WILL HERO

By: Shivam Jindal (2020125) and Aayush Gakhar (2020006)

WILL HERO

Design and Implementation

1. The Game starts at the Main menu. We added several buttons on the main menu for starting a new game, loading a previously saved game, controlling sound and music and exit.
2. The saved games menu show the 5 previously saved games. Saving a game store the hero's current position, helmet with weapons unlocked and the coins collected till that point.
3. For implementing the game, we used various OOPS concepts extensively such as, inheritance, interfaces, abstract classes, exception handling, serialization and deserialization for saving and loading a game etc.
4. We used Java FXML for all the screens. There are various .fxml files for different screens, such as main screen, pause screen, screen for loading a game, win game screen etc.
5. Basic concepts of physics are used for jumping of hero and orcs and collision between game objects. The GameObject class inherits from imageview class.
6. Various design patterns such as factory, iterator, singleton are used for implementing the game.
7. We used animationTimer for the animations during gameplay and timelines for the animations on main menu. The collisions were detected by checking for intersection in different objects in each iteration of the timer loop.
8. The animations were done by adding the speed of each object to its position in each iteration.

WILL HERO

Individual Efforts

Aayush Gakhar (2020006)

1. Created all the FXML pages and GUI designs for all screens
2. Handled the collision between game objects
3. Created weapons and implemented them
4. Implemented Serializing and Deserializing to save and load game.
5. Added the buttons and their functionality
6. Implemented game physics for all game objects for realistic movement.

Shivam Jindal (2020125)

1. Created all the islands and placed them in the game via scenebuilder
2. Added two types of orcs: Red Orcs and Green Orcs and their position in the game
3. Jumping of hero and orc at their position
4. Boss implementation
5. Added Coin chests and Weapon chests and placed TNT.
6. Added music and different sounds on different things happening

Bonus Implementation

1. Added music in the game, making it more enjoyable while playing it.
2. Utilized concepts of linear algebra and basic physics concepts such as momentum, projectile motion for ultra smooth gameplay and soothing user experience.
3. Added the speed feature in the game, that is game progressively becomes faster, making it more challenging to win.
4. Added sounds for user interface and for clicking any button.
5. Added an instruction screen to help users know and understand the gameplay.
6. Implemented adaptable MediaPlayer with different music for different screens.