S/W Eng. Group Project 1 - Arcade Game

# Introduction

The output of this assessment is for your team to implement an arcade style game in C++.

Project ground rules

* Teams of 4
* Use TDD
* Code should not be merged if there are failing tests
* Only code in master branch can be assessed
* Each developer(student) will be assessed on two stories that they have completed.
* Use github for source control (KP will create repos)
* Trunk based development
  + git rebase origin
  + Merge conflicts? => get team to help resolve conflicts
  + RUN ALL TESTS
  + Tests green? => git push
  + Tests red? => fix tests
* Use github project board (Basic Kanban) for story tracking
* Implemented in C++, graphical application.

# 

# Phase 0 - Team formation deadline 3rd December

1. Each team sends me an email with team member real name and github username
2. Anyone not in a team at this stage will be assigned a team

# Phase 1 - (no code) Deadline Fri 10th December

1. Set up team repo with these steps (read all steps before executing) :
   1. Find your team name [here](#_pjsv1qr8s8e9).
   2. Go to this Github classroom link (<https://classroom.github.com/a/G5M4JoVs>).
   3. If you see your team name, join it. Otherwise, create a new team with the EXACT team-name you were given (no additional spaces, punctuation, uppercase/lower case the same)
2. Research report (team document)
   1. Introduction
      1. Who are we
      2. What is the purpose of this document
      3. Why are we doing this project
   2. Game description
   3. Game rules
3. Choose a Minimal Viable Product (MVP) (team document)
   1. Describe the simplest version enjoyable to the user. List the features of the MVP
   2. Define "out-of-scope"
   3. List of stories needed to complete the MVP
      1. Large stories(epics) should be broken into smaller stories
      2. Aim for at least 4 **equally** sized stories for the MVP
      3. Stories need detailed acceptance criteria
   4. Stories should be written up as cards in Github project board in ToDo column (Ken will set up the Github project boards) . See this [example story](https://github.com/kenpower/BoggleExample/issues/1) from a text based game
   5. Each story must contain a list of acceptance criteria (with checkboxes)
   6. Each story must contain a list of implementation tasks (with checkboxes). Tasks that will be done in other stores don't need to be listed.
4. These documents must be saved as markdown files (research\_report.md & mvp.md) at the top level of your team github repo. [Markdown guide](https://guides.github.com/features/mastering-markdown/)

# 

# Phase 2 - Development - Deadline (early-mid Feb, exact date TBA)

1. Team decides which 4 stories from the MPV will give the most value, and work on these first. (At least **4 more stories** will be needed during development)
2. Each student will take ownership[[1]](#footnote-0) of a minimum of 2 stories.
3. Stories will be implemented by pair-programming
4. A student cannot begin work on a second story until the first one is DONE.
5. Stories CANNOT be worked on unless in the "Ready" Column

Ken is product owner (PO)

* PO Decides if a story is ready to develop
  + Story approval will happen in the lab session. If a story needs further analysis then will have to wait for another week for approval.
  + Story has technical tasks listed
  + All unknowns and assumptions are clarified
  + Dependencies on other stories are known (try to keep dependencies to a minimum)
  + ~~Story is estimated by the team in story points~~
* Team developers can move stories from "Ready" to "In progress". **But each developer can only have one story in "In progress" at a time.**
* TEAM Decides if a story is done
  + Done if story feature is working
  + Done if sufficient test coverage
  + Done if has been merged
  + Done if ready to grade (stories not done will not be graded)
* Notes
  + Stories in done column can be graded at any time

# 

# Phase 3 - Reflection and Reporting Deadline TBA (End Feb?)

Each person chooses 2 stories that they "own". Everyone must choose different stories.

For each of the two stories you will add an issue comment (similar to the comment in this [story](https://github.com/kenpower/BoggleExample/issues/1), copy that comment's format).

The comment should contain a link to every test that was created for that story. Use the "Copy permalink" to grab a link to each test.



The issue comment must contain a link and embedded image of a class diagram showing the classes and relationships involved in this story (use <http://www.plantuml.com/plantuml/uml/> to generate a link to the diagram).

The issue comment must contain a link and embedded image of a sequence diagram showing the messages and collaborators involved in this story (use <http://www.plantuml.com/plantuml/uml/> to generate a link to the diagram).

Write a **two** brief descriptions (<100 words)

1. Describe how the story was implemented to follow SOLID principles (you can focus on either the SRP or OCP)
2. Describe how the story could be implemented to better follow SOLID principles (you can focus on either the SRP or OCP)

Finally add a **reflection section,** Where youthink about the design of the story, pairing, tests, implementation, lab session etc.and explain what things you might do differently if you were doing it again.

Also try and identify positive aspects that you will do in future stories

# Grading

| 10% | Attendance at Project sessions |  |  |
| --- | --- | --- | --- |
| 20% | Team Research Report |  |  |
| 10% | MVP Design Doc |  |  |
| 30% | Story 1 |  |  |
| 30% | Story 2 |  |  |
|  |  |  |  |

Story graded on

* Appropriate encapsulation
* TDD
* Use of Humble object where needed
* Clean code/ SOLID principles
* Ability to onboard new developer
* UML class diagrams for the story

# Teams 2021/22

|  |  | TeamName |
| --- | --- | --- |
| Kmitas | Lukas | Arcade2021A |
| Kuzmanovic | Patrik | Arcade2021A |
| Mc Donagh Rollo | Conor | Arcade2021A |
| Rondoleanu | Alin | Arcade2021A |
| Boyko | Yaroslav | Arcade2021B |
| Daly | Kevin | Arcade2021B |
| O Keeffe | Joseph | Arcade2021B |
| Price | Benjamin | Arcade2021B |
| Fogarty | Stephen | Arcade2021D |
| Lawlor | Emmett | Arcade2021C |
| Sumkauskas | Lucas | Arcade2021C |
| Xiong | Haotai | Arcade2021D |

1. Your code quality mark will be based on the code for the stories you 'own' [↑](#footnote-ref-0)