## **Cube Crawler**

**Game Summary:**  
A hyper casual game in the art style of most Mobile .IO games, the player takes third person control of a cube and must slowly progress along a path while avoiding obstacles in red, in order to reach their goal.  
  
**Core Mechanics:**

* Respawn points
* Red boxes hurt
* Blue do nothing

**Gameplay:**   
the player can controller their box with WASD or controller, they have to navigate through moving platforms, if they hit a red one, they go back to a respawn point.  
  
**Music:**   
will use a repetitive style of music, something the player can drown out but could be annoyed by purposely if they are losing

**Art Style:**

Very flat, inspired by .IO games.

**About:**

Lots of animated shapes moving across the screen, the player must navigate through them, they then hit a green colored checkpoint marker, with no indication to what it is but with no choice to hit it. once they die, they go back to it.

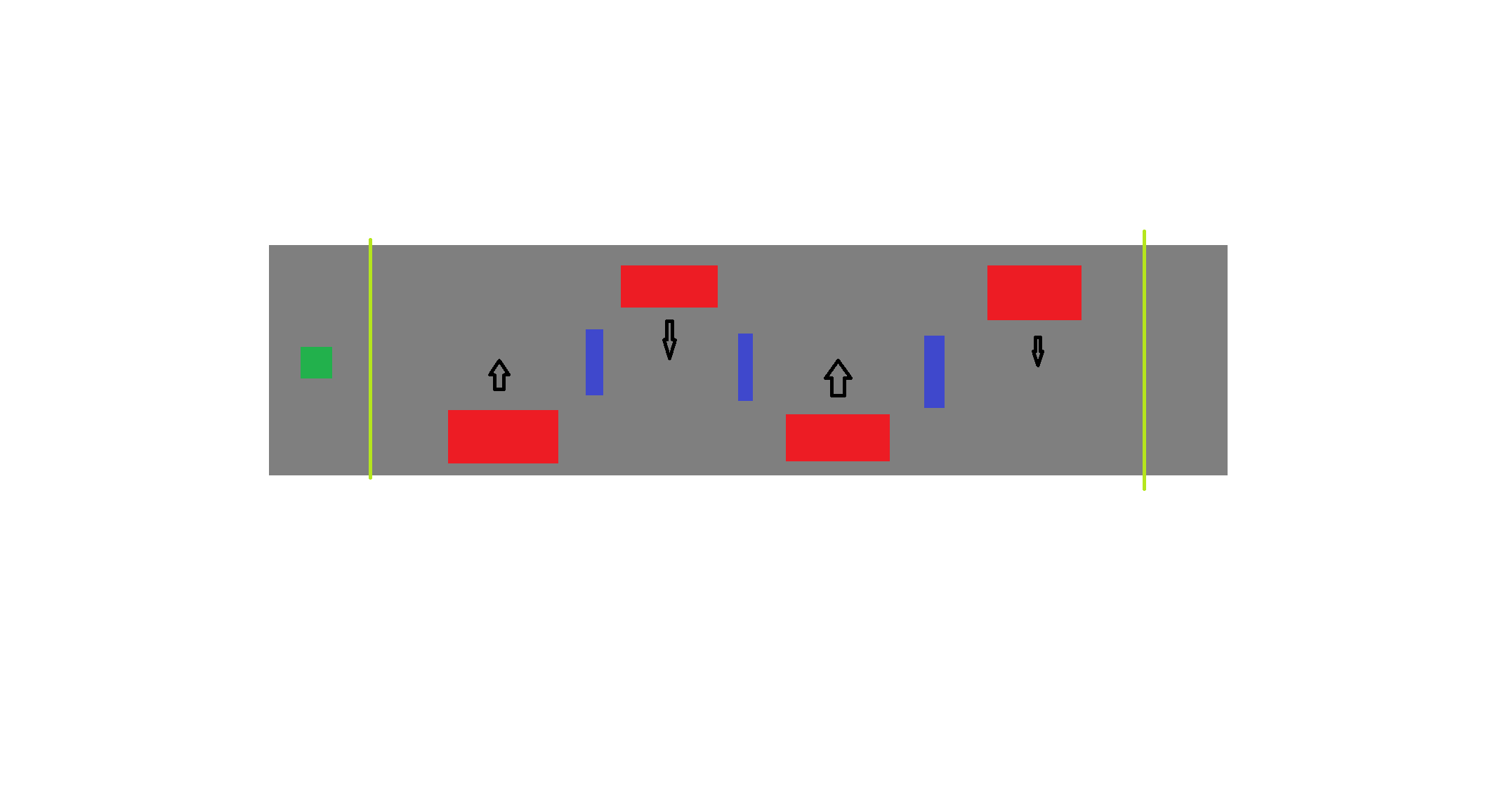
## Scope Check

## Part 1: Visualization and Implementation

**Visualization**

Answer these questions **based on the moment you have sketched** (not the whole game).

