CopyWrong is a 2D puzzle platformer game with various levels players can challenge. The player runs and jumps to collect the gem and attempts to dodge enemies on the way back. Remember, your movements toward the gem are very important. Tread carefully as enemy clones copy your exact movements.

The team's initial idea was a more flashy platformer with more action and less puzzles. The game is inspired by Super Mario Galaxy's cosmic clones and Mario games tend to have more action and be more fast paced than puzzle platformers. The initial plan was to use enemy clones to chase after you immediately when you start the game forcing you to move quickly. However, the team wanted to gravitate towards a more original idea rather than the plain inspiration from Mario. Thus, decided clones were to spawn after you reach a checkpoint and must carefully path backwards.

For the player impact, the team believes that this element of having to move backwards towards the clones is meant to create a more puzzle game versus the action platformer Mario games are. In fact, with the change from the initial plan where clones follow you immediately upon game start, the team also changed the character movement to be more precise. This deviation from an action platformer to a puzzle platformer required players to be accurate with jumps rather than diving into a fast action game. The team switched from slippery movements to precise movements to allow the player to control their path better for emphasis on the puzzle element.

The two week process consisted of a large amount of experimenting with different player movements (speed and jump height and slippery controls) to make right. Additional testing on clone spawn rate and max time per level was done in each level to ensure the right amount of difficulty. The gold spike consisted of a couple of levels that demonstrated the clones' copy

mechanic but not ideal movements nor clone spawn rates. There were also not a lot of platformer elements such as spikes or invisible blocks.

To properly implement movement and clone spawn rates, many tests were done. The team conducted sample play sessions with different movement physics and clone spawn rates and determined how much each change would benefit or harm the level's difficulty and player experience. With much testing, public variables such as "CloneSpawnRate", "PlayerSpeed", "MaxTime", etc were used. Overall, the movement was made more precise so players could experience more of the puzzle side of this game and clones spawned at a good rate every level.

Playtesting was done with close friends who are more likely to criticize bad gameplay elements as well as other 494 students and a 494 staff. Many found that the game required different movements and found the lack of platformer elements nowhere to be found. Thus, feedback was taken into account and these were changed. In general, many also found the game to be quite difficult and restarting the game after messing up took too long. Thus, improvements such as the lowering of clone spawn rates, increasing time, and a restart button for quick access was implemented. Perhaps one of the best feedback changes was a pause to the game timer until the player was ready to start. This gives the player more time to think about their path before actually attempting to challenge the level, defining the puzzle game as actually a puzzle game.

A notable part of the project progress was the team's time management. The team started early, and thus finished early. In the end, they had time to check for mistakes and fixed a good number of bugs without being in haste.

Unfortunately, what went wrong was the team's lack of planning before working on the project. The backend could be a jumbled pile of disorganization. The design was inconsistent with certain scripts linked to the player and others to different objects. For example, collecting

the gem has trigger code in the player gameobject whereas hitting a lever has the trigger code in lever prefab. Many times the team found itself searching everywhere for a component.

To ensure that the final project returns a better result, the team will take into account the importance of planning and will look at everything the game will consist of before putting in time to develop the game itself. The team will also make a spreadsheet or likewise to keep track of these elements if the system gets too large.