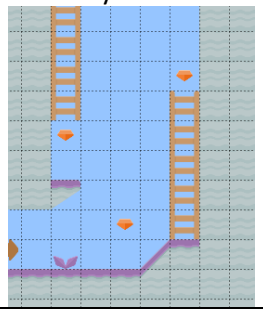
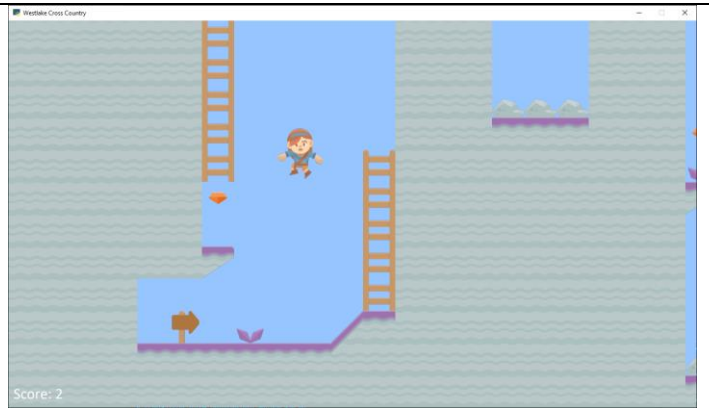
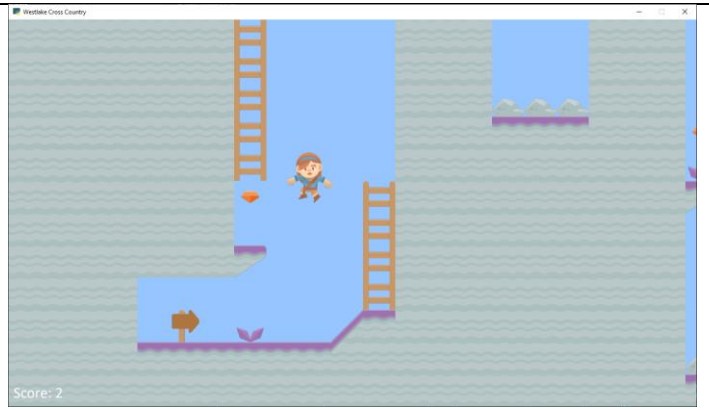
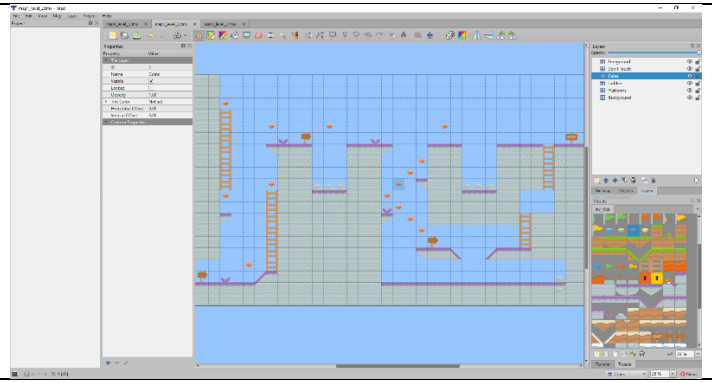
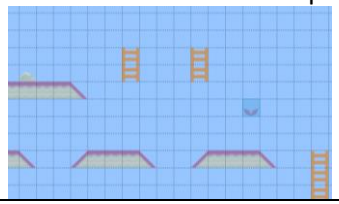
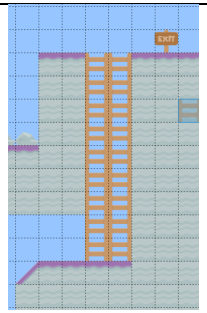
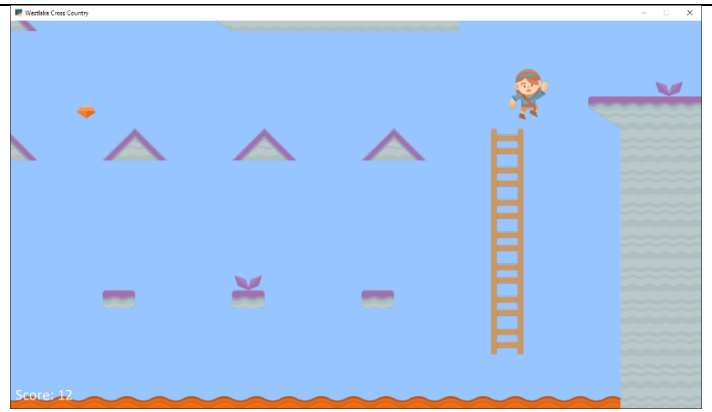
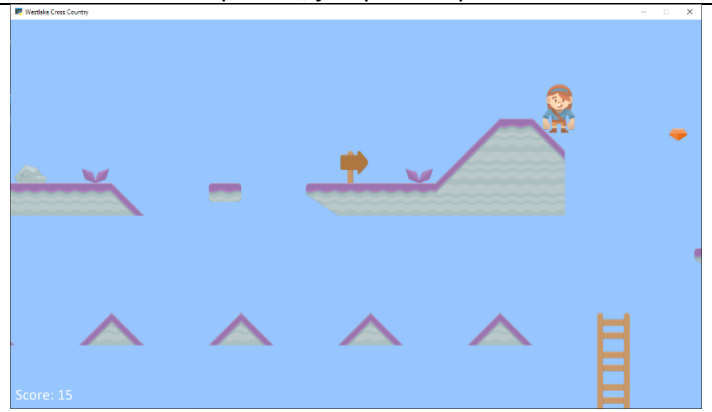
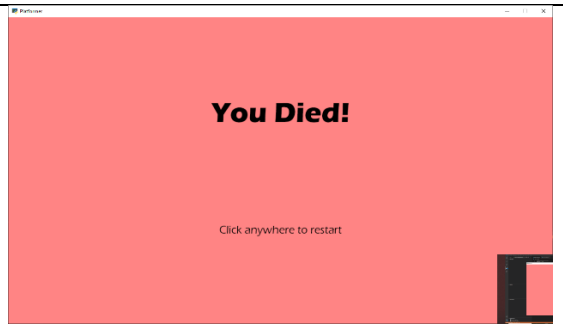
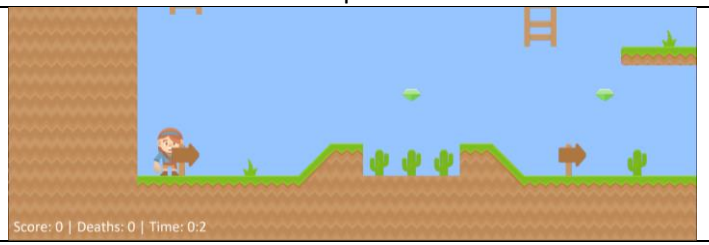
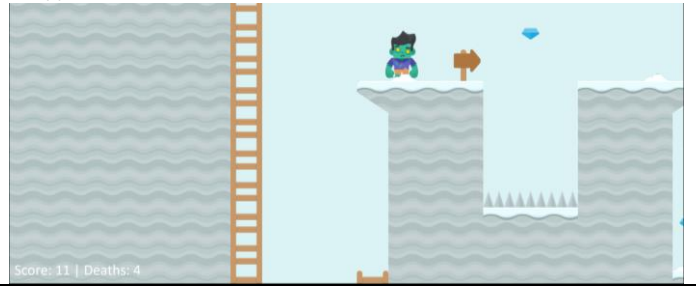
Test Log

