GAME 352 - Final Project   65.5/70

Step 1:

Two options, you can continue with your game from assignment 1 pending my approval or you can create a new game.

If you create a new game, you will have to:

- ☐ Create a GDD for the game.

- ☐ Discuss the challenges of making this as a mobile game. - not discussed

- ☐ Outline the controls

- ☐ Explain how you will acquire the art assets - not explained

If your are continuing your game:

- ☐ Flesh out your original description making a proper GDD. ✔️✔️

- ☐ If you were previously working in a group explain how your game will differ from your partners. ✔️✔️

- ☐ Bonus: continue improving the title until it passes app review by apple.

Step 2:

Migrate your code to your own repo.

- ☐ Create your own repo on Github and add me as a collaborator. ✔️

  - it can be private or public.

  - just because a repo is public doesn't mean it isn't protected by copyright.

Step 3:

Incorporate the following into your game.

- ☐ abstraction of the games rules from the game scene ✔️

- ☐ particle effects ✔️

- ☐ collision detection (regular or using physics bodies) ✔️

- ☐ traditional sprite animations or character animations ✔️

- ☐ a menu using UIViewControllers & UIButtons ✔️

- ☐ actions (SKAction) ✔️

- ☐ abstraction ✔️

- ☐ implementation obfuscation ✔️ in kid factory

- ☐ a factory pattern ✔️

- ☐ a delegate pattern ✔️

- ☐ an observer pattern - not attempted

- ☐ capture user input ✔️

- ☐ at least 1 struct - not attempted

- ☐ at least 1 enum ✔️

- ☐ proper use of inheritance ✔️

- ☐ and polymorphism ✔️

- ☐ at least 1 protocol ✔️

- ☐ some form of networking eg. match making, leader board, cloud saves etc. - not attempted

---

Step 4:

Present your game to the class; presentations will take place Week 12 and Week 13.

- ☐ Discuss any problems you ran into making the game. ✔️

- ☐ Show some code you are particularly proud of and explain it to the class. ✔️

- ☐ Showcase the final product. ✔️

Step 5:

Code is due by week 13. I'll be deducting marks for poor use of the following:

- Efficiency ✔️
- Maintainability / Readability ✔️
- Structured / Architecture ✔️
- Follows Standards ✔️
- Extensible ✔️
- Completeness ✔️