# IT 140 Design Document Project 1

**Raymond Agosto**

## Storyboard (Description and Map)

Somewhere in the galaxy is a planet full of mechanical junk and spare parts. These parts are being harvested and melted down by large steam-powered robots, and these scrappers don’t seem to care if the “junk” is alive. It all goes into the furnace.

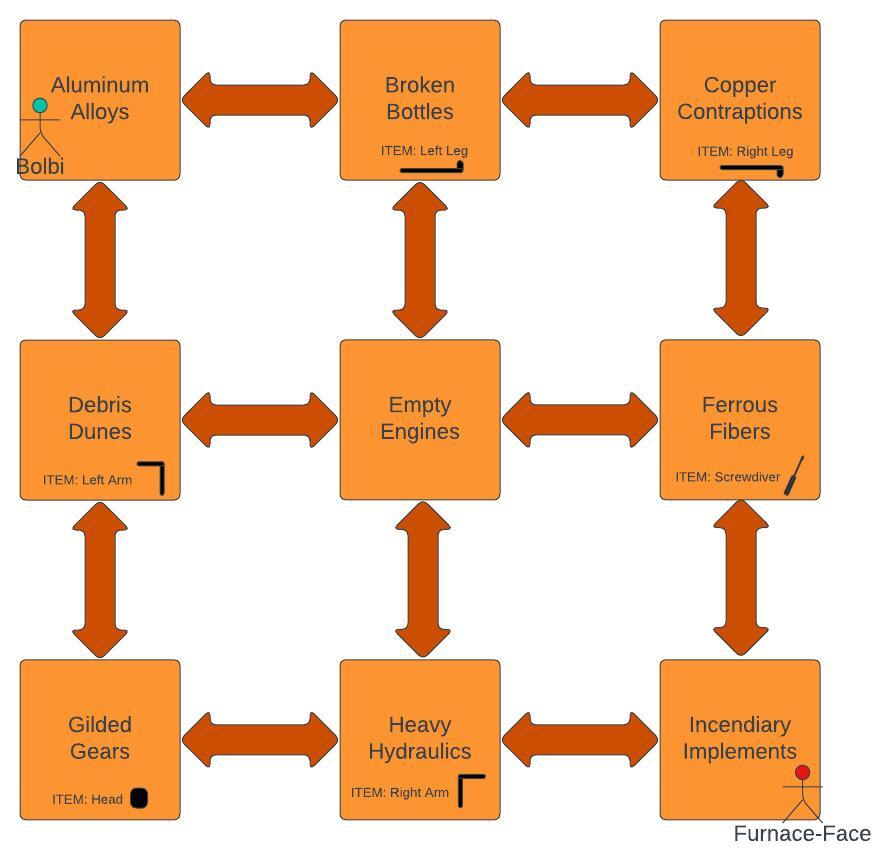
The player controls Bolbi, a small, modular robot who wakes up and realizes that his parts are scattered around the junkyard. Bolbi’s head is unable to move, but it catches a glimpse of a rather large scrapper bot with a furnace for a face. It seems that Furnace-Face is heading straight toward Bolbi’s torso! The torso can roll around at will and has only basic awareness of immediate surroundings.

The player needs to roll the spherical torso to each of Bolbi’s parts and attach them before trying to take on Furnace-Face. Running into Furnace-Face before collecting all of the parts results in Bolbi being crushed and melted down to nothing. To make matters worse, Furnace-Face is continually, but slowly, giving chase! Bolbi needs to collect his head, two arms, two legs, and a screwdriver to fight Furnace-Face.

The junkyard is composed of 9 sections or “blocks” much like a tic-tac-toe board. It’s a junkyard, so there is not much variation in the décor so to speak. However, the junkyard has sectioned off piles for different types of scrap, so Bolbi will still be able to navigate around the board based on junk piles.

Gameplay is turn-based, and most actions will consume a turn. The player can **MOVE** to an adjacent block in the cardinal directions, (N,S,E,W). The player can **COLLECT** a part, and the player can **WAIT** in place, which also consumes a turn. Every 3 turns, Furnace-Face will move one block. The player gets a free **SCAN** every turn, which reveals any item (part) on that block, and alerts the player if the villain is in an adjacent block.

