Game Design Document

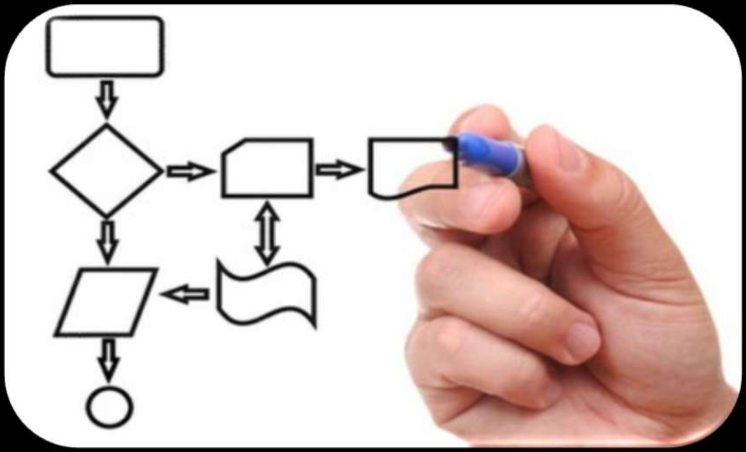
CSC404 Supplemental Notes

The Design Document

- The design document describes all aspects of your game, without actually creating code.
  - Acts as a contract between designer and client.
  - Also acts as a blueprint for future developers.

From Software Engineering

- Based on the idea that computer scientists should create software the way architects create buildings.



  - Understanding of what the software will do.
  - Analysis of the necessary software components.
  - Planning of the development of each component.
  - Coordination of the team and the development.

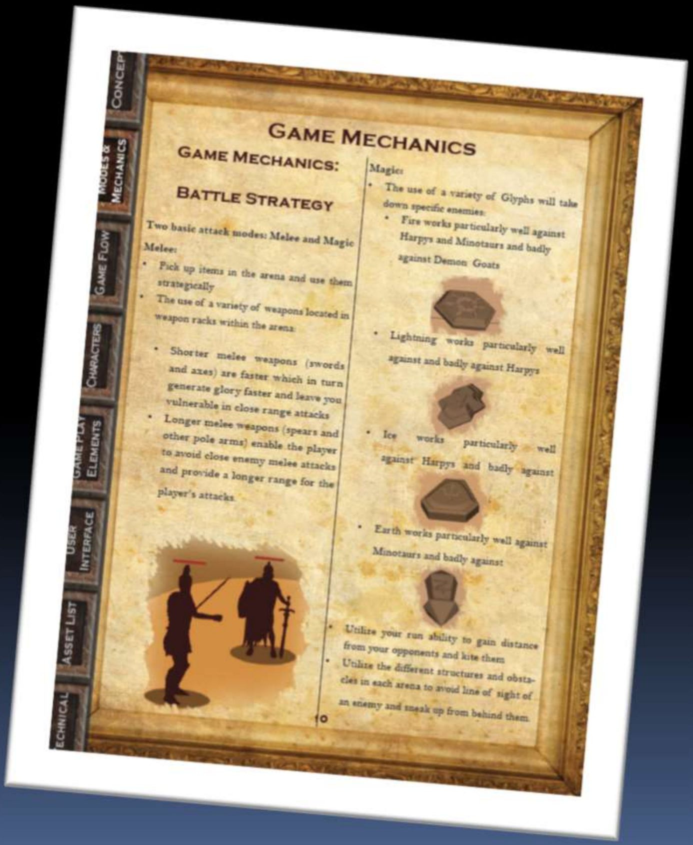
Design Document (background)

- Software Design Documents (SDDs) in the game industry often outline the following:
  - High level summary.
  - Background on project domain (definition of terms, etc).
  - The game requirements, and how to achieve them.
  - Constraints (both technical and non-technical).
  - Development procedures and coding guidelines.
  - Languages and tools that will be used.
  - Definitions of variables and a description of their usage
  - Logical structure and logical processing steps.
  - Error, alarm and warning messages.
  - Performance & reliability.

