[NU Robot RPG]

# **[Team Members]**

* Jacky Luo
* Zac Snowbank
* Kyle Murphy

# **[Project Description]**

This program is a robot customization and RPG combat game. You will be able to make a base robot and customize it based on loot from destroyed robots. The end game is to liberate the planet from a super robot boss.

The game is an RPG with speed-derived turn-based combat. The parts of the robot determine the complete robot’s stats. You will have the capability to face off the boss between each exploration of a map. You can explore maps of different sizes to collect parts and improve your robot to combat the boss.

# **Part 1: Functional Requirements**

## **Glossary**

* “Player” - The person piloting and building robots in an effort to clear maps and get to bosses.
* “Boss” - An NPC robot of the same level as the player to provide a challenge at the end of each level.
* “Level” - A series of maps the player explores.
* “Robot” - A collection of parts (Head, Torso, Arms (multiple depending on torso), Legs) That together define the Player’s and the NPC’s stats.
* “Parts” - The heads, torsos, arms and legs that make up the robot.
* “Enemy” - Non-boss NPCs the player fights.
* “Depot” - A check point where the player can heal, change out parts and save their progress if their robot is destroyed.

## **Priority**

Each requirement has a priority level of [1], [2], or [3].

1. Must-have functionality critical to the problem solution.
2. Highly desirable feature that should be included.
3. Optional requirements that will be completed if time allows.

## **Basic RPG**

1. The game has all necessary parts and features to run, which are:
   1. Single-player campaign
   2. Robot Components to represent
      1. Head
      2. Torso
      3. Arm
      4. Leg
   3. Map system
      1. Randomly generated size

1.3.1.1 Grid tiles from a single number 10-20 as 10\*10, 15\*15, 13\*13, etc.

1.3.2. Enemies based on difficulty with part rarity

1.3.2.1. Easy with 75% common drops, 25% uncommon drops

1.3.2.2. Medium with 25% common drops, 50% uncommon drops, 24% rare drops, 1% experimental drops

1.3.2.3. Hard with 50% uncommon drops, 30% rare drops, 20% experimental drops

* 1. GUI

(buttons are denoted in quotes)

* + 1. Startup Screen
       1. “Create New Game”
       2. “Load Saved Game”
       3. “Exit”
    2. Create New Game Screen
       1. User selects one basic item from each type
       2. User selects difficulty
       3. “Start Game”
       4. “Back to Previous Screen”
    3. Load Saved Game Screen
       1. Navigate to or prompt for file path to saved game
       2. “Load Game”
       3. “Back to Previous Screen”
    4. Play Game Screen
       1. Display areas for stats and equipped objects for Robots
       2. Display area for status updates and user communication
       3. Display area for map system
       4. Buttons for map movement (“up”, “down”, “left”, or “right”)
    5. Depot Screen
       1. Display for equipped parts properties
       2. Display for inventory parts properties
       3. “Change Parts”
          1. Menu of equipped parts
          2. Menu of inventory parts
          3. “Switch Part”
          4. “Done”
       4. “Continue”
       5. “Save Game”
       6. “Exit”
    6. Combat Screen
       1. “Attack”
          1. “Attack Type”
       2. “Defend”
          1. “Defend Type”
    7. Loot Screen
       1. Display area for loot type and properties
       2. “Keep Loot”
       3. “Continue”

1.5. Basic Combat system

1.5.1. Each robot takes a turn to choose an attack which does damage minus the armor of the

other robot

1.5.2. Robot speed will only determine who goes first

2. More Randomization and utility options

2.1. Save files will be stored so you can load your previous progress

2.2. Adding different attack types to add weakness and resistance

2.1.1. Ballistic

2.1.2. Beam

2.1.3. Explosive

2.1.4. Fire

2.1.5. Electricity

2.3. Parts have completely randomized stats instead of a pre-determined set of pieces

2.3.1. The enemies will be pre-made except the piece they will drop

2.3.2. The rarer the item the better it will be, with experimental parts named to show how good they are

2.4. The robot parts have set requirements to combine together

2.4.1. Legs can only support so much weight

2.4.2 Some torsos must be attached to tread legs

2.5. Speed based action using a base robot speed

2.5.1. Robots with faster speed will eventually have an extra turn in combat

2.6. Parts have special actions attached to them

2.6.1. Torsos and Heads have more variety, like allowing additional arms to be attached

2.7. Adding a high score system

2.8. A system to save robots so they can quickly pull up different load-outs

3. In-depth expansion on functions and storyline

3.1. The player can scrap pieces to make get resources and buy pieces at depots instead of only

finding them

3.2. Parts are made of different material that determine their resistances, weight, and armor

3.3. Adding equipped weapons to use

3.4. Adding a story mode

3.5. More planets to free with higher difficulty

3.6. A shop system with currency to buy and sell more parts

# **Part 2: Objective Coverage**

Of the topics designated as topics discussed in the OOP class, we will use the following:

* Inheritance will be used with a base parts class to the arms, legs, torso, and head
* Will be using a multidimensional array to represent the map
* Using a hashmap to save favorite robots
* The save file system will need to have exception handling and file I/O

# **Part 3: Basic Design**

# **Part 4: Project Plan**

1. Week 1
   1. Complete Tier 1: core features in robot, parts, and the map
   2. Start GUI interface
2. Week 2
   1. Finish and debug Tier 1 features
   2. Implement systems to GUI
   3. Start tier 2 requirements for robot parts and combat
3. Week 3
   1. Finish game systems and implements tier 3 elements as time allows
4. Week 4
   1. Present project

