horizontal line

**Ranken Technical College**

4431 Finney Ave

St Louis, MO 63113

Roguelite Midterm

**Due 15th November 2017**

# OVERVIEW

The goal of this assignment is to create a lightweight roguelike game. You can make this game using your choice of console, windows forms, or Unity (as a 2D game).

**These projects must be submitted and graded through a bitbucket repository for credit. This is an individual assignment and should not be worked on in groups. Your teacher can clarify the requirements and expectations but will not help write/debug the code for this assignment.**

# GOALS

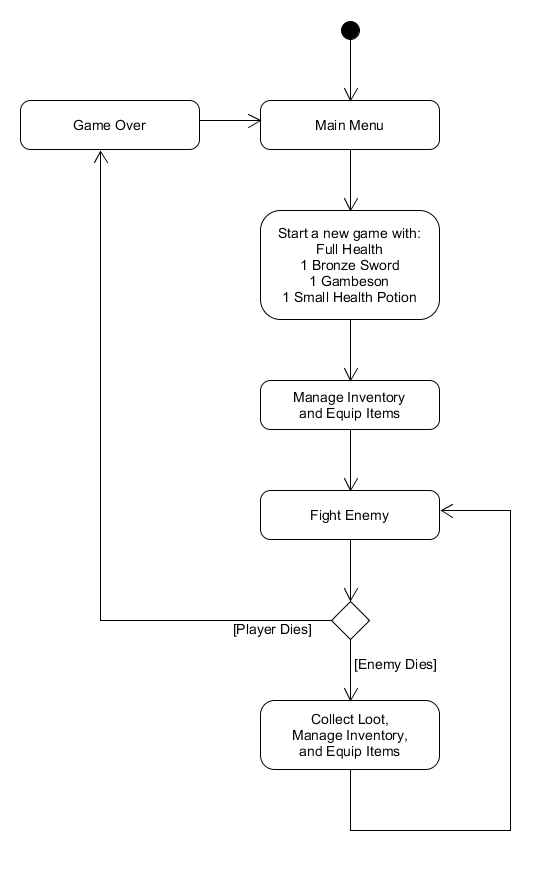
1. The player can fight randomly generated enemies in simple turn based combat.
2. The player can collect loot from defeated enemies.
3. The player can manage their inventory and equipped items.

# 

# 

# SPECIFICATIONS

The basic gameplay loop will be as depicted in the below UML Activity Diagram.



# Equipment

The player can carry up to 20 unequipped items.

The player has 7 slots to equip the following:

* Helmet
* Chestpiece
* Grieves
* Vambraces
* 1 Weapon
* 2 Potions

# Combat

During combat the player and their enemy will take turns in alternating order.

The player will always get the first turn.

Drinking a potion is a free action. (Doesn’t cost a turn.)

Damage is calculated as follows:

Damage Dealt = (Total Attacker Attack Value) - (Total Defender Defense Value)

When an enemy is defeated they will drop all of their equipped items plus one bonus item.

After every five regular enemies the player will fight a miniboss.

Minibosses will drop their equipped items plus three bonus items.

# Enemy List

Here are some enemy templates you can use (you can also create your own if want). Implement at least 3 templates for regular enemies (at least one of which must be a wizard) and at least 1 miniboss template. Enemies with a range of stat values will spawn with a fixed number in that range.

1. Rat
   1. Natural Weapon
   2. Attack Value 1-3
   3. Defense Value 0
2. Dog
   1. Natural Weapon
   2. Attack Value 2-4
   3. Defense Value 1
3. Bat
   1. Natural Weapon
   2. Attack Value 1-3
   3. Defense Value 3
4. Kobold
   1. Bronze Sword *(see Item List for stats)*
   2. Gambeson *(see Item List for stats)*
5. Goblin
   1. Iron Sword *(see Item List for stats)*
   2. Gambeson *(see Item List for stats)*
6. Orc
   1. Iron Sword *(see Item List for stats)*
   2. Iron Chestpiece *(see Item List for stats)*

# Item List

Any time you need to spawn random gear or loot, use the following item list.