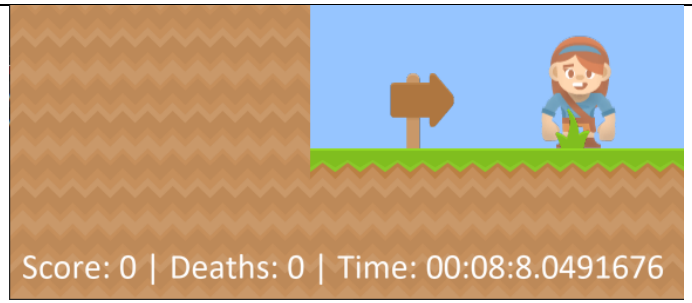
Date	What is being tested	Test type	Expected result	Actual result	Changes made
08 Sep 20	Testing that it is possible for the sprite to travel the distance required to pass the spawn ladders section on level 2, after extending the width to increase difficulty 	Boundary	The physics engine allows the sprite to travel the distance between the ladders in order to continue with the map	 Becomes either impossible or far too difficult to complete the jump	Reverted changes, brought left side platform/ladders back right one grid space
08 Sep	Testing the minimum possible ladders that can be on the right side to make the jump challenging however still possible; tested this by removing one from the initial count of 5 ladders	Boundary	Would make the jump more challenging than it was with 5 ladders; however still possible and not too challenging to the point of being destructive to the game	 The jump becomes far too difficult; almost impossible to complete	Reverted back to 5 ladders on the right side

