*Andrew Perrault 18 September 2014*

# Sci-fi Text Game GDD

## Story

The main story is about an astronaut mission to explore an asteroid. As the astronaut attempts to land on the asteroid his vehicle fails and he crashes. The game begins with the astronaut waking up in a heavily damaged ship at the bottom of a deep crevasse. The object of the game is to survive long enough and make contact with the rest of the mission and get rescued. The main conflict of the game is very much man vs nature and as such very little will be explained to the player about the main character. He/She does not even have a name and will have no communication with any NPCs at all until the final sequence when you actually make contact and win the game.

## Possible Titles

I have a number of possible titles but I have not decided which to go with yet. I will decide when more of the actual writing is completed.

Marooned

In the Distance

On the Rocks

## Puzzles

At the start of the game everything is dark except for a red blinking alarm signal. The signal says that oxygen is leaking from the ship.

First you must find a space suit before the oxygen is gone. The suit seems to be working but only has 3 percent oxygen in the tank. Oxygen will act as a turn counter and you will die and the game will end if you cannot win before the number of turns complete.

There is a flash light and a communication station on the ship. Leaving the ship without the flashlight will result in you getting lost and dying. The communication station is working but cannot get a a signal from the bottom of the crevasse.

The ship main power is broken and you cannot open the door to leave. You need to change from main power to auxiliary power and open the door. When the door opens the auxiliary power unit fails and the ship goes dark. Examine the auxiliary power unit and remove a long section of wire. Take the wire.

Outside there are a number of tunnels and a spare equipment bay. In the equipment bay there is an oxygen tank that you can use to fill the suit. There is also a solar power generator and a roll of fibre optic cable. The communication antennae appears to be undamaged. Take the cable and power generator.

Enter the tunnels.

You find a room with a massive vain of platinum and huge nugget of platinum on the floor. Take the nugget.

You find a huge crystal grotto. Light from the sun is refracted through the crystals and this room is the bright. Drop the solar generator. Attach the wire to the generator. Take the other end of the wire.

Go back to the ship. Attach the wire to the auxiliary power generator. You see the battery charging light come on.

Exit the ship. Take the antennae.

Go back into the tunnels and find your way to the surface.

Find the top of the crevasse. Setup the antennae. Attach the fibre optic cable to the antennae. Tie the cable to the platinum nugget. Throw the nugget into the crevasse.

Go back to the ship. You will find a platinum nugget attached to a cable that disappears up into the darkness. Take the nugget. Take the cable end. Go into the ship. Attach the cable to the communication station.

Use the station. Contact the rest of the mission crew in orbit around the asteroid.

Win game.

## Interface

The player interacts by typing commands at a prompt.

n, s, e, w, up, down can be used a short forms for go north, go south, etc

inventory or i displays the players inventory

look or l displays a description of the current location

examine or x displays details of an object

get and drop picks up and drops items

help will display instructions

quit will exit the game

## Technical Systems

### Game Object Class

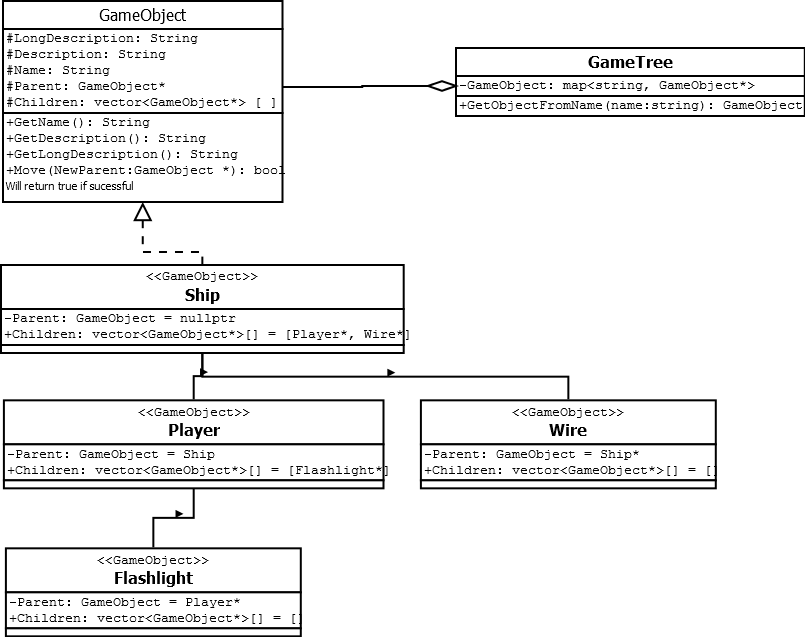
All items, rooms and the player will derive from the game object class. The game objects will be related to each other in a tree like structure. Each object will have one parent that may be null – rooms will have null parents. All objects will have a vector of children which indicates the game objects contained within that object.