Items with a range of stat values will spawn with a fixed number in that range.

(You do not have to implement all of these items. For full credit you need 10+ items.)

1. Bronze Sword
   1. Weapon Slot
   2. Attack Value 4-10
2. Iron Sword
   1. Weapon Slot
   2. Attack Value 8-14
3. Steel Sword
   1. Weapon Slot
   2. Attack Value 12-18
4. Buckler
   1. Weapon Slot
   2. Defense Value 3-6
   3. Attack Value 4-10
5. Gambeson
   1. Chestpiece Slot
   2. Defense Value 0-3
6. Iron Chestpiece
   1. Chestpiece Slot
   2. Defense Value 2-5
7. Steel Chestpiece
   1. Chestpiece Slot
   2. Defense Value 4-7
8. Bronze Helmet
   1. Helmet Slot
   2. Defense Value 0-3
9. Iron Helmet
   1. Helmet Slot
   2. Defense Value 2-5
10. Steel Helmet
    1. Helmet Slot
    2. Defense Value 4-7
11. Spiky Helmet
    1. Helmet Slot
    2. Defense Value 3-6
    3. Attack Value 4-10
12. Bronze Grieves
    1. Grieves Slot
    2. Defense Value 0-3
13. Iron Grieves
    1. Grieves Slot
    2. Defense Value 2-5
14. Steel Grieves
    1. Grieves Slot
    2. Defense Value 4-7
15. Bronze Vambraces
    1. Vambraces Slot
    2. Defense Value 0-3
16. Iron Vambraces
    1. Vambraces Slot
    2. Defense Value 2-5
17. Steel Vambraces
    1. Vambraces Slot
    2. Defense Value 4-7
18. Small Health Potion
    1. Potion Slot
    2. Heal Value 10-15
19. Large Health Potion
    1. Potion Slot
    2. Heal Value 20-25
20. Poison Vial
    1. Potion Slot
    2. Heal Value -10 to -15

# Screen Mockups

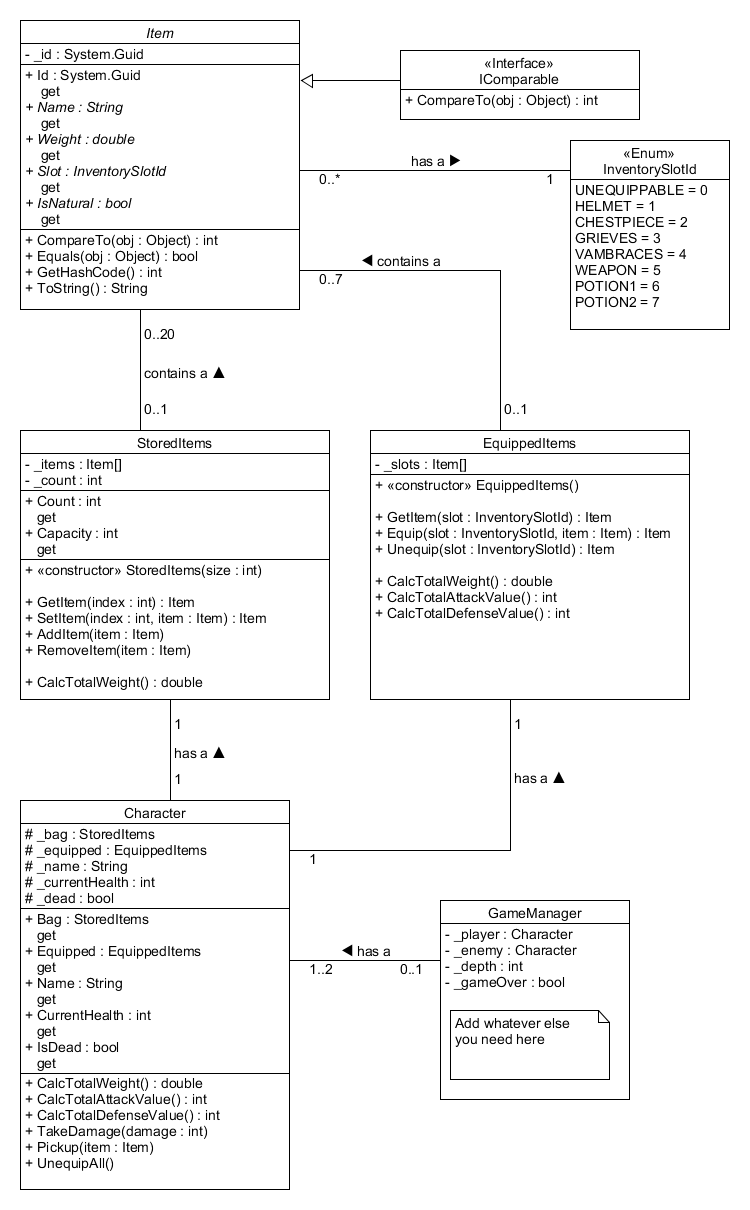
Below are the 4 screens that you should have in your game. You are free to take artistic liberty with how these look and are laid out, but make sure the basic functionality is there.

