in a triangular direction.
  3. Scoring Criteria ( First Contact Points, Marking points, Tracking Points) - It is defined by the differences in the points based upon the priority of the target.

Target properties set in the config file:

id= What the game refers to the target as

imagePath= image used on the map

Priority = Initial priority assignment

trueIdentity= Actual priority/id assignment

status= active or inactive. Is the target currently visible in the game

identityLevels= Classification levels of the id, seen by the user when inspecting

positionX= start X

positionY= start Y

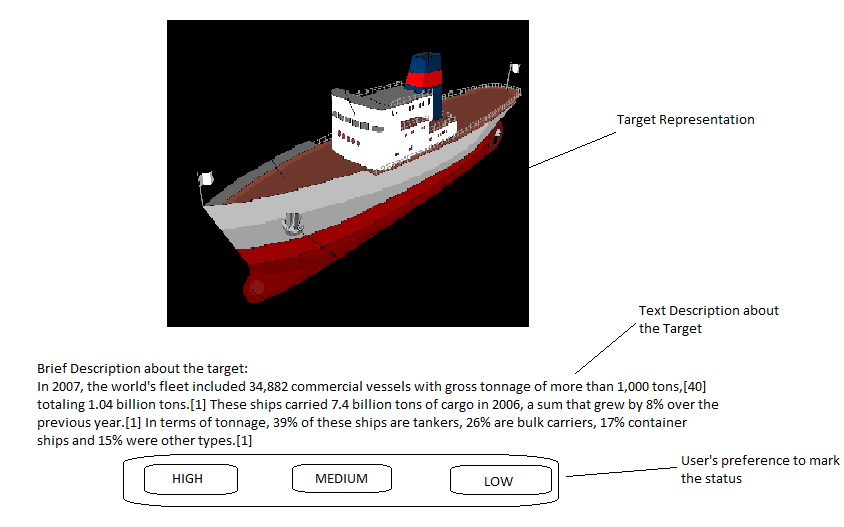
speed= rate of movement

description= Text that is visible when inspecting the target

## Target Inspection:

Once the target is detected, it becomes available on a list of different targets for inspection by the user. The availability lasts as long as the target is being detected and for a short period of time thereafter. The user has to select the appropriate target which he had detected. Inspecting the target reveals an image and text corresponding to the target identity characteristics. The user can then mark the target with different options. These options usually consist of marking the target as ‘low priority’ or ‘high priority.’ If you have correctly matched the marker with the target identity, you gain additional score (see below).

High priority targets when detected and marked correctly, tend to score higher points as opposed to low priority targets.



## Scoring Criteria:

The scoring is determined by the how quickly the user directs the units to identify the target and track the same.

First Contact Points:

The user gets first contact points when the target is detected by the user for the first time. The first contact points can only be obtained once.

Marking Points:

Once the user detects the target, the user can inspect the target and will be provided with different target marking options. Marking the target with the appropriate priority grants a one time point bonus to the score.

Most of the time a low priority marking bonus will be zero.

Tracking Points:

After the user obtains the first contact points for a target, the user can choose to track the target more than once to get the tracking points. Tracking points may vary depending upon the priority of the target. Correctly marked high priority targets will have high tracking points as opposed to low priority targets.

The user’s ability to identify all the targets and track them in an effective way with the best utilisation of the available resources gets him maximum score.

## Background & Grid

The background image will display a customizable graphic in the background. Additionally the size of the grid will be set by the administrator. Ideally the grid locations will be used by the activity feed to report objects.

## Overlay:

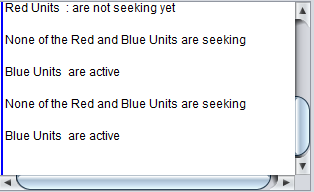
The overlay is a togglable information screen used to display information to the user. The most basic level is the display of the group search areas, but it should also be able to display customizable ‘overlay only’ objects. It is one such property which helps the user to understand the game in a better sense. It is used to visualise the search area boundaries.It helps the user to see the boundaries and further play the game based on the calculative boundary assumptions. At any point during the game, the user will be able to see the target boundaries as the spatial positions for the search area may require to be changed quite often.

Overlay-only objects - An overlay only object is a graphical information display that is only visible when the overlay is toggled on. The most common use of this would be a flight path that displays where a target might be. Other overlay only objects would be storms, other visualizations of unit likelihood.

Currently Overlay is buggy with additional commands.

## Activity Feed:

The activity feed is a series of text based user alerts designed to inform the player about what is happening in the scenario. It is a live feed which keeps on updating itself as the user plays the game. It is analogous to a virtual game commentator. Every action is recorded and displayed on the activity feed. For example, the user will constant notification about all the active units. When the units are transferred from one group to another, it gets notified in the activity feed. It enables the user to verify his actions and correct himself for any errors.



Example List of activity feed:

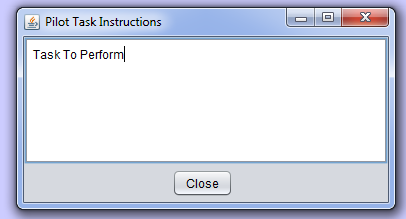
* newTargetSpotted
* oldTargetSpotted
* newTargetInspect
* oldTargetInspect
* targetOutOfRange
* unitLaunchStart
* unitLaunchComplete
* unitRelocateStart
* unitRelocateComplete
* newTargetView
* oldTargetView
* targetDetectionSuccess
* targetDetectionFailure
* unitSwitchStart
* unitSwitchComplete

The admin are able to set the level of messages that are applied. Each class of feed items should be able to be turned off or on in the configuration file and the script file should be able to send custom messages or turn on/off classes of feed items.

