**Rogue-Bot**

A traditional turn-based Rogue-like emphasizing treasure-hunting and combat through a procedurally generated labyrinth.

**Game Design Documentation**

A Terminal Audacity Production

Connor Hilarides

Technical Director

Troy K. B. de Magro

Lead Designer

Claire Robsahm

Systems Engineer

Enrique Rodriguez

Gameplay Architect

Leonardo Saikali

Producer

GAM 200

Fall 2014

**Table of Contents:**

[High Concept:](#h.vfrzuy7m2vlg)

[Introduction:](#h.p1l419bfbn4z)

[Game Flow:](#h.lnl7hpv57izs)

[Game Mechanics:](#h.kmj0sxanomfv)

[Game Characters:](#h.j0vu576wn95v)

[Game Resources:](#h.j0vu576wn95v)

[Game Environment:](#h.stkg4sqt56d8)

[Game Controls:](#h.pjla5xpz4c40)

[Audio Design:](#h.pjla5xpz4c40)

[Physics Design:](#h.xdnsh74jo4gh)

[Multiplayer Design:](#h.xdnsh74jo4gh)

[Appendix A (Story):](#h.pjla5xpz4c40)

[Appendix B (Art List):](#h.iqb24j9m1yhs)

[Appendix C (Audio List):](#h.2dynu8qgelew)

[Appendix D (Interface Flow):](#h.p9df7np5ezf4)

# High Concept:

Rogue-Bot is a traditional real-time Rogue-like emphasizing exploration and combat through a semi-randomly generated labyrinth.

# 

# Introduction:

Rogue-Bot takes place in the depths of a ruined and unexplored dungeon. The player assumes the role of robots built by the Rogue-Bot factory, who are sent out one at a time to explore these dungeons and reclaim the loot for the company. Each robot is controlled manually (by the player), who must use scattered materials, tools, armour, & weapons to explore the dungeon, combat hostile forces, and reclaim as much treasure as possible.

Exploration will take place over a constant ground-level hub that connects to fully randomized dungeon-esque complexes, making each play-through unique. When a player robot is destroyed, it can send back some of the resources & items it collected back to the factory so that the next robot can be built stronger, faster, and more versatile. Robots can also be upgraded while active through equipment upgrades found while exploring.

# 

# Game Flow:

To start, the player will build a robot. This feature will expand in depth as the game is played and more upgrades can be built with materials & items found by previous robots. Then the robot is sent into the (game HUB area name not decided), a consistent area with a few enemies that scale in difficulty as the game progresses. From the HUB the player can access the randomly generated dungeons, where they can find various items & loot, fight enemies, (and as a stretch goal, a boss at the end of each one), and come across certain special rooms. Regardless, at the end of each dungeon there will be a special item needed to explore more of the (game HUB area name not decided), progress the story, and access new dungeons.

# 

# Game Mechanics:

Core Mechanics:

Grid-Based Movement ~ Up, down, left, & right for both the player and AI entities. Weapons can be fired in the eight cardinal directions.

Item-Based Leveling ~ Instead of focusing on stats, items will confer different status effects

Attacks Appear On Grid ~ For instance, when a missile is fired it also moves across the grid to its target (instead of instantly hitting). Different attacks can have different speeds/effects on the grid and its inhabitants.

Progressional Permadeath ~ The player gets to use their robot for one “life”. After that they’ll have to build a new one using parts & items salvaged during the previous robot’s exploration. The extent of the progression aspect depends on stretch goals.

Damage Types ~ Different weapons deal damage in different ways. For instance, a flamethrower would deal fire damage, a rusty wooden sword would deal melee damage, and so forth. Armour is used to protect against certain damage types.

Simplified Player Health ~ The player always takes one damage when successfully hit. This may or may not apply to enemies (something to be tested) but the idea is that the player shouldn’t have to focus too much on their current health, knowing how much damage they are guaranteed to take if hit.

