Comp Sci NEA Analysis

Introduction:

My project is a “speed-runny” based platformer in which the player – the “Star” – will initially race through a level of platforms and other obstacles, with a single goal of reaching the end as fast as possible. After the first run through, the player’s “Shadow” will follow them, but unlike certain games like Mario Galaxy, where the player’s shadow follow their movements with a 3s delay, this player’s shadow will recreate the exact same path as the prior run through, either will recreate the player’s inputs or will use the player’s position data from the prior race to recreate the movements in a way. The overall aim is to beat your previous time like most speedrunners do in their respective games, looking for millisecond improvements in places others wouldn’t think to try.

The basis of the movement is going to be similar to that of “Dead Cells” by Motion Twin, with its very fast horizontal movement and some movement aspects from that of “Haste” by Landfall. Although haste is a 3d game, its slope acceleration mechanics would be a good feature to try and implement. Dead Cells is a good example of movement mechanics as along with its movement being nice for general use, the devs used it for things like hidden parkour challenges that test the player solely on their movement capabilities.



My target audience for this is people who enjoy platformer games like “Hollow Knight” and “Dead Cells” and also enjoy the repetitiveness that these games can bring, with combat devoid gameplay and much more movement based than some of the other types of platformers out there.

The main inspirations for this project are as follows:

- Super Meat Boy:

a similar platformer game by Edmund McMillen in which the player plays as meat boy and intends to save bandage girl from the many traps and other obstacles set around, includes a list of selectable characters with different traits and gameplay styles. The current \*any% world record is **17m 27s 267ms** by Matte, for reference, the second closest is 283ms slower, just over a quarter of a second.

- Trackmania:

a racing game in which avid players strive to perfect the maps that they race on, aiming for a perfect time with the aforementioned millisecond improvements in the tiniest input differences, sometimes borderline impossible for humans to achieve as they may be only 1 or 2 frame windows (1/60or 2/60ths of a second for most players) Trackmania doesn’t necessarily have an \*any% record as there are maps that the players can choose to play on.

On the right is an example of what is called a “ghost” in Trackmania, this is a past race someone has ran on the same track that the player is currently racing on, this person may be the player themselves past run, or a world record for a track.

speedrun.com:

the home of speedrunning, where all the speedrunners will upload and view everyone else’s times, all games with runs uploaded for them will be listed here along with their categories and sub-categories

\* any% is the typical speedrunning category for borderline all games and constitutes simply beating the game as fast as possible, by almost any means possible, the “almost” is because most games are ran with the “NMG” or “No Major Glitches” ruleset, meaning game breaking bugs that can get you to the end of the game within minutes. Some games do allow this like The Legend of Zelda: Ocarina of Time, where in the world record, they use a glitch to warp to the end of the game, beating it in around 3-4 minutes.

Dead Cells:

Basic Movement Mechanics:

Dead Cells has very snappy movement that is responsive and fast, the player has quite quick acceleration but you can avoid that with its rolling mechanic, I aim to recreate the movement in some effect, with at least the acceleration and the relatively quick speed. I do plan to have a “soft velocity cap” in which if you are going too slow, you will accelerate to reach it, but if you are over the limit, you will have a slowly increasing deceleration that will return you to speed. The progress of the deceleration’s increase will be reset in a manner of ways, like if you are maintaining a higher than cap velocity by constantly dashing, it won’t be able to begin properly decreasing the velocity. This is similar to Haste in a way as there is a lot of accelerating down slopes and you are given a velocity meter, that shows your current speed.

Haste:

Acceleration/ Deceleration:

As previously mentioned, I want to implement some form(s) of acceleration similar to what Haste has, this includes, basic acceleration up to an exceedable limit, this will require, the player to have a variable velocity, which at the current time is implemented, a soft velocity cap, which I currently haven’t implemented, and a few different forms of acceleration, these will be, on ground acceleration, which will progressively add to the players velocity up to the soft cap assuming no other acceleration is used, item acceleration, which will either: temporarily increase the soft cap so the player can go faster on average, or, add a large number to the players velocity but will also immediately begin max soft cap deceleration\*

\*Soft Cap Deceleration is a feature I plan to implement to counteract the players speed going too high, how it will work is, when the player’s velocity goes above the soft cap, a separate value will begin to increase over time as long as they are above the cap, this value is affectively an accelerating deceleration. Assuming the player is above the soft cap, the SCD value will begin increasing, progressively slowing them down more and more, until they are pushed below the soft cap, where SCD will begin decreasing rapidly, not instantly to avoid some buggy abuse of the feature.