

AP Computer Science Final Project - README Template

Instructions:

The first step in creating an excellent APCS final project is to write up a README. At this stage, this README file acts as your **project proposal**. Once you've filled in all components, Shelby will read through it and suggest edits. Ultimately, you need a document that adequately describes your project idea and **we must agree on this plan**.

Have one member of your group **make a copy of this Google Doc**. Then, they should share it with all other members **and with Mr. Shelby** so that every group member has edit permissions, and Shelby can add comments on your ideas.

There's a lot of parts of this document that you might not have full answers for yet. Because you haven't written the program yet, it's difficult to think about the **instructions** or **which group members will do which parts**. Even though this is hard to think about, you must have something in these sections that acts as your current plan. However, during the course of the project, you'll **continuously update this document**. This means that you will not be *held* to exactly what you put here - components of this document can change (and it's pretty common!).

There is one exception: the **Features List** section. Once Shelby OKs your README, the Features List section **cannot be modified**. For this reason, it is most important that you get a solid idea of what you want to make and the primary features it will have *now*.

Talk with your group. Consider drawing some pictures of what you think your project might look like. Be precise. When you're ready, fill this out together. Each component in brackets below ([these things]) should be replaced with your ideas. Note that there are several sample READMEs posted on this assignment for you to use as guidance.

-----When README is finalized, remove everything above this line-----

[Reaper's Scythe]

Authors: Ashwin Senthilvasan & Ido Haiby

Revision: 4/28/22

Introduction:

The program is a free world role-playing game with arcade fighter game aspects such as multiple characters and movesets/weapons. It will be a 2D game as you explore the world and you can go up to characters to initiate fights in order to progress. Once you interact with an enemy character you will be sent to a different themed window that models an arcade fighter game. There will also be special enemies or bosses that must be defeated to complete the game. There will be at

least 1 biome. There will also be obstacles in the world such as trees, cliffs, and mountains. This game combines a typical role-playing game with an arcade fighting style that differs from normal role-playing games and gives it a unique edge.

Story: In a vain attempt to gain a soul, a soulless commits an unthinkable act, taking the Reaper's Scythe and attempting to take another soul for himself. With the Reaper's Scythe now missing, death has disappeared and the world's natural order is flipped. The once strong become the weak and the weak become strong. The unnamed soulless goes into a 100-year sleep, unable to contain the power of the Reaper's Scythe. He wakes up to a world, a hundred years later, without order and in his quest to regain power and return the scythe, he must progress through the realm.

Instructions:

Game firsts launches up to a main menu with a settings option where you can adjust basic options such as volume and brightness and the option to quit/close game.

During Exploration:

- Can move in 4 directions up, down, left, right.
- Left-click - interact with characters (npcs and enemies) and options buttons.
- Escape key - opens ups main menu screen.

During Battle:

- Only able to move forward and backwards
- Limited to what weapons were equipped before the battle
- Left Click - attack with whatever weapon equipped
- Space - jump

Features List (THE ONLY SECTION THAT CANNOT CHANGE LATER):

Must-have Features:

- At least 1 biome that includes different enemies, items, and bosses. A 2D open-world screen where you move your character through different areas and can interact with other characters. By interacting you can prompt the battle screen or you can interact with NPCs that help you progress. The world will also include obstacles you have to find your way around - mountains, cliffs, trees, walls, etc.
- At least 1 boss enemy that has unique fighting styles and movesets compared to normal enemies. These will be significantly harder to defeat than a normal enemy and should pose a challenge to the player. You should only be able to complete the game after defeating each of the bosses. There should be a certain order in which you defeat them in order to progress to different areas.
- A system to level up stats. There would only be three possible stats to upgrade. These would be strength, endurance and health. Every level would add a constant amount of health, endurance or strength.
- Characters in the open world that you can interact with. These can be enemies you fight to gain items. NPCs you talk to learn about the story and progress.

- A fully working battle screen that includes all controls for each move. Also should be easy to use for beginner players. The battle screen would have a health bar and stamina bar for yourself and a health bar for the enemy.

Want-to-have Features:

- A way to save the game at different checkpoints. Players would be able to defeat certain enemies or unlock certain rooms and then reach a checkpoint where their progress would be saved.
- Multiphase boss battle. You would have to defeat the bosses first phase to fight its second phase. You would keep the same health you had in the first phase and you would have to restart both phases if you die.
- Weakness - some bosses will have inherent weaknesses (fire enemies may be weak to water damage, etc.) - players will be able to tell a difference in damage done to the boss if a weakness exists but it will not be openly shown
- Status effects such as poisoning, frostbite, and confusion. Poisoning will be a slow-tick status effect that does small amounts of damage over time. Frostbite, when proc'd, will slow down the enemy significantly. Confusion makes an enemy behave randomly (random movements, attacks, etc.) There will be signs on the UI that show if bosses are affected by the status effects as only bosses can be affected by status effects.
- Multiple stats in order to increase possible builds (different types of characters such as a mage upgrade intelligence and mind more, or melee - upgrade strength and endurance more). Weapons have different stats such as attack power and attack speed.

Stretch Features:

- Networking/in-person multiplayer. PvP duels or co-op gameplay to play with other players over different computers and cooperate or duel each other.
- An autosave feature, meaning you are able to quit the game wherever you want and be able to return to the exact same spot. You would also have multiple characters/saves so that you can go back and replay the game in different ways. You would not be able to save during fights and would be reset to your last open-world position.
- 3D graphics during the open-world experience. This would enhance the feeling as you play the game and the look of the game. It would also add different interactions between the world and the characters.
- Special animation for different weapons, characters, etc. These would be different for each major type of weapon (sword, bow, staff) and would be different for each major type of enemy or boss.
- 2 additional biomes that include different npcs, enemies, items, and bosses. Both biomes will have items and characters that correspond to the type of biome.

Class List:

[This section lists the Java classes that make up the program and very briefly describes what each represents. It's totally fine to put this section in list format and not to use full sentences.]

- Biome
 - A super class that would provide the framework for building specialized biomes.

- It will represent a general biome that has obstacles and decorations.
- Will include the ability to load the biome as the player moves.
- Sprite
 - A super class that would provide the framework for making specialized npcs.
 - It will represent a general character that could be built upon to make enemies, bosses, and other npcs
 - Will include general animations and properties.
- Player
 - A sprite that the user controls.
- Enemy
 - A sprite that is an enemy you would fight in the battle screen
- NPC
 - A sprite that is an NPC you can talk to.
- Weapons
 - A superclass that would provide the framework for creating specialized weapons.
 - It will represent a general weapon that could be built upon to make specific weapons such as swords and bows.
 - Will include general animations and properties.
- Screen
 - Superclass for general screen that would be used to create the game screen and menu screen.
- FirstScreen
 - Subclass of screen. The main menu screen of the game.
- SecondScreen
 - Subclass of screen. The open world game screen.
- ScreenSwitcher
 - An interface that defines switching between screens.
- BattleScreen
 - Subclass of screen. The battle screen.
- MapScreen
 - Subclass of screen. Displays the map.
- Main
 - Would run the game using a drawing surface.
- DrawingSurface
 - Displays the game and the graphics components of the game.

Credits:

[Gives credit for project components. This includes both internal credit (your group members) and external credit (other people, websites, libraries). To do this:

- Processing
- Mr. Shelby - Game physics demo
- Finalbossblues - player images
- Andy Sowards - Game background
- Hyper Light Drifter - Main menu background

- Ido - animations, NPC, second screen, sprite, drawingsurface
- Ashwin - maps, enemy, battlescreen, player
- Craftpix - npc image