

Analysis

Let the headaches begin on what game we should do and what mechanic we should use. Everybody has their idea and everybody wishes to implement it. We spoke a lot on what game we should do and we came up with a few such as a car chasing game, clicking game (CookieClicker made us do it) and then we came up with the Gravity Running game.

We all liked different approaches but after all we will be using the Iterative Model which Maria introduced to us as she used it in her previous project. This model worked perfectly with our game as this game had to be finished as a whole level or else it wouldn't be able to work. The game had to have all the mechanics for it to be able a functional game. Without having the enemies, it wouldn't have been a fun game to play. This process includes the repetition of prototypes, playtests and revaluation of the game before releasing it. Compared to the waterfall model where the steps are followed one by one after each other, the iterative model focuses more on a cyclical development where certain steps are repeated numerous times. When using the iterative model there are 5 steps we need to follow:

- 1. Plan everything and see what we need:** Before starting to code and finishing our project we must first plan how the work will be conducted and also divided amongst us and we will also check what needs to be produced first. During this step those who will be programming the game will have to check what they need for the game to be complete at the end, which at this stage consists more of brain storming. One mustn't either forget what software is required for the final product.
- 2. Analysis and Design:** After finishing the requirements and planning required for the game we would then move on to analysing the appropriate business side of the game and how the game data would be stored and produced. During this step the designer would also start to plan the design and layout of the game, like sketching out the characters and sprites they would be using throughout the game.
- 3. Implementation:** After finishing the research of what is required the programmers would commence to code the game bit by bit. Here the designer would start seeing the game fall into place step by step and also help fix any glitches in the sprites.
- 4. Testing:** Once the coders finish crushing their brains we would then move on to testing our game ourselves first to check for any bugs and then move on to test it on other people who can help fix little problems they might have found in the game
- 5. Evaluation:** After the testing is done we would move on to evaluation where a thorough search and testing is done to check what needs to be changed and fixed. During this process both the coders and the targeted audience are tested to see what they think and what could be improved.

After all these steps are finished they are repeated numerous times until the game is close to perfect. The iterative model has its ups and downs as all things do. Some of the pros are:

1. It is more flexible to make changes and also the least costly.
2. It is easier to test and find the problems that occur.
3. Less risky situations as they could be identified while testing.
4. Easier to program and see it working immediately.

On the other hand the disadvantages of an Iterative Model are:

1. Problems with the device might come out later in the production of the game.
2. The project's succession depends on user's feedback
3. Takes time to develop the changes as they have to go through all the steps.

To produce our game we will first start doing some research on what the games require and how we could code them. We would also check if certain components are able for us to make with our little experience, we can't bite more than we are able to chew. We will check all the required material which we would need and read about the code we would be using to produce our mechanics. In conjunction the designers of the game will research and design all the sprites that we will be using for the game as they would be needed to start coding and trying out the game. After the designs are finished they would be passed on to the coders so that they can start to code the game with the sprites that we would be using in our final game. We would then continue on to test the game and each time change whatever bug that one might find while testing. After the testing is done and we have gained a bit of knowledge on what we need to change and fix and hopefully fix them all we would then continue to finalise the game and export it to be played on PC.

Components:

1. **Background:** For background we will need to use BoxCollider2D so that the screen size would be the size of the background.
2. **Bear:** The bear will be a 2D sprite which when it moves in the game it will have moving hands and feet to make it more realistic, we would need to use an animator for this. The bear will also have a BoxCollider2D and a Rigidbody2D. It includes the controller script which will make the movements of the player, and also the scoring when the bear hits a colour drop.
3. **Drops:** The game would also feature 2 types of drops. Both drops would have a Rigidbody2D and also a CapsuleCollider. When the bear hits the colour drop he would gain a point but if the bear collides with a black drop the game would end.
4. **Land:** The game will also feature a top platform and a bottom one so that the player could walk on it. This would have a BoxCollider2D as the player has to collide with it but can't pass through it.
5. **Scoring System:** The game will also have a scoring system which adds 1 score every time a player picks up a Drop. Also when a black colour drop is hit it will load the game over scene.
6. **Storyline:** Story will be shown on a canvas which will have a button to skip through it until the instruction page, when pressing play the game will start.
7. **Collector:** A script with a rigidbody 2D that when the bear collects a colour drop it will call the background and reposition it for the infinite background effect.
8. **Event system:** This will start the game when the play button is selected.