
The Making Of
CUBED
A Reflective Practice
by Hien Quy Tran

A Reflective Practice

by Hien Quy Tran

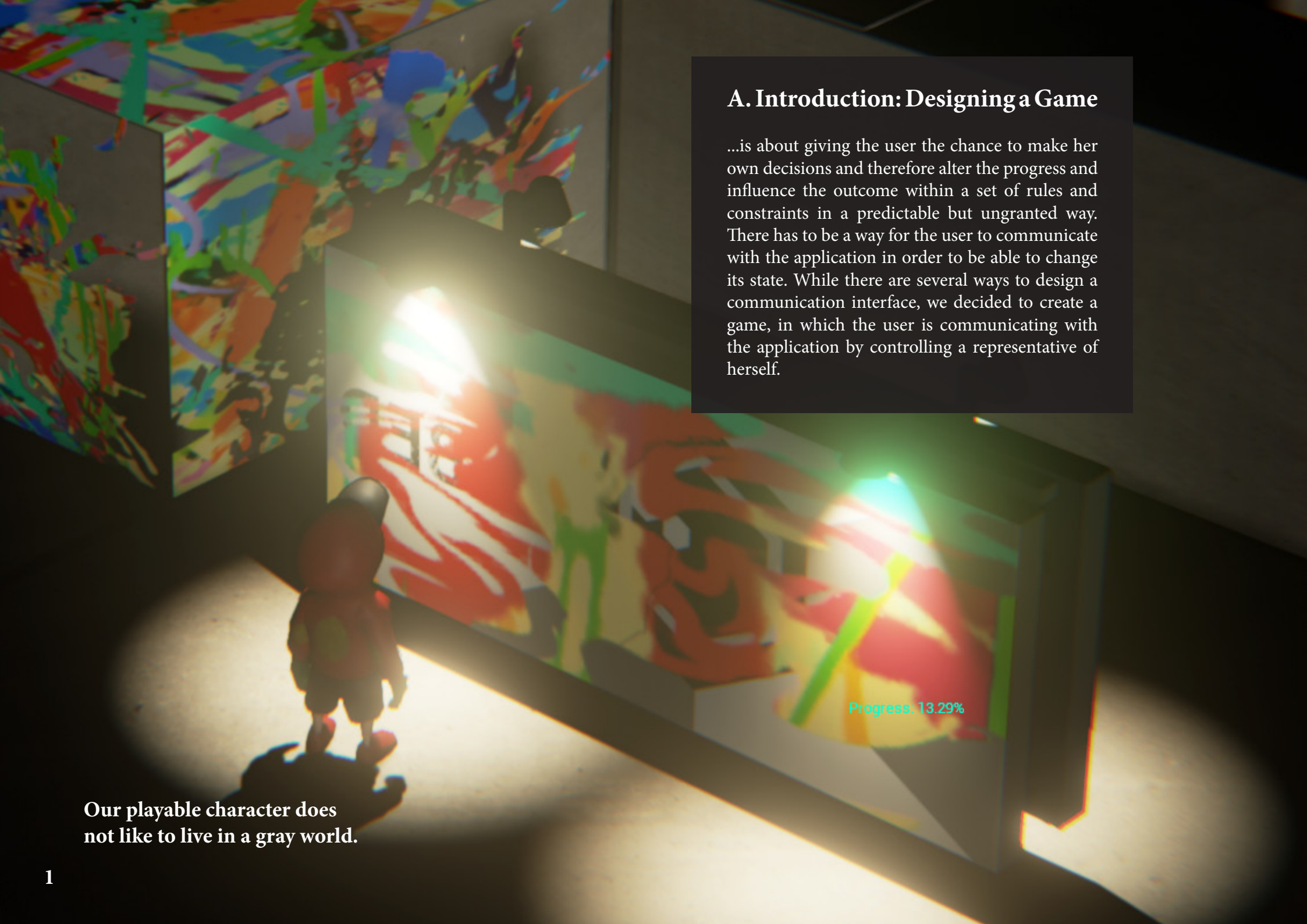
Date: 02/02/2015

Student Number: 543800

HTW Berlin - University of Applied Sciences
Lecture: Game Design
3rd Semester

Project: Development of a modular based game in Unreal Engine 4

The Making Of Cubed Content



A. Introduction: Designing a Game

...is about giving the user the chance to make her own decisions and therefore alter the progress and influence the outcome within a set of rules and constraints in a predictable but ungranted way. There has to be a way for the user to communicate with the application in order to be able to change its state. While there are several ways to design a communication interface, we decided to create a game, in which the user is communicating with the application by controlling a representative of herself.

Our playable character does not like to live in a gray world.

His mission is to color in the whole cube without being caught by the agents.

Making Decisions
...are about evaluating dependent factors and predicting the result in regards of a prefered outcome. While people have no choice to consume the reality, games have to be motivating in order to be consumed. Looking at the term motivation, the understanding of human behaviour still remains unsolved, but one of the most important factor is arguably the presence of a desired outcome. While there are different ways to promote a desired outcome, we decided to design our game around an obvious goal.

