Santa Run

Game Design Document

Nathan Tubb
12-28-2020

Table of Contents

1. Santa Run

Santa Run is a 2d platform-based endless runner that utilises simplified dynamic difficulty adjustment to assist players of all skill levels in having an enjoyable and balanced experience. This unique system analyses players performances each time they play, to build an accurate assumption overtime on whether the difficulty of the game should increase or decrease to suit their needs.

2. Story

The player steps into the shoes of Santa Claus in his fitness preparations for Christmas eve. To prepare himself Santa is attempting his home-made assault course in the north pole, jumping his way over slabs of rock coated in snow and floated with reindeer dust. To help him keep his energy up, the elves have floated chocolate coins along the way, Santa can pick these up for extra reward though he must take care of the added risk.

3. Gameplay and Level Design

3.1 Player Mechanics

The player can control Santa using either keyboard or controller, allowing for a greater preference amongst users, the game uses unity's axes functionality to allow for a seamless transition between the two.

To traverse the in-game level Santa can use up to two consecutive jumps at a time, these jumps can be used at varying power by either tapping or holding the jump axes input, giving the player more control.

Platforms can be passed through from below, giving the player more freedom to choose their path or collect coins that have spawned in difficult places. Jumping through a platform from below will reset the players jumps allowing them to continue upwards to reach more coins or higher platforms.

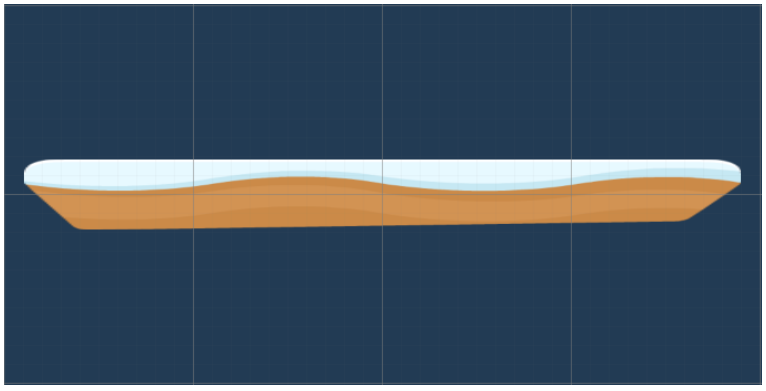
3.2 Procedural Generation

To give the player a unique and ever-challenging experience each time they play the game will procedurally generate platforms and coins. This to produce an experience with greater playability and challenge.

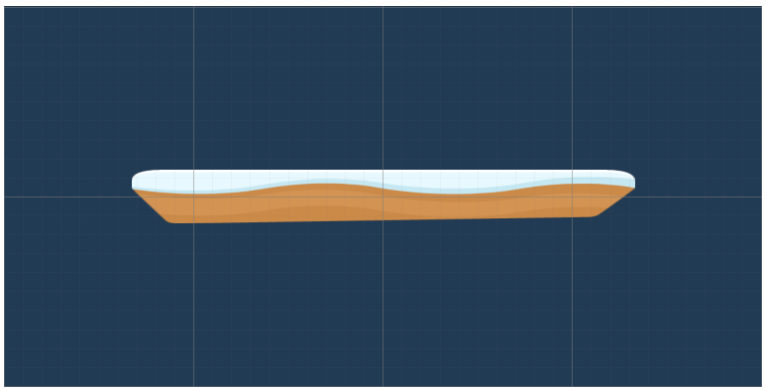
3.2.1 Platform Generation

To add variety, platforms will spawn in three different sizes. These sizes are selected at the random by the system each time it spawns a new platform allowing for infinite combinations and arrangements.

