Manual Tecnico:

Para los analizadores que programa necesitaba se utilizo la herramienta de jflex y cup, siendo la principal el de carga de mapa que tiene un estructura .Json. El analizador sintactico de la carga de un juego realiza la tarea de ir formando el objeto “Mapa” para luego ser usado en el juego, esto lo hace recorriendo producciones, estas producciones se pueden ver en la carpeta gramatica y el archivo sintacticoJson. Si en algun momento se da un error sintactico se le notifica al ususario, mostrandole el token que fallo y la posicion en la que se encuentra con los metodos syntax_error y unrecovered_syntax_error.

```
public void syntax_error(Symbol s){
    String lexema = s.value.toString();
    int fila = s.right;
    int columna = s.left;
    fp.agregarTextoAcciones("Error Sintactico Recuperado\n");
    fp.agregarTextoAcciones("\t \tLexema: "+ lexema+"\n");
    fp.agregarTextoAcciones("\t \tFila: "+ fila+"\n");
    fp.agregarTextoAcciones("\t \tColumna: "+ columna+"\n");
}

public void unrecovered_syntax_error(Symbol s) throws java.lang.Exception{
    String lexema = s.value.toString();
    int fila = s.right;
    int columna = s.left;
    fp.agregarTextoAcciones("Error Sintactico Panic Mode\n");
    fp.agregarTextoAcciones("\t \tLexema: "+ lexema+"\n");
    fp.agregarTextoAcciones("\t \tFila: "+ fila+"\n");
    fp.agregarTextoAcciones("\t \tColumna: "+ columna+"\n");
}
```

Si no se encuentran error y el mapa se construye correctamente se empieza el juego con metodos sencillos que muestran el tablero al usuario, utilizando objetos que extienden de jPanel que son los objetos “Casillas” que nos seran de utilidad ya que contendran los planetas que tambien genera el mapa.

Este programa tambien tiene la necesidad de crear archivos Json que en algun otro momento pasaran por los analizadores. Los encargados de esta tarea se encuentran en el paquete “Escritores” los cuales son simples escritores de texto pleno como por ejemplo:

```java
public class CreadorArchivoReplay {

    public void crearArchivo(File file, ArrayList<Planeta> planetas, ArrayList<EnvioDeFlota> envios) {
        String textoAEscribir = "";
        textoAEscribir += "{\n\tterminado: true,";
        textoAEscribir+=escribirPlanetasIniciales(planetas);
        textoAEscribir += "\n\tRONDAS: {";
        int ronda=0;
        boolean segundo=false;
        System.out.println("enviosss "+envios.size());
        for (int i = 0; i < envios.size(); i++) {
            if (i==0) {
                ronda=envios.get(i).getRonda().getNumero();
                textoAEscribir += "\n\t\t"+ronda+": [";
            }
            if (envios.get(i).getRonda().getNumero()==ronda) {
                if (segundo) {
                    textoAEscribir+=",";
                }
                textoAEscribir += "\n\t\t\t{";
                textoAEscribir += "\n\t\t\torigen: \""+envios.get(i).getOrigen().getPlaneta().getNombre()+"\",";
                textoAEscribir += "\n\t\t\tdestino: \""+envios.get(i).getDestino().getPlaneta().getNombre()+"\",";
                textoAEscribir += "\n\t\t\tnaves: "+envios.get(i).getNaves();
                textoAEscribir += "\n\t\t\t}";
```

Guardar Partida y Replay

El programa tiene la capacidad de guardar una partida en proceso asi como una que ya concluyo, para esto se generan archivos de la misma manera como anteriormente visto de texto. Para luego ser leido por el mismo Analizador Sintactico.