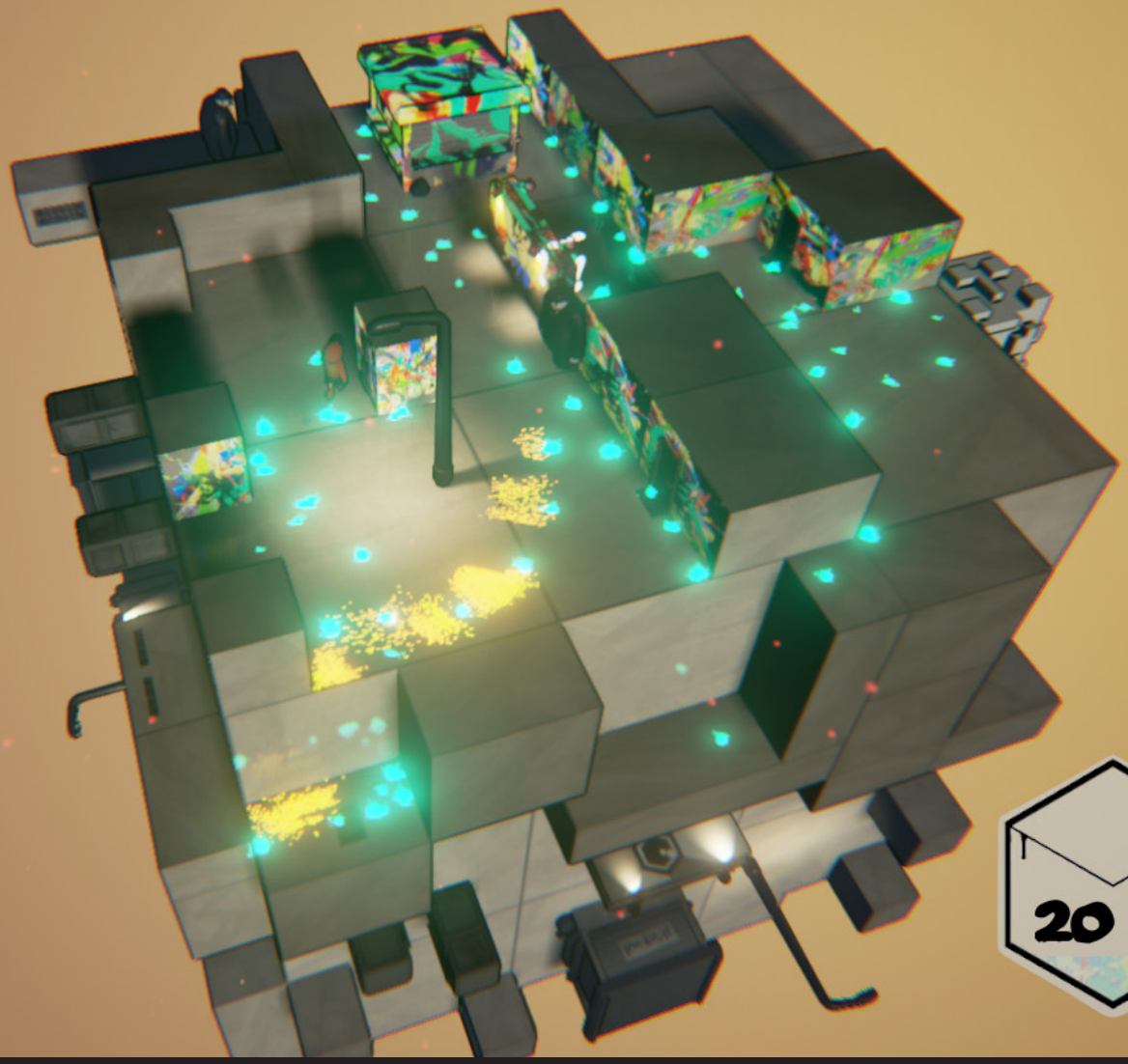
Competitive scoring system in which the multiplier is getting reduced by time.

SCORE 234,690

MULTIPLIER 29.1

1. Dual Scoring System (2 h)

The goal of our game is to color in as many faces as possible in order to set a high score. We wanted the game to be highly competitive, while still remaining rewarding for casual players. This is the reason, why we decided to implement two major different progress feedbacks. While some players will put more attention to the color progress bar, which shows the current overall percentage of the colored faces, others will put more importance to the scoring system which is designed to be more competitive. The major difference is, that the scoring system is time related, while the color progress bar is not.

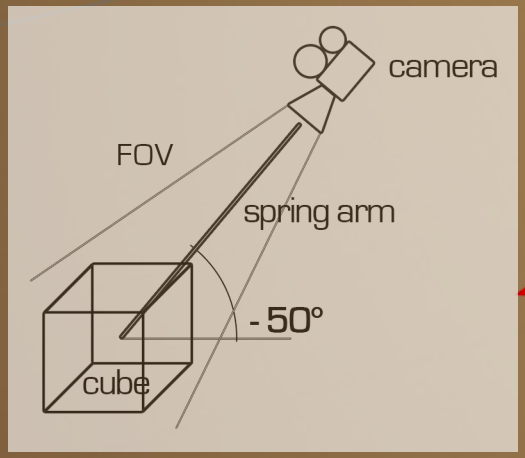


Progress bar is time unrelated. There is no need to hurry in order to reach the 100%.

2. Making all Sides of the Cube Accessible (20 h)

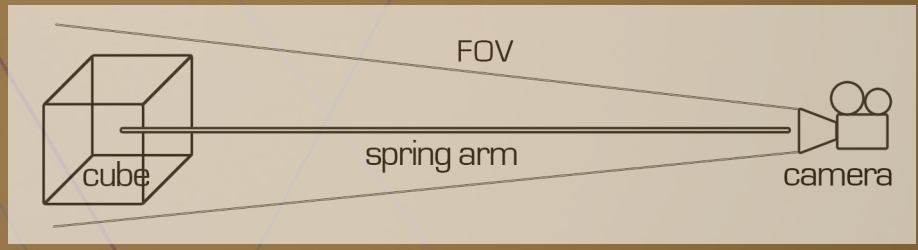
Because the playing field of our game is cube shaped and we did not want to limit the players movement to the top surface, we had to think of a solution for the transition from one surface to another.

The first step to make all sides of the cube useable, was to implement a camera system, which provides best overview over all sides of the cube. In order to do that, I placed a rotation point in the center of the cube and attached a spring arm which holds a camera to it. The camera is set up to face towards the the spring arm and the rotation point. Depending on the players position in world space, the rotation point is set to interpolate to a specified orientation.