Secondary Mechanics:

Loot/Currency ~ The player can collect what is effectively money to spend in shops and carry over from a defeated bot to upgrade the next bot the player sends out into the dungeons.

Real-Time Traps ~ In the turn-based system there can be real-time traps to change the pace of the game. Currently debating whether or not to have enemies be affected by them as well, and how much if so.

Destructible Terrain ~ Certain terrain can be destroyed by certain weapon effects. Ideas include drill weapons that are meant to be used on terrain, or really explosive weapons (like high-quality missiles or explosive gauss shells) destroying weak walls. Ideally there will be more than one destructible terrain type where its strength against terrain-destroying weapons increases.

Robot Creation ~ Players build their robots before sending them out into the game. Previous robots can bring back items/loot to help build later robots, increasing the player’s abilities within the game despite the permadeath.

# Game Characters:

Rogue-Bot:

A bot manufactured by D-INC to explore dungeons. This is who the player controls in-game.

Purpose: Spelunking the dungeons to acquire loot for D-INC.

Abilities: Carrying lots of weapons and loot, using lots of weapons in combat, exploring and mapping deep underground labyrinths.

Shop Owner:

There appear to be a few peaceful inhabitants of the dungeons. They seems to subsist off of a primitive marketing system, in which they trade their goods for other goods with the various dungeon denizens.

Purpose: Running a store so that the player can buy and sell things.

Abilities: Buying/selling goods, storing goods, repairing goods, creating goods of their own.

Jellatin:

Many types of Jellatin have been documented as inhabitants of most dungeons. They are generally imbued with an element such as coldness, electricity, or radioactivity to name a few. They have a nasty tendency of attacking Rogue-Bots.

Purpose: A common obstacle & foe of the Rogue-Bots.

Abilities: Being radioactive and/or acidic, damaging robots.

We also plan to implement many other enemies, but have decided that testing with them will take place in GAM 250. Here is a draft list of potential enemies and enemy types.

Skeleton/Zombie  
 Move: Will chase if player is close.  
 Attack: Attacks player if able.  
  
Imp  
 Move: Will try to maintain a 3 space distance from the player.  
 Attack: Shoots 2 spc/turn fireballs.  
  
Mole  
 Move: On odd turns, Will dig underground and show a dirt pile 0-5 spaces from previous location.  
 On even turns, will pop up from dirt pile.  
 Attack: Will attack player if player is on top of or adjacent to dirt pile when it pops up.  
 Will move player if pops up underneath.  
 Extra: Cannot be hurt when underground.  
  
Living Armor  
 Move: Will chase player if near. Moves slowly.  
 Attack: Will attack player if able.  
 Extra: Carries a shield. Cannot be hurt from direction the shield is facing.  
  
Leaper  
 Move: Will chase player if near. Can instantly move 1-5 spaces in a straight line to attack player.  
 Attack: Will leap at player and attack player if able.  
  
Treasure Imp  
 Move: Will run from player if close. Moves Quickly.  
 Attack: None.  
 Extra: Will ALWAYS drop an item when killed and money.  
  
Cultist  
 Move: Will try to stay close to enemies. Runs if player is close.  
 Attack: None.  
 Extra: Heals enemies.  
  
Statue  
 Move: None.  
 Attack: Will constantly shoot projectiles in the direction it is facing.  
  
Bomb Imp  
 Move: Will chase player if near.  
 Attack: Will suicide explode if next to player, damaging everything around it.  
 Extra: Will explode when killed, damaging everything around it.  
  
Scavenger  
 Move: Will chase player if near. Will try to align itself with player when engaged.  
 Attack: If aligned with player and NOT adjacent, will shoot projectiles. If adjacent to player, will attack with melee.  
 Extra: "Looks like an imp with robot parts similar to what the player has attached to it."  
 High chance to drop loot.  
  
Competitor  
 Move: Will chase player/monsters if near. Will try to align itself with player/monsters when engaged.  
 Attack: If aligned with player/monster and NOT adjacent, will shoot projectiles. If adjacent to player/monster, will attack with melee.  
 Extra: "Evil looking robot that looks fairly similar to the player. Implies other robots similar to the player exist in the dungeon; though, not necessarily friendly."  
 High chance to drop loot and lots of money.  
  