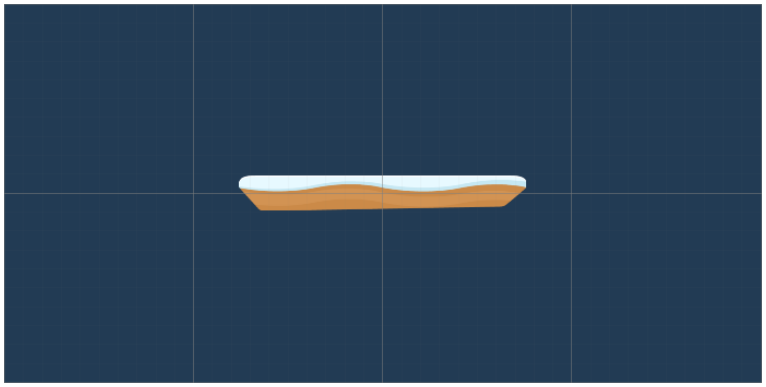
Large: The largest and easier platform to land on



Medium: Slightly more difficult though still has room for error



Small: The smallest and most difficult platform, often needing to be bunny hopped at higher speeds



Platform spawning takes several variables into account. Firstly, the maximum distance the player can cover with a single jump given their current speed is used to limit the maximum distance at which a platform can be spawned, this is capped at a distance that will always allow players to see oncoming platforms before they must jump. Using this system allows for platform gaps to feel proportionate and balanced, whilst ensuring the game does not become impossible or sparse.

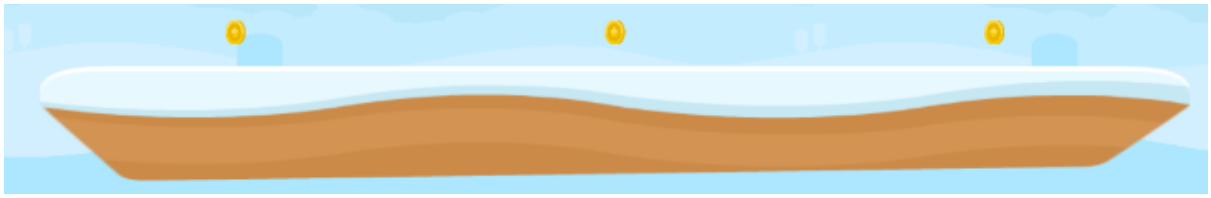
Secondly, the maximum height change between the two platforms is also capped so that there is never a height that is physically impossible for the player to reach. The value of this variable is determined through testing.

Finally, to optimise the generator, platforms will be pooled and reused rather than destroyed.

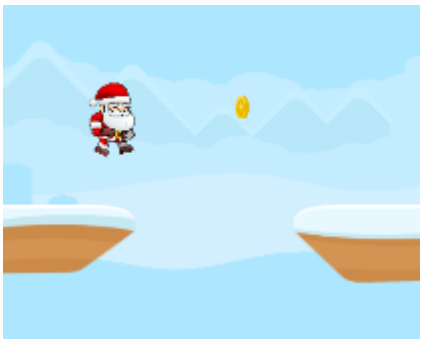
3.2.2 Coin Generator

Santa's chocolate coins are generated in three different patterns, this again adds variety and creates random new combinations for the player to try and collect. The generator runs separately to the platform generator though uses its position and information about the last platform generated to place its coins accurately.

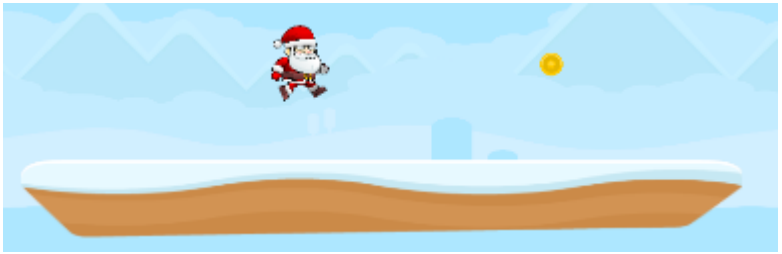
Platform Spawned Coins: These coins spawn in threes across different platforms. Being placed in thirds across the width of the platform they provide the easiest source for Santa to collect, provided he can land and run the width of the platform before needing to jump to the next one. This pattern of coins has a 20% chance to spawn each time a new platform is generated.