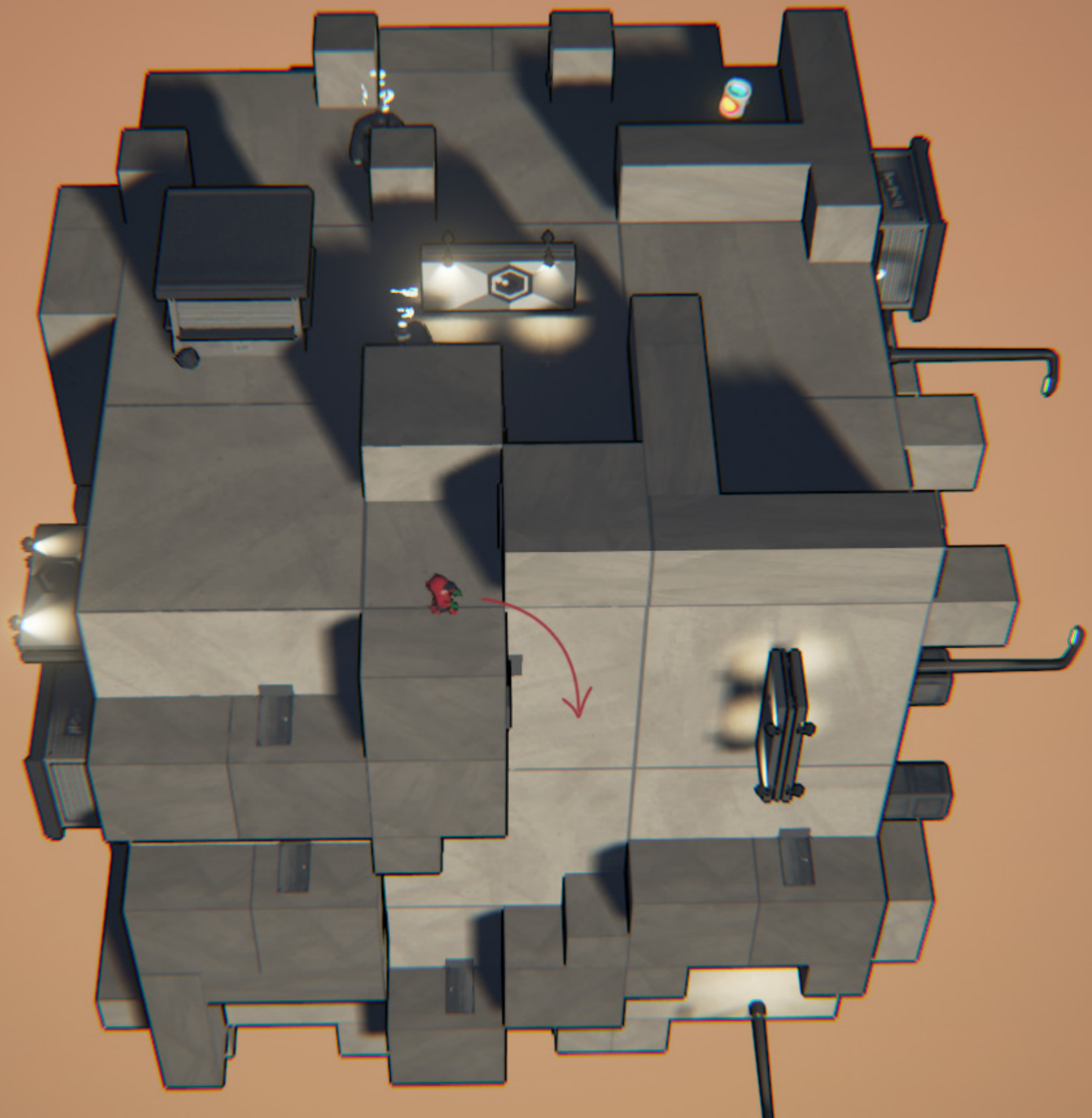


While the player is on the top surface the rotation points pitch orientation will always be -50°.

As soon the player leaves the top surface the rotation point will automatically orientate itself orthogonally to the corresponding side surface.

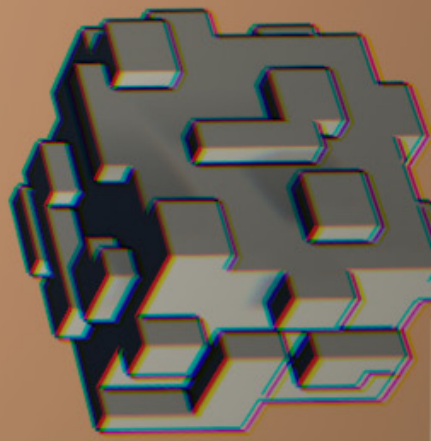


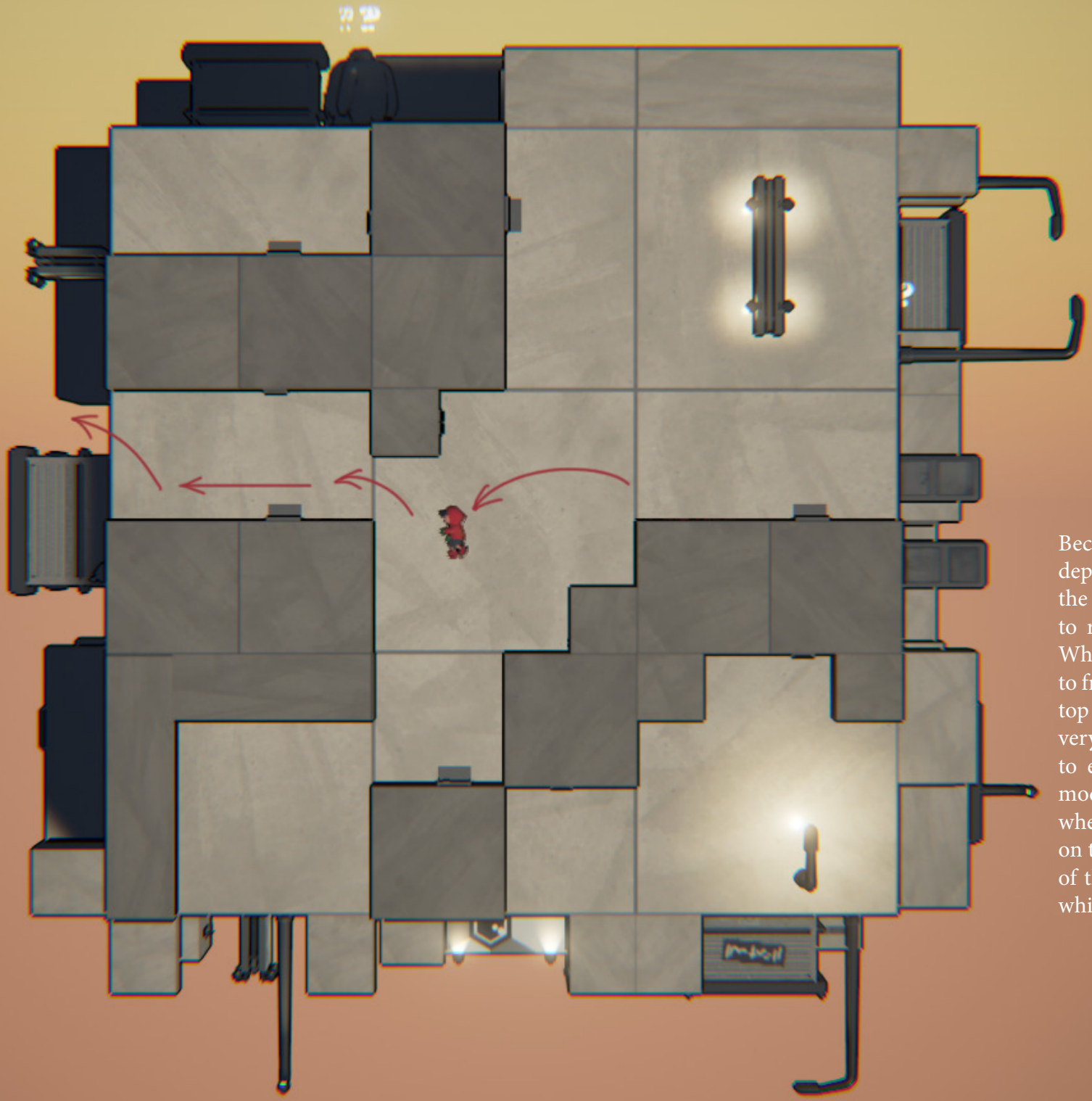
In addition to that, the length of the spring arm is being extended when facing the sides resulting in a higher distance for the camera to the cube. While the spring arm changes the distance, the cameras field of view is being adjusted to compensate the change. This allowed me to blend smoothly between perspective and orthogonal camera projection modes.