|  |  |
| --- | --- |
| **Main Menu.png** | **Game Over.png** |
| **Combat.png** | **Equipment.png** |

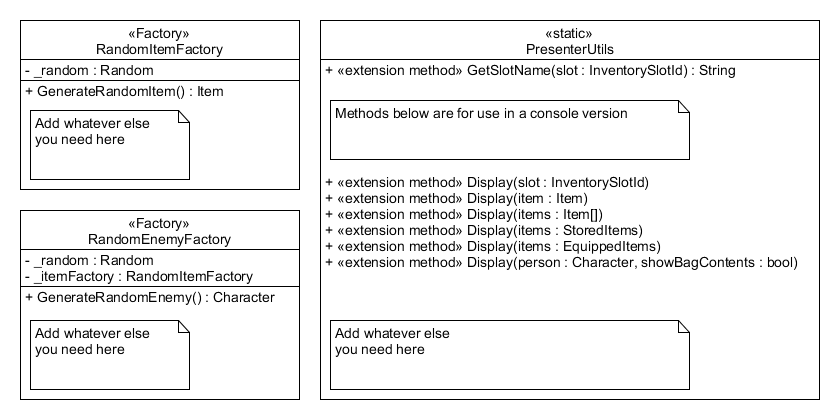
# 

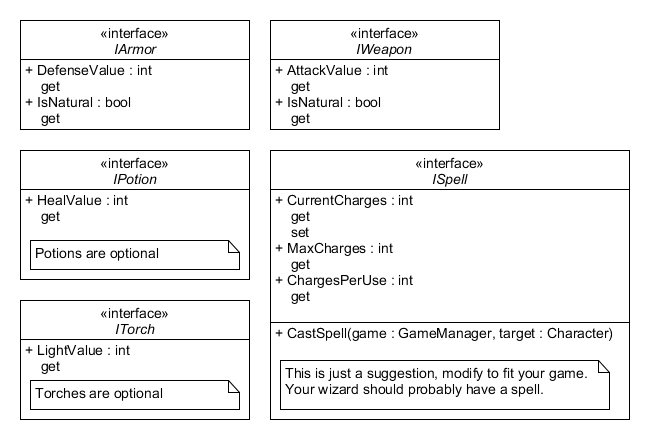
# Architecture

The UML Class Diagram below gives an overview of the core classes in the system.

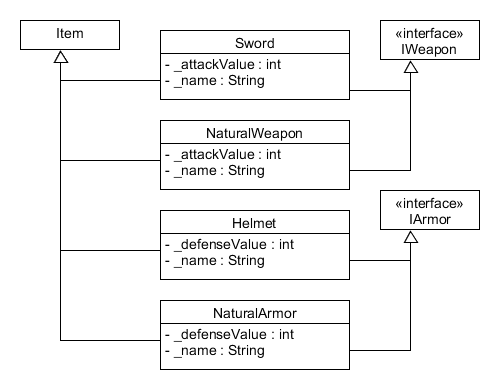


Below are some optional utility classes that you may want.

