Brandon Wright

CW2 C++ Documentation

Contents

[NOTE: make sure to look at 16. on the requirements sheet for the game plan 1](#_Toc130668727)

[Pre planning phase 1](#_Toc130668728)

[States 1](#_Toc130668729)

[GUI 1](#_Toc130668730)

[Player 1](#_Toc130668731)

[State pattern design 2](#_Toc130668732)

[FileIO (Input/output features) 2](#_Toc130668733)

[Displayable object features 2](#_Toc130668734)

[AI 4](#_Toc130668735)

[Collision detection 4](#_Toc130668736)

[Animations, foreground scrolling and zooming 7](#_Toc130668737)

[Game plan (LOOK AT 16) 8](#_Toc130668738)

# NOTE: make sure to look at 16. on the requirements sheet for the game plan

# Pre planning phase

## States

### GUI

**Start up - Splash screen**

**Pause - Stop all things updating**

**Running - Allow things to run**

**Win - Show win screen**

**Lose - Show losing screen**

### Player

Could use these for powerups?

### State pattern design

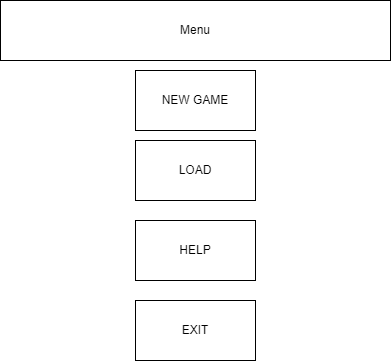
## FileIO (Input/output features)

**Load level data to construct levels with**

**Load saved data for where player left off (can be saved in pause menu or saved on close event), this data would contain things like player position, health, possible power up state, where they are in the level, ect. This would also load the level they are currently in ect.**

**Store statistics of player like steps taken, enemies killed and other stuff.**

**MENU in which you can load data**

****

## Displayable object features

**Player**

**Enemies**

**Projectiles**

**3 displayable object classes – Player- enemy – Projectile**

**These classes need a parent class that ISNT displayable object, the parent class WILL need to inherit from displayable objectDiagram

Description automatically generated**

**Objects must be able to be created and destroyed. Not all objects require to be destroyed. This has to be able to happen over time. Object can not interact with anything before its created and after its destroyed, it can only interact while its alive. NOTE:** You must not just re-create the entire object array contents to do this, although you could add to the end of it. (Please see the methods on DisplayableObjectContainer.) To meet these requirements, you MUST add to and remove from the displayable object array. NOTE NOTE: notify the program when you change the contents of the DisplayableObjectContainer. If you change the object array you need to tell anything using it that you did so, hence why this is in as a separate requirement – it can be trickier to get right. Please see the drawableObjectsChanged() method and investigate how it works to get this right

## AI

AI must be considered complex, minimum criteria is something that isn’t just randomly moving; however, this won’t be considered good marks. Enemy could move in based on a radius/distance to player. More marks would be making the AI think smartly, like cutting off the player. An algorithm for finding the shortest like djikstras or A\* would be extremely high marks here.

## Collision detection

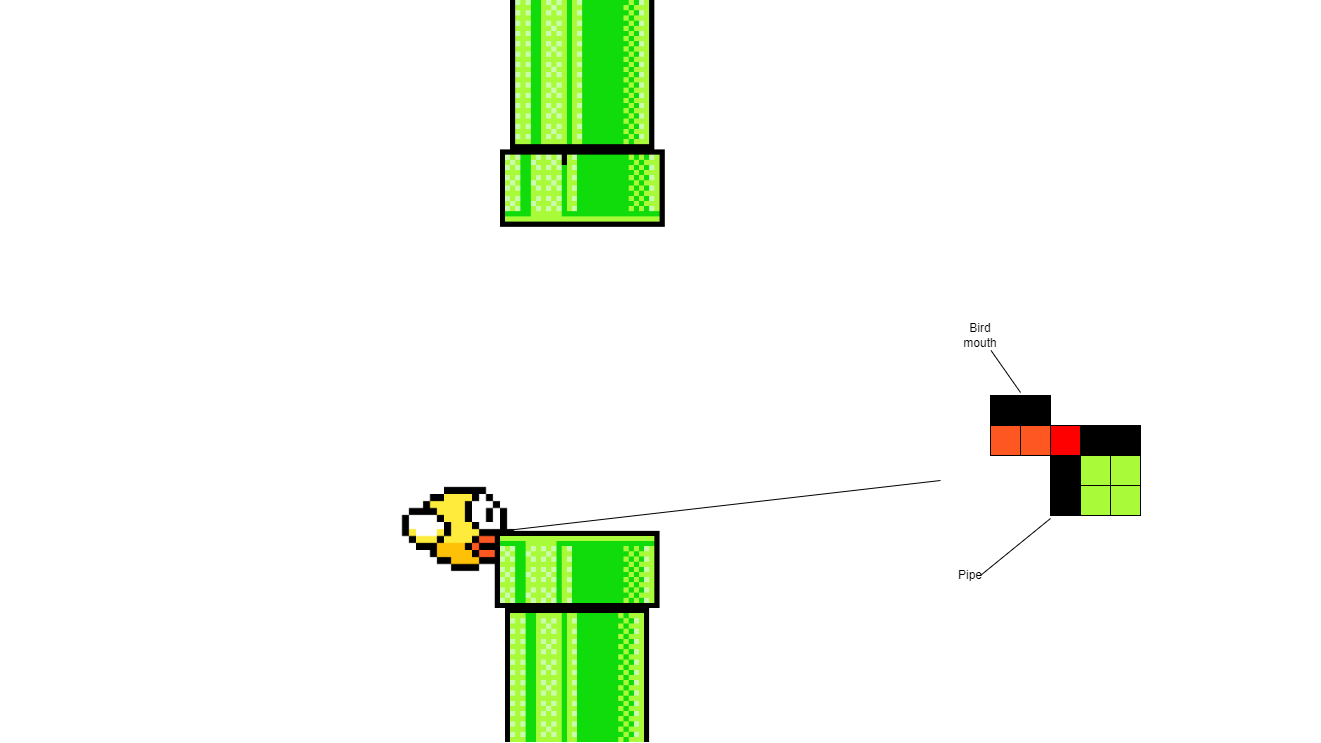
To ensure good collision detection, all objects should be CONVEX and not CONCAVE. I.e the angles of each point are ≤ 180°