Platform Gap Coins: Singular coins spawned in the gaps between platforms these coins present a skilful challenge to collect often needing the player to jump in a way that will allow them to collect the coin whilst still landing on the next platform. These coins spawn halfway between two platforms at a random height, in relation to the next platform. Their spawn rate is also set to 20%



Random Coins: Random coins spawn at a random position and height above platforms, often presenting the player with a choice on which coins to collect, as they can spawn alongside either of the other two patterns. These coins have a spawn rate of 70% making them far more common than the other two types.



Finally, Like the platform generator, the coin generator also pools its coins for better performance.

4. Dynamic Difficulty System

A dynamic difficulty system allows for the game to adjust its difficulty outside of the developer's hands to account for the current user's skill or success in completing the task. Santa run utilises a simplified DDA system, that modifies the time it takes for Santa's movement speed increase to its maximum, based on the players average performance every three attempts. The average performance is compared against several threshold values to determine what decision the system should make.

4.1 Performance Calculations

Each attempt made by the player produces a performance value in the DDA system, this value ranges from 0 to 100 and is then stored ready to be used when calculating the players average.

The base performance for each run is set to the midpoint of this range (50) and represents what is expected of the player. This value is then affected by two modifiers that indicate whether the player has performed above or below expectation. The modifiers themselves are weighted as such that they can only increase the performance value to 100 or reduce it to 0.

4.1.1 Time Modifier

The first of these two modifiers is the time modifier. This modifier use tracks how close a player is to a target time at their time of death, to estimate whether they have performed better or worse than expected.

A decimal range from -1 to 1 is used to track this as it can then be used to multiply the weight of the modifier and produce the result.

Decimal Range Break Down:

-1 = 0 seconds: Player died in the warmup or when the game starts

0 = survival time target: In this instance 50 seconds

1 = (2 * time alive target): Player dies at or past the 80 second mark

Time Modifier Calculation:

a = Time Modifier

b = Range Value at time of death

c = Modifier Weight

$$a = b * c$$

4.1.2 Coin Modifier

The coin modifier calculates what percentage of the coins presented to the player they collected. It then uses this percentage to find a value within a decimal range from -0.5 to 1 and multiplies the weight of the modifier with this value. The reason for this modifiers range starting at -0.5 is that the coins generate in such a way that it is impossible to collect them all. Therefore, the player is only expected to collect 25% to avoid a penalty.

The calculation works out the percentage of overall coins collected based on the number of coins presented, and then finds the value of that percentage in the range.

Percentage coins collected

a = coins collected

b = coins presented

c = percentage collected

a / b = c

Value of percentage in range Calculation

Mathf.lerp = find a value in a range based on a percentage

Mathf.lerp(-0.5, 1, percentage collected)

Modifier Calculation

a = value in range

b = modifier weight

c = resulting modifier value

a * b = c

4.1.3 Average Performance Calculation

An average performance calculation is made every three attempts. This calculation takes the stored performances of these past three attempts and finds their average or mean.

Average Performances

a = total value of all three performances

b = total number of performances

c = average performance value

a / b = c

4.2 Difficulty Thresholds

To determine what the system should do once it has calculated the players average performance several thresholds are used as categories. The average performance is compared with these thresholds as seen below.

Difficulty Threshold Values:

0-22 = - 2 difficulty levels

23-45 = - 1 difficulty level

46-55 = no change

56-77 = + 1 difficulty level

78-100 = + 2 difficulty levels

A different set of thresholds are used for a testing diagnostic to determine the skill level of players when they first begin the game.

Diagnostic Threshold Values:

Novice: Participants who performed between 0 and 40

Intermediate: Participants who performed between 41 and 70

Advanced: Participants who performed between 71 and 100

4.3 Changing Difficulty

To increase the challenge over time the game uses a milestone distance system that multiplies Santa's current speed by 1.18 each time a milestone is reached. Milestones themselves are multiplied each time they are reached so that it will take the player the same amount of time or longer to reach the next one even if they are travelling faster.

Traditionally these milestones would stay constant every time the player attempted the game, regardless of how well they performed. Santa Runs DDA system instead changes these milestones based on whether it decides to increase or decrease the difficulty.

Increasing difficulty: To increase the difficulty the value of the first milestone is divided by 1.25 for 1 level increase or 1.5 for 2 level increase.

