IMY 300 – Project deliverable 2

Implementing the core mechanic

Description

This is the first deliverable for the basic interaction of your game. It will require you to implement a working prototype which demonstrates the **core mechanic** of the game in a test environment.

In other words, although your assets (visuals, sounds, environment, etc.) might not have been implemented yet, a player must be able to use this prototype and understand the workings of the core mechanic in context.

For example,

> If you are creating Tetris you will need to implement the stacking mechanic of the block, the rotation of the blocks, and how a lane clears once a row is full. You do not need to implement the point system, the variety of blocks, or even the win-and-lose conditions.

Although most of the assets you use in this deliverable will still be placeholder assets, your prototype must have some in-game assets that reflect the choices made earlier in Deliverable 2. These include:

1. A main character
2. One another in-game character (e.g. enemy, NPC, etc.)
3. Background music/ambience
4. At least one sound effect

The prototype does not have to be situated in the final working environment, but the environment should be set up in a way to demonstrates the core gameplay and experience you are trying to create.

You will be penalised for any bugs that prevent the core mechanic from being used properly.

Words from the lecturer

Use this deliverable to really fine-tune the core mechanic, as this is the main interaction that will drive the play of the game. By understanding which interaction is connected to what entity, you will be able to (better) sort out the behaviour of each class. The aim is to begin creating an adaptable base system so that the new entity you wish to include can comfortably fit into the existing system you are creating, making the expansion of the game less tedious.

This is also a good time to start thinking in fragments, in that the "effects" of interaction or even whole animations are not a single entity. As such it would be wise to add classes for effects or even animations, providing you with more flexibility on how to grow your system. A basic example of this can be seen in Figures 1 and 2.

Figure 1: Basic punch animation for a character

Figure 2: Cloud of dust effect for each punch

Marking criteria

- Quality of Implementation [5]
- Naturalness and Functionality of Core Mechanic [5]
- Absence of Bugs in Core Mechanic [5]
- Clear Indication and Expandability of Core Mechanic [5]
- Placement of Visual Assets [3]
- Setup of the test environment [3]

Submission instructions

You must upload a working executable of your prototype on ClickUP.

No booking is required as feedback will be given during the workshop session in the Yellow lab.