1. Sketch



1. Player has to move though obstacles, green/yellow lines are checkpoints, red boxes hurt blue are just there to get in the way to slow down the player
2. Red boxes move side to side, and later on could have more complex animations.
3. Player uses WASD to move around, the camera will follow them around, they have to get from one side to the other.

**Implementation**

1. Imagine how you would write the code for each thing in your last two lists (things that move and player interactions/feedback).

Movement: can reuse old code from a past assignment (already very simple to add in)

Animated platforms: probably won’t use too much code, will use animation system instead video link here: <https://youtu.be/rO19dA2jksk>

Camera controller, I would use a video I found a while back: give it an offset and it will follow the player: <https://www.youtube.com/watch?v=MFQhpwc6cKE>

Check point system: rather than use an array I can just replace a variable storing the last check point by updating a singular one: <https://youtu.be/ofCLJsSUom0>

Damage: give the red boxes a tag, if the player hits that tag they go back to a checkpoint.

1. Coding Question: Pause menu, best way to do it is to stop time and just gray out the game,

Will use Brackeys: <https://www.youtube.com/watch?v=JivuXdrIHK0&t=333s>

1. Most of my issues I found fixes for on YouTube with video tutorials.
2. Make a list of every visual and audio asset in the sequence (i.e. you don’t need to think about the whole game). Be thorough – don’t forget things like animations and particles, or UI elements like score.
   1. Music:
      * Main game music, repetitive and funny
      * Menu music, something classic, maybe a Mario song
   2. Will uses unity’s simple material renderer to give that .IO feel.
   3. All where in built apart from the background image in the main menu which will be a generic blank background.

## Part 2: Scale, Challenges and Resources

**Scale**

Now step back from that sequence mentally and think about the whole game. Think about all the parts of the game that can be numbered and grouped. For example:

1. Make a list. Go ahead and put down a number for each that seems reasonable.
   1. Movement
      * Controller support
   2. Checkpoints
      * Checkpoint location and style
      * Is there an indication
   3. Damage system
      * Respawn location
      * Reloading screen
   4. UI and menus
      * Pause menu
      * Main menu
   5. Animations
      * Tags given to the boxes
2. Mentally cut each number in half. Is the game still playable? Does it still create your core experience? Now try reducing each number to one, and ask yourself the same questions.

Write down your final list, with numbers, here.

* 1. Movement
  2. Checkpoints
  3. Damage system
  4. UI and menus
  5. Animations

**Challenges**

Based on all of the above, write down the top three challenges you foresee in the process of making your game over the next few weeks. Be specific, and phrase them as questions. These are examples of answers that are too vague to be useful to you:

* Coming up with unique level design
* Balancing the levels to not be too easy or hard
* Finding the right kind of music

**Resources**

|  |  |  |
| --- | --- | --- |
| Week | Area to work on | Description |
| Week 1 | Basic how to play screen with in which I shall:   * Add player controller and camera control * Add the floor and some obstacles that move * Add walls that do nothing | Create an environment where I can move and test out everything the player can do in the environment I have planned, once done turn this into the how to play screen. |
| Week 2 | Build the first few levels and add in check points | Make up about 3 levels and add in the begging of the checkpoint system |
| Week 3 | Pause menu and other Screens. Find music and a nice backdrop for the main menu | Add in the pause menu and the main menu and organize them |
| Week 4 | Add more levels  Play test what I have with friends | Add a few more areas with more complex Puzzles and ask people to try them out |
| Week 5 | Balance the game and adjust what I need to, put in the last levels | Add the last levels in along with the win screen.  Finalize the player speed and levels, get people to play it start to finish and see how they like it |
| Week 6 | Documentation | Finished all the documentation for the game. |

Animated platforms <https://youtu.be/rO19dA2jksk>

Camera controller <https://www.youtube.com/watch?v=MFQhpwc6cKE>

Check point system: <https://youtu.be/ofCLJsSUom0>  
Pause menu <https://www.youtube.com/watch?v=JivuXdrIHK0&t=333s>