The Idea

We will make a 2D platformer game, like Mario or Hollow Knight. The aim will be to reach the end of the level without dying. Along the way the player will face enemies that will try to stop the player, as well as parkour to reach the next stage.

The game will be for PC only and written in Java.

Interaction with the player

The player will be able to interact with the game with 2 main methods:

* Playing the game, this involves the movement controls, the game window and the GUI on screen.
* Through menus, these involve the start, pause and options menu.

Having a simple interface means that it will be easy for any player to be able to navigate the menus game knowing exactly how to edit a specific setting or getting back to playing the game.

Start menu concept:

A picture containing diagram

Description automatically generated

The play button leads to a menu selector where you select which level to play (levels unlock after finishing the previous level - the first level is a tutorial level that will teach the player the basic controls and, some of the enemies and the basic game concepts). The Options menu will have settings like the video and audio settings.

System Architecture:

The game will have 7 main classes:

* Rigidbody: includes functions for collisions as well as data such as the position and rotation.
* DynamicRigidbody (extends Rigidbody): contains functions for movement.
* Platform (extends Rigidbody): contains rendering of platforms.
* MovingPlatform (extends DynamicRigidbody): contains rendering, movement script
* Entity (Extends DynamicRigidbody): contains information about health, etc
* Player (extends Entity): includes movement controls as well as attacking.
* Enemy (extends Entity): includes enemy damage, Enemy movement AI.

Most other classes with be small variations of these classes (e.g. enemy or platform types). The gameplay will be made up of 2 main loops, the main “mainloop” and the secondary “gameloop”. The mainloop will be render the main menu, level selector as well as take inputs etc. Once a level is opened, the gameloop is called, this is a loop that deals with everything in the level, from rendering, movement, pause menu etc.

This is a basic flow chart outlining the barebones of the game loop and handeEvents subprocess. The gameloop will be run when playing a level. The mainloop will call this when a level is clicked on in the menu selector.

Diagram

Description automatically generated

Requirements

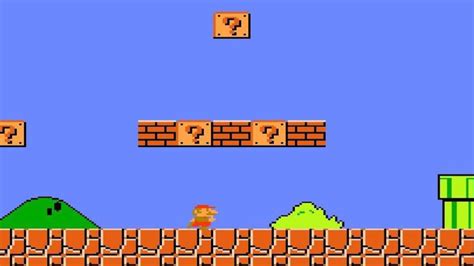
System Requirements:

* Java installed on any operating system (Swing is platform independent)
* Keyboard
* Mouse

User requirements:

* The menus must be easy to navigate.
* The controls must be simple enough that anyone can use them.
* Must have enough gameplay to keep the user entertained.
* Should be able to have the window full screen or windowed.

Other platformers:



Super Mario Bros, one of the most iconic platformer games ever. The player traverses the levels with only movement controls, killing enemies by jumping on their heads. This game will have a large influence on what our project will be like as it is a successful example of what this kind of game will feel like.