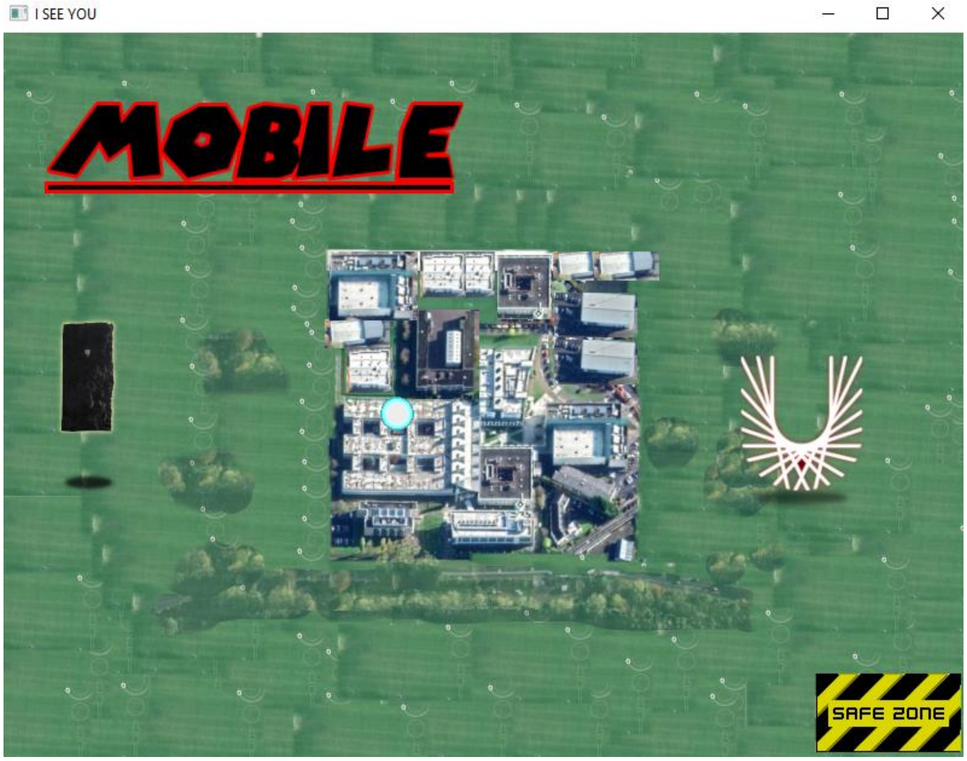
Apologies for my rambling video. I hope you find it useful.



Notes from Mario&Luigi class

https://setuo365-my.sharepoint.com/:i:/g/personal/c00302210_setu_ie/EddxcU3cZJhDvDkJa-lo0PcBG6vB9va3kOvwRlsbgx5Tvg?e=tiQ7qB

https://setuo365-my.sharepoint.com/:i:/g/personal/c00302210_setu_ie/EWcs1Uc4dlVIjzk68TgqwzIBn3rfn3KLPi2e6Q_WUj3q2A?e=dp3tDI

Reflections on Project

Aside from 1) the abrupt change of pace during the first Game Design lab compared to what had gone before and 2) getting my project to run on Library computers, I found the Mario&Luigi project to be straight-forward.

I was glad to see a header file demonstrated (having coincidentally looked it up the night before!). This makes a great deal of sense in terms of organising projects, and differs strongly from Unity/C#, where the standard practice is to incorporate most variables (including global variables) into script files associated with one or two classes. I imagine the header file will lead to a more data-driven architecture, similar to what Unity recently implemented with DOTS https://unity.com/dots. (Related - excited to use threads with SFML!)

The execution of Game.run from inside Main was illuminating - need to memorise a lot of this kind of material!

Due to my slow handwriting in the lab (see linked scanned notes), I was forced to rely to a degree on Pete's Github commits. In order to compensate and spend some time stretching my knowledge, I heavily customised the original project. These customisations included:

* Original (own work) object collision code for a central square object based on rect bounds

* Original 'collision' code based on radius between hostile actor and player character (dependent on player state)

* Extended player state switching to create a 'mobile but vulnerable - 'Mobile'' state and an 'invulnerable but immobile - 'Ghost'' state

* Turned off drawing background when player is in 'Ghost' state to create primitive 'vfx' to emphasise change in player state

* A patrolling enemy following a path between waypoints (using arrays), which fires projectiles at the player

* A gameover/reset function that returns the game to its initial play state on player 'collision' with hostile actor

* A 'win condition' of reaching a safe zone, that turns off the game (no time for a fancier end, sadly)

* Most of the art has been replaced with custom images.

* A soundtrack and additional sound effects has been added

The most difficult thing to build was the custom collision around the SETU campus. The behavior and construction are awful but was it satisfying to create 'collision' using the methods taught in the classes so far. The code checks set positions at the top, bottom, left and right of the player character against the bounds of the RectangleShape placed over the SETU campus, and if it is detected that the player has entered these rect, the player position is reset to the position just before entering the rect volume.

As ever, scope is my nemesis (also GGJ, also external work), leading completion to drag for a few days. I'm annoyed to find that it's easy to speed-run the game, but given the time and priorities, c'est la vie.

I've tried to stay within the bounds of the style guide - at least, the first half of the guide, I haven't read the other half yet! A major sin is that I haven't split the project into multiple Source Files - but, again, trying to stay within the bounds of what was taught in classes so far.

I hope you have fun trying to get to the safe zone! Watch out for that darn Logo!