Below are diagrams for the different interfaces that your Item subclasses should implement as needed.



So for example you might implement Swords and Helmets as follows. (Note that these class diagrams are intentionally incomplete. Fill in details as needed, but follow good coding practices.)



# MILESTONES - These are suggested project iterations and an estimate of how long we believe it could take an average student. Bold items are what’s being added to each Milestone.

## M1 - 1 day

* **Fight a single randomly generated enemy using only your starting gear.**

## M2 - 2 days

* **Fight randomly generated enemies until you die using only your starting gear.**

## M3 - 5 days

* **Fight randomly generated enemies until you die.**
* **Allow the player to loot and equip items between battles.**
* **Allow the player to drop items they don’t want.**

## M4 - 7 days

* Fight randomly generated enemies until you die.
* Allow the player to loot and equip items between battles.
* Allow the player to drop items they don’t want.
* **Allow the player to equip and use potions.**

## M5 - 9 days

* Fight randomly generated enemies until you die.
* Allow the player to loot and equip items between battles.
* Allow the player to drop items they don’t want.
* Allow the player to equip and use potions.
* **Scale the difficulty of enemies based on the player’s current depth.**
* **Spawn minibosses after every five regular enemies.**

## M6 - 11 days

* Fight randomly generated enemies until you die.
* Allow the player to loot and equip items between battles.
* Allow the player to drop items they don’t want.
* Allow the player to equip and use potions.
* Scale the difficulty of enemies based on the player’s current depth.
* Add minibosses after every five enemies.
* **Allow the player to leave the dungeon and bank their items after defeating a miniboss.**

## 

# Core Grading Rubric (95% see bonus points for additional points)

## +35% for building a working combat system

* 5% first enemy class
* 2.5% for second enemy class
* 2.5% for third enemy class (must be a wizard)
* 5% for miniboss class (must be stronger than a regular enemy)
* 2.5% for keeping track of and displaying player health during a battle
* 2.5% for keeping track of and displaying enemy health during a battle
* 2.5% for player attack (initiated by player action)
* 2.5% for enemy attack (initiated automatically)
* 10% for loot system

## +45% for building a system to equip and unequip items between battles.

* 10% Allow the player to equip items between battles
* 10% Allow the player to unequip items between battles
* 5% Implement the StoredItems class (indexer optional)
* 5% Implement the EquippedItems class (indexer optional)
* 2.5% Implement the Item class
* 2.5% Implement the Character class
* 2.5% Implement the GameManager class
* 2.5% Implement the PresenterUtils class
* 2.5% Implement at least 5 Item types
* 2.5% Implement at least 10 Item types

## +15% for allowing the player to drop items they don’t want between battles.

* 5% user interface for dropping items
* 10% implementing the logic and collapsing empty inventory space (without creating a new array)

# Bonus Points (max +30%)

## +5% for making the game in Unity

## +2.5% for creating your own art assets

## +2.5% for integrating your art assets

## +2.5% for creating your own sound effects

## +2.5% for integrating your sound effects

## +2.5% for allowing the player to equip/unequip potions between battles.

## +2.5% for allowing the player to use potions in battle.

## +2.5% for scaling the difficulty of enemies based on the player’s depth within the dungeon.

## +2.5% for scaling the quality of loot drop based on the difficulty of the enemy.

## +5% for allowing the player to leave the dungeon and bank their items (and XP) after a miniboss. (Items and XP are lost when the player dies.)

## +2.5% for adding an XP system to the game. (Defeating enemies grants XP, and leveling up increases your stats or grants bonus gear.)