08 Sep	Testing whether the number of padding grid spaces occupied by the 'background' platform textures was sufficient to ensure viewport did not clip outside to the sky	Boundary	Two blocks of padding on the bottom was likely to be insufficient		Indeed was insufficient; therefore shifted map up by one block to ensure viewport never clipped outside the stone textures
09 Sep	Testing whether ladder mechanics was feasible/possible to allow for this section of the map 	Valid	Able to 'surf' through the air and fly to the next ladder	As expected	Removed in favour of different platform design to increase emphasis on game mechanics; ladders were too easy
09 Sep	Testing whether character can pass through a one block gap while climbing a ladder	Valid	Able to climb through a one-block gap	 Player sprite is too wide to fit through, can't pass through a one-block gap	 Widened ladder path to two blocks width

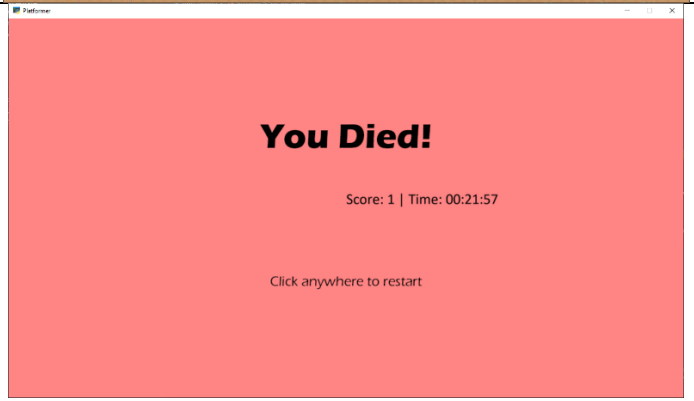
10 Sep	Testing whether you can cut straight to the end on the third map	Invalid	Player character unable to make the jump	 As expected; jump is not possible	None; performed as expected
10 Sep	Testing that the final jump is possible	Valid	Player can make the jump as long as is it performed correctly as per the game's mechanics (i.e. if standing on an 45 degree angle, you must first jump upwards before translating left or right)		Jump is possible as expected

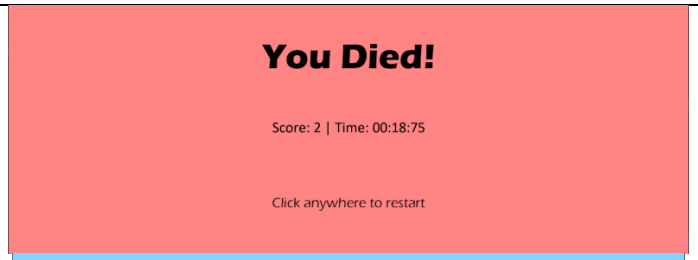
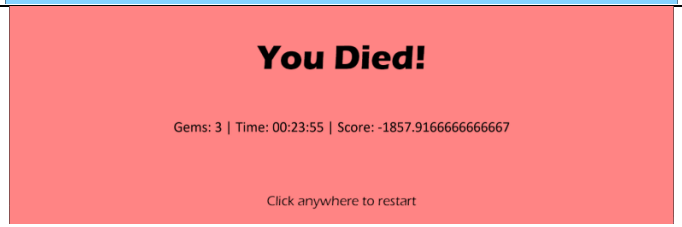
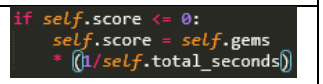
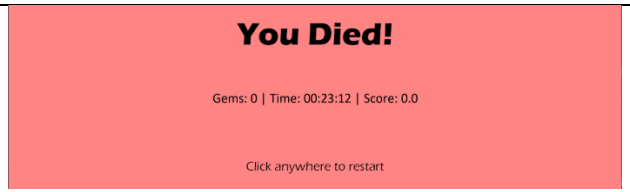
14 Sep	Testing that the game over screen displays correctly with different images; i.e. You Died and Congratulations screens	Valid	'You died!' screen displays on death; 'Congratulations' screen displays on completion of final level	 As expected	None
14 Sep	Testing the display of game time is correct	Valid	Should display the time elapsed since start of the program as minutes:seconds , each to 2 dp		`Time: {:02d}:{:02d}` Added python formatting string, which formats to 2 dp and adds leading/trailing zeros as appropriate

14 Sep	Testing whether score/deaths reset after each new level	Valid	Score and deaths do not reset after progression to a new level, only after game restarts	Score resets after each level
				// **AFTER CHANGE** – score and deaths do not reset
				 
					`def setup(self, level):`
					`  # set score to 0`
					`  self.score = 0`
					Removed self.score = 0 and self.deaths = 0 in setup() so that they do not reset each time setup() (used to progress levels) is run

