

Lance's Script Elements

Using Officer Rank

```
eOfficerRank blah;
```

```
OfficerRank = OrionBaseTeam->mGetFirstShip()->mGetOfficerRank(kSecurityOfficer);  
OfficerNumber = OfficerRank * 5;
```

Display a Message

```
fMission.mDisplayMessage(kLNF_msg, kMissionScriptMessage);
```

Display a Random Message

```
fMission.mDisplayMessage (fMission.mRandomInt32 (kOrionTaunt1_msg,  
kOrionTaunt3_msg), kMissionScriptMessage);
```

Set a Comm Button

```
Ship->mSetCommButton (1, kButtonHail_text);
```

Set a callback event

```
Ship->mSetShipCallbackEvents (kNotifyTransportedItemTo);
```

****List of callback events in scriptstructs.h****

Damage a ship's components

```
Ship->mDamageSystem(kDamageMaxShields, 100);
```

Create a pointer to a Ship

```
tShipInfo* Ship;  
Ship = OrionBaseTeam->mGetThisShip(1);
```

Pointer to a Team

```
tTeamInfo* PlayerTeam = fMission.mGetTeamHandle (kPlayerTeam);
```

Set team relations

```
fMission.mSetTeamRelations (kOrionTeam, kImperialTeam, kTeamAtWar, kGoalDestroy,  
1.0, kTrue);
```

Create fleet

```
OrionTeam->mCreateFleet (5, kMaxBPV, kClassFrigate, kClassFrigate, 50,  
kStartPosition_0);
```

Create Ship

```
OrionBaseTeam->mCreateShip (kMaxBPV, kClassStarBase, kClassStarBase, 100,  
kStartPosition_V);
```

Track a distance from a grid point

```
Ship->mTrackTacticalDistance(50.0, 13, 8);
```

```
case kNotifyAtDistance:  
    if (Ship->mGetTacticalDistance(OrionBaseTeam->mGetFirstShip()) < 50.0)  
  
        if (Ship->mGetTrackingDistance() == 50.0)
```

Determine a distance from an object

```
float dist = Ship->mGetTacticalDistance (PlayerTeam->mGetFirstShip());  
float dist = Ship->mGetTacticalDistance (mapObject);
```

create ships during the mission

```
mCreateMissionShips (kPCInOrbit);
```

Determine the Player's target

```
tShipInfo* PlayersTarget = fMission.mGetShipTarget( fPlayerShip );
```

```
if (target == StarTapTeam->mGetFirstShip())
```

Set a timer

```
fMission.mSetTimedInterval (kTimerLimitSet, 20);  
fMission.mSetTimedInterval (kTimerLimitSet_10, 20);  
fMission.mSetTimedInterval (kTimerTurnsSet, 10);  
fMission.mSetTimedInterval (kTimerCountdownSet, 5);    // set countdown timer to  
expire in 5 turns
```

```
switch (fEventID)  
{  
case kTimerLimitSet:  
case kTimerLimitSet_10:  
case kTimerTurnsSet:  
case kTimerCountdownSet:  
}
```

Tell a ship to disengage

```
Ship->mSetShipCallbackEvents (kDisengageShip);
```

```
case kNotifyDisengaged:
```

Tell a team to disengage

```
Neutral_ATeam->mDisengageShips( kByMapEdge, kVeryHighPriority );
```

Disengage method

```
kByAcceleration,  
kByMapEdge,
```

PRIORITY

```
kNeverPriority = 0,  
kVeryLowPriority = 1,  
kLowPriority= 3,  
kMediumPriority = 5,  
kHighPriority = 7,  
kVeryHighPriority = 9,  
kAbsolutePriority = 10,
```

Set Final Destination

```
Ship->mSetFinalDestination(); // to lowercase map point
```

Deep Scan

```
case kNotifyDeepScan:  
    if (fIsDeepScanOn == kTrue)
```

Place a ship near a grid point

```
Ship = fPlayerShip;  
int32 OldX = 0;  
int32 OldY = 0;  
Ship->mGetGridPosition (&OldX, &OldY);  
int32 NewX = OldX + fMission.mRandomInt32 (1, 2);  
int32 NewY = OldY + fMission.mRandomInt32 (5, 10);  
BostonTeam->mCreateShip(kMaxBPV, kClassLightCruiser, kClassLightCruiser, 20,  
kStartPosition_Specified, NewX, NewY);
```

change the Name of a ship

```
Ship->mChangeName(kBoston_name);
```

Check ships Damage percent

```
if (Ship->mGetDamagePercent() < 50.0)
```

schedule a mission

```
mScheduleMission (int32 MsgID); // file name of mission to schedule (including  
extension)
```

Tractor - Tell a ship to tractor something

```
mTractorEntity(tShipInfo* Ship, int32 X, int32 Y, aiPriority)
```

Defend - Tell a ship to defend something

```
mDefendEntity ( tShipInfo* Ship, ThreatRadius, FollowRadius, aiPriority)
```

