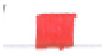
Design: An Introduction of the Mechanics

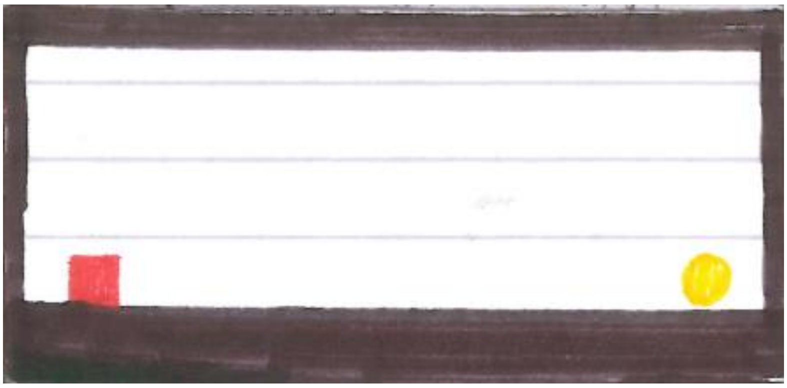
I guess for all explanation purposes; I will make a hyper-simplified example of the type of game I am talking about when I say a 2D platformer.

Let's say that the playable character is a red Square: 

The square has a limited set of controls- this would usually be the 3 arrow keys <^>, for the respective movement and an action button [usually spacebar].

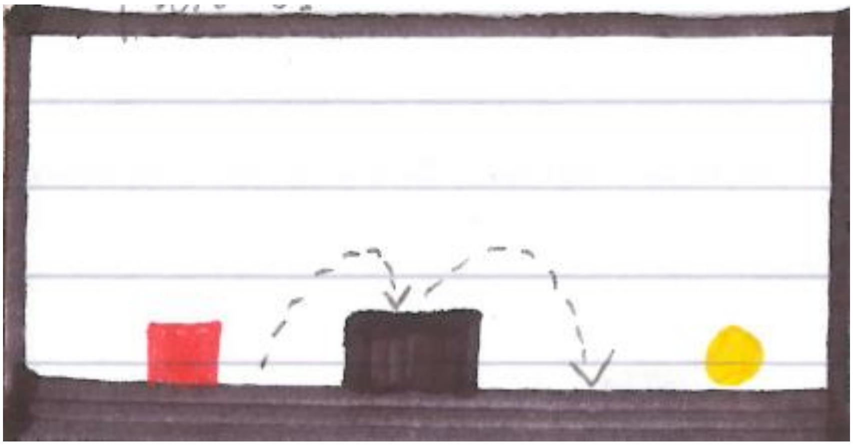
Sticking with the example let's say that the objective is a coin:

Now we can design a level:

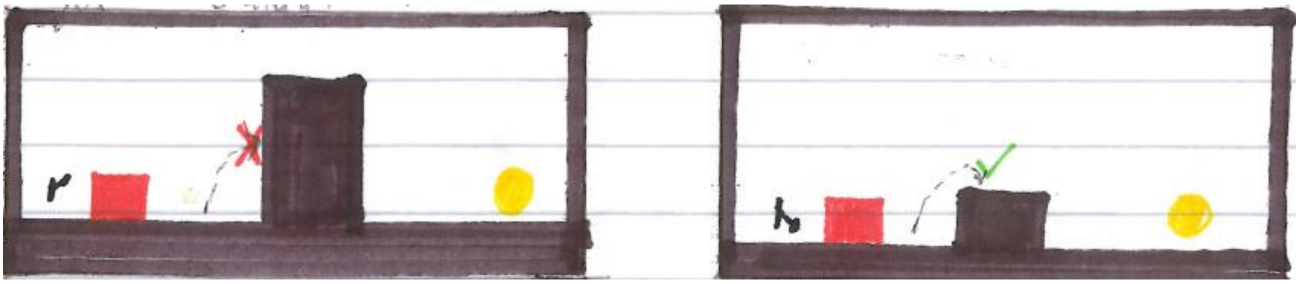


This is a very simple level- all you'd have to do is move right. While this would fulfil the requirements of a game (the user(s) control elements to achieve an end goal or current set objective). This doesn't really immerse the user or provide any type of interest.