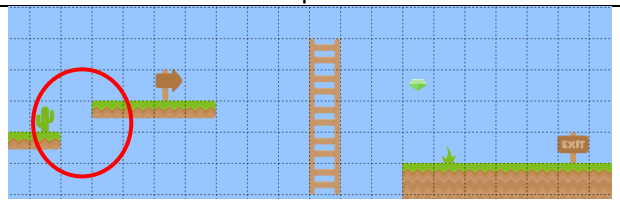
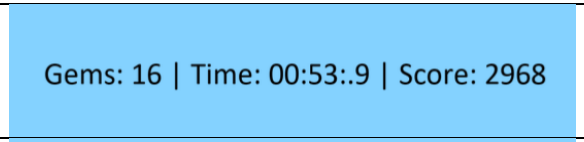
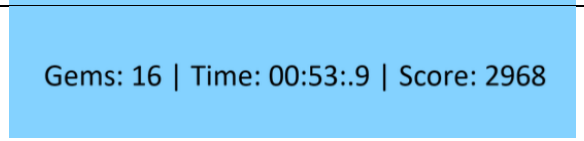
14 Sep	Testing display of milliseconds in the game text	Valid	min:sec:millis (to 2 dp) e.g. 01:42:30 In this iteration, I am expecting to just see total_time printed in place of the milliseconds	 As expected: displays minutes and seconds correctly; displays millis with seconds and to too many digits of resolution	Changed method of calculating milliseconds to *millis = float(self.total_time) – int(self.total_time)* i.e. round total_time to the next lowest integer (*not* proper rounding) and minus that from the float value of total_time, which should result in giving only the decimals
14 Sep			In this iteration, I am expecting to see the milliseconds displayed in a form such as 0.xxxxx, where the full seconds have been removed and replaced by a leading 0	 As expected	Next step is to round the value to 2 dp, displaying as 0.xx I did this by applying a round(x, 2) to the milllis value, where x represents the above calculation.

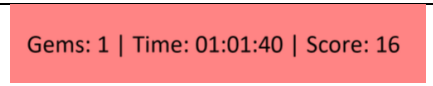
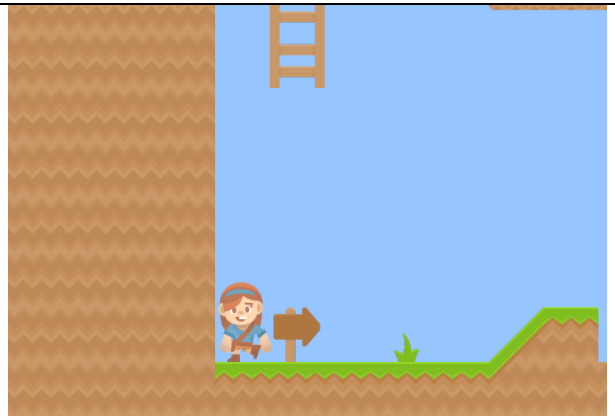
14 Sep			In this iteration, I am expecting that the millisecond value will display as 0.xx, rounded to two decimal places.		Next step is to remove the leading zero so it displays fully correctly in the form xx:yy:zz, where xx = minutes; yy = seconds; and zz = milliseconds. I did this by converting the millis float value to a str(), allowing me to use the index [-2:] to only display the last two digits of the number; i.e. cutting off the leading 0 and decimal point.
14 Sep			This should be the final iteration of testing, where the time should be displayed as expected as min:sec:millis, all to 2 dp.		None; functions as expected
15 Sep			Correct display of millis on screen	Correct display, however is far too laggy // **AFTER CHANGE** – millis is still calculated, however is not drawn	Game still calculates and stores the milliseconds elapsed for display on the end screen (for hardcore

					speedrunners), however it is not drawn on screen as it has a significant effect on performance.
15 Sep	Testing the display of game stats on the game over screens	Valid	Display of score and time on the game over screen	 Displays on the screen, however is not centred correctly	`align='center', anchor_x='center', anchor_y='center'` Added above arguments to arcade.draw_text()

15 Sep				 	None; functions as expected
18 Sep	Desmos equation to calculate score from both gem count & time elapsed functions correctly	Valid	Correct display – positive integer value	 Score displays negative	 If the output from the score equation results in a negative value[1], use a simpler, linear equation to calculate the score
18 Sep	Testing score equations when gems = 0	Valid	Score = 0 and displays correctly		Calculates correctly to 0, however needs to be displayed as an int to remove decimal point 

