GUI Report

The GUI for our game was inspired by The Binding of Isaac (*TBoI*) [1], it uses a fairly minimalist interface design - each area is a room in birds-eye view, with doors to travel other areas. Big Duck on Campus shares several gameplay aspects with *TBoI*, which is a very popular game – over 1.35 million owners on the gaming platform Steam [2] – the GUI used must be successful and playable. Therefore we found it a suitable model to use to improve the GUI design we inherited [1]. We emulated this with our GUI design – each university location is made up of a map of these rooms , randomly generated. This basic idea was unchanged from our inherited game based on the GUI report available at: https://github.com/teal-duck/teal-duck/raw/gh-pages/Assessment%202%20swap%20files/gui2.pdf. However, the GUI as a whole required refining, as will be detailed in this report. Numbers in brackets starting without letters refer to related requirements.

The GUI needs to give the player the exact information needed, as they need it – so that they can they have the required knowledge to play the game sufficiently. But also they must not be overloaded with irrelevant information which distracts from the gameplay [3]. We felt that the GUI we inherited (Fig 1.) did not do so sufficiently, or in a clear/aesthetically pleasing way. The information displayed was either small and formatted off-centre - Health and Score(4.3) - or absent entirely, in this case the player's current location and objective. Whilst being generally unhelpful to not have the location of the level (4.1), it is actually detrimental to the player, and consequently their experience, to not inform them of the objective they are supposed to be carrying out.

Requirements 4 through 4.4 (available online at: https://github.com/teal-duck/teal-duck/raw/gh-pages/Assessment%202%20swap%20files/req2.pdf) detail the compulsory information to be displayed be the GUI: location of the player within the level, part of the University where the level is located, amount of points the player has, and the location of obstacles. This GUI report will show how we fulfilled these requirements. We also made additions to the GUI wherever they were made necessary due to the changes and inclusion of new features made by our further implementation.

We added the location and objective information at the bottom of the screen (Fig 2.) Wall size was increased to make the rooms look more aesthetically pleasing as the doors no longer reached to the very edge of the window, making them looks less flat. It also provided an area to use to display information to the player without obstructing any part of the active GUI; i.e., the area of GUI where gameplay occurs. Therefore the information would be visible to the player when needed, but not distracting or annoying to the player when trying to play through the game. We also increased the size and changed the font and colour of the information text to make it more clear and aesthetically pleasing. It also subsequently 'fit into' the overall GUI more once changed, providing a more seamless effect.

The reason we chose to maintain such a simple design for our GUI is based upon the principle of allowing the player, and by extension gameplay, to be the most important factor in interaction with the system. By reducing the cognitive burden of the user, the game becomes more intuitive, more responsive to the desires and more satisfying to play [4]. This was achieved by allowing much of our GUI to be self-explanatory, even to a beginner user; closed door being impassable and open doors not, for example. We also changed the colour of enemy projectiles so that they were different from the player's. This made the difference more immediately apparent to the player; it also allowed them to deduce that they would be damaged by enemy projectiles, but not their own, with less cognitive burden. Therefore it fit with our intentions for the GUI.

We added a mini-map (Fig 3.) to the GUI in the top-right corner. This map demonstrates the layout of the randomly generated map to the player as they travel through it; this avoids making the game too easy by revealing the entire map to the player immediately, but also allows them to see where they have already been to avoid them getting lost, or not knowing where to backtrack to to access the next unexplored room. The mini-map is clearly visible but out of the way, again, of the gameplay area, and doorways are clearly indicated. Rooms are normally coloured white if explored and grey if unexplored, but are blue to indicate a 'boss room', again improving the usability of the GUI.

We added a health pickup resource, which was represented by a very commonly-recognisable red cross first aid bag. We also changed the obstacles (4.4) with touch-damage to be spikes rather than bins; these changes were made to improve the clarity during gameplay, thus improving usability. This was in response to our initial playthroughs after picking up the game, where some of our group members did not realise that the black bin obstacles caused damage on collision. Bomb pickups were also added, represented by a bomb sprite which, again, we considered to be easily recognisable. The information of how many bombs the player currently had was also needed to display to the player; we did this by inserting a picture of a bomb in the top right corner followed by how many the player currently has. This is a simple and obvious way of representing this value, while also being aesthetically pleasing and, again, not overlapping the area of gameplay.

More buttons were added to our main menu (Fig 5.) in accordance with new functionality, but was kept in the same style; additionally, a pause menu (Fig 6.) was added, similarly styled for coherence. This was a large change from the inherited GUI, which had an unclear pause screen, with only the word 'Pause' displayed in the middle of the screen in red text which did not stand out sufficiently. Our new pause menu also makes the game more usable as it provides the player with options to save and quit the game.

