COMP30540 Game Development

Assignment 1 Report

Weronika Wolska – 17301623

Game: Bird Tower

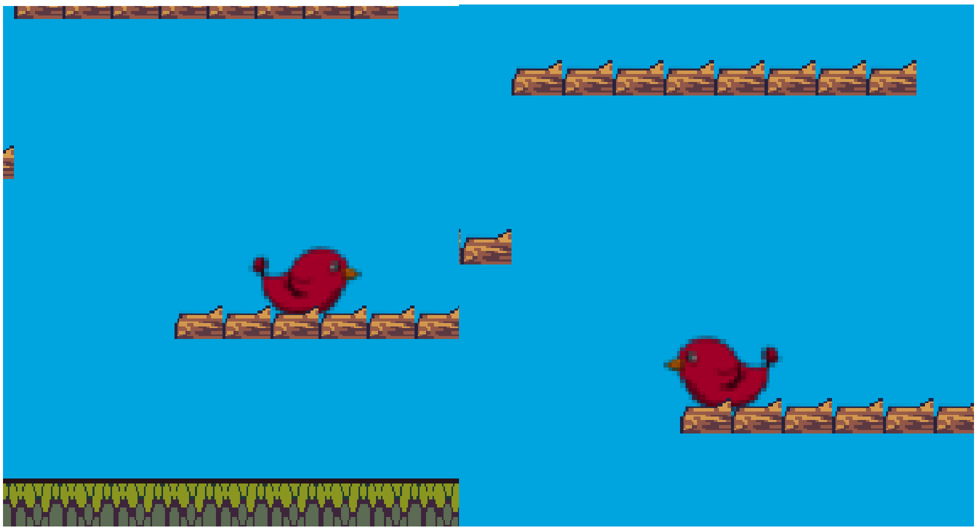
Theme: Platformer

Objective:

The objective of the game is for the player to reach the top platform without falling down. The player starts at the bottom of the screen, indicating that they need to move up the platforms on the screen. Once the player reaches the centre, the branches start moving down and more branches are produced. The aim is to not fall to the bottom of the screen before reaching the 200th platform. Since the platforms are randomly generated, each new game played acts as a different level.

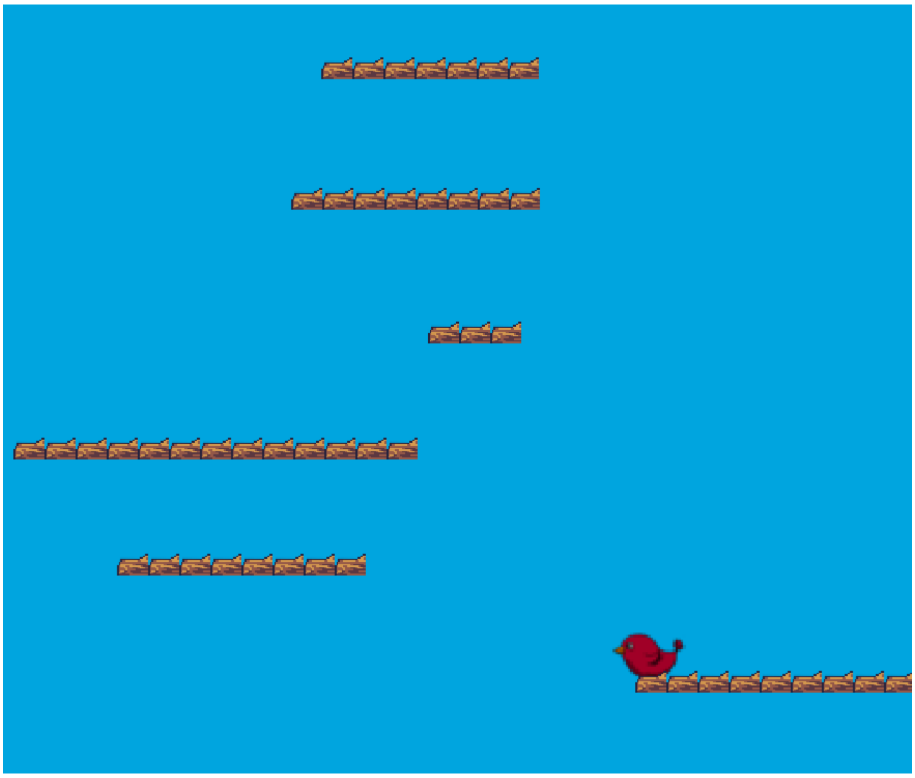
Features:

- The player object has the ability to move to the right (‘d’ key) and left (‘a’ key), as well as being able to jump onto the various platforms by pressing space.
- Appropriate animation is applied to the player to change the direction the bird is facing when a and d keys are pressed.



- The jump physics is implemented by applying a vector to player centre with the value of y as a jump value. The jump value is multiplied by a constant (0.9) to make sure that the jumping slows down over time and, once the jump value reaches >0.01, the space key pressed value changes to false and gravity is applied.
- Once the player reaches the centre of the screen, the jumping is implemented through the platforms moving downwards rather than the player moving upwards, to create the illusion of verticality within the game, however the logic behind the jumping remains the same.
- When the player is located on the bottom half of the screen, both of the jumping values are included in the jumping physics. This works similar to a double jump and is to make sure that the player is not stuck at the bottom of the screen for the remainder of the game once they fall onto a lower platform, and allow them to get back into a winning position.
- The above can also be used strategically to get back into advantageous positions. This is especially helpful since the game was designed to implement a mouse listener* that works

as a timer-based vertical movement, so some platforms are out of reach of the player with the standard jump.



This adds more challenge to the game, as you have to time the jumps accordingly, so that the player object does not reach the bottom of the screen, losing the game.

*The MouseController was implemented, registers the clicks in the game window, and is mapped to functionality within the Model class. However, the information that a mouse click was made is not registering in the Model class.

- The first 9 platforms are statically generated when a model is first created (and when the game is restarted). The rest of the platforms are randomly generated as the game progresses. A new platform is added if there are less than 6 platforms visible on the screen (unless the game is reaching the end).
- The platforms are removed from the list as they reach the bottom of the screen.
- For each platform, the game checks if the player object is within its hit box. If the player lands on a platform, the y coordinate of the player is fixed to that on top of the platform (as long as the jump key is not pressed) and the gravity is set to false so that the player does not fall through the branch.
- If the player collides with a platform that's above the one that is last visited, the score is increased. The score variable determines when the platforms stop being generated, i.e. when the game is reaching its end.

ART CREDITS:

- Start menu screen and player sprite: Eoin Abbey-Maher
- Branch sprite: Jesten (twitter @jestenpixels)
- Ground tiles: Craftpix (https://craftpix.net/freebies/free-swamp-game-tileset-pixel-art/)