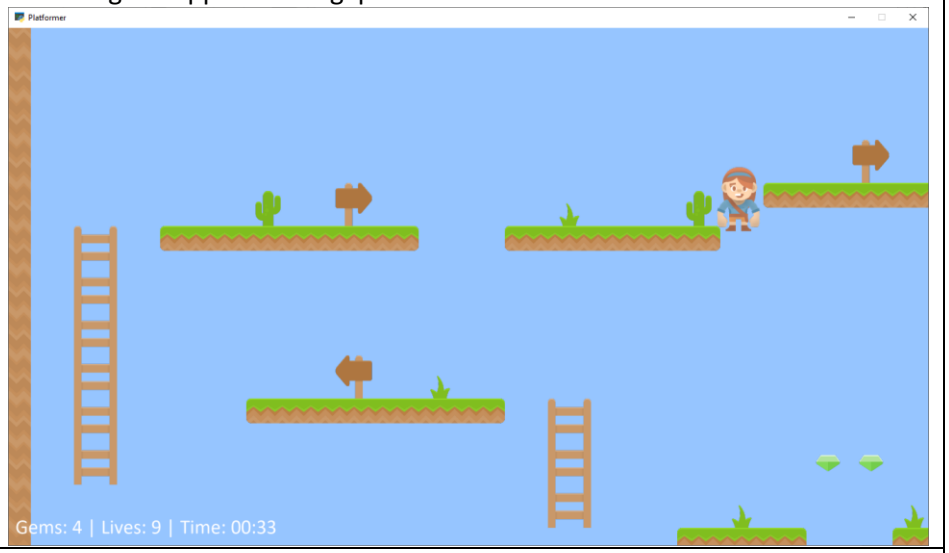
[1] As explained below, the purpose of the complex Desmos equation is to ensure that the score is fair to players regardless of whether they decide to speed-run or decide to collect as many gems as possible, ensuring players of similar calibre in their chosen style will receive similar scores. If the result from the equation is negative, it means that it is no longer relevant if the outputted score is fair to the player's style, as it is likely a player that is newer or less skilled in the game. In this situation, use a simpler equation to calculate the player's score.

18 Sep	Testing score equation where Desmos function would return a negative value however where gems != 0	Valid	Score is non-zero and is a positive integer value	 Is a positive value, however needs to be converted to int() value and given a multiplier to give a reasonable score in the tens-hundreds range	Multiplied by 1000, and converted to int() form
21 Sep	Testing score display if not following score function	Valid		 As expected	None
21 Sep	Testing whether player gets stuck when falling through the one block gap on level 1	Valid	Player falls through	 Player could get stuck on the edge of the left block, without falling through and without dying on the cactus in rare instances	Moved the block to the Obstacles layer, so even if the player manages to clip the edge of the block they will be treated as if they touched the cactus
22 Sep	Testing display when game stats suffice that Desmos equation is used	Valid	Positive, integer value		None, as expected
22 Sep	Millis display error when is a round 10s value, i.e. 10/20/30 milliseconds	Valid	Displays with a trailing zero	 Displays incorrectly with a leading decimal point	format([millis], '.2f') Use python's formatting strings to format to 2dp, ensuring a trailing zero is added if needed