Collaborator  
 Move: Will move randomly but chase monsters if they are near.  
 Attack: If aligned with monster and NOT adjacent, will shoot projectiles. If adjacent to monster, will attack with melee.

# Game Resources:

Most of the items in-game will be created via random item generation following the below formula:

[Attribute] + [Material] + [Base Item] + of [Type]

(Optional) (Needed) (Needed) (Optional)

So as to support items such as these examples:

Napalm Missile

Iron Headbucket

Lead Cannonball of Explosions

Sturdy Diamond Arrow

Rusty Wooden Gloves of Flames

Shoddy Plastic Axe of Ice

Attributes, materials, and base weapons all can affect the item stats, while materials and types can affect what damage type the item is of.

The idea behind damage types is that weapons can deal damage in one or two ways. For instance, a Radioactive Rock could deal Radioactive or Physical damage. Meanwhile armour can provide resistances (1/2 damage) or immunity (no damage) from a damage type. So an enemy that's immune to Radioactive type weapons would not be affected by the Radioactive part of the Radioactive Rock but would be affected by the Physical part and would still take damage.

And yes, silly things like a Wooden Flamethrower of Ice are possible in this setup. That's how this game is going to play out; silly, over the top, and audacious.

# 

# Game Environment:

Core Areas:

The HUB Level ~ The player starts on a mostly/fully persistent level. It’s large and contains a number of special purpose rooms, including the entrances to the randomly-generated dungeons.

Dungeons ~ The bulk of the game (and the other special rooms) take place in dungeons. These are randomly-generated strings of multiple floors (the amount in each dungeon is determined semi-randomly). Floors will have differing themes (ruins, volcano, ice cave, et cetera).

Final Floors ~ The last floor of each dungeon can be more specialized. These will contain key items towards completing the game, other powerful items, and powerful enemies/bosses and trap-filled rooms depending on accomplished stretch goals.

Secondary Areas:

Shops ~ Using the currency system the player can buy various (semi-random or fully random) items here. Depending on scope these shops could be specialized in weapons or armour, or potentially even items sold based on damage types.

Trap Halls ~ Special rooms filled with traps. Will have valuable items/loot at their ends.

Enemy Zoo ~ Special rooms packed with enemies. Will have valuable items/loot at their ends. Could either contain lots of enemies or contain enemy spawners that need to be destroyed.

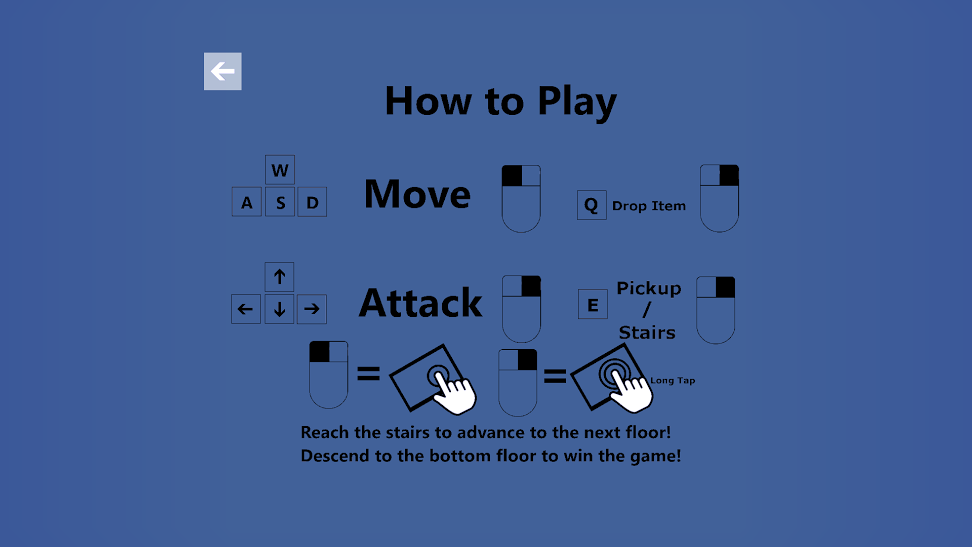
Deconnected Rooms ~ Rooms separated by large quantities of destructible terrain. The strength of the destructible terrain can depend on the value of the items/loot inside.

