Team name: Team 28
Game name: Temporary

Team ideas:

- 28 studios
- Pufferfish Games
- Fugu
- 28th element - nickel
- Dime
- 28 in ascii - file separator control char
- 0x28 in ascii

Game name ideas:

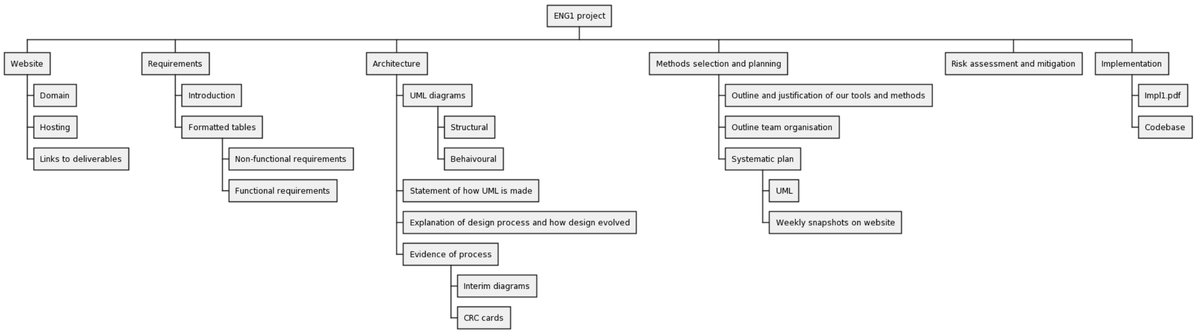
- Heslington Hustle :(

Meetings: whenever needed
Every monday:

To do:

- Name team
- What platforms do we want to target - desktop definitely, web browser, android, and iOS all possible
- Get a free domain name for the website from github students. Need to be sure we're happy with the name e.g. team28.tech. Discuss at meeting
- Set up deliverable documents and zipping
    - Lectures recommended these:
    - http://www.projectlibre.com/
    - https://plantuml.com/
- Choose a license for game and website
- Choose website theme
- Formatter and linter - checkstyle, google code format, spotbugs, intellij?

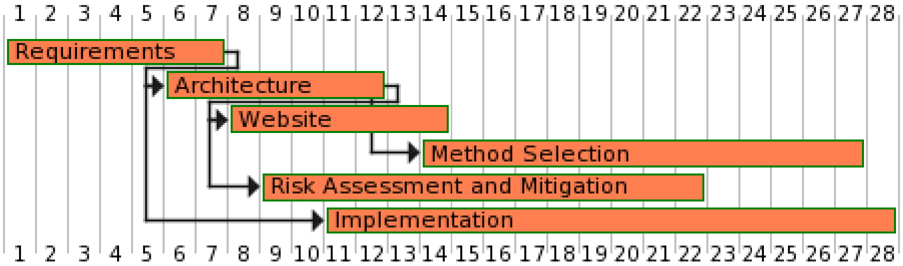
These should go into the Plan1 document
These should be snapshotted weekly to be included on the website as evidence of our planing process
plantuml task diagram


ENG1 project
Website
Domain
Hosting
Links to deliverables
Requirements
Introduction
Formatted tables
Non-functional requirements
Functional requirements
Architecture
UML diagrams
Structural
Behaivoural
Statement of how UML is made
Explanation of design process and how design evolved
Evidence of process
Interim diagrams
CRC cards
Methods selection and planning
Outline and justification of our tools and methods
Outline team organisation
Systematic plan
UML
Weekly snapshots on website
Risk assessment and mitigation
Implementation
Impl1.pdf
Codebase

plantuml gantt chart

Basic Architecture - Alex Spencer / Rhys

Risk (5a 5b) - Alex Firth / Jamie Rogers
Code - Joel / Maciek
Method Selection and planning (4a,4b, 4c) - Muhammed



Game engine options

- Libgdx is the best one
- Fxgl also exists
- Slick2d seems dead?
- Java no libraries standard library
- JWJGL is the low level framework the other engines use
- Jmonkeyengine is 3d

Libgdx selected!

- Wiki: https://libgdx.com/wiki/
- Javadoc: https://javadoc.io/doc/com.badlogicgames.gdx

How to use project:

- Gradle command line
- Intellij (installed on uni computers) or android studio (version of intellij with android build help, recommended by libgdk)
- Eclipse (recommended by course, bad)
- Vscode extensions

Java documentation: https://docs.oracle.com/en/java/javase/11/docs/api/index.html

Website

- GitHub pages docs: https://docs.github.com/en/pages
- Jekyll docs: https://jekyllrb.com/docs/

I'm experienced with both Git and GitHub Pages feel free to ask me about either on the discord - Joel

Github organisation: https://github.com/eng1team28

For sharing assets, the brief etc.
Google drive folder: ENG1 Team 28

- Req1

- Arch1
- Plan1
- Risk1
- Impl1

Deliverable Max mark Page limit File name and format
1. Website (submit only the URL) 3 — url1.txt
2. Requirements 20 1 + 3 Req1.pdf
3. Architecture 22 6 Arch1.pdf
4. Method selection and planning 10 2 + 1 + 2 Plan1.pdf
5. Risk assessment and mitigation 10 1 + 3 Risk1.pdf
6. Implementation 25 1 Impl1.pdf
+ Code
+ Executable JAR

Inspiration - stick rpg
Fishing minigame
For minigames - bullet hell undertale style?