Few examples

1) The ship interior object has two children - the player and a roll of wire. When the player picks up the wire the wire node is moved from a child of the ship to a child of the player.

2) When the player goes to another room the player's node will move from a child of one room to the child of another.

Many if not most of the actions within the game can be completed by reading or manipulating this tree structure. If you want to use an item it must be a child of the player node or a child of the players parent. Changing rooms is just moving the player node. Picking up items is moving the item node. Generating a description of a room would be the players parent description with its child descriptions appended.



### Lexer

All input to the game will come from the player typing commands at a prompt.

Some examples of commands would be:

Go north

Inventory

Open Door

Take Wire

The lexer is the class that reads in the commands in English and converts it to a form that can be used by the game. The lexer is not intended to be a natural language processor. The puzzle in a text game should not be trying to figure out which words the game understands and the lexer will value getting the most likely intentions of the player over grammatical correctness.

The number of verbs that the game understands will be vary limited but there will be a much larger number of aliases to these verbs. The verbs the game understands will include Go, Put, Get and Use.

The lexer will read in the words of the sentence and replace the words with a token. Token will come in a number of types including verb, noun, list separator and sentence separator.

For example:

The player types: ***Take wire and antennae***

The lexer will search a dictionary and find take is an alias for the 'Get' verb. Wire and antennae are both nouns also found in the dictionary.

The command would be translated to : **{verb:Get}{noun:wire}{listSeparator}{noun:antennae}**

The verb will then be used as the key in a map of verbs to function pointers and the function Get will be called with an array of items as a parameter :**Get( [{noun:wire}, {noun:antennae}] )**

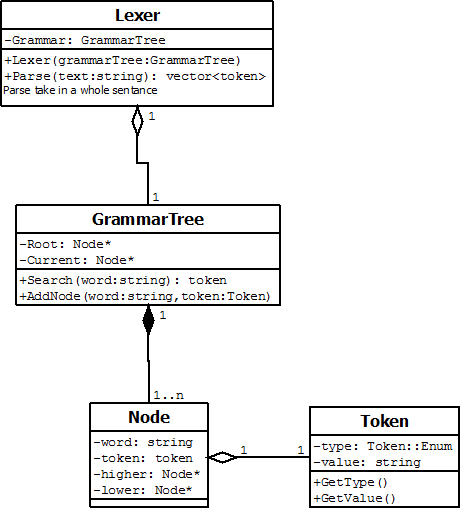
Many words like 'with' and 'on' will be ignored.

***Use key on door*** and ***Open door with key*** and ***Use key door*** would have identical translations of:

**Use( [{noun:key},{noun:door}] )**

Obviously ***Use key door*** is not grammatically correct but ignoring small words makes the code much simpler and that is most likely what the player intended anyway.

I will need to keep an eye out for specific edge cases where the system may break down. For example the lexer should not confuse ***Pick up rock*** with ***Go up*** or even just ***Up***.



## Object Importer

The game objects will be imported from text files rather than hard coded. The files will have the object name and a list of properties and values on each line. The room importer will make use of the lexer with an alternate grammar dictionary to interpret the text files. The intention of the room importer is to make adding content to the game much quicker than having to rewrite headers or building long list of hard to maintain code for every new object. The object importer will build the game object tree and also add a token to the command interpreter grammar tree automatically on import.

e.g.

object cockpit

{

description : blah blah blah

parent : null

children : player, computer

exits :

{

south : cargo bay

west : cavern

}

}