Our added functionality of the flight ability also lead to an addition to the GUI. The user needed a simple way to see how much flight they had left, and so a 'flight bar' was added; labelled with the word flight for clarity, this bar runs out as the player uses up their flight speed. A border was also added around the bar to avoid its colour clashing with the colour of the walls in the GUI, therefore making it always clearly visible.

The information we provided to the user through the GUI, as detailed above, allowed us to fulfil user requirements 16.1 through 20 (available https://github.com/teal-duck/teal-duck/raw/gh-pages/Assessment%202%20swap%20files/req2.pdf).

Bibliography

[1] E. McMillen and F. Himsl, *The Binding Of Isaac*. 2011.

[2] SteamSpy - All the data about Steam games, "The Binding of Isaac: Rebirth -", 2016. [Online]. Available: http://steamspy.com/app/250900. [Accessed: 09- Feb- 2016].

[3]D. Quintans, "Game UI By Example: A Crash Course in the Good and the Bad - Envato Tuts+ Game Development Tutorial", *Game Development Envato Tuts+*, 2013. [Online]. Available: http://gamedevelopment.tutsplus.com/tutorials/game-ui-by-example-a-crash-course-in-the-good-and-the-bad--gamedev-3943. [Accessed: 09- Feb- 2016].

[4] Fei Hu; Lixia Ji, "On the Peak-Experience in the Game GUI Design," in *Management of e-Commerce and e-Government, 2008. ICMECG '08. International Conference on* , vol., no., pp.204-207, 17-19 Oct. 2008

Appendix

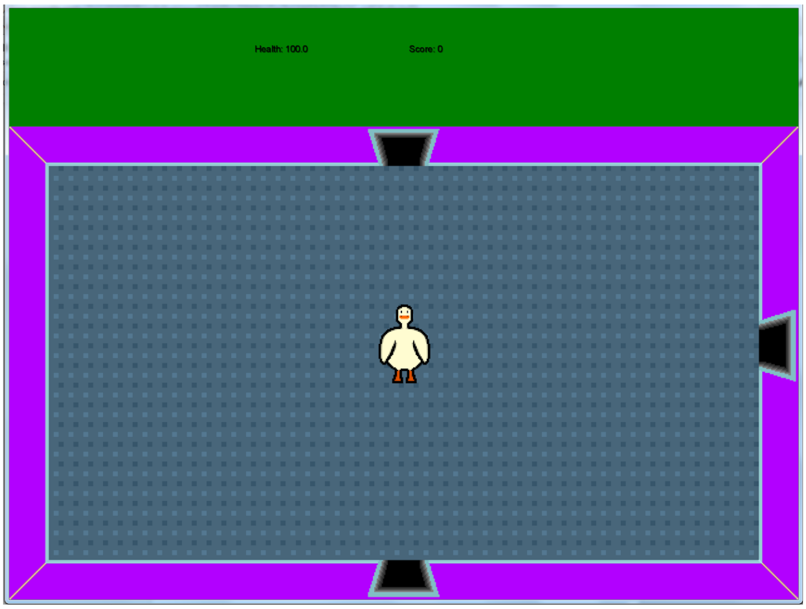
Health: 100.0
Score: 0

Fig 1. The inherited GUI

The green bar at the top is doesn't resize well with the game screen causing the room to be too squished vertically on a 16:9 window which is the most common aspect ratio size for modern screens.

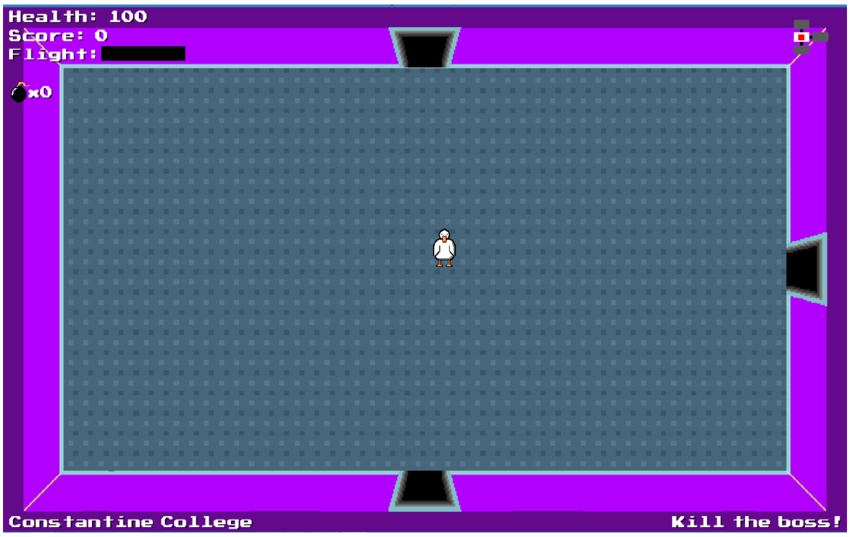
Health: 100
Score: 0
Flight:
x0
Constantine College
Kill the boss!