## +2.5% for adding a torch and light system to the game. (See Darkest Dungeon.)

## +2.5% for adding diseases to the game. (See Darkest Dungeon.)

## +2.5% for adding cursed items to the game. (Items that cannot be removed, and may have a mix of negative and positive modifiers.)

## +2.5% for adding a stamina system (heavy, quick, etc attacks that use different amounts of stamina)

## +2.5% for adding a range system (player and enemy can be 1-4 steps apart and weapons can only attack within certain range bands)

# 

# 

# FAQ

## Q: How much time do I need to build this?

A: Plan to work on it for 2+ weeks (at 2-4 hours per day). Don’t procrastinate. You will need to do a lot of this work outside of class.

## Q: If I make a console version can I still get max points?

A: Absolutely. Look through the bonus points section for ideas.

## Q: Can I get more than 125% on the midterm?

A: No.

## Q: Can the player change their equipped gear in the middle of combat?

A: No.

## Q: Can I make the combat time based instead of turn based?

A: There are are a few pitfalls with a time based system that are beyond what we will cover in this class. There are also some tools that you will need which will not be covering in class. ***So I would not recommend it.***

## Q: Can I add a dungeon map to the game?

A: Not for the midterm. The problems, algorithms, and skills that go into making a map that the player can walk around are not a good demonstration of the competencies that we want you to demonstrate. Also, building a fully functional dungeon map could easily take 1-3 months, there is just a lot to it. ***One of the big non-obvious hurdles with making a map is that you have to get into AI and pathfinding.***

## Q: Can I add feature X?

A: Maybe. Be wary of [“feature creep.”](https://en.wikipedia.org/wiki/Scope_creep) This assignment will take a fair amount of work just to get 100% without adding extra bells and whistles.  ***I would recommend building the game to where you will be at least getting an 80% on the midterm before adding bells and whistles.***

If the feature detracts from the skills/competencies that we want to see from you, then definitely skip those features. (Time based combat, maps, shops, crafting, and mini games all fall under this categorization.) If feature X isn’t on the list and you want bonus points for feature X, please talk to Paul Smith before you start building it.

## Q: Can I keep working on this in my own time and possibly release it to the app store?

A: Yep. If you can get this playable on a phone or simply as a Youtube video it might even be a good thing to show in an interview.

## Q: If I want to use Unity, what are the key things I need to know?

A: Most of the classes in the UML diagram will be scripts that are subclasses of MonoBehaviour.

Also, there are two user interface systems in Unity. There is the old GUI way of doing it (don’t use this) and the newer UI way that was introduced in Unity 4.3 with all of the 2D functionality (use this one).

<https://www.youtube.com/watch?v=4qE8cuHI93c>

<https://unity3d.com/learn/tutorials/topics/user-interface-ui>

## Q: Can I add more fields, properties and methods to the GameManager class?

A: Yes, in fact you will probably have to. You can even change the signatures or drop methods on the GameManager class as you implement it. What the GameManager ends up looking like is highly depend on how you decide to solve the problem.

***However, we do expect fairly strict adherence to the spec for the Item, StoredItems, EquippedItems, and Character classes as defined in the UML diagram.***

## Q: What is the Factory pattern?

A: A factory is just a class that helps facilitate polymorphism by constructing object instances for use by other parts of the system.

<https://en.wikipedia.org/wiki/Factory_method_pattern>

## Q: Can I use the same controls for both the Main Menu and Game Over screens?

A: Yes. But if you want to add artwork or music to either I would suggest keeping them separate.

## 

## Q: Can I use somebody else's artwork/sound?

A: We aren’t grading on the quality of your artwork or sound effects. We just want to see that you played around with the tools and have at least a basic understanding of how to use them. ***So please create your own.***

## Q: What tools would you recommend for creating graphics?

A: Photoshop, Illustrator, Gimp, Krita. ***I would recommend learning and using something other than MS Paint.***

## Q: What tools would you recommend for creating sound effects?

A: Record sound effects from your own voice and/or miscellaneous kitchen utensils with Audacity.

## See something that you think should be clarified in an FAQ? Please let us know.