Locked Rooms ~ Rooms with more items/loot than usual guarded by locked doors. The strength of the door can depend on the value of the items/loot inside.

# Game Controls:

Our game is grid-based, with the player & other beings moving around the mine being restricted to the cardinal directions (up, down, left, & right) with projectile entities that can move in the diagonal directions as well. The player moves their Rogue-Bot with WASD control and can use melee attacks using the arrow keys. The player can also use the number pad to fire projectiles, Q to drop items and E to pick up items or interact with nearby objects.

We also plan to make the game fully playable with the mouse only (for tablets specifically). In this case, left clicking a tile moves you there and right clicking does an action (attacking, picking up items) there.



*An Alpha how-to-play screen.*

# Visual Design:

As a base the game is top-down and 2D (though a 3D option can easily be made available in the current engine build). Either way, sprites are going to be 2D and generally cartoonish in nature. This is meant to tie in with the comedic nature of the game, the retro audio style, and our dynamic item systems (which will use tinting to re-colour items that are of the same type but have different attributes).



*An Alpha gameplay screenshot.*

As a stretch goal we could utilize various shaders and/or lighting effects in special circumstances (extreme temperature, status ailments, nearby explosions, et cetera). Otherwise our main concern with the visuals are how they affect player feedback, so we want visuals to be easily discernable from each other.

# Audio Design:

The current plan for both sound effects and music is to compose both in 8-bit, emulating the SID chip from the Commodore 64. Sounds will be a crucial aspect of our player feedback system, and so we plan to utilize sound effects for most player actions (everything from selecting weapons to moving to picking up items). If it’s a thing the player can do, chances are that there will be a sound effect tied to it.

As far as music, we want to give it a constant presence in the game. For this we’ll need music tracks for special areas (the menu, shops, bosses battles, et cetera) and some general purpose songs for when the player is simply exploring the dungeons. These tracks will be used in said specific areas to ensure variety during lengthy play sessions.

All in all, sound & music should be happening constantly, but also smoothly. Sounds should not be jarring, and the sounds & music should not be working against each other. The point of this to create a sense of immersion with the sound.

# Physics Design:

Our game will not need physics. All of the gameplay takes place on a grid, and so we will be using that grid for positioning functions and all of the space elements that go with it. More minor details like particle effects won’t need an entire physics engine to implement either.

# 

# Multiplayer Design:

Currently we have no plans to introduce multiplayer into our game. A high score system is a possible stretch goal, but nothing more.

# Appendix A (Story):

While story is not a major element of our game, there is still a general plotline.

In the world, there are many ruins leading into deep underground labyrinths. While most of the people of the surface don’t have so much as a passing thought of these ruins in their high-tech society, Dungeoneering Incorporated does. The fine folks at D-INC have realised that these ruins tend to contain lots of treasure, and treasure still turns quite the profit in the modern market.

There ruins are dangerous though; even beyond the superstitions and curses, many dangerous creatures call the ruins home, and the people who built these structures put in many traps to keep looters from stealing the loot that was once theirs. As such, D-INC has a factory that produces “Rogue-Bots” to do the dirty work of fighting the chaotic hordes and retrieving the treasures. You are one of these Rogue-Bots. Good luck on your mission!