To give the player some control over the camera, I added the ability to rotate the camera around the cube by changing the yaw value of the rotation point and the ability to tilt the camera up and down by changing the pitch value of the rotation point.

While it would also have been possible to design the game to continue the same way, after transition to another surface, I wanted to give the unique shape of our playing field more relevance and make the game feel different after transition from top surface to side surfaces. The idea was to focus on being an isometric top down game when on the top surface, while focusing on being a jump and run sidescroller game when on the side surfaces.

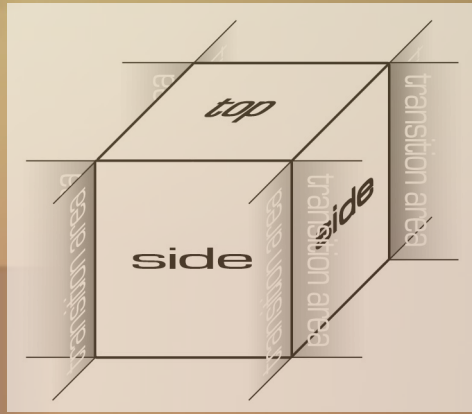




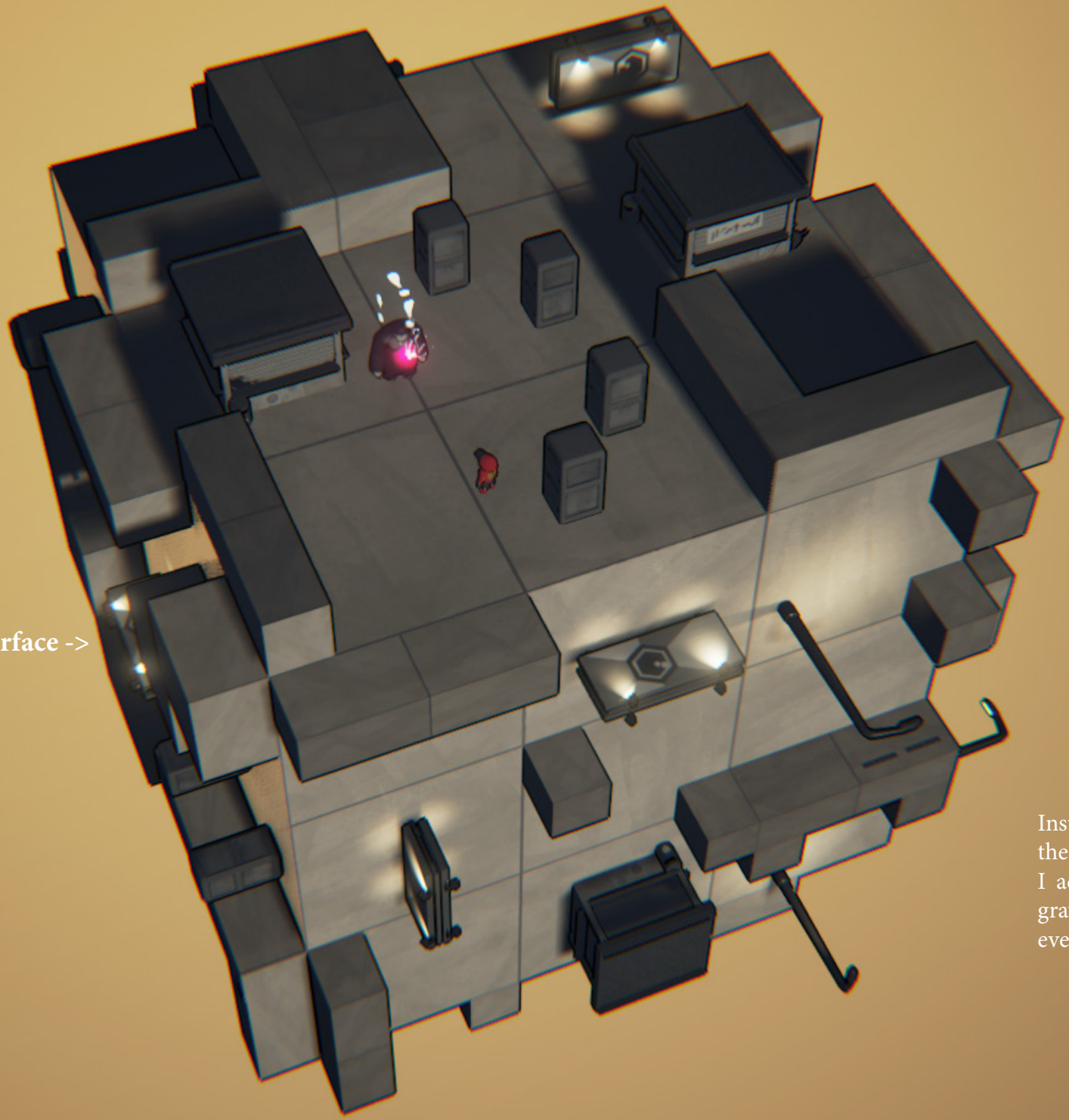
Because of the different game modes, depending on which surface of the cube the playable character is active on, I had to restrict the movement accordingly. While the playable character is designed to freely move along the X/Y axis on the top surface, its movement is designed very differently for the sides. In order to emphasize the different movement modes, I lowered the height of the jump when on top and raised it when being on the sides. Also movement along one of the axis is being locked accordingly while on the side.