## Briefing:

Briefing is a group of text based information focussing more on the important tasks that user needs to carry out. It is advisable for the user to see his briefing notes before the user actually starts implementing his tasks. It comforts the user to play and understand the game in a better way and also it can be used to help prioritise the tasks in an efficient way.

In scripted scenarios the briefing may change at certain points to reflect updated task goals and priorities.



## Resource pool (lower priority):

An abstract pool of points used to purchase additional resources.

It includes all the resources that a user could potentially use while he plays the game. All the resources will be of limited usage to a normal user beyond which the it will be required to buy the resources. Some of the available resources that a user could use are units, groups, overlay information etc.

# Experiment Setup & Administration

Following below is a brief summary indicating different access controls for Admin and User.

## Configuration file:

In order to provide dynamic experience to the user, we use a configuration file from where the specifications for different scenarios are stored. Administrator will be the only person who has access to the configuration file. To reduce complexity and to increase modularity, we separate certain specifications about the game, split it up and store in a scripting file. The configuration and the scripting file could either be a csv file or an xml file.. Following below are certain properties that could be stored in configuration file.

1. Starting Number of Units for each Search Area
2. Unit
   1. Detection Range for each Unit
   2. Mobility for each unit
   3. Reliability for each unit
3. Graphics for objects and background
4. Time Period for the game
5. Target characteristics
   1. Target identity
      1. True identity
      2. Marked status
      3. Image
      4. Text
   2. Target mobility
      1. Speed
      2. Behavior
   3. Scoring Criteria ( First Contact Points, Marking points, Tracking Points)

## Scripting:

A script would contain triggers that allow for changes in object or game properties when certain conditions are met. Most commonly this would be tied with time and would allow for the introduction of an interruption task. A few examples for properties that could be stored in scripting file would be appearance/disappearance of objects, properties changes and other target management controls.

* Target waypoints - It defines the target’s trajectory. For example, a target might move in a hexagonal direction or triangular direction. All the points that connect the target’s trajectory are referred to as target way points.
* Object Characteristics changes - All the characteristics changes can only be an administrator level change and the user has no control over any object characteristics.
  + Object appearance/disappearance - The administrator has the privilege to control the appearance/ disappearance of any object. For example, the administrator can choose to create a weather storm which can eliminate certain select targets to disappear or to introduce select targets for specific scenarios.
  + Behavior- Any objects behavior can be controlled by the administrator. For example, the administrator can choose to eliminate any unit from any group or to introduce any number of units for any scenarios
  + Detection- All the active objects that fly around are referred to be flying with ‘ON’ state. The administrator can pick any object and make it as inactive object (‘OFF state’).
  + Scoring- Each target has its own scoring pattern. The scoring criteria can be changed by the administrator to increase/decrease the importance of the target in the context of the game.

## Data Collection

Data should be collected in a csv file with the relevant data in a standardized file format. Ideally, data collection would occur at a variable rate set by the administrator but a 500 ms sample rate is acceptable.

Data collection includes all locations and statuses of objects as well as the state of all buttons (number of times activated). It does not include the user mouse or keyboard input independently of button activations.

# User Experience

Group level control

1. Proximity - It is measured by nearness in space among different units that fly across a group.The user can choose to increase or decrease the level of proximity control by moving the slider to right or left accordingly.
2. Cohesion control - It is measured by the act of forming a united whole. The user can choose to make the units to combine together or distant themselves from different units by moving the slider to right or left accordingly. The proximity and the cohesion control has to be controlled depending on various standings of the game
3. Unit Transfer - The user may choose to transfer units from one group to another in order to track the target in a better way. A group may have 0 units

Object level control

1. Inspecting targets (zooming) - After the target is detected, the user has to click on the “View Target” button. A picture and text description about the target will be displayed. The user will be asked to mark the identity of the target. Once identity of the target is marked successfully, marking bonus points will be added to the total score (if it matches the true identity of the target).
2. Unit purchasing- The user can purchase the units if he/she feels that the units that are already to provided are not sufficient enough to detect and track the targets.
3. Overlay viewing - It helps the user to track the search area boundaries. The user can toggle the “Overlay” button to see the search area boundaries.

# General Mechanics

## Object Movement

Object movement speed needs to be scalable to allow for very slow speeds as well as many levels of control. This should be for both Units and Targets, as well as any other moving objects. Movement should be smooth, updating on the screen at least 10 hz or more.

## Timer

A timer which automatically ends the game is implemented. The timer can count up or down in multiple formats.

# Things to fix

Completed --Turning off unit launch only removes button but not drop down or section (all other turn off features should remove headings as well as drop downs)

Completed -- Credits need to be able to be turned off completely.

Selecting the Current Group buttons makes things visible

Completed -- Overlay not working

**Completed --Timer counting downwards**

**Completed --Different image for target inspection**

**Already implemented-Ability to change delay for disappearance of items**

**Persistent freezing issues (new) - Usually post unit launch or when a search area is relocated close by**

**Sensors are not working properly in some situations**

Selecting the Current Group buttons makes things visible

When a unit is added in Red group it freezes and doesn’t seek (doesn’t get added to the unit list, doesn’t make the red group option available)

SEEK BEHAVIOR CHANGE

RELOCATE BEHAVIOR CHANGE

Post launch unit movement needs to match Relocation behavior in terms of speed (right now it is moving very slow).

**Scoring Criteria review (target identity/levels of marking)**

**Multiplier for target?**

**(clarify)**

TARGET MOVEMENT BEHAVIOR - WAYPOINT SYSTEM?