# Appendix B (Art List):

Sprites

In Short: We will have a sprite for each enemy, NPC, menu icon, projectile, particle, item, and animation in the game. This number depends on the amount these entities we implement.

|  |  |
| --- | --- |
| **Sprite** | **Purpose** |
| player.jpg | The player sprite. |
| how\_to.jpg | The how-to play panel. |
| back\_arrow.jpg | Menu icon for going back. |
| checkmark.jpg | Menu icon for continuing with an action. |
| option.jpg | Menu icon for options. |
| credits.jpg | Menu icon for credits. |
| jellatin.jpg | The sprite for the Jellatin enemies. |
| *other enemy sprites* | Every enemy needs a sprite. |
| shop\_owner.jpg | The sprite for Shop Owners. |
| *other NPCs* | Should we implement more. |
| bomb.jpg | The sprite for bomb projectiles. |
| *other projectiles* | Every projectile needs a sprite. |
| *particles* | Particle effects are not a priority for spriting. |
| team\_logo.jpg | File for the team logo |
| digipen.jpg | File for the DigiPen splash. |
| dagger\_anim.jpg | Animation for attacking with a dagger. |
| saber\_anim.jpg | Animation for attacking with a saber. |
| spear\_anim.jpg | Animation for attacking with a spear. |
| axe\_anim.jpg | Animation for attacking with a axe. |
| drill\_anim.jpg | Animation for attacking with a drill. |
| *any other attacks* | All attack animations need a sprite. |
| dagger.jpg | Icon for a dagger. |
| saber.jpg | Icon for a saber. |
| spear.jpg | Icon for a spear. |
| axe.jpg | Icon for a axe. |
| drill.jpg | Icon for a drill. |
| *any other items* | All items need a sprite. |
| stairs.jpg | Icon for stairs (to the next floor!) |

Models

Our game does not use models of any sort.

Textures

In Short: We will have a set of textures for each set of walls, floors, and destructable walls in the game. This number depends on the amount of dungeon varieties we implement.

|  |  |
| --- | --- |
| **Texture** | **Used in...** |
| ruin\_floor.jpg | Ruins areas (the floor) |
| ruin\_wall.jpg | Ruins areas (the walls) |
| ruin\_destruct.jpg | Ruins areas (the destructable terrain) |
| dungeon\_floor.jpg | Dungeon areas (the floor) |
| dungeon\_wall.jpg | Dungeon areas (the walls) |
| dungeon\_destruct.jpg | Dungeon areas (the destructable terrain) |
| cave\_floor.jpg | Cave areas (the floor) |
| cave\_wall.jpg | Cave areas (the walls) |
| cave\_destruct.jpg | Cave areas (the destructable terrain) |
| final\_floor.jpg | Final areas (the floor) |
| final\_wall.jpg | Final areas (the walls) |
| final\_destruct.jpg | Final areas (the destructable terrain) |
| shop\_floor.jpg | Shop floor (is a special area denoting special funtionalities). |

Fonts

We will be using Segoe UI for the game and Consolas for the debug console.

Source

All visual artwork is planned to be drawn by Leonardo Saikali (so within our team).

# Appendix C (Audio List):

Sounds

Note: All sounds will have stereo capabilities, will not loop, and will be in .ogg files for the final product. Sampling frequencies and bit rate are to be determined during sound production.

|  |  |
| --- | --- |
| **Sound** | **Plays when...** |
| menu\_cursor.ogg | The mouse hovers over a menu option. |
| menu\_select.ogg | A menu option is clicked on. |
| move\_mech.ogg | A mechanical entity in the game moves. |
| move\_sludge.ogg | A non-solid entity in the game moves. |
| move\_default.ogg | Any other entity in the game moves. |
| item\_get.ogg | An item is collected. |
| hit\_mech.ogg | A mechanical entity is hit with a physical attack. |
| hit\_sludge.ogg | A non-solid entity is hit with a physical attack. |
| hit\_default.ogg | Any other entity is hit with a physical attack. |