The Game Design Document

Design Doc Requirements

- Your design document must have:
  - A description of your game
  - A description of your team
  - Gameplay description
  - Character designs
  - Level designs
  - Music & sound design
  - Control flow diagrams
  - Development timeline



GAME MECHANICS

GAME MECHANICS:

BATTLE STRATEGY

Two basic attack modes: Melee and Magic

Melee:

- Pick up items in the arena and use them strategically
- The use of a variety of weapons located in weapon racks within the arena:
  - Shorter melee weapons (swords and axes) are faster which in turn generate glory faster and leave you vulnerable in close range attacks
  - Longer melee weapons (spears and other pole arms) enable the player to avoid close enemy melee attacks and provide a longer range for the player's attacks.

Magic:

- The use of a variety of Glyphs will take down specific enemies:
  - Fire works particularly well against Harpys and Minotaurs and badly against Demon Goats
  - Lightning works particularly well against and badly against Harpys
  - Ice works particularly well against Harpys and badly against
  - Earth works particularly well against Minotaurs and badly against
- Utilize your run ability to gain distance from your opponents and kite them
- Utilize the different structures and obstacles in each arena to avoid line of sight of an enemy and sneak up from behind them.

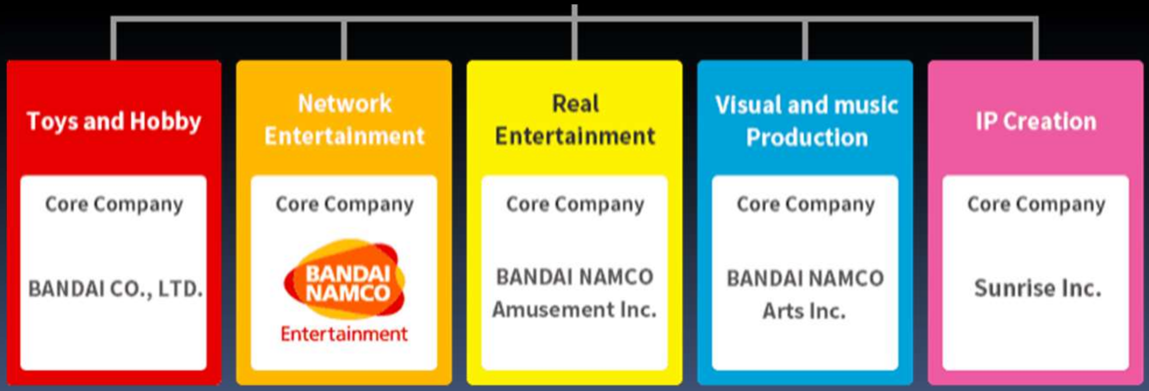
10

Game Description

- Start with the pitch of your current game.
- Elaborate with more details:
  - Aesthetics (reference samples > mood boards)
  - Dynamics (player experience)
  - Controls (player actions, interface devices, etc)
  - Inspiration (reference specific mechanics from specific games)

Team Description

- For each team member, state the strengths and the areas of responsibility.
  - Remember that these roles indicate who takes the lead for certain game elements.
  - The entire team must work on the entire game!