Website (3 marks) - End of W3/4
Requirements (20) - End of W2
Architecture (22) - End of W3
Method Selection and Planning (10) - End of W4
Risk Assessment and Mitigation (10) - End of W4
Implementation (25) - End of W4

Constraints

- Java …
- Appropriately-licensed 3rd-party libraries, tools, and assets
- Accommodate stakeholder

Assessment 1 Requirements (from Product Brief)

1. Game Duration: The game will last for 7 days, with each day ending when the player goes to sleep. The game ends after the 7th day.
2. Game Avatar: The player will control an avatar representing a second-year computer science student. The avatar can move around the map and interact with various objects.
3. Control Scheme:
   - Moving the avatar around the map.
   - Interacting with what is in front of the avatar.
4. Activity Locations: Implement one of each interaction/activity location:

    - One place to sleep (e.g., a student accommodation building).
    - One place to study (e.g., the CS building or the university library).
    - One place to eat (e.g., the Piazza building or restaurants in town).
    - Three recreational activities (e.g., going to the sports centre, spending time at the lake feeding ducks, etc.).
5. Map: The game must contain a map for the avatar to move around in. This map should contain at least some of the Heslington East campus (CS building, student accommodation building, Piazza building, etc.).
6. Activity Tracker: Implement a tracker for the 7 days the game lasts, with a mechanism to move to the next day.(sleeping?) The tracker should also include a simple counter denoting how many of each activity has been performed so far.
7. Game Objective: The primary objective of the game is to ensure the avatar has studied enough for exams, taken some time for themselves, and had enough rest every day.
8. Resource Consumption: Each activity will consume a varying amount of the avatar’s two available resources: time and energy (X%). Each day contains 16 hours, and each day the avatar has an energy bar/capacity.

Questions for stakeholder

1. Specific platform? Desktop only? -
    a. answer: Windows/Mac/Linux
    b. - Window Size Scaling - Discrete options, large screens possibly
    c. - NOT Mobile/HTML
    d. Real target customer is prospective students at an open day
2. Give SSON pls
    a. “You are to build a single-player game that requires managing the activities performed by a second year computer science student the final week before their exams are to begin.” - from brief? Or something else
    b. ^^^^ “More or less what i want” - Tommy
3. Requirements and Architecture for whole assessment or just assessment 1?
    a. Just Assessment 1
4. Additional non-functional requirements? E.g. timings and usability
    a. “Enjoyable as possible” - requirement could be 7/10 players enjoyed the game
5. Game ? Should users be encouraged to replay for high scores/achievements?
    a. High score leaderboard - “not a feature for assessment 1, maybe assessment 2”
6. Difficulty of the game? Different modes e.g Easy, Normal, Hard, Hardcore?
    a. Fairly easy - 5/10 minutes playtime
7. What's most important graphics/gameplay?
    a. Usability/gameplay is important

   b. Gameplay - potential to add animations if necessary
   c. “Make the game as enjoyable as possible, how can you design things and make it more enjoyable”
   d. Potential for music/background music - not required - make sure its mutable if implemented
   e. Sound effects

8. Should the game have additional gameplay for entertainment that doesn't directly impact the main ‘story’ for example having other minigames/sidequests?
   a. Possibly Assessment 2
9. Game file size limitations? (i.e. any hardware limitations?)
   a. Answer: Submittable on the teaching portal - up to 512MB
   b. Quick download
   c. Usability
   d. Minimum hardware requirements - not a concern for such a simple game, our discretion
10. Size of map preference?
   a. None, should be small (game is short)
11. Licence - our discretion?
   a. More about asset licences
   b. Up to us
12. NPCs?
   a. Not necessary
   b. Potential side feature
13. Game style? E.g pixel art
   a. Our choice - no specific request from him
14. Any other games we can take inspiration from?
   a. Stick RPG
15. Ways to level up? E.g mini-games/tasks to do
   a. No mini-games…
   b. Essentially cutscenes/skips
   c. More assessment 2 based
16. Tutorial?
   a. Quick and easy (1min?)
   b. “If the tutorial is too long, the player will lose interest and not play the rest of the game”

Use cases

- **Name:** User
- **Precondition:** User downloads the game/loads in web browser
- **Trigger:** User begins playing the game
- **Main Success Scenario:**
    - 1. User progresses through the game
    - 2. Completes activities and tasks
    - 3. Levels up and grows their attributes

    - 4. Passes the exam
- **Secondary scenarios:**
    - 1.1. User does not find the game enjoyable and gives up
    - 1.2 User fails the exam
    - 2.1 User fails to complete the tasks, potentially too difficult
    - 4.1 Fails the exam
- **Success Postcondition:**
    - User replays the game after finishing on a possible high difficulty?
- **Minimal Postcondition**
    - User finishes the game or doesn't finish and doesn't replay.