|  |  |
| --- | --- |
| **Sound** | **Plays when...** |
| hit\_flames.ogg | A flame-based attack is made. |
| hit\_freezing.ogg | A menu option is clicked on. |
| hit\_radioactive.ogg | A mechanical entity in the game moves. |
| hit\_zapping.ogg | A non-solid entity in the game moves. |
| hit\_resist.ogg | Someone is hit by an attack type they resist. |
| hit\_immune.ogg | Someone has immunity to whatever just hit them. |
| player\_shot.ogg | The player/ally shoots a projectile. |
| enemy\_shot.ogg | An enemy shoots a projectile. |
| explosion.ogg | An explosion occurs in any circumstance. |
| kill\_mech.ogg | A mechanical entity is destroyed. |
| kill\_sludge.ogg | A non-solid entity is destroyed. |
| kill\_default.ogg | Any other entity is destroyed. |
| weapon\_select.ogg | A different weapon is selected. |
| armour\_on.ogg | A piece of armour is put on. |
| armour\_off.ogg | A piece of armour is taken off. |
| heal.ogg | Someone is healed. |
| trap.ogg | A trap is set off. |
| drill.ogg | A drill is used and doesn’t fail. |
| drill\_fail.ogg | A drill fails to do its job. |

Music

Note: All music will have stereo capabilities and will be in .ogg files for the final product. Sampling frequencies and bit rate are to be determined during sound production. Music not labeled as **Crucial** is a stretch goal. Areas labeled *(Maybe)* are of undecided importance. Music file names are tentative.

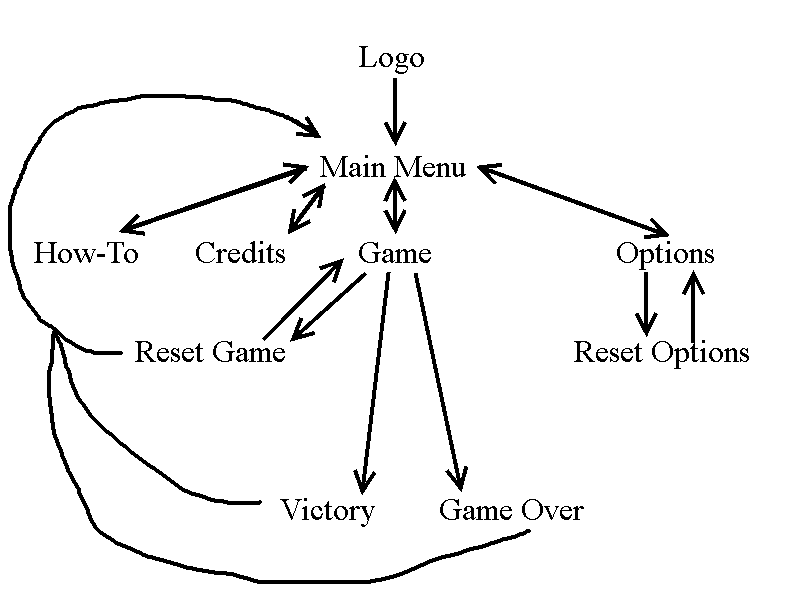
|  |  |  |  |
| --- | --- | --- | --- |
| **Music** | **Intended Setting(s)** | **Crucial** | **Looping** |
| logo.ogg | Startup logo for DigiPen/Terminal Audacity. |  |  |
| title.ogg | Title screen. | Yes | Yes |
| buildabot.ogg | When building a new Rogue-Bot (new game). |  | Yes |
| hub\_level.ogg | The HUB level. | Yes | Yes |
| spelunkying.ogg | Standard level theme. | Yes | *(Maybe)* |
| ancient\_depths.ogg | Standard level theme (ruinous areas). |  |  |
| lost\_halls.ogg | Standard level theme (non-ruinous areas).. |  |  |
| labyrinthia.ogg | A secondary standard level theme. | *(Maybe)* |  |
| *other level themes* | A stretch goal for 1-3 songs per area type. |  |  |
| shop.ogg | In shops. | Yes | Yes |
| the\_horde.ogg | Upon discovery of an Enemy Zoo. |  | *(Maybe)* |
| boss.ogg | A general boss theme. | Yes | Yes |
| *unique boss themes* | A stretch goal for a song per boss. |  | Yes |
| conquered.ogg | When an entire dungeon is cleared. | Yes |  |
| final\_floor.ogg | A stretch goal for a “final area”. |  | *(Maybe)* |
| victory.ogg | The game is completed. | Yes | Yes |
| defeat.ogg | The game is lost. | Yes | Yes |
| hall\_of\_fame.ogg | A high-scores list. |  | Yes |

Source

All audio is planned to be composed by Troy K. B. de Magro (so within our team).