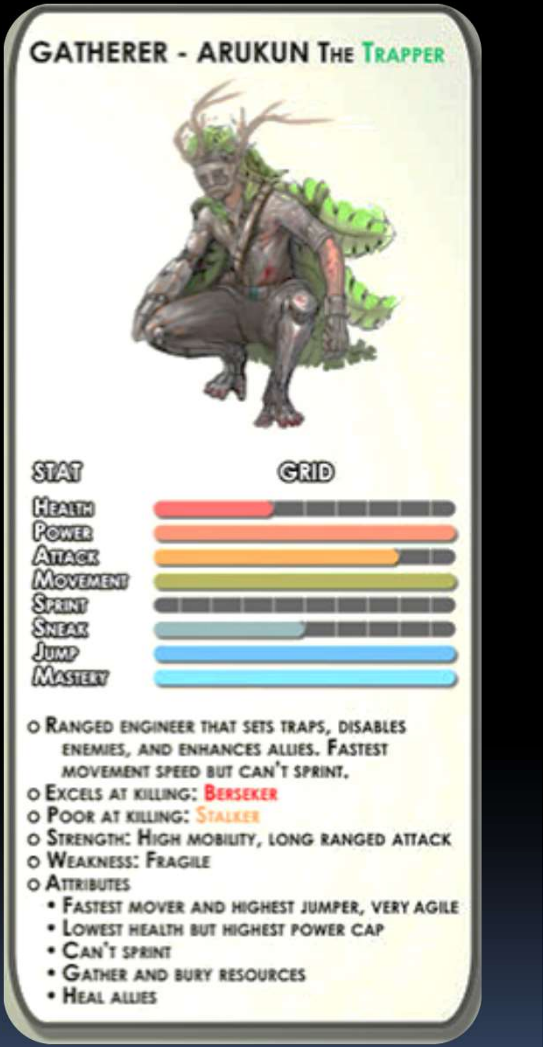
Gameplay Description

- How is the player meant to play the game?
  - Overall game arc
  - Player goals, as tied to the mechanic
    - Find the keys, light the cauldrons, etc.
  - Mechanism that the player uses to achieve these goals.
  - What are the main fun elements?
  - What else will make this fun?

- Illustrations help tell this story.

Character Designs

- Player character (detailed)
- Non-player characters (less detailed)
- Character development systems (stats, powerups)
- Other level assets
  - Items that are key to the gameplay.
- More than just sketches, please!
  - Prove that you can make these.



GATHERER - ARUKUN The Trapper

- Ranged engineer that sets traps, disables enemies, and enhances allies. Fastest movement speed but can't sprint.
- Excels at killing: Berseker
- Poor at killing: Stalker
- Strength: High mobility, long ranged attack
- Weakness: Fragile
- Attributes
  - Fastest mover and highest jumper, very agile
  - Lowest health but highest power cap
  - Can't sprint
  - Gather and bury resources
  - Heal allies

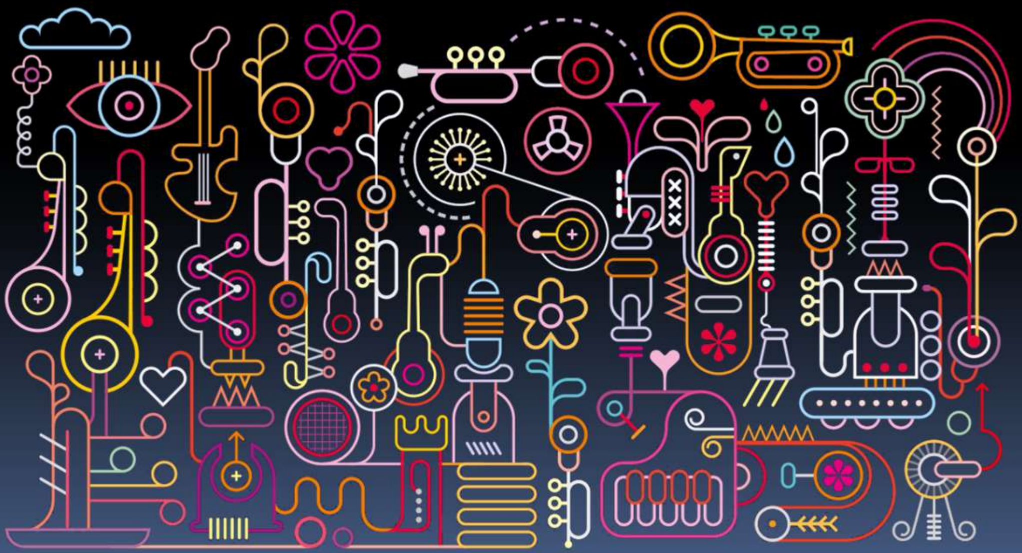
Level Design

- Invest time here.
  - Overall level progression.
  - Show the purpose of each level, and how these levels introduce skills and build up the experience.
  - Show the thought and polish you've put into your level designs.
    - Be sure you can make them!