3. Think around Corners (30 h)

While it is nearly always possible for the player to make a transition from the top surface down to the side surfaces, the easy transition back up is not always granted. To compensate this, I implemented the easy transition from one side surface to another and the ability to change gravity.



In order to make the playable character being able to jump around corners the game has to understand the timing when the player wants to make a transition. For that, the position of the playable character is always being tracked and saved into a variable. As soon it enters a corner area, the application reads out the last position of the player and predicts, that she wants to make a transition to the adjacent side. In this moment the game is slowing down to readjust the camera and controls.



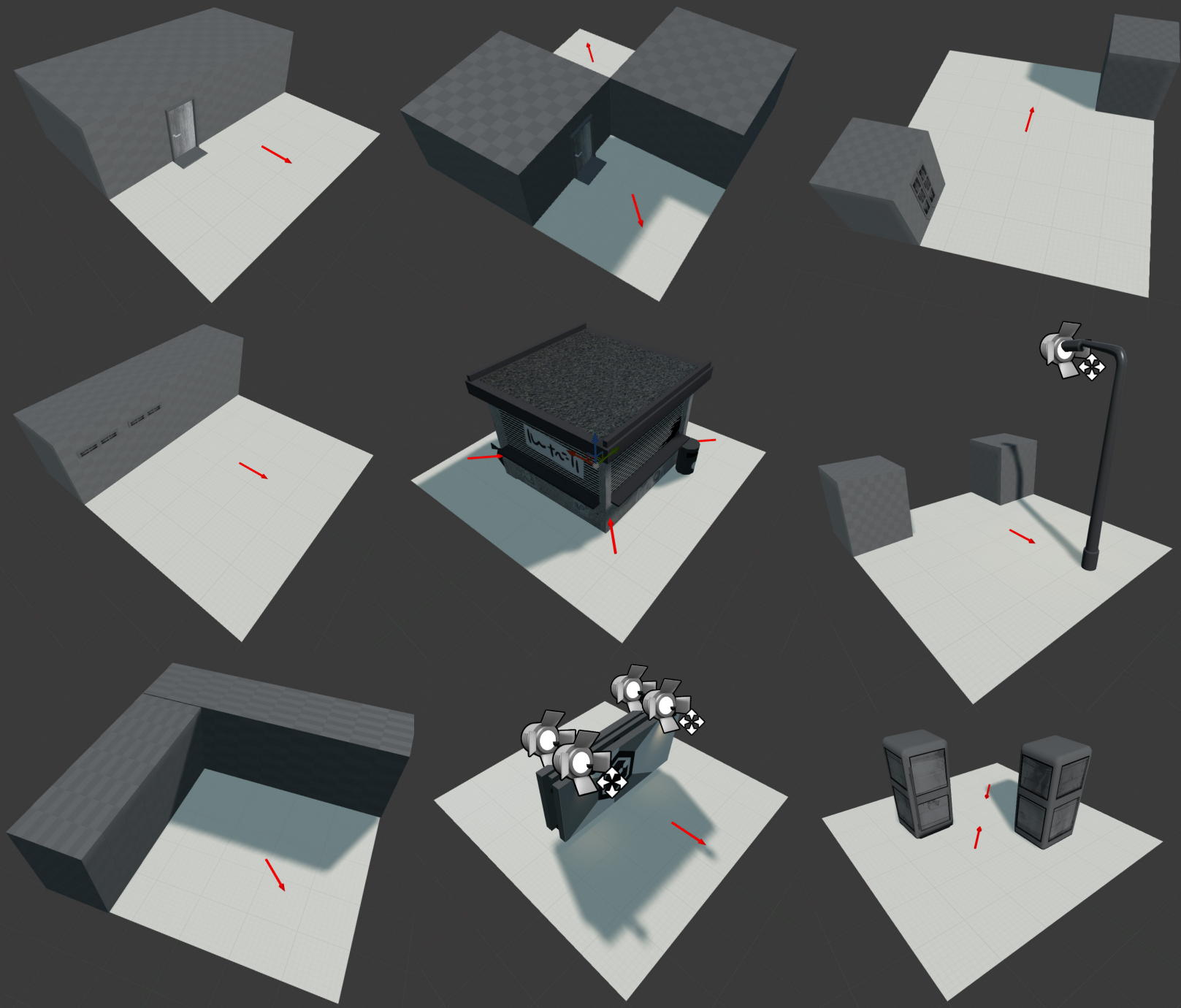
<- the side surface becomes the top surface ->

Instead of actually changing the direction of the gravity, I achieved the illusion of a gravity change by flipping everything instantly by 90°.