Using the term 'platformer'- we can increase the difficulty by forcing the user to not only pay attention to their surroundings to achieve victory, but to also introduce a new feature:



While it may seem simple. An algorithm is now required to solve this level. Not only would you have to now jump over the block, you will also have to figure out when the acceptable timing for this action is. You would also have to consider if you can jump over the block, in this case you can jump on top on it, but if you couldn't…

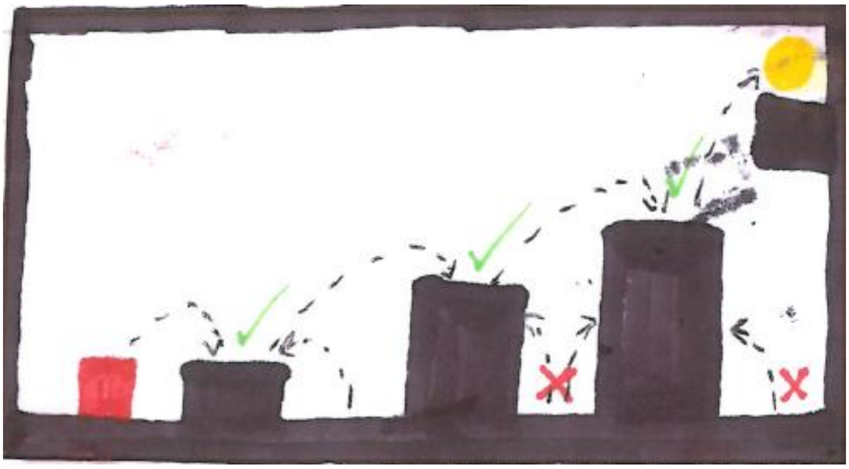


This introduces the final input the player will have- the spacebar for interactions. In this case the intractable object was a switch that lowered the middle section, allowing for the level to be completed. This forces the user to actually start thinking. They have a problem and logically, they must find a solution.

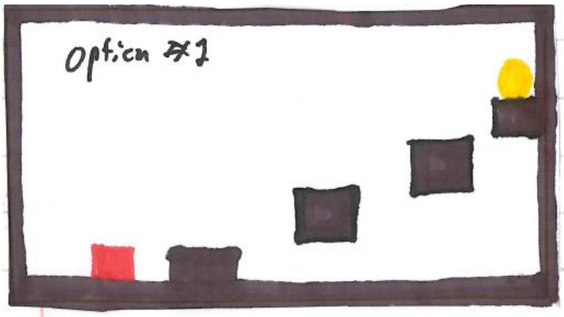
This is an example of the type of puzzles in platformers, not so much brain puzzles as it is puzzles of logic and assumption. Each puzzle takes the form of something affecting the path to you end goal, rather than a puzzle game where solving the puzzle is the end goal or the only means of progression.

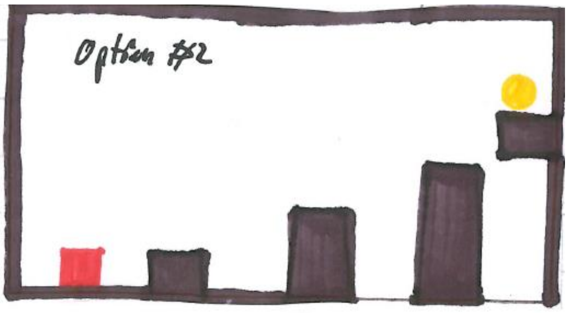
While I have explained most of the core mechanics of platformers, there is one large (but optional) aspect I left out (there is one more, but that will be brought up in a later aspect)- Failure.

Let's look at this layout:



This is a pretty stable level, the only problem is that if you miss-time a jump and land where the red crosses are, you just broke the game. As there is no longer a win condition, this no longer fulfils the requirement to count as a game, there are three ways of solving this problem.





Option number one avoids the concept of failure by making the platforms float. This means that rather than being trapped if you miss a platform, you just start from the beginning by moving back to it. This idea of designing levels to reset progress when you mess up via pushing you back to an earlier section rather than having a failure revert the game to an earlier state.

Option number 2 creates the concept of death. In the gaps of the platforms that would trap you I have removed the floor. This creates 'bottomless pits', allowing there to be a 'death' situation. In a death situation, the level will automatically revert to a previous state. This would either be the start of the level with all interactions reset, or for longer levels a checkpoint system may be implemented.

By collecting a checkpoint: 🚩

you create a situation where on death the game reverts to the state it was when you touch/collected the checkpoint.

The third option would be to create a manual restart option: ↻

what this would do is create a 'death situation button', triggering a death situation. Personally, I don't like this concept, as the idea of having to 'commit suicide' every time you miss-time a jump could lead to a very rage inducing game. Therefore, while I think the idea of a reset button is a good one, it should be used as an additional feature.

These are the features that lay at the heart of most 2D platformers. By adopting these simple mechanics, I can focus on themes and topics in the levels rather than coming up with a whole new game mechanics.