5-minute paper sketches do not inspire confidence.

Music & Sound Design

- What are the music and sound elements that your game will feature?
  - If you're not all in constant contact, this is a good time to set up an agreed mode of communication.



Control Flow Diagrams

- Do you have a levelling system?
  - Show how the progression works.
- Do you have a puzzle to solve?
  - Show how the player needs to solve it.
- Do you manage a complex set of inputs?
  - Show how these inputs come together.

Development Timeline

- Assign somebody (usually team lead) to coordinate development from the GDD to the playable prototype.
  - Set up SMART goals for everybody on the team.
  - Have the team commit to these goals.
- Always have something playable.
  - Take lessons from the game jams. Don't assemble all the parts just before the next presentation!
- Work on things that are easy and important first, with a priority on important.

General GDD considerations

- The #1 item when creating a document:

  Do not approach any document like a checklist.

  - The list items on the previous page are a reminder of what you need, not a linear set of instructions to follow.
  - There has to be a sense of flow and cohesion.
- The #2 item when creating a document:

  Create the document with your reader in mind.

  - Should answer all questions on how to create the game.

Design Doc Presentations

Presentation considerations

- Biggest issue from past presentations:

Too much telling. Not enough showing.

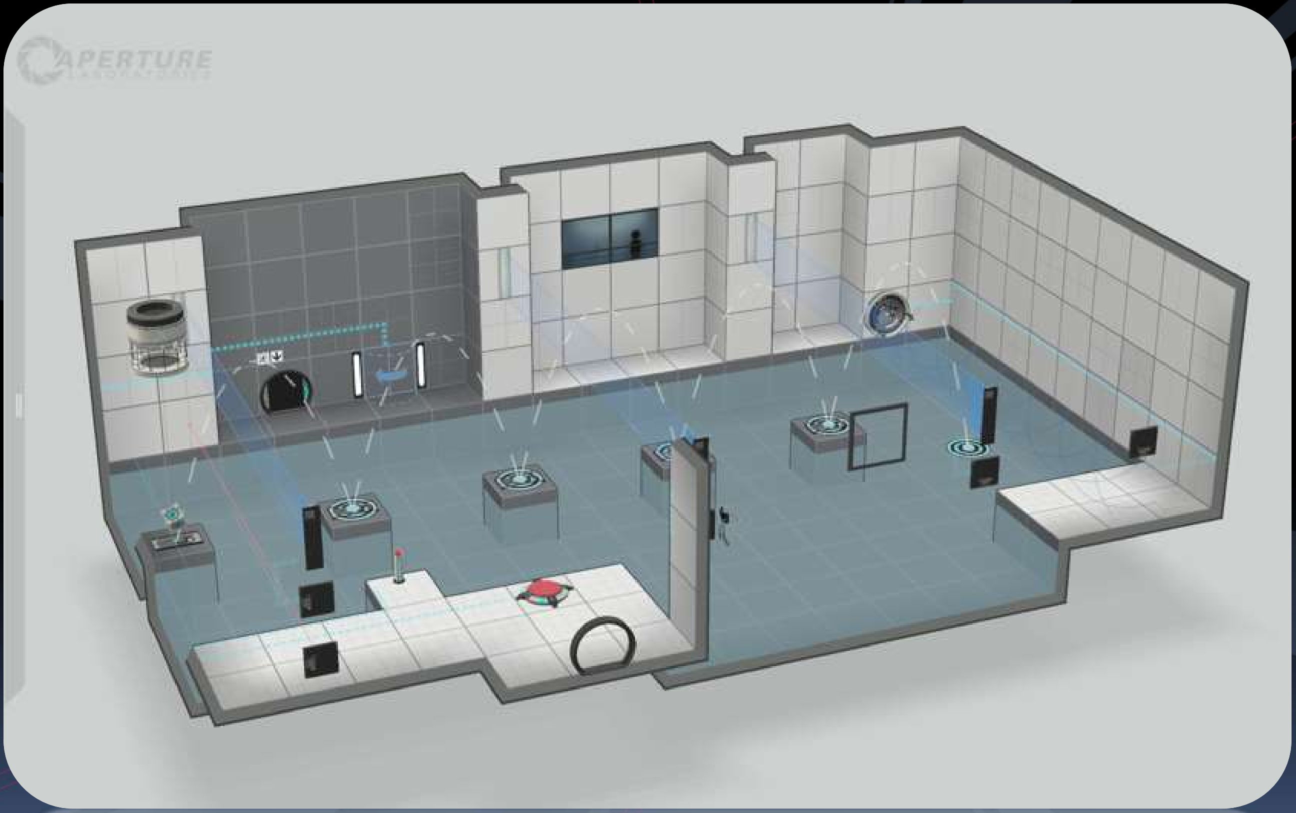
- Examples:
  - No level designs, just level descriptions ☹
  - Reading blocks of text off the slides ☺
  - No models or tech demos, just “inspirations” ☹