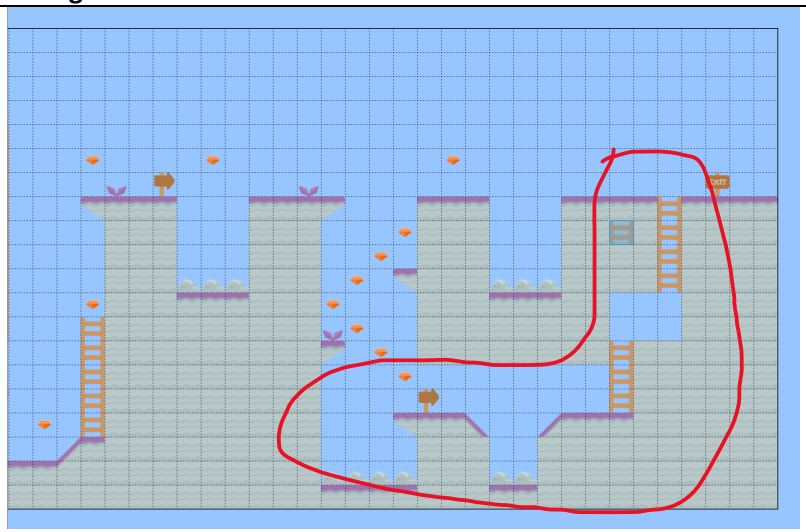
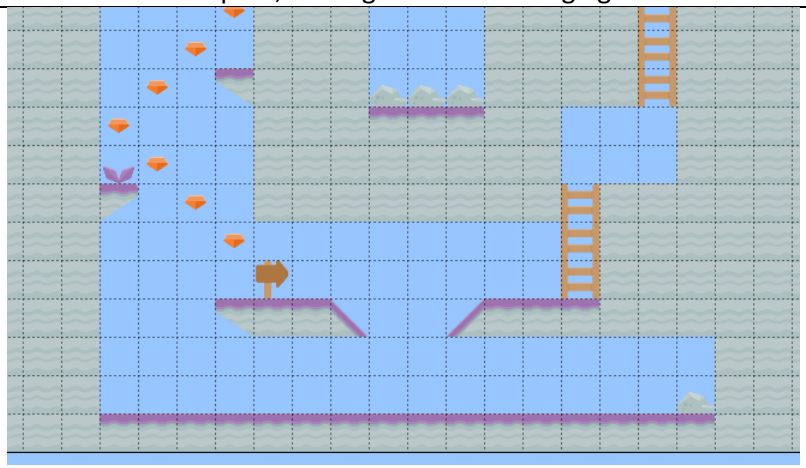
22 Sep	Millis display when is a round 10s value	Valid	Displays with a trailing zero	 As expected; displays as :40 rather than :.4	None
23 Sep	Testing keypresses	Valid & Invalid	Only valid inputs are WASD/arrow keys and mouse click	As expected, no unexpected behaviour from pressing random keys	
23 Sep	Level incrementation works as expected	Valid	On reaching the EXIT sign on each level, the character is transported to the next map, or the game ends on the final map	As expected, Level 1 > Level 2 > Level 3 > Game over	
23 Sep	Testing that the user cannot fall through the world or progress levels in invalid locations	Invalid	Walls on all maps to prevent 'noclipping', the player cannot progress levels anywhere except at the EXIT sign	 As expected, example of testing is above	

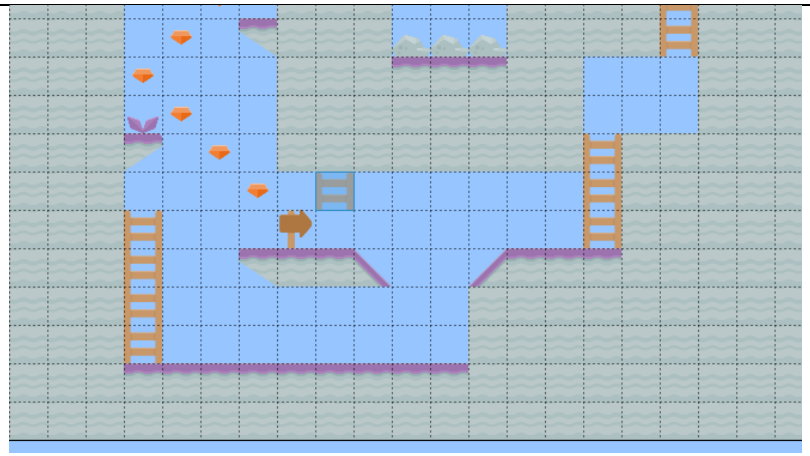
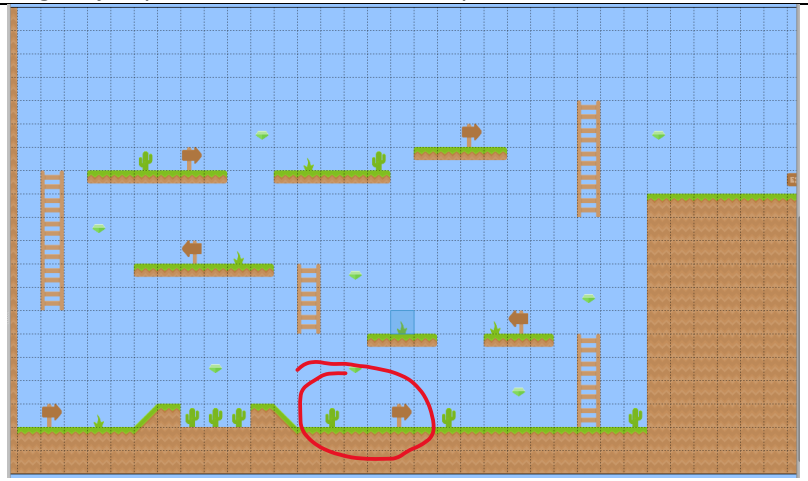
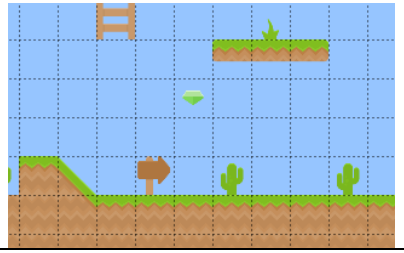
User Feedback