For example, the first milestone distance is 55 meters. If the difficulty is increased the new first milestone becomes 44m, meaning that the player must travel 11 meters less before their speed is increased. This change in the first milestone affects all milestones afterwards, resulting in the Santa reaching higher speeds far quicker.

Decreasing difficulty: To decrease difficulty the value of the first milestone is multiplied by 1.25 for 1 level increase or 1.5 for 2 level increase.

For example, the first milestone distance is 55 meters. If the difficulty is decreased the new first milestone becomes 68.75m, meaning that the player must travel 13.75 meters further before their speed is increased. This change in the first milestone affects all milestones afterwards, resulting in the Santa taking longer to reach higher speeds.

4.4 Results Document

Whilst the experiment is running the game creates a text file on the user's computer that records data about DDA systems decisions and the group the participant is in. It also shows the players average scores in each between each difficulty evaluation to show if they are improving or are finding the game more difficult.

Results Document Example:

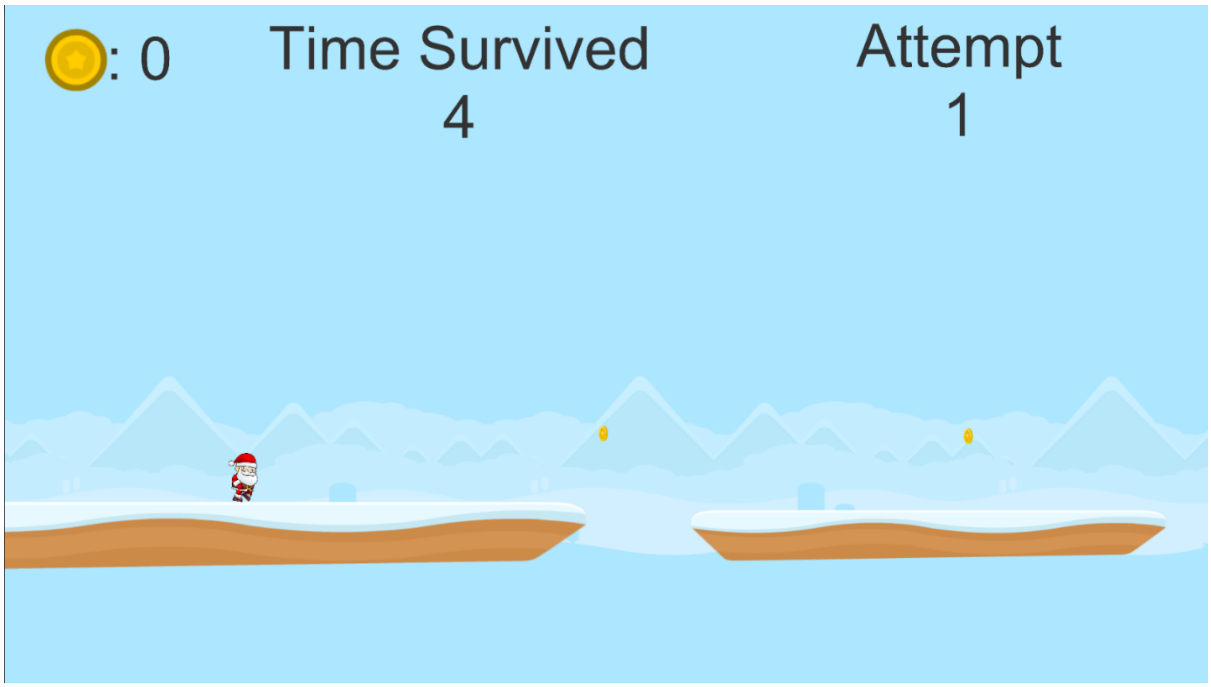
```
Participant X: DDA Results BreakDown

Group: Advanced
Average Time Survived: 79.60374
Average Coins Collected: 84

Difficulty Evaluation: 1
Result: Difficulty Increase 1
Average Time Survived: 55.91281
Average Coins Collected: 56

Difficulty Evaluation: 2
Result: Difficulty Increase 1
Average Time Survived: 64.96394
Average Coins Collected: 62

Difficulty Evaluation: 3
Result: Difficulty Increase 1
Average Time Survived: 59.54985
Average Coins Collected: 66

Difficulty Evaluation: 4
Result: Difficulty Decrease 1
Average Time Survived: 40.14948
Average Coins Collected: 44

Difficulty Evaluation: 5
Result: Difficulty Increase 1
Average Time Survived: 58.54477
Average Coins Collected: 64
```