Present the blueprint, not just an advanced pitch.

Other GDD considerations

- Organization is very important.
  - Break down your game into parts, and create a section for each part in your document.
- Be both general and specific.
  - Outline motivations as well as details.
  - Avoid hand-waving.
- A picture is worth a thousand words.
  - Include diagrams, sketches, screenshots and/or storyboards.
- Every design document is different.

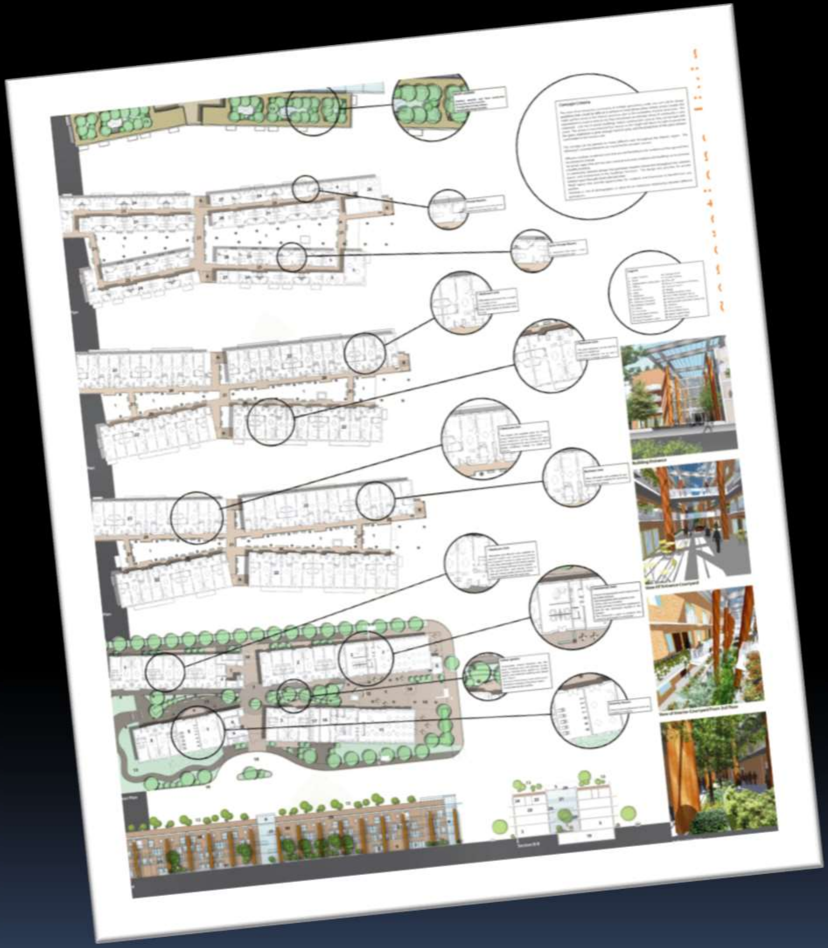
Level Design Exercise



Breaking down your levels

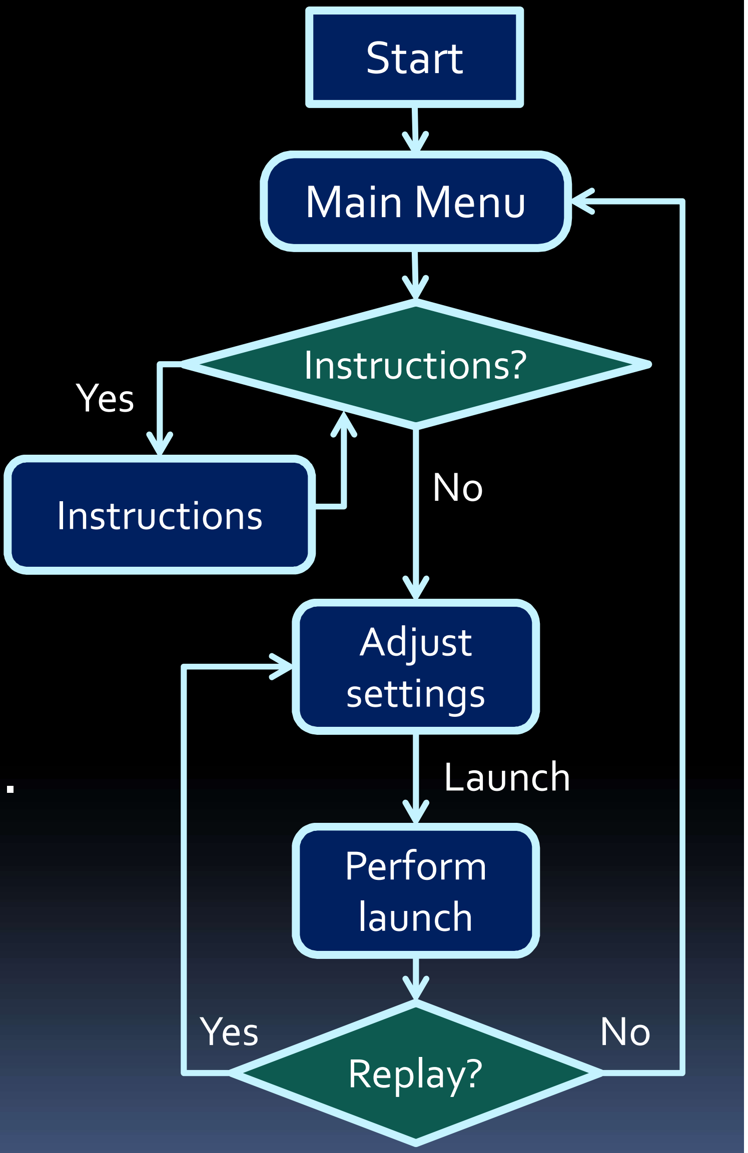
- Step #1 (to do alone):
  - Describe the features of an easy level, a hard level, and a medium difficulty level.
- Step #2 (with somebody from another group):
  - Describe the skills needed to perform your medium difficulty level.
  - *"The player needs to know how to move around the level, how to jump, how to pick up and throw crates, and how to combine these together to open the door by placing the crate on the floor switch."*
- Step #3 (with the people in your group):
  - Compare the lists of skills that the player needs to do, and order levels based on those skills.

Example Game Design Presentation



