

CSCU9N6 Assignment Report

By: 3141979

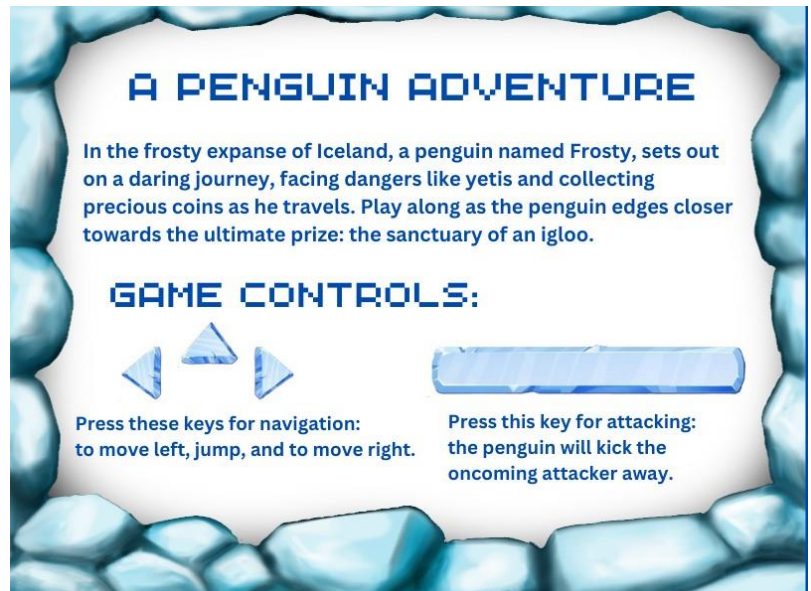
INTRODUCTION

The game, “A penguin adventure” has been implemented according to the requirements of the assignment. This game is 2D platform-based and features a character navigating through two levels, collecting coins and defeating enemies.

KEY FEATURES:

This game uses the GameCore library to implement the required elements of the game. The game follows a penguin [player character] as it passes through different levels, with actions such as, walking, jumping and attacking enemies.

The image attached, has a description of the storyline of the game and the necessary controls to be used for various movements of the player.



Animations and Sprites

The player has five different animations to showcase different actions, such as for the idle state, moving towards left, moving towards right, jumping, and also for attacking [the player kicks the enemy away]. The penguin is the player-controlled sprite in this game.

The enemy sprite is a yeti, which has an animation for moving towards the player. There are two enemy sprites that interact with and pursue the player. There are also 3 cloud sprites that move around in the game environment.

Collision Detection

There are three methods to handle collisions in this game, `checkTileCollision`, `handlingTileCollision`, and `attackingCollision`.

- `attackingCollision`, is the `boundingboxcollision` method that has been modified and checks for a collision between the two sprites, with the given distance in between them. It checks if the bounding boxes intersect and returns true if there is a collision and false, if otherwise.

- `checkTileCollision`, this method handles the collisions between the sprites and tile map. It clears any detected collision tiles from the `collidedTiles` list. It then checks if the two rectangle objects (one for tile and other for the sprite) intersect. If a collision is detected, then they are handled accordingly such as by playing sounds for the coin tile characters and removing them after, incrementing levels for “x” tile, and calling the `handlingTileCollision` method.
- `handlingTileCollision`, this method handles the collision between the sprites, and the tiles. The collision is solved by adjusting the sprites position and velocity based on the tiles bounding box.

Sounds

- In this game, three sounds are used along with a midi track for background theme music. There is a sound when coins are being collected, when the player attacks the enemy, and a sound for when a level has been successfully completed.
- A sound filter has been used when playing the attack sound to fade it so that sound isn't too harsh on the ears and doesn't disrupt the game flow.
- The MIDI track is played with the help of `backgroundMusic` method. The method loads a midi sequence, after which a sequencer is initialised and then music is set to loop so that it plays throughout the game in both levels.

Game World (tile maps/interactions/parallax)

- This game consists of two levels, level one is centered on the player defeating yetis and collection of few coins afterwards, level two focuses on collecting coins and navigating through the platforms by jumping, and other actions.
- Parallax scrolling: I have implemented parallax scrolling, using four different layers to ensure a visually appealing game background. When creating the game world, each tile and background layer was carefully selected to capture the snowy landscape well.
- There are mainly 4 keyboard events with different player actions such as moving left, right, jumping and attacking.

CONCLUSION

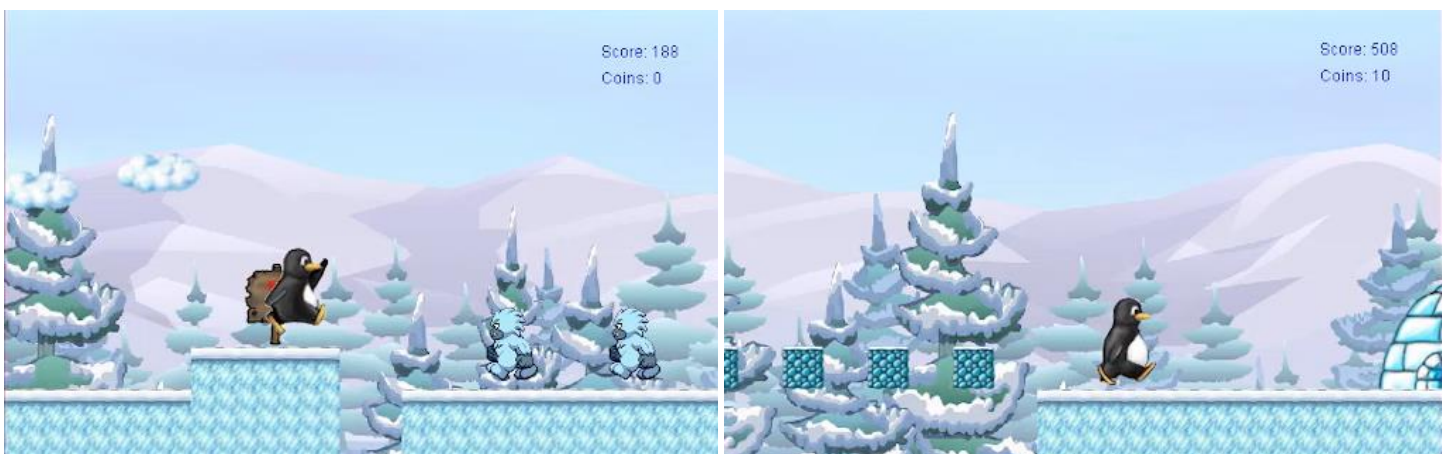
The 2D platform game developed using the GameCore library incorporates the required key features to create an engaging experience for the player.

Multiple animations and sprites for the player, enemies, and other elements contribute to the game's environment. Collision detection has been implemented to handle interactions between the player, enemies, and the tile maps.

Sound effects and background music have been added to enhance the overall gaming experience. The implementation of parallax scrolling with use of multiple layers also helps to create the visually captivating game world.

To improve the game further, more computer-controlled sprites could be added, along with more sound effects to make the game engaging. More levels could also be added with new player characters too.

Level 1:



Level 2:



Resources used:

<https://www.sprisers-resource.com/> [Sprite and tiles]

<https://pixabay.com/sound-effects/> [Sounds]

<https://craftpix.net/> [Background]