5. User Interface

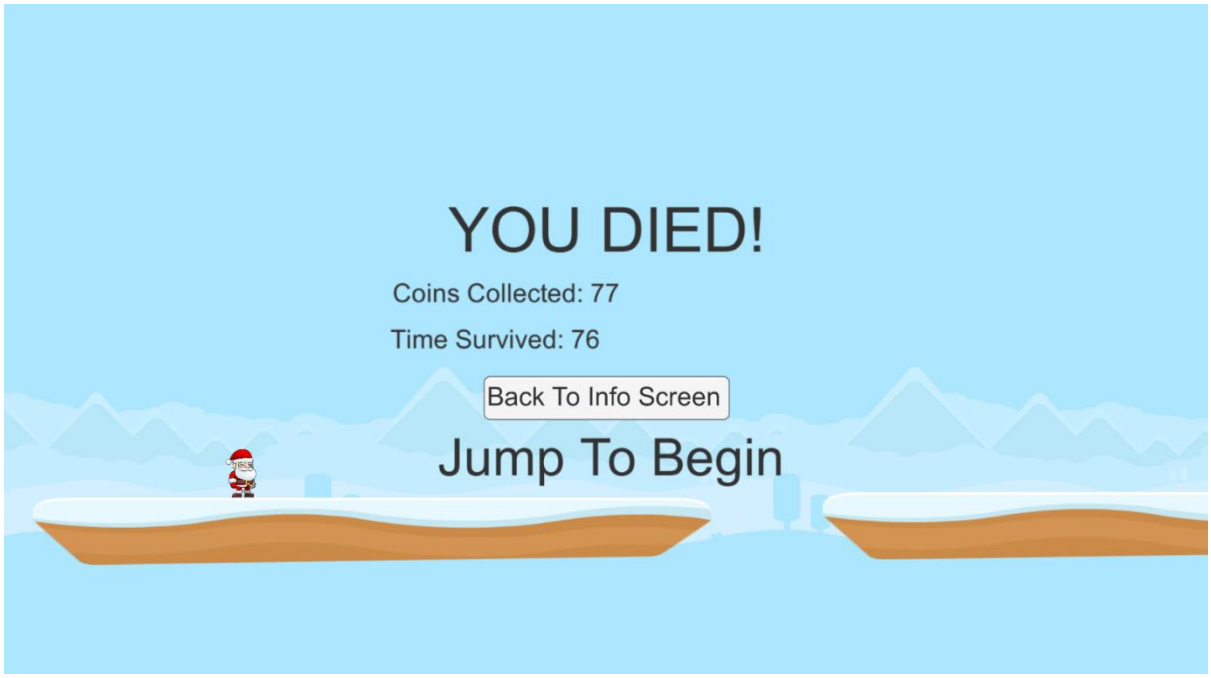
5.1 Game UI

As the focus of this game is to test the use of DDA systems within the genre the Gam UI is simple and straight forward to provide the player with some basic information.

Gameplay Screen: The game UI displays information about the player's current attempt as well as the total time they have survived and the number of coins they have collected.

