Night Painter – Graduation game at DADIU

About this project

During my internship at DADIU I we had a six weeks production with the goal of developing a mobile game to launch at the Google Play Store and AppStore. We were a group of 18 people divided into roles e.g. game designer, level designer, programmer etc. The game was implemented in Unity with C#

My roles

I was a programmer during the entire stay at DADIU. We were six programmers, who each took different programming roles. My roles were:

- Production pipeline programming
- Designated Audio Programmer
- Checkpoint/life system programmer
- Optimization

Pipeline

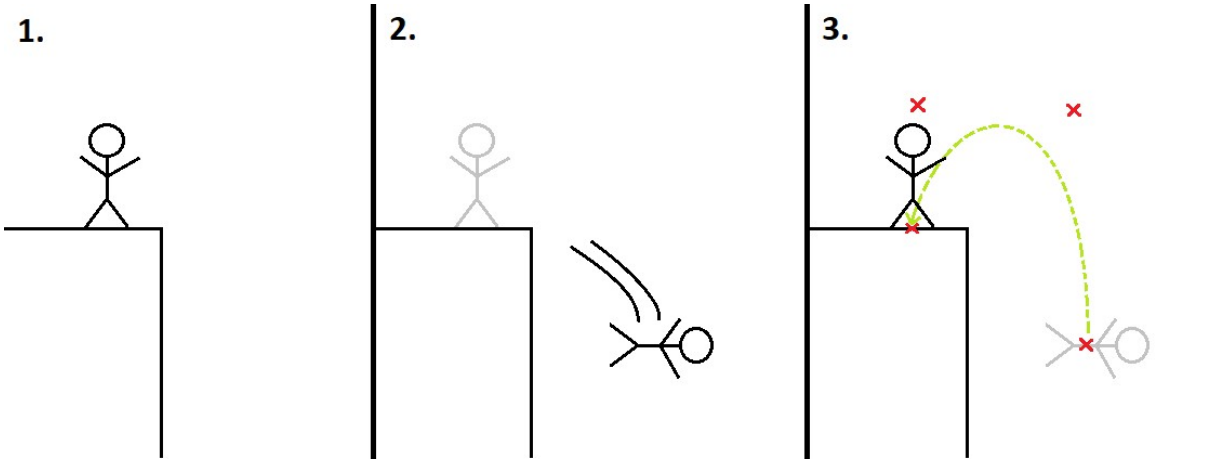
For the production pipeline we used bitbucket for the git repository and Jenkins to automatically build the game, every time a new commit was pushed to the master branch. We had a computer only responsible for building the game. Whenever the game was built, QA would go to the pc, move the game to their phones, delete the old cache and install the game. To optimize this process, I created a batch file, which would do the previous mentioned steps in one go, using the Android Debug Bridge.

Audio programming

To implement the audio, Wwise was used instead of Unity's own audio system. This would save the programmers a lot of work as the only thing needed, to implement the sound is a few lines of code. Instead of changing the source in the Audio Source every time, a new clip was available, the new audio is automatically updated in Unity, when updated in Wwise. My main task as the designated audio programmer was to be the communication between the programmers and the audio designer, and to implement the audio, whenever it swas finished.

Checkpoint/life system

The game was developed iteratively. Halfway through the process – based on feedback from visitors - the team realised that the game had too many confusing factors. Therefore, some of the things I worked on was changed. However, one thing which was kept was when a player jumps over a platform, they would move back to where they last touched it. This is illustrated below, with approximate positions for the points in the curve. The code for moving the player back to the platform can also be seen below.




```
IEnumerator ReturnPlayerToPlatform() {
    playerController.canMove = false;
    lastKnownCube = cubeInteraction.lastKnownCube;

    CalculateRoutePoints();

    while (tParam < 1) {
        tParam += Time.deltaTime * speedModifier;

        player.transform.position = (   Mathf.Pow(1 - tParam, 3) * route[0]
                                      + 3 * Mathf.Pow(1 - tParam, 2) * tParam * route[1]
                                      + 3 * (1 - tParam) * Mathf.Pow(tParam, 2) * route[2]
                                      + Mathf.Pow(tParam, 3) * route[3]
                                    );

        yield return new WaitForEndOfFrame();
    }
    tParam = 0f;
    playerController.canMove = true;
}
```

Optimization

Before the development of Night Painter, the team had already developed two games. We experienced the outcome of not including optimization in the design process. My responsibility within optimization was therefore mainly to look into the most optimal way of implementing different features. When getting closer to the deadline, I would be the one analysing what would cause a lower framerate, but also discuss with the lead programmer and the production manager, which ones there were high priorities and which ones there was time to fix.