# Appendix D (Interface Flow):

Flow Chart:

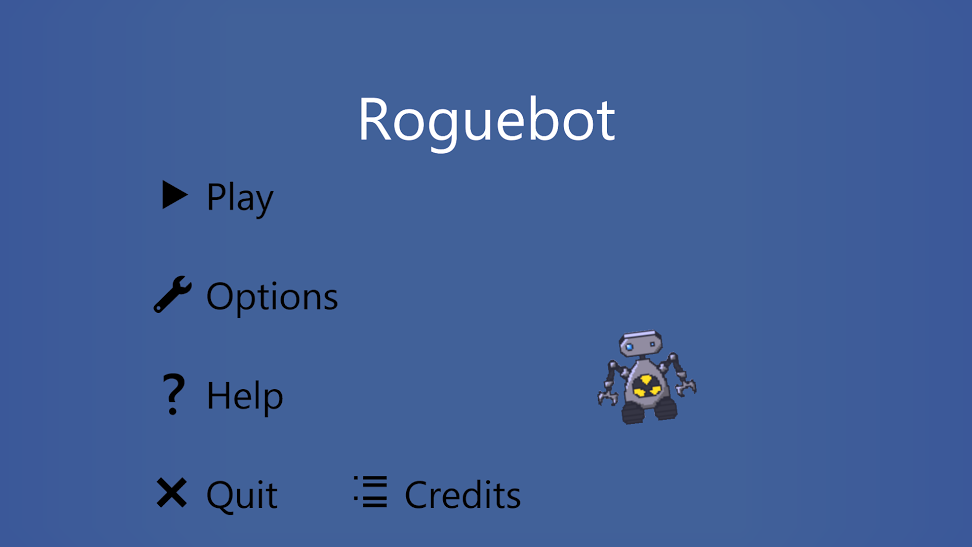


Alpha Screenshots of the above listed screens:

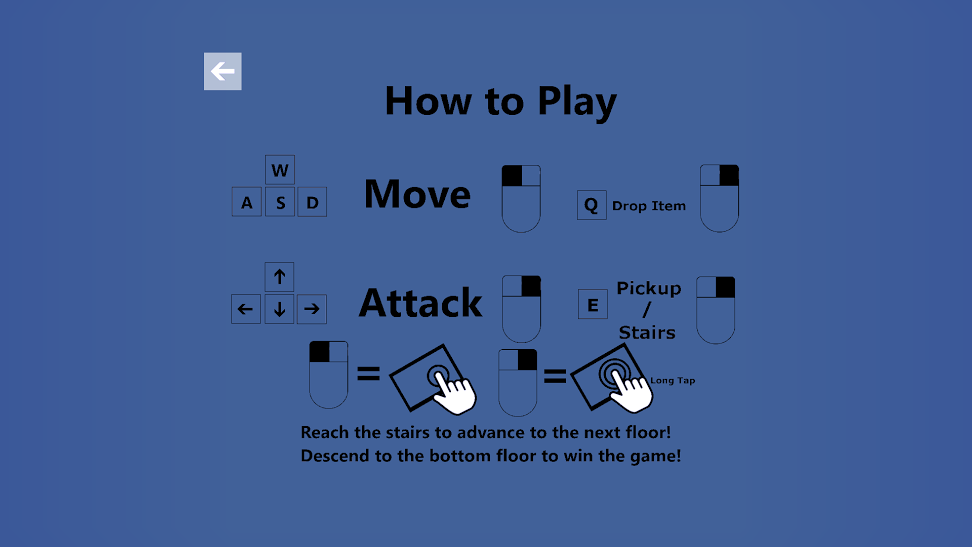
Logo



Main Menu



How-To



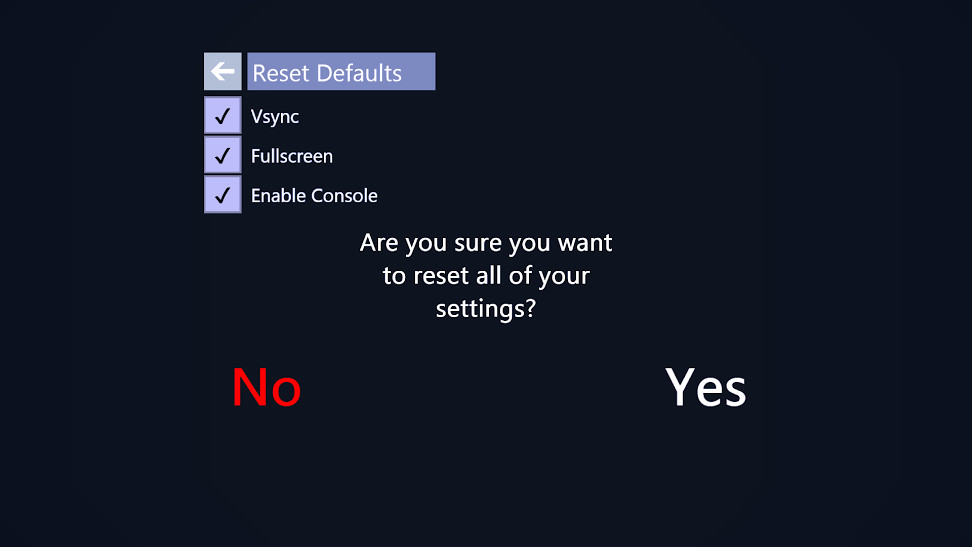
Credits



Options



Reset Options

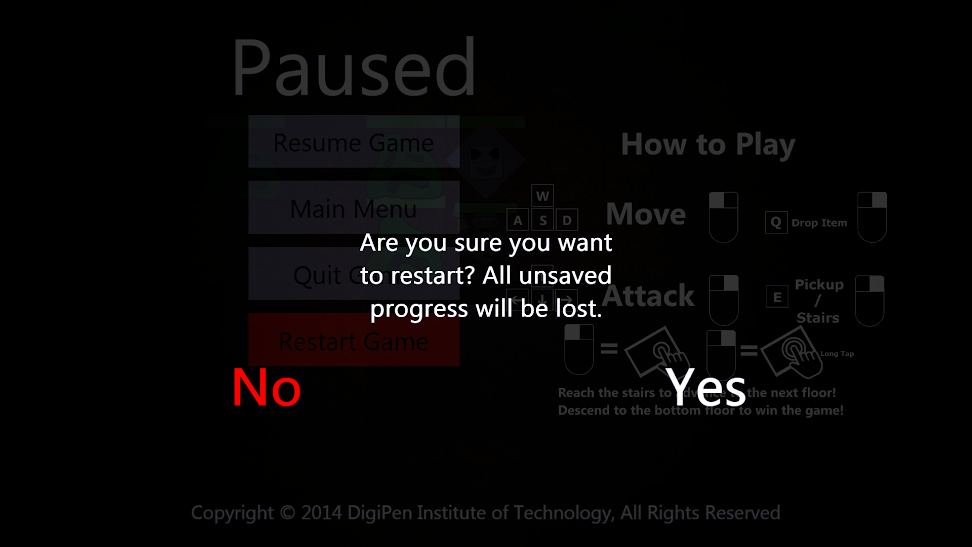


Game



*(The Game also has its own menu, from which it accesses the Reset Game screen)*

Reset Game



Victory



Game Over