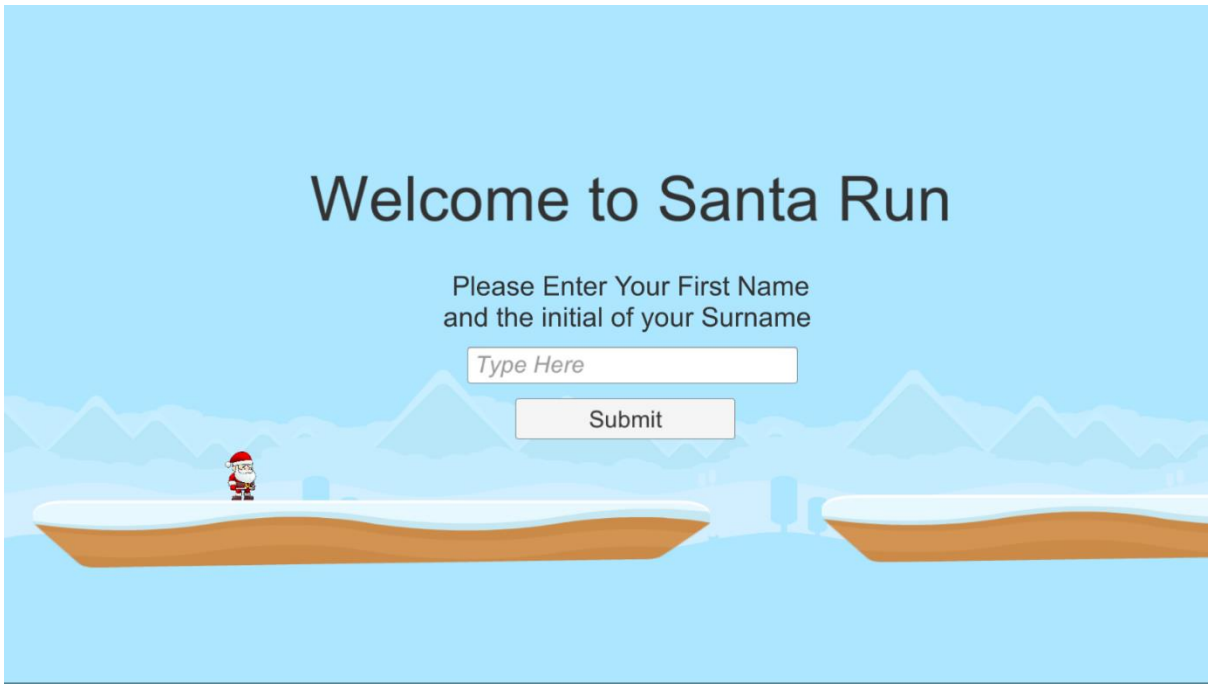


Death Screen: The death screen shows players how long they survived and how many coins they collected. It also allows them to navigate back to the information screen or start a new attempt.