Patrol - set patrol routes

Change the amount of supplies aboard a ship

```
Ship->mAccessSupplies (kSetSupplyCount, kSupplyScatterPack, 0);
```

Add transporter items to a ship

```
Ship->mAccessSupplies (kSetTransporterItemCount, kTransGornEgg, 1);
```

gets the shooter

```
in kNotifyDamage
```

```
tShipInfo* HostileShip = fMission.mFindShipID (fEventID);
```

sets a navigational goal arrow

```
fMission.mSetGoalIndicator( Ship );  
fMission.mSetGoalIndicator( iTruth, X, Y );
```

figure out what a player is orbiting

```
in kNotifyBeginOrbit:
```

```
tShipInfo* OrbitObject = fMission.mFindShipID (fEventID); // what the player is  
orbiting
```

transport an item to a ship

```
PlanetTeam->mGetFirstShip()->mSetShipCallbackEvents( kNotifyTransportedItemTo );
```

```
tShipInfo* TransportTarget = fMission.mFindShipID (fEventID);
```

```
set the callbacks on the ship that is doing the transporting, either to or from.  
set mSetShipCallbackEvents( kNotifyTransportedItemTo );
```

the '**Ship**' in the Events switch is the ship that is **issuing the command to do** the transporting

in other words, the fEventID is always the **other** ship, and **Ship** is **fPlayerShip**

this is the item that is transported

```
fEventItem1 == kTransGornEgg
```

the transporter can only hold one of any item

check out Klingon 9 for some groovy details

figure out closest map edge

```
void FindClosestMapEdge(tScript& tScript, tShipInfo* fPlayerShip);
```

```
FindClosestMapEdge(*this, fPlayerShip);
```

```
//-----
```

```
// figure out closest map edge to fPlayerShip
```

```
// and set the goal pointer to that edge
```

```
//
```

```
void FindClosestMapEdge(tScript& tScript, tShipInfo* fPlayerShip)
```

```

{
    // LNF maybe turn off goal pointer when the player gets close to the border?
    // LNF or just have a goal pointer toggle button
    const int32 MapEdgeX = 35;
    const int32 MapEdgeY = 26;
    int32 ShipPositionX, ShipPositionY;
    bool MapLeft = false;
    bool MapTop = false;
    int32 XDiff, YDiff;

    fPlayerShip->mGetGridPosition (&ShipPositionX, &ShipPositionY);

    if ( ((MapEdgeX / 2) - ShipPositionX) >= 0 )
    {
        MapLeft = true;
        XDiff = ShipPositionX;
    }
    else
    {
        XDiff = MapEdgeX - ShipPositionX;
    }

    if ( ((MapEdgeY / 2) - ShipPositionY) >= 0 )
    {
        MapTop = true;
        YDiff = ShipPositionY;
    }
    else
    {
        YDiff = MapEdgeY - ShipPositionY;
    }

    if ( XDiff <= YDiff )
    {
        if ( MapLeft == true )
        {
            tScript.fMission.mSetGoalIndicator( kTrue, 1, ShipPositionY );
        }
        else
        {
            tScript.fMission.mSetGoalIndicator( kTrue, MapEdgeX,
ShipPositionY );
        }
    }
    else
    {
        if ( MapTop == true )
        {
            tScript.fMission.mSetGoalIndicator( kTrue, ShipPositionX, 1 );
        }
        else
        {
            tScript.fMission.mSetGoalIndicator( kTrue, ShipPositionX,
MapEdgeY );
        }
    }
}

```

for DeepScanTracker - gets the player's target

```
( fObjects[i] == fScript->fMission.mGetShipTarget( fScript->fPlayerShip ) )
```

```
-----
```

test the status of a ship

```
int32 blah = Ship->mTestShipStatus( kShipCaptured );
```

```
kUndefined  
kShipCreated  
kShipAtFinalDestination  
kShipCaptured  
kShipDisengaged  
kShipDestroyed
```

```
-----
```

Balance tricks

randomize enemy ships based on the player's fleet size

```
int32 NumberEnemyShips = fMission.mRandomInt32 (2, PlayerTeam-  
>mGetShipsInPlayCount());
```

switch on the Difficulty Level (three levels)

```
int32 DifLev = fMission.mGetDifficultyLevel();  
  
switch ( DifLev )  
{  
    case 0: // easy  
        break;  
    case 1: // medium  
        break;  
    case 2: // hard  
        break;  
}
```

use these BPV levels - total for team : adjust the multiplier

```
int32 LowBPV = PlayerTeam->mGetCombatBPV() * .8;  
int32 HighBPV = PlayerTeam->mGetCombatBPV() * 1.1;  
int32 TotalBPV = fMission.mRandomInt32 (LowBPV, HighBPV);
```

average player ship BPV - adjust the multiplier

```
int32 LowBPV = (float) PlayerCombat / (float) NumPlayerShips * 0.8f;  
int32 HighBPV = (float) PlayerCombat / (float) NumPlayerShips * 1.3f;  
int32 TotalBPV = fMission.mRandomInt32 (LowBPV, HighBPV);
```