Como funciona el mismo Analizador:
En el archivo Json para estos dos componentes se encuentras tres atributos importantes para la lectura de ambas:
Termindao: Indica si la partida esta en proceso o ya concluyo con un simple boolean true o false.
Planetas: Esto representa el estado inicial de todas las planetas del mapa en la ronda 1.
Rondas: En cada ronda se encuentra un conjunto de envios que han realizado los jugadores, lo cual nos ayudara al ir recorriendo los turnos.
Un ejemplo :

```
{
        terminado:true,
        PLANETAS:[
                {
                        nombre: "A",
                        fila: 10,
                        columna:1,
                        dueño: "Jose",
                        naves: 10,
                        produccion: 5,
                        porcentajeMuertes : 0.5
                },
                {
                        nombre: "B",
                        fila:10,
                        columna:4,
                        dueño: "pc1",
```

```
                naves: 20,
                produccion: 5,
                porcentajeMuertes : 0.5
        }
    ],
    RONDAS:{
    1:[
        {
            origen: "A",
            destino:"B",
            naves: 10
        },
        {
            origen: "B",
            destino:"A",
            naves:3
        }
    ],
    2:[
        {
            origen: "A",
            destino:"B",
            naves: 1
        },
        {
            origen: "B",
            destino:"A",
            naves:4
        }
    ]
    }

}
```

Al momento de pasar por el analizador el boolean terminado indicara si es un replay o no, ademas al convertir las especificaciones de ronda a EnviosDeFlota se podran utilizar para ingresarlos a la logica del juego, para hacer mas facil ambos procesos. En el caso el replay el juego simplemente comenzara desde la ronda uno y los envios se haran hasta que el usuario indique termino de ronda. Ahora prar el guardado de partida se tendran que saltar todas las rondas hasta que se llegue a la ronda en la que se guardo.

Online

Para el jugabilidad con otra persona se tuvo la idea de que cada aplicación contara con un servidor y con un cliente que se comunicara con el servidor de la otra persona estas clases se envuentran de la siguiente manera.

```java
public class Servidor extends Observable implements Runnable{
int puerto;

    public Servidor(int puerto) {
        this.puerto = puerto;
    }




    public static String obtenerTextoDeFile(File file) {
        String text = "";
        try {
            BufferedReader br = new BufferedReader(new FileReader(file));
            while (true) {
                String aux = br.readLine();
                if (aux == null) {
                    break;
                } else {
                    text += aux;
                }
            }
        } catch (FileNotFoundException ex) {
            Logger.getLogger(Servidor.class.getName()).log(Level.SEVERE, null, ex);
        } catch (IOException ex) {
            Logger.getLogger(Servidor.class.getName()).log(Level.SEVERE, null, ex);
        }
        return text;
    }



    @Override
    public void run() {
     try {
```

```java
public class Cliente implements Runnable{

    private  String host ;

     private int puerto;
     private String mensaje;


    public Cliente(String host,int puerto,String mensaje) {
        this.host=host;
        this.puerto = puerto;
        this.mensaje=mensaje;
    }



    @Override
    public void run() {
        try {
            Socket sc = new Socket(host, puerto);
            DataOutputStream out=null;
            out = new DataOutputStream(sc.getOutputStream());
            System.out.println(mensaje+" A enviar");
            out.writeUTF(mensaje);
            sc.close();
        } catch (IOException ex) {
            Logger.getLogger(Cliente.class.getName()).log(Level.SEVERE, null, ex);
        }

    }
}
```

Como podra haber notado estas clases implementan runnable ya que el metodo run nos ayudara a controlar los observadores que tambien se implementaros, cada aplicación ya sea el host o el guest cuentan con un observador que notificara los cambios realizados del la persona rival. Estos cambios que se observan son mensajes String que representan una entrada para el AnalizadorSintacticoOnline, las producciones de esta gramatica son muy sencillas y tienen similitudes con la de Guardar partida y Replay, ya que simplemente se mandan los envios de esa ronda como por ejemplo:

```
{
	{
		origen: "A",
		destino:"B",
		naves: 10
	},
	{
		origen: "A",
		destino:"C",
		naves: 2
	}

}
```

Esto simple representa el planeta origen su destino y las naves que seran partes de la flota.
El analizador simplemente convertira el texto en un envio y asi.