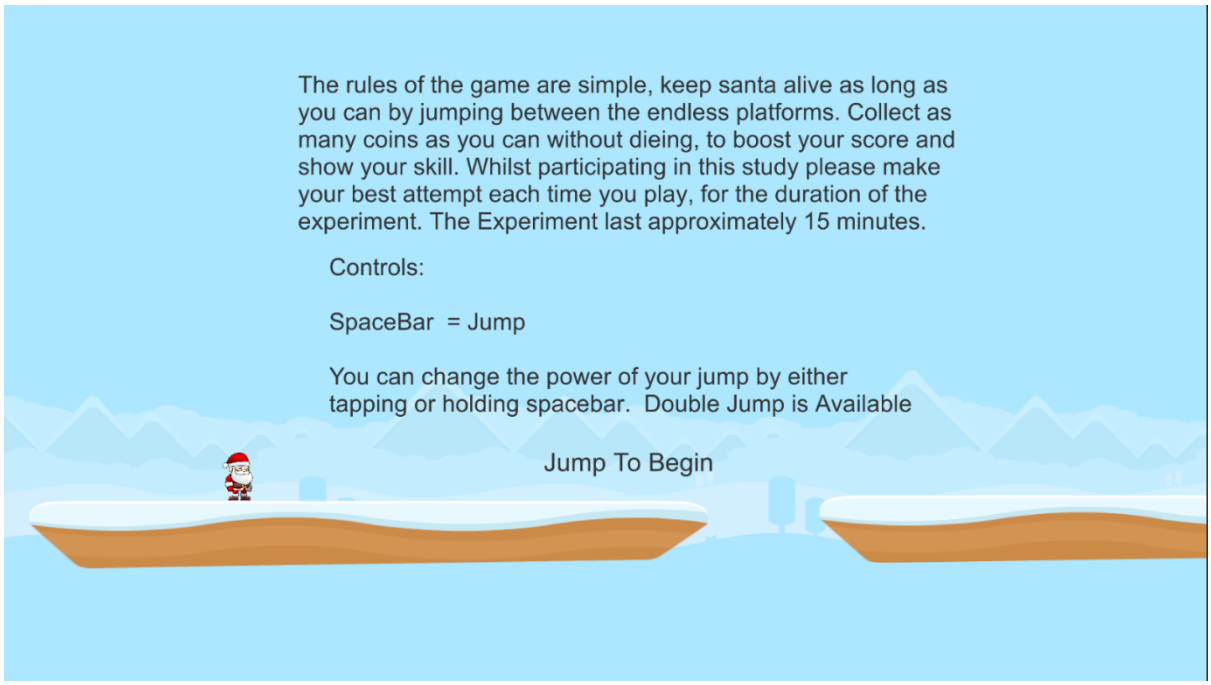


5.2 Menu UI

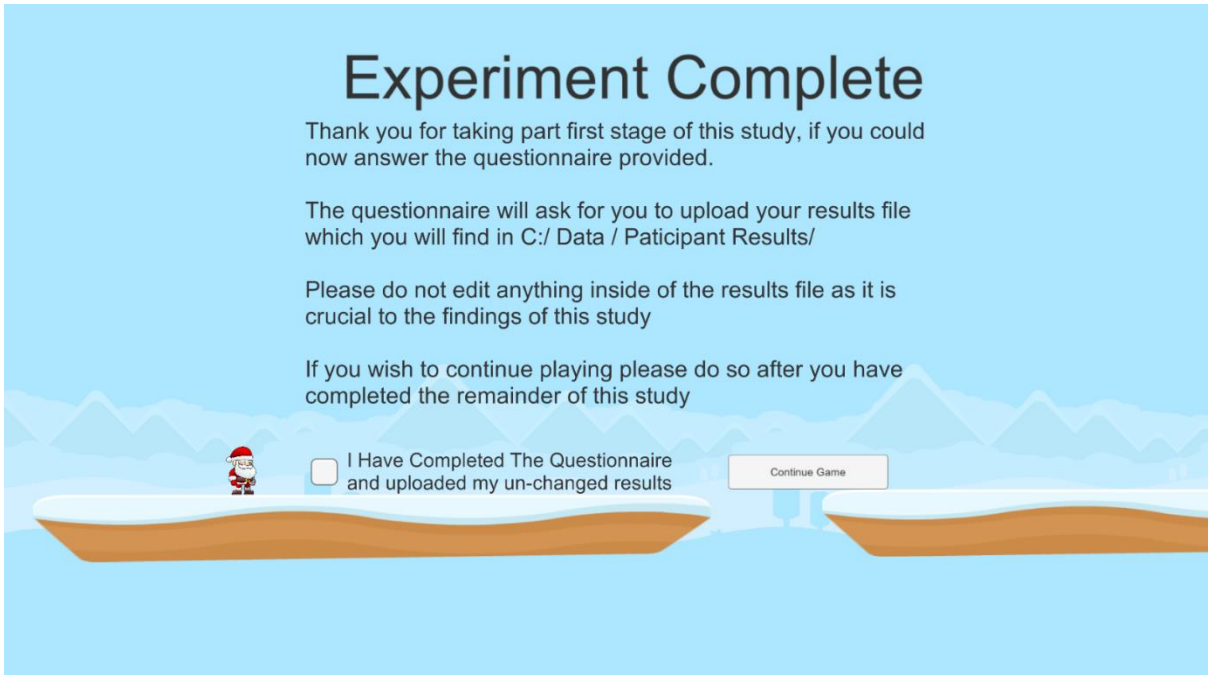
Name Screen: This screen is intended for participants of the study in which they enter their name which is then saved.



Info Screen: The info screen informs the player on what the rules of the game are, what the controls are and how long the study takes.



Experiment Complete Screen: This screen informs the participant that they have completed the experiment and guides them on what to do next.



6. Art and Sound

All art and sound for this game are externally sourced to decrease production time. Artistic style is intended to be cartoon and non-serious to attract more casual audiences.

Parallax Background: To give the 2d world depth and increase the feeling of motion the game uses a parallax background, in which, layers of the background move at different speeds to the player, giving the impression that they are in fact in front of or in a 3D environment.