-or-

```
int32 NumberOfPlayersShips = PlayerTeam->mGetShipsInPlayCount();  
int32 AveragePlayerBPV = TotalBPV / NumberOfPlayersShips;
```

Always give the player a fixed number of ships

```
gNumShipsAddedToCommand = 6 - PlayerTeam->mGetShipsInPlayCount();
```

Outgunned BPV

get the difference between the enemy's BPV and the players

"5" is the number of enemy ships you want

```
int32 EstimatedEnemyBPV = 5 * TotalBPV + 200; // 5 enemy ships and the bats  
int32 OutGunned = EstimatedEnemyBPV - PlayerCombat;
```

Determine the player's largest ship class

```
// this line in declaration
eClassTypes PlayerShipClass = kClassShuttle;

// determine the biggest ship class the player has
for (Ship = PlayerTeam->mGetFirstShip(); Ship != NULL; Ship = PlayerTeam->mGetNextShip())
{
    if (PlayerShipClass < Ship->mGetClassType())
    {
        PlayerShipClass = Ship->mGetClassType();
    }
}
```

Use casting to increment/decrement the ship classes

```
// this line in declaration
eClassTypes PlayerShipClass = kClassShuttle;

this line will take PlayerShipClass which is type eClassTypes, cast it to an int32,
increment it, then cast it back as eClassTypes
```

```
((eClassTypes) ((int32)PlayerShipClass + 1))

OrionTeam->mCreateFleet (NumberOfPlayersShips, TotalBPV, kClassFrigate,
((eClassTypes) ((int32)PlayerShipClass + 1)), 50, kStartPosition_Specified, 25,
17);
```

I think this is good, but I haven't tested it

use TotalBPV which is dependent on difficulty level, then choose the largest ship class based on the largest player's ship and make that dependent on diff level

```
// put this in the declarations section
eClassTypes LargestEnemyShipClass = kClassFrigate;
```

put this line in the difficulty switch statement, and change -1, 0, +1 ship class according to difficulty level

don't forget to check the 'ends' of the ship classes so the enemy ships aren't less than frigates or greater than battleships! (unless you want base stations and freighters)

```
LargestEnemyShipClass = (eClassTypes) ((int32)PlayerShipClass - 1);
```

also, in the create ship line, use TotalBPV that is also dependent on difficulty level

```
OrionTeam->mCreateFleet (NumberOfPlayersShips, TotalBPV, kClassFrigate,
LargestEnemyShipClass, 50, kStartPosition_Specified, 25, 17);
```

this is all I have so far....I would like something where if I have multiple teams, I can balance numbers and ship classes across all teams....on both sides

```
/**      balance      **
eClassTypes PlayerShipClass = kClassShuttle;
eClassTypes LargestEnemyShipClass = kClassFrigate;
eClassTypes NewEnemyShipClass( tScript& tScript, eClassTypes LargestEnemyShipClass
);
```

```

//***          ***

// determine the biggest ship class the player has
for (Ship = PlayerTeam->mGetFirstShip(); Ship != NULL; Ship = PlayerTeam-
>mGetNextShip())
{
    if (PlayerShipClass < Ship->mGetClassType())
    {
        PlayerShipClass = Ship->mGetClassType();
    }
}

switch ( DifLev )
{
    case 0:
    {
        TotalBPV *= 0.60f;

        // set LargestEnemyShipClass to one SMALLER than the player
        if ( PlayerShipClass > kClassFrigate )
        {
            LargestEnemyShipClass = (eClassTypes)
((int32)PlayerShipClass - 1);
        }
        break;
    case 1:
    {
        // do nothing
        LargestEnemyShipClass = PlayerShipClass;
    }
    break;
    case 2:
    {
        TotalBPV *= 1.5f;

        // set LargestEnemyShipClass to one LARGER than the player
        if ( PlayerShipClass < kClassBattleship )
        {
            LargestEnemyShipClass = (eClassTypes)
((int32)PlayerShipClass + 1);
        }
        break;
    }
}

int32 AveragePlayerBPV = TotalBPV / NumberOfPlayersShips;
-----
Message if player uses ESC key to End Mission

bool PlayerUsedESCKey = false; // if the player Ends Mission with the ESC key

    case kNotifyPlayerLeaving:
        PlayerUsedESCKey = true;
        break;

if ( PlayerUsedESCKey == false )
{
}
else // player exited improperly, display a message
{
    fMission.mAddToDebriefingList( kImproperlyExitedMission_msg );
}

```



```
kImproperlyExitedMission_msg,
```

```
// kImproperlyExitedMission_msg
```

```
"You improperly exited the mission.",
```

```
-----  
-----  
-----
```