

AP Computer Science Final Project - README Template

Downfall

Authors: Kevin Ren, Aviral Vaidya

Revision: May 6, 2022

Introduction:

[In a few paragraphs totaling about ½ page, introduce the high-level concept of your program. What this looks like depends a lot on what type of thing you are making. An introduction for an *application* will look different than one for a *game*. In general, your introduction should address questions like these:

What does your program do?

What problem does it solve? Why did you write it?

What is the story?

What are the rules? What is the goal?

Who would want to use your program?

What are the primary features of your program?]

Downfall is a 2-dimensional platformer in which you move your character side to side as you fall downwards through the sky.

The objective of the game is to survive as long as possible by landing on platforms to avoid dying and shooting to destroy enemies which move toward you. The platforms are both flat and angled, and hitting a platform causes you to bounce off along a path normal to the platform (perfectly elastic collision).

In addition to this, you may also compete against a single other player in a multiplayer game mode, with customizable usernames for each player, where the two players compete with the goal of surviving for a longer duration of time than the other player. The player who falls down through the sky first loses, and the other player is declared the winner, bringing up the end game menu. Falling down through the sky with the aforementioned rules.

Instructions:

[Explain how to use the program. This needs to be **specific**:

Which keyboard keys will do what?

Where will you need to click?

Will you have menus that need to be navigated? What will they look like?

Do actions need to be taken in a certain order?]

The game will implement a standard menu interface for level starting or quitting, with buttons activated by mouse clicks. Upon launch, there will be a start menu with a start level button and quit button.

In game, you will use the left and right arrow keys to move the character left and right, respectively. You can also use A and D. Upon dying, you will be directed back to the menu described above with a death message.

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

Must-have Features:

[These are features that we agree you will *definitely* have by the project due date. A good final project would have all of these completed. At least 5 are required. Each feature should be fully described (at least a few full sentences for each)]

- Main menu at the beginning
- Platforms that move from side to side
- Enemies which loosely follow your path and kill you upon collision
- Shoot enemies with projectiles
- Angular platforms (when you hit it you bounce off at an angle)
- Screen wrap around (when you go off the edge of the screen you appear on the other side)
- Menu when you die

Want-to-have Features:

[These are features that you would like to have by the project due date, but you're unsure whether you'll hit all of them. A good final project would have perhaps half of these completed. At least 5 are required. Again, fully describe each.]

- Enemies which loosely follow your path and shoot projectiles at you
- Multiplayer (play against an opponent to see who lasts for a longer time)
- Platforms that disappear after one use
- Collectible power-ups which enhance your attacks (more projectiles, faster fire rate, etc)
- Collectible defenses which give you temporary invincibility

Stretch Features:

[These are features that we agree a *fully complete version of this program would have, but that you probably will not have time to implement*. A good final project does not necessarily need to have any of these completed at all. At least 3 are required. Again, fully describe each.]

- Varying wind (x acceleration) with animation
- Parallax effect with background
- Chat in multiplayer mode

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.]

Player.java

Represents your character which bounces on the platforms

Enemy.java

Represents the enemies

Platform.java

Represents the platforms which you land on (might have a 2nd constructor for angular as well)

Projectile.java

The projectiles launched which kill whatever it touches

Powerup.java

Power-ups that can be collected by colliding with them, they give different abilities as described above

Screen.java

The specific screen which the game is drawn on, including menu and game

DrawingSurface

This class handles the key and mouse presses that the user uses to interact with the game environment, as well as change the screen being displayed (like when you navigate through the main menu or die)

Main.java

The class with the main method. Running this class launches the program.

Game.java

Game screen

Menu.java

Menu screen seen when a player launches the game

ScreenSwitcher.java

Interface to help change screens

Sprite.java

Superclass for all objects on the screen (enemy, player, etc.)

Credits:

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

Alpha checkpoint:

Aviral:

- Added needed jar files to project (shapes and processing)
- Set up skeleton class for Sprite
- Debugged player.java
- Updated uml diagram

Kevin:

- Created skeletons for subclasses of sprite
- Set up drawingsurface.java
- Screenswitcher interface

External Libraries:

Processing java

Aviral's Shapes library

- List the group members and describe how each member contributed to the completion of the final program. This could be classes written, art assets created, leadership/organizational skills exercises, or other tasks. Initially, this is *how you plan on splitting the work*.
- Give credit to all outside resources used. This includes downloaded images or sounds, external java libraries, parent/tutor/student coding help, etc.]