

## Games development report - ID 2831609 | Dino-Jump

I have created a game called Dino-jump that features a cute tyrannosaurus-rex called Dino, who travels the desert inspired environment looking to grow his coin collection. The goal of the game is to collect all the coins per level. When all coins are collected, in a fun ironic twist an asteroid strikes the player causing an explosion which updates the environment to a new level. Interaction with enemy sprites is optional however they reward substantial points, the user can choose to move the player away from danger. Enemy sprites can be eliminated by accurately jumping on the centre of their heads, however if you are jumping from Infront of them there is a high chance their vigilance betters Dino. An enemy hit instantly finishes Dinos. The user should gain higher ground and jump onto the centre of the enemy's head to defeat them, attempting to jump from low ground in either direction will not work.

### Controls

Game controls		Debug controls for testing	
<b>A, arrow left</b>	Moves the player sprite left.	<b>1</b>	Turns on debug mode.
<b>D, arrow right</b>	Moves the player sprite right.	<b>2</b>	Teleports player to end of map.
<b>Space, arrow up</b>	Makes the user jump.	<b>3</b>	Kills player to restart game.
<b>Left click</b>	Used to control the start menu. Used to restart the game when player dies.	<b>4</b>	Spawns an asteroid to change level.
<b>V</b>	Toggle audio (mute/unmute).	<b>Left click</b>	Sends enemies towards players general direction.

### Successful features

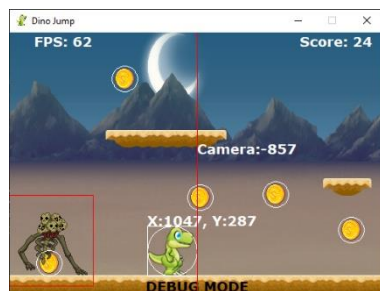
#### Requirements

<i>"Animations with suitable animation changes for player and controlled sprites"</i>	My program contains a total of 11 sprite sheet animations and an extra 4 images loaded in as an animation so I could control the parallax background as a sprite to avoid excess implementation.
<i>"One or more moving player-controlled sprites"</i>	My program contains one player-controlled sprite.
<i>"Multiple moving computer-controlled sprites that interact with the player and environment."</i>	I have multiple enemy sprites that feature multiple animations, interact with the player, and change direction when reaching a border.
<i>"Correct collision detection and handling for all sprites and tile maps."</i>	I have collision detection applied to all my character sprites.
<i>"Multiple sounds that are appropriate to in game events."</i>	I have five different audio files used in my game (1 midi track and four wav files).
<i>"Use of a novel sound filter."</i>	I have a fade filter for all sounds except the background music.
<i>"Control of background theme music that uses a MIDI track."</i>	I have a midi track that can be paused and played.

"Multiple keyboard and mouse events."	My program features many keyboard and mouse events as shown in the control section of my report.
"Interactive use of 2D tile maps with at least 2 different levels."	I have implemented two different levels.
"Parallax scrolling of the background to give an illusion of depth"	My parallax background features four images. The first two which is the sky and the moon do not move. The mountains then moves when the player reaches the middle of the screen. The desert floor moves even faster than the mountains to give an illusion of depth.

## My favourite game features

### Mario style movement



I wanted to implement a classic Mario style camera movement. The camera is fixed on the screen until the player reaches the middle and then the player is fixed running in the centre and the camera follows along. When the player reaches the end of the screen it will be released from centre lock and be able to reach the run up to the far edge of the map. The red line in the middle of the screen drawn in debug mode represents a line that the player cannot cross until near the maps end.



### Double jumping

My game will only allow for two jumps at a time. After pressing the space bar, it will be 1.5 seconds before the player regains a jump. With the gravity value the player will not be able to leave the ground. I have writing above the player that is colour coded. It will be red if player has none left.



### Asteroid strike

My game has an asteroid strike animation once all the coins are collected that kills the player, any remaining mobs and clears the enemies.



### Debug mode

By pressing 1 on the keyboard, debug mode is enabled which shows what I like to call the "Mario movement line", hitboxes of the mobs, frame rate per second, coordinates of mobs, and the camera number which is the offsetX variable used for drawing tile map and updating actual mob locations. All of this is colour coded. When a mob is defeated, its hitbox turns white as it is no danger. There are other debugging controls to compliment debug mode.

### Enemy movement

Every 10 seconds the enemies stalks the player, if enemies collide, they change directions. In a weird way I decided to not take the implementation further as I found this made their movement unpredictable compared to if the enemies were to just follow the player consistently.

Overall I feel like I exceeded when it came to game movement and aesthtic. The game is very colourful, easy and fun. I have also tested the game on my laptop and achieved a high frame rate

due to my implementation of Graphics2D clip which allows me to only render the items within camera bounds.

### Trial & error

During the development I struggled greatly figuring out collisions with the tile map and perfecting the parallax scrolling. My issues with collision were mostly down to player hitboxes. I tried many different free image sheets alas the shape of many sprite resources I found were wasteful. The sprite would be centred and surrounded by vast space resulting in inaccurate collision maths due to sprite getting its bounds from the image size. I eventually found many individual images of Dino which I cropped to remove excess white space, reduced in size to gain a 64x64 copy and then merged the images into an image sheet. I am now however very happy by the attack and spawn animations for the enemy.

I struggled with the sprite animations as sometimes they wouldn't reset. I also had strange bugs with the order my update loop was calling methods which would result in the sprites flying off the screen once the player died. At one point the dinosaur got hit in the head repeatedly by an enemy before slouching up, sliding along the screen towards the next enemy which would hit it again and make it appear passed out on the floor once more. I almost wanted to keep that as a feature as it made me laugh hysterically.

### Future implementations

In future I would like to improve the collision as truthfully it is not perfect. I would also like to make the map more creative and have it automatically generated. I think it wouldn't be too difficult as the map concept is quite simple (floating islands). That would also allow for a random and constantly expanding player experience however I estimate it would take a few weeks to perfect.

I put substantial time into this project however I feel I lacked when it came to the map designs, in future I would like to add vertical walls that block players and mobs. Perhaps have smaller mobs appearing on the floating islands as well. This would make the game more challenging.

I currently achieve a high frame rate average but in future I would like to remove the cap and make a faster game.

FPS: 62

### Summary

During the implementation of this project, I encountered many challenges and bugs. I found the time frame very challenging due to personal reasons. I started the project early and invested a substantial amount of time into the implementation of flappy bird and adding extra features. However, it wasn't till later I realised that game idea was too simplistic to gain a good mark in this assignment. If I had less confusion about the assignment, I feel like the time wasted would have made a significant contribution to the quality of Dino-Jump.

Despite the stress I felt from this project, I have gained a good understanding of how simple game engines are created and reused and I am happy with my creation. I will return to improve my game in future.

**Instructions to compile on windows command line. CD to the top layer of the zip folder and type "java -jar Dino-Jump.jar", the program will execute. I have tested this on another device.**