<https://www.youtube.com/watch?v=Ap5eBYKlGDo>

<https://www.youtube.com/watch?v=7Ik2vowGcU0&list=TLPQMjUwMzIwMjMlHMt6fp6VkA>

For concave collisions, although it be more intensive on the application, you could use pixel overlapping detection, i.e is pixel A on the same XY and pixel B

An example of this collision:



## Animations, foreground scrolling and zooming

This is seen in 9.

**Scrolling background for the menu screen would be the most appropriate here.**

**Make it appear that the background has been animated on the menu**

**Animate moving objects.**

**This could easily be moving enemies and the player moving too. Also, on the main menu there should be some animating object too. This I think could easily be done by using a tilemap for a player and changing the tilemap x and y every n seconds**

**Image rotation**

**This could easily be done for some projectile attack like a keyboard flying through the air changing every so few centimetres (space on the screen)**

**Tile manager usage**

**This could easily be used to build the levels that exist in the game. Since I can use a tilemap, the tiles of the tilemap can easily be converted to a format of a tilemap**

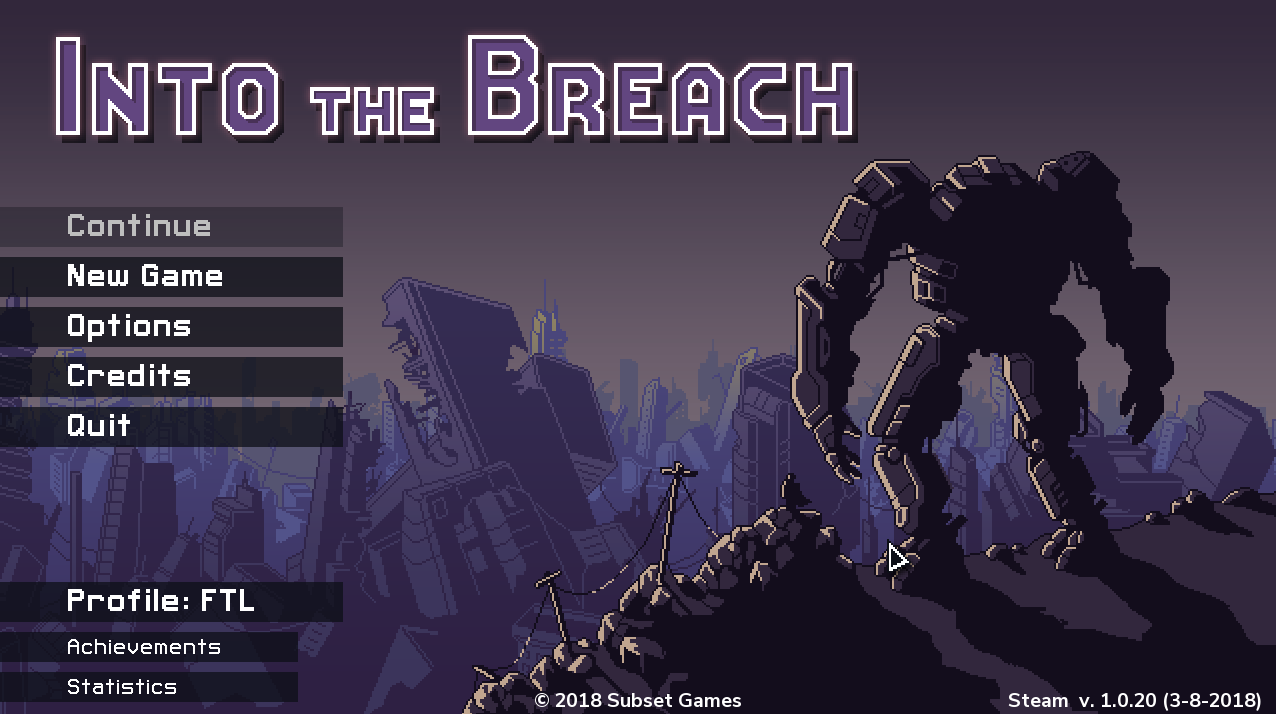
**Allow user to enter text which appears on the graphical display (max 2 marks):**

This would be done by allowing the user to enter their name when they click new game, this will attach their name to save data so they can see their name on the save data and be able to see that one save data is theirs and another might be their friend that was playing the game or something.

# Game plan (LOOK AT 16)

## MENU

### Base Screen

Menu idea: 

My version:



The triangle indicates what part of the menu is currently selected.

This character animates to fit a requirement (Have an animated or changing background by utilising multiple images (2 marks))

### New game

This will prompt the user to enter their name. Upon pressing ESC, they will be taken back to the main menu.



## Win screen

Have some scrolling shapes or images to meet the scrolling requirement.

## Lose screen

## TileMap

**Use this to render tiles. When all enemies are gone, change a tile to green and thatll send the player to the next level.**