Each robotic part gives a unique benefit to Bolbi. Collecting both legs lets Bolbi run instead of roll, allowing him to move 2 blocks at a time. Collecting an arm allows Bolbi to COLLECT other parts without using a turn. Collecting Bolbi’s head allows him to see FF’s smoke trail over the piles of junk, meaning that the player can see exactly where Furnace-Face is on the board. The screwdriver is the only weapon available to Bolbi, and in order to collect it, he needs at least one arm. Ideally the placement of the parts will be randomized, so repeated playthroughs will be different from the first.



## Pseudocode or Flowchart for Code to “Move Between Rooms”

GameplayLoop

{

3TurnPlayerLoop

{

…

...

Player has input “move” after decision prompt

Player position is at this point already saved as pos\_player (x,y)

Check movement options saved as move\_op (north,south,east,west) against board

#will always be +1 OR –1 to x OR y exclusively i.e. “north” corresponds to (0,-1)

#dictionary string directions with duple coordinates should suffice

{

IF proposed x AND y (after addition) are still in range of x,y values for existing blocks on board

Add move\_op strings to prompt asking which direction to go in

}

Loop {

Print prompt only with available move options. On (0,0) starting block, prompt would be “Move which direction? You can type [east] or [south]”

IF input is valid,

update pos\_player with new coordinates.

ELSE

Print “Bolbi can’t go that way.”

}

IF pos\_player is equal to pos\_boss

IF player has all items

Print winner screen “congratulations you beat Furnace-Face”

Break out of loop

ELSE

Print loser screen “Bolbi has been melted down for scrap”

Break out of loop

}

1TurnBossPhase

{

Print warning to player “Bolbi senses movement; Furnace-Face is closing in!”

Compare pos\_boss to pos\_player #boss will only have up to two options for approaching player

{

IF pos\_boss(x) is greater than pos\_player(x)

Store “west” in boss\_move\_h #boss is east +x of player, wants to move west

ELSE IF pos\_boss(x) is less than pos\_player(x)

Store “east” in boss\_move\_h

ELSE

Store “none” in boss\_move\_h #can possibly add “none” to dict with (0,0) for WAIT

}

{

IF pos\_boss(y) is greater than pos\_player(y)

Store “north” in boss\_move\_v

ELSE IF pos\_boss(y) is less than pos\_player(y)

Store “south” in boss\_move\_v

ELSE

Store “none” in boss\_move\_v

}

#Choose which move for boss to make, either horizontally or vertically

IF “none” found in (boss\_move\_h)

Add boss\_move\_v coordinate values to pos\_boss

ELSE IF “none” found in (boss\_move\_v)

Add boss\_move\_h coordinate values to pos\_boss

ELSE

choose randomly between the two

Add those coordinates to pos\_boss

} #end of boss move

}

## Pseudocode or Flowchart for Code to “Get an Item”

…

Item names have been saved as variables with True/False values, initialized as False

Upon board generation, items are assigned to random blocks not already containing Bb or Ff

GameplayLoop

{

…

Give player appropriate upgrade(s) based on which parts are TRUE

...

Player has input “SCAN” after decision prompt

BossScan #check if boss is on adjacent block comparing coordinates

{

IF pos\_boss(x) is equal to pos\_player(x)

IF pos\_boss(y) subtracting pos\_player(y) gives an absolute value of 1

IF maximum of pos\_boss(y) and pos\_player(y) is pos\_boss(y)

Print “Bolbi senses FF, to your South!”

ELSE

Print “Bolbi senses FF, to your North!”

IF pos\_boss(y) is equal to pos\_player(y)

IF pos\_boss(x) subtracting pos\_player(x) gives an absolute value of 1

IF maximum of pos\_boss(x) and pos\_player(x) is pos\_boss(x)

Print “Bolbi senses FF, to your East!”

ELSE

Print “Bolbi senses FF, to your West!”

}

ItemScan

{

IF the block contains a string that matches one of the parts in the item list

Print “Bolbi’s [part name] is here.”

ELSE

Print “There are no parts here”

}

Player has input “COLLECT” after decision prompt

{

#check player position, if it matches the block an item was placed in

IF the block contains a string that matches one of the items in the item list

Set that item variable to TRUE #player now has that item

Print “Bolbi has acquired his [part name].”

ELSE

Print “Cannot collect, nothing here”

}

}