Fig 2. The new game screen GUI

Information is stored for the player at the edge of the frame which requires less eye travel the player. It also works better with the 16:9 aspect ratio. Health, score, flight bar and number of bombs are displayed on the top left. The minimap is on the top right, location is on the bottom left and the objective is on the bottom right.

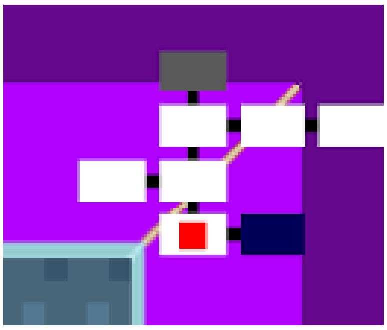

Fig 3. Mini-Map Closeup

The mini map helps the player track progress through the various levels. Colours are used to differentiate special rooms e.g. the boss room, from other normal rooms. The red square shows which room the player is currently on. New rooms on the map are added when the player discovers a path to thatroom.

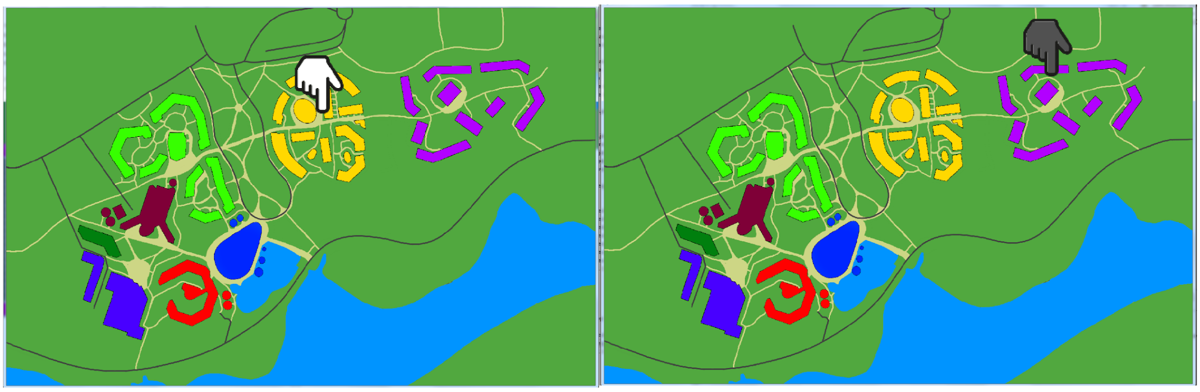

Fig 4. Level select screen

The hand now changes to a dark colour when a level has been completed and is white when the level hasn't been completed.

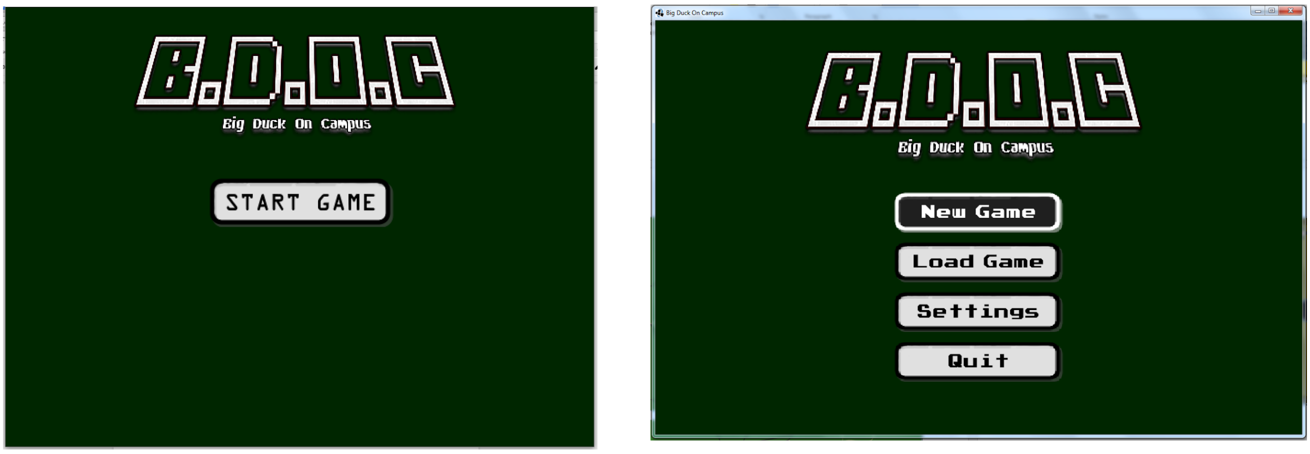

Fig 5. Main menu (old on left, new on right)

Before the main menu had a static non interactable image of a button but it now has a interactive button menu that will change colour depending on which button is currently selected.