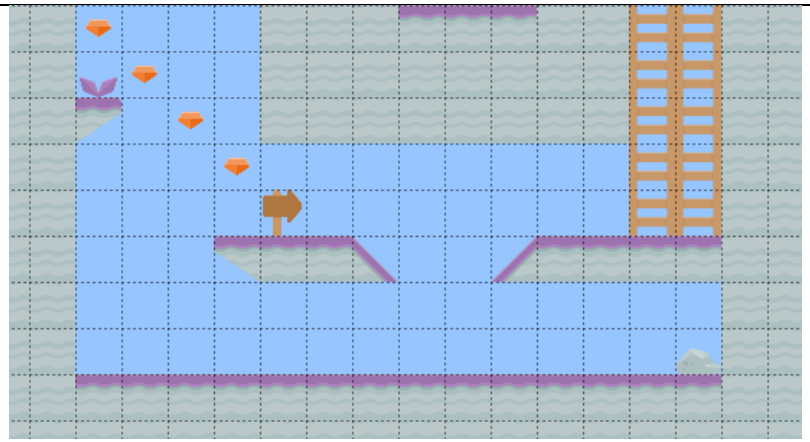
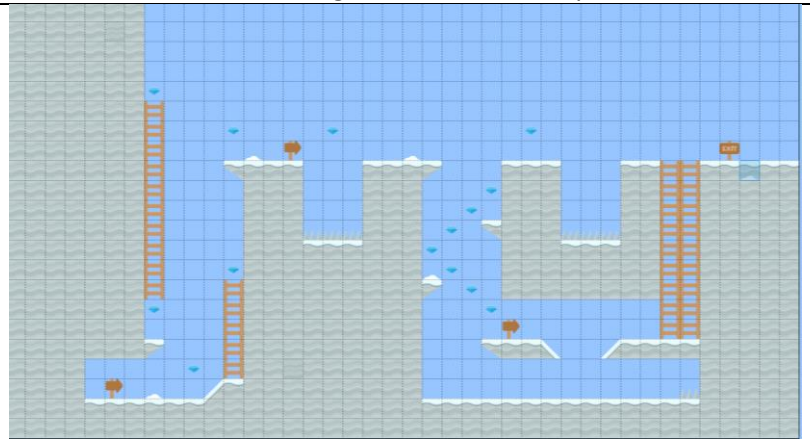
Tester	What aspects of the game do you like?	What do you think could be changed or improved?
Euan	Hidden shortcuts	Try hide the shortcut better, I found it accidentally by trying to jump over the cactus
Cole	I like how you have the option to choose between going for a fast time or collecting lots of gems	The structure of the bottom of level 2 is a little weird, it feels too easy for the amount of gems you can get
William	Changing player and map sprites between levels	I think it could be made better if the sprites followed a better progression, maybe going from summer > winter > space?
Grant	Dynamic game with a 'choose your own adventure' feel	Maybe change the death counter to a lives left counter, and make the player die at the end so they can't respawn infinitely
Sam	Variation in the map design across the levels	The game feels really laggy, it might be the timer?
Mrs Smith	Changing characters Challenging	High scores?? Multiplier of time x scores
Zinzan	The animations	The cactus doesn't stand out enough The score needs fixing
Luke	Jump physics are good and the hit boxes are not too big	The switching of obstacles through levels makes it hard to know what to avoid in the level and just a suggestion but you could add "extra" hearts people can get if the need like on a longer or hard to get too route. A restart button would also be nice.
Brad	Physics are good, momentum etc	Art could be better, obstacles blend in.