4. Designing the Tiles (30 h)

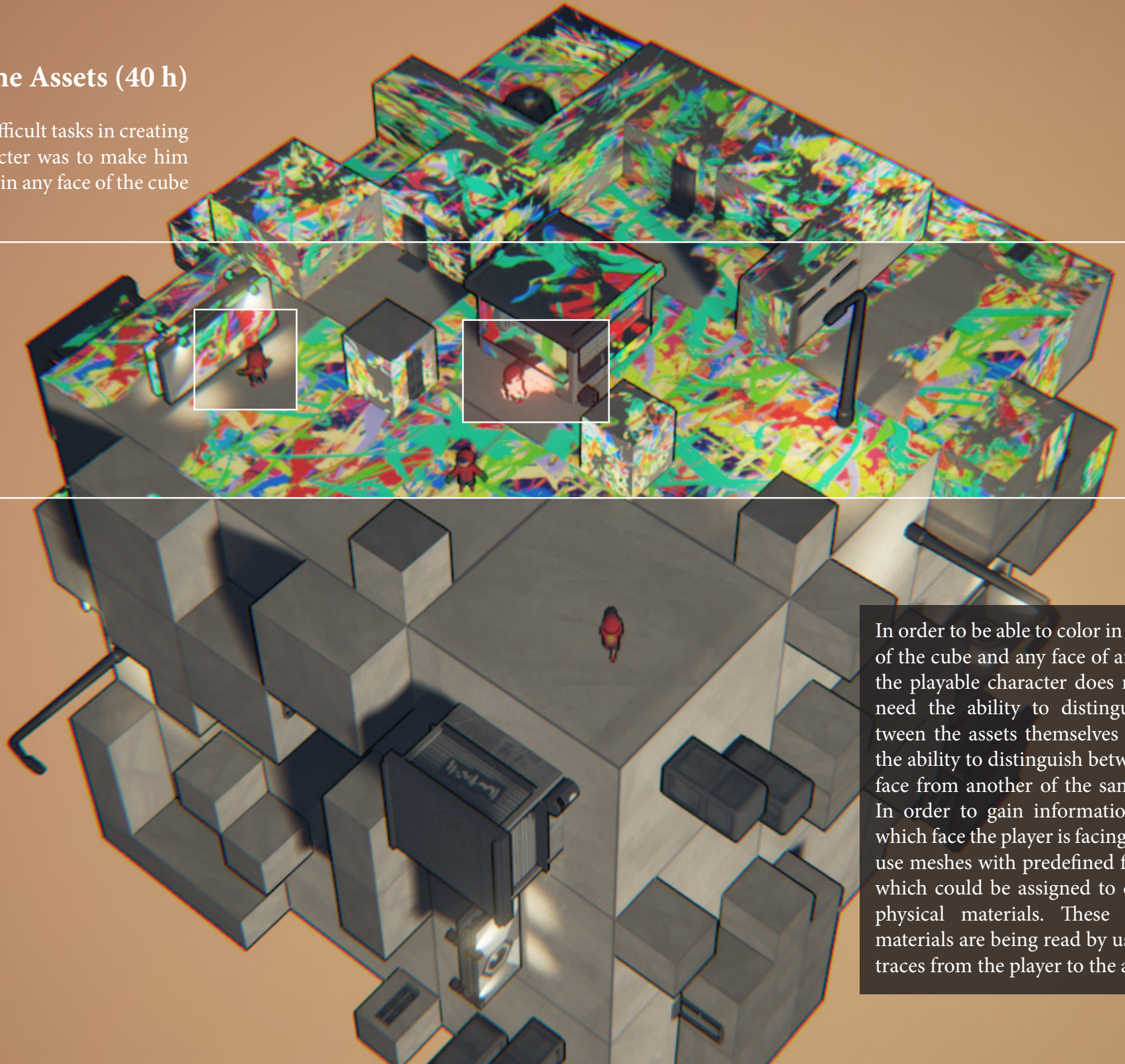
The tiles which are predefined and then shuffled up and placed around the cube had to be designed to provide interesting shapes and hiding spots on the top surface, as well as providing platforms and obstacles for the side.

To make sure, that the tiles will be equally distributed around the cube, I was setting up an array containing each tile once. This array is being shuffled up and the tiles are being laid out accordingly for each side of the cube.



5. Painting the Assets (40 h)

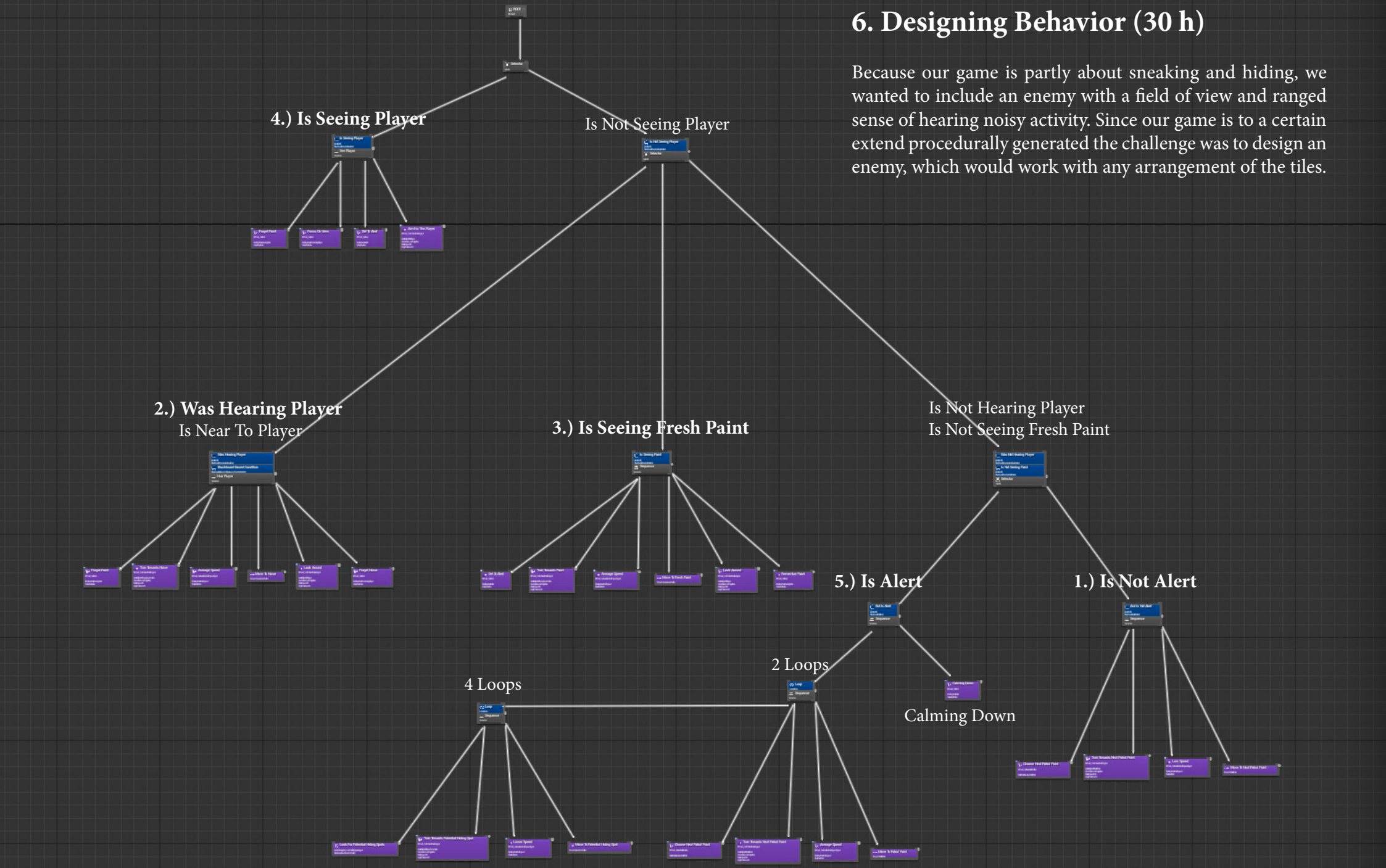
One of the most difficult tasks in creating the playable character was to make him being able to color in any face of the cube and the assets.



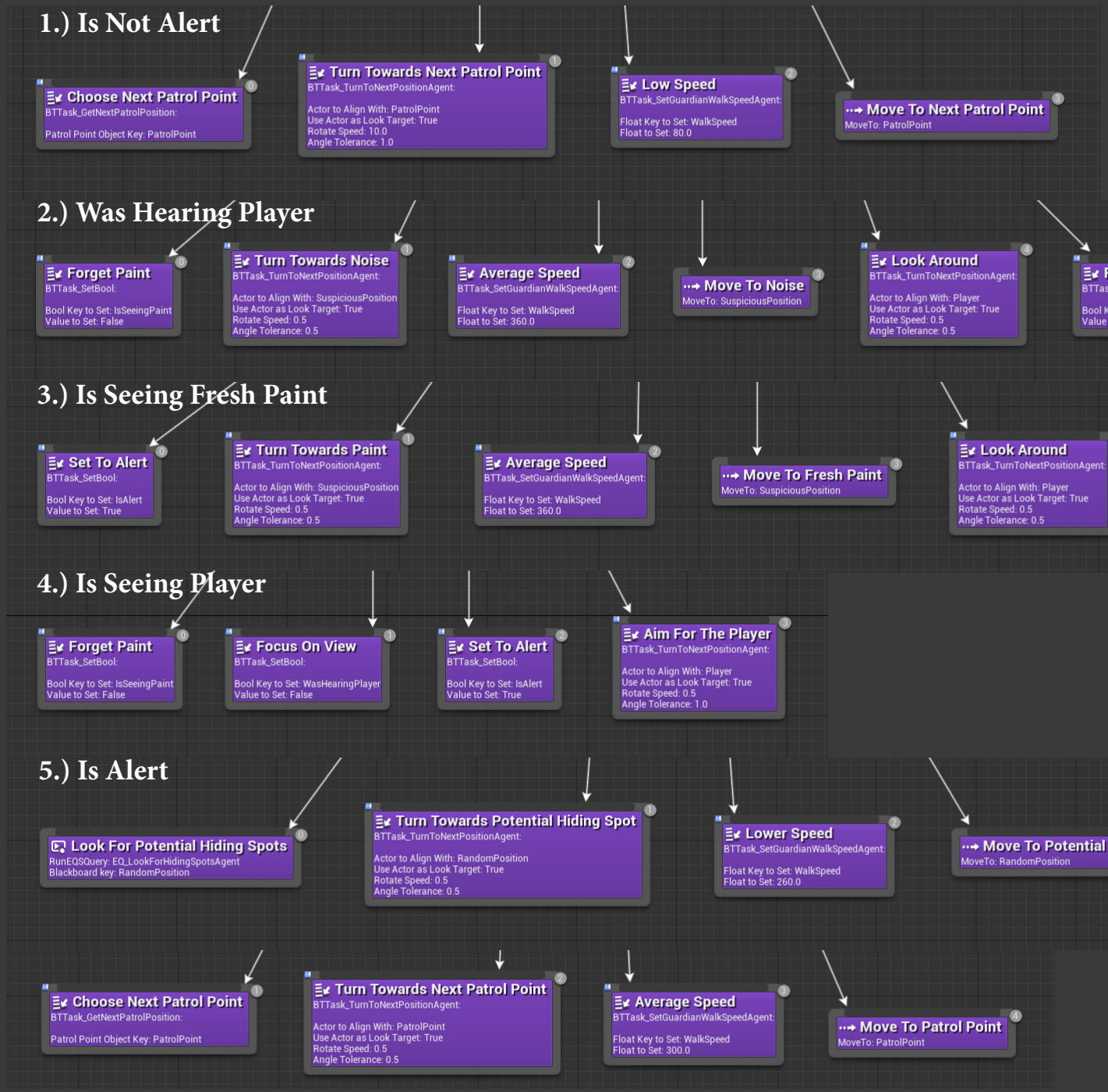
In order to be able to color in any face of the cube and any face of any asset, the playable character does not only need the ability to distinguish between the assets themselves but also the ability to distinguish between one face from another of the same asset. In order to gain information about which face the player is facing I had to use meshes with predefined face IDs, which could be assigned to different physical materials. These physical materials are being read by using line traces from the player to the assets.

6. Designing Behavior (30 h)

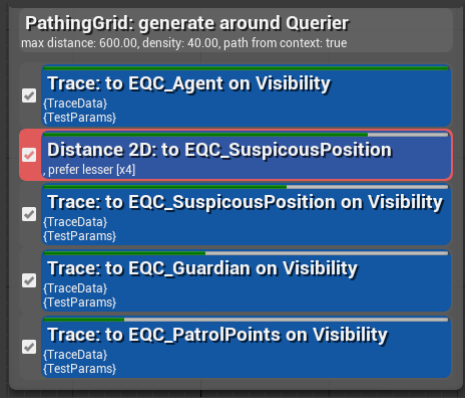
Because our game is partly about sneaking and hiding, we wanted to include an enemy with a field of view and ranged sense of hearing noisy activity. Since our game is to a certain extend procedurally generated the challenge was to design an enemy, which would work with any arrangement of the tiles.

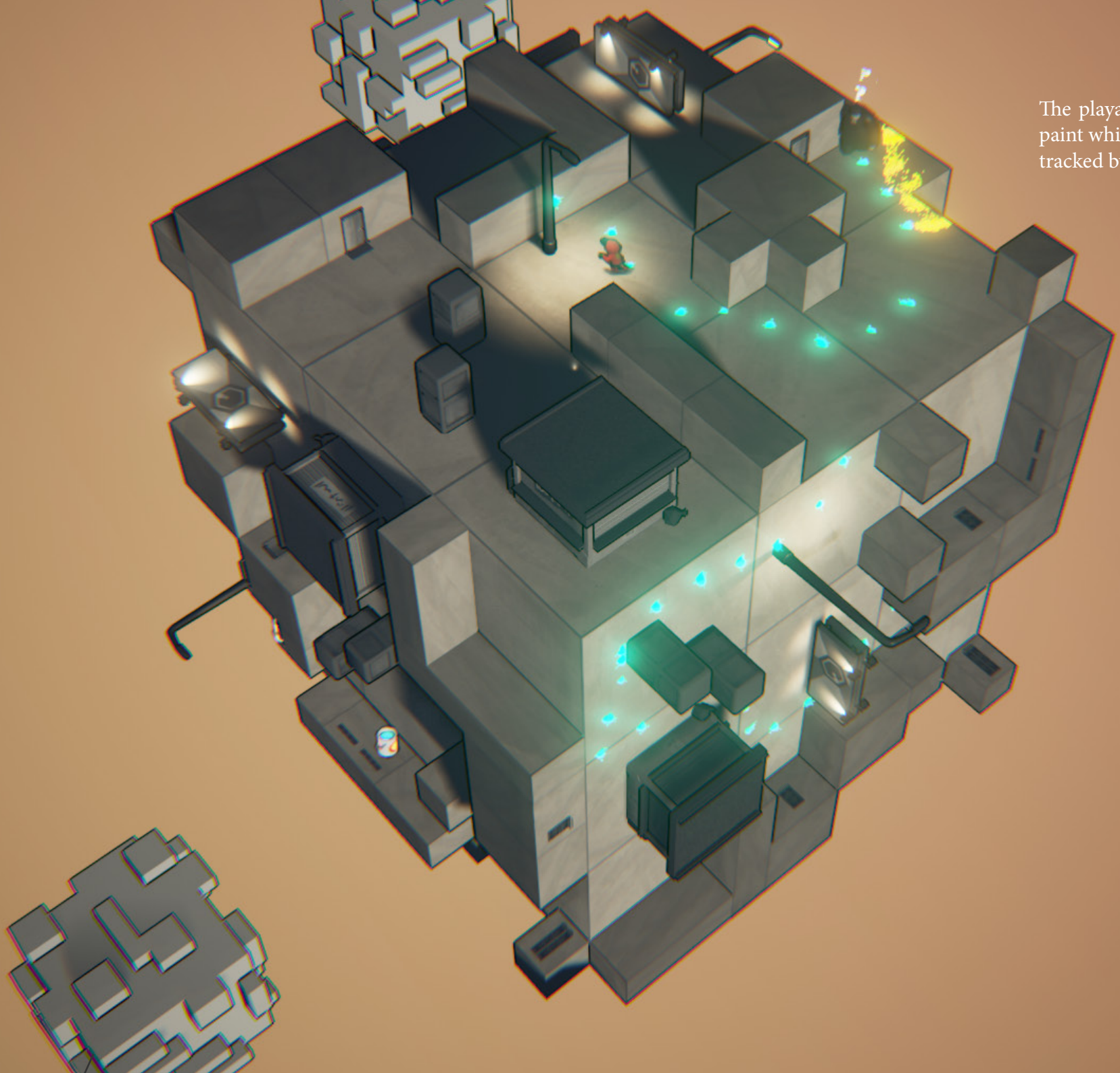


BEHAVIOR TREE

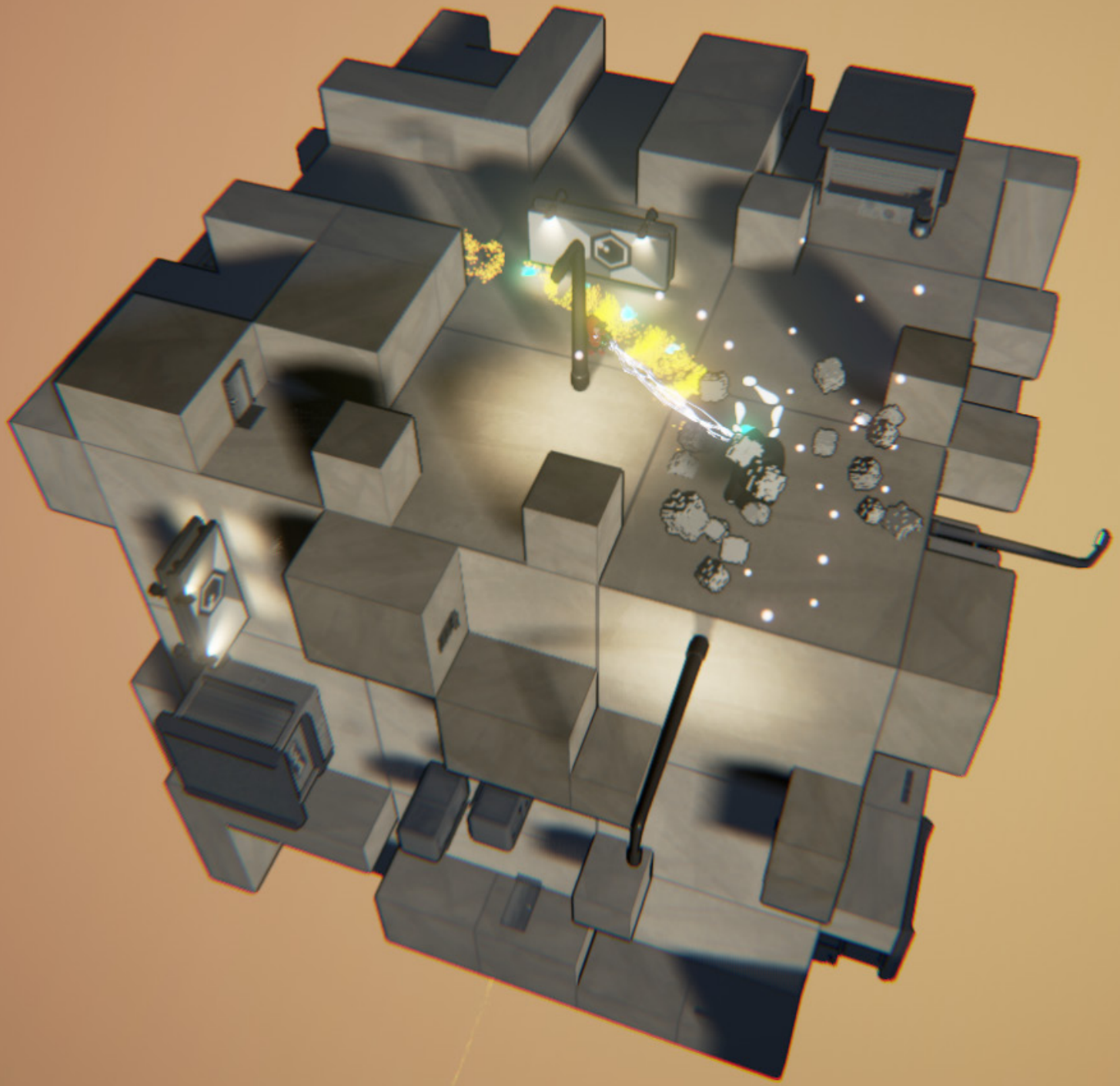


The enemy is using an environment query system which lets him to break out of his way points and identify potential hiding spots. He is preferably checking positions, which are blocked from his view and are near to known suspicious locations.





The playable character is droping paint while running, which can be tracked by the enemy.



The enemy will aim at the playable character on sight and charge his taser for an electric shot.