11sdd assessment: AREA51

# Defining and understanding

## CONTEXT DIAGRAM

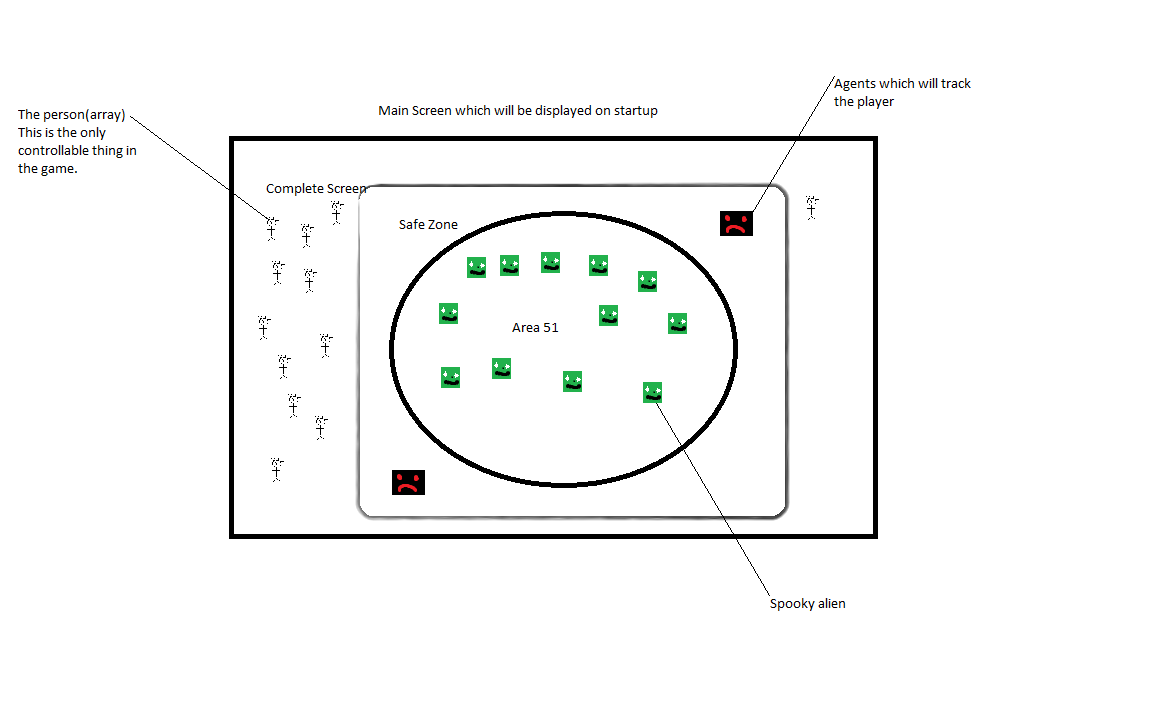
## Data flow diagram

# planning and desgining

## Gannt Chart

## Test Data

## Screen Design



## Structure Chart

## Data dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Data Type | Scope | Description |
| X | Int | Global | The x value of various variables |
| Y | Int | Global | The y value of various variables |
| dx | Int | Global | The xSpeed value of various variables |
| dy | Int | Global | The ySpeed value of various variables |
| inTrackingZone | Boolean | Global | Whether or not the variable is in the tracking zone |
| inArea51 | Boolean | Global | Whether or not the variable is in area51. |
| personTouchingAgent | Boolean | Local to movePerson | Whether a person is touching an agent |
| personTouchingAlien | Boolean | Local to movePerson | Whether a person is touching an Alien |
| closestPerson | String | Local to moveAgent | Checks who the closest person is to any given agent |
| AliensDiscovered | Int | Local to personTouchingAlien | How many aliens have been discovered |
| peopleAlive | Int | Local to personTouchingAgent | How many people are still alive |
|  |  |  |  |

## algorithm design & ipo chart

**Discover Aliens:**

**BEGIN**

**FOR I=0 TO peoples Length**

**For j=0 TO aliens Length**

**If PersonTouchingAlien is TRUE**

**Set Aliens.x = -20000**

**DiscoveredAliens = DiscoveredAliens + 1**

**NEXT j**

**Next i**

**END**

|  |  |  |
| --- | --- | --- |
| **INPUTS** | **PROCESSING** | **OUTPUTS** |
| **A key Pressed** | **Checks whether keyPressed is a number, if so 🡪**  **IF key is not a number, checks if that key was an arrow, if so 🡪** | **Selects the player assigned with that number**  The arrows change the selectedPersons dx and dy based on which arrow has been pressed |

## language & ide features

In Processing there are several different features, some of which I have used in my program. These include classes, the drawing of shapes such as ellipses, the ability to load images, many sub programs, different types of loop and nested loops.

A class is used to define many different variables to a single entity, this was used in my program to define variables for my people, aliens and agents, this was also much easier than defining each variable individually.

A very useful feature in processing is the ellipse mode feature which is used to allow circles to be draw from the centre. This made it a lot easier to make my game as I didn’t have to worry about my circle being in the wrong spot or the wrong size since it was drawn in the middle of the screen. The ellipse mode helped me save time and eliminate errors.

The ability to load images allow me to be more creative in my game design, letting me make my own characters such as the aliens, people and agents. The useful thing with this neat feature was that it was easy to use and allowed me to set the size of the image after it was imported which was very nice.

Throughout my program I used many different subprograms, this allowed me to spilt up my work into easy, manageable chunks instead of one huge wall of text. The two subprograms unique to Processing are the Draw and Setup subprograms. The setup is run once at the beginning of the program and is used to define all the environmental boundaries such as screen size, background colour, the importing of images and the first subprogram to be called. The draw subprogram is called directly after the setup and continuously runs all its code until the program is closed. Draw () is called automatically and so doesn’t need to be called on its own.

I used many different types of loops such as while loops, for loops and if statements. These are the backbone of my program and without them this game wouldn’t have been possible.

Another neat feature that I used was the use of nested loops which provided to be very handy in that they would call both of my people and aliens at the same time. This was used in detecting collisions between the aliens and people and the agents and people. Originally I thought these would be very difficult to implement but quite the opposite they were very easy to use and they were very helpful.

# testing and evaluating

## Test data checked

# maintaining

## contest versioning