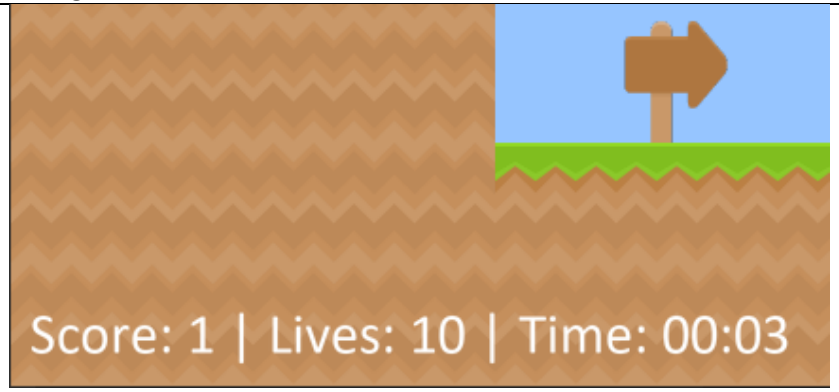
Mr Douglas	Good mix in the third level, variety in mechanics	You can get trapped in the gap 
Janith	Very nice aesthetic, really good level of difficulty	Maybe decrease gravity a tiny bit to make it a tad easier :p

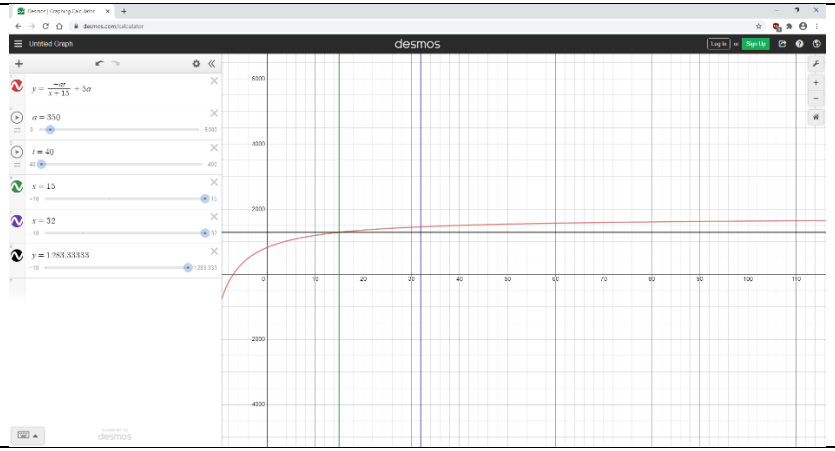
Changes from User Feedback | Development Log

Date	Changes made	Reasons for change
08 Sep	 Added to bottom path, making it more challenging	Added the 45 degree platform jump and staggered ladders in order to balance out the 7 additional gems from following the lower path, in an attempt to further increase the sacrifice required in the time taken to complete the game.
08 Sep	 Removed direct pits to die in, forcing the player to walk to the end	By removing the 'death pits', the player is forced to use even more time if they were to fail the jump, further balancing out the decision between gems vs. game time.

08 Sep	 Remove the 'easy way out', forcing them to need to make the 45 degree jump in order to exit from the pit	Weighing up the time impact of the ladder forcing them to make the jump to the time impact of forcing them to teleport back to the start, especially as they must learn how to make the 45 degree jumps for the third level.
08 Sep	 Before change to cactus and sign order	I changed the order of the cactus/sign following user feedback as they found the ladder shortcut far too quickly as they were forced to jump from the raised platform in order to clear the cactus, often unintentionally finding the shortcut. By changing the order of the sign and cactus, it means the player is much more likely to walk down the ramp and then jump, missing the shortcut, making it more of an easter egg to find.  After change

09 Sep	 Removed ladders and brought back rock to teleport back to start	Following user testing, I observed that it often took longer for the player to have to teleport back to the start in order to reach the end of the level than for them to make the 45 degree jump, and therefore I decided to bring this layout back to ensure the most balance between the gems available and time cost.
09 Sep	 Reskinned maps	I reskinned the maps following user feedback such as from William to make the maps follow a more logical progress in their use of sprites, so I skinned the first map in a grassy theme, the second map in a snowy theme, and the third in a space theme.

15 Sep	 Changed ‘deaths’ -> ‘lives’	User suggestion, as it makes the purpose of the life counter clearer, and allows for the game to end after x deaths
15 Sep	 Stopped drawing the millisecond count to the regular game text	Although it is still calculated for display on the game over screens, it is no longer displayer following user complaints that it made the game excessively laggy

18 Sep		Using a Desmos equation to calculate the final score as a function of both the total time elapsed and the gems collected. It is unfair to use a linear equation in this situation as it would heavily favour one playstyle over the other, by using such an exponential graph it allows for a curve such that a fast time is rewarded similarly to a reasonably paced game with lots of gems, but equally 'punishes' more mediocre performances depending on time and gems.

Given more time, I would have actioned some of the other feedback I received such as from Mrs Smith to add high scores, which would make the score system even more interesting, as people would be incentivised to beat their records. I would also definitely look into the feedback received from Luke to add an instruction screen or a level introduction screen detailing which objects are harmful, and potentially adding power-ups such as extra hearts or similar.