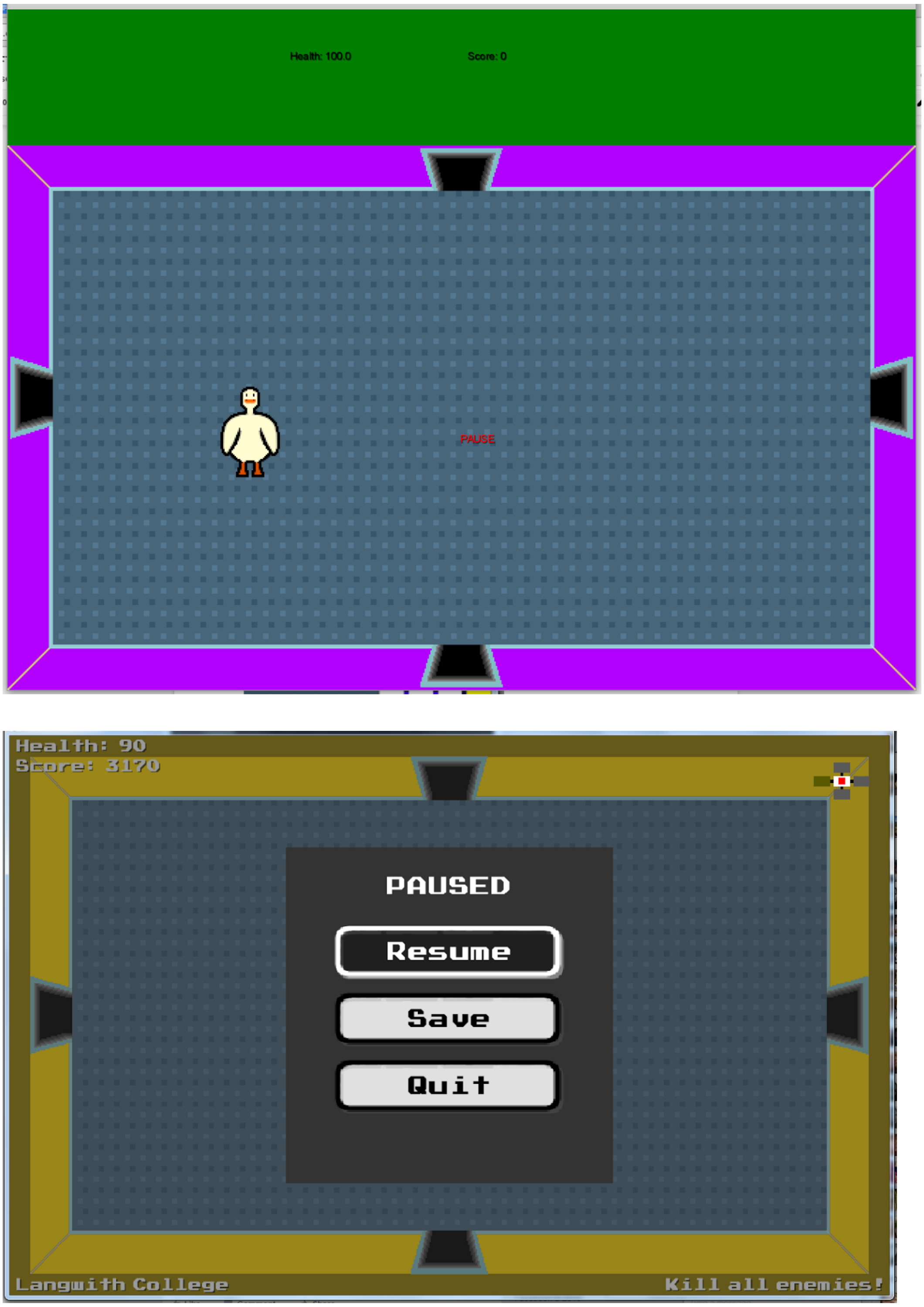

Fig 6. Pause screen (old on top, new on bottom)

The old pause screen only says “PAUSE” on the screen in the centre of the game screen when paused. It is very small so is not very clear that the game has paused a part from the fact that the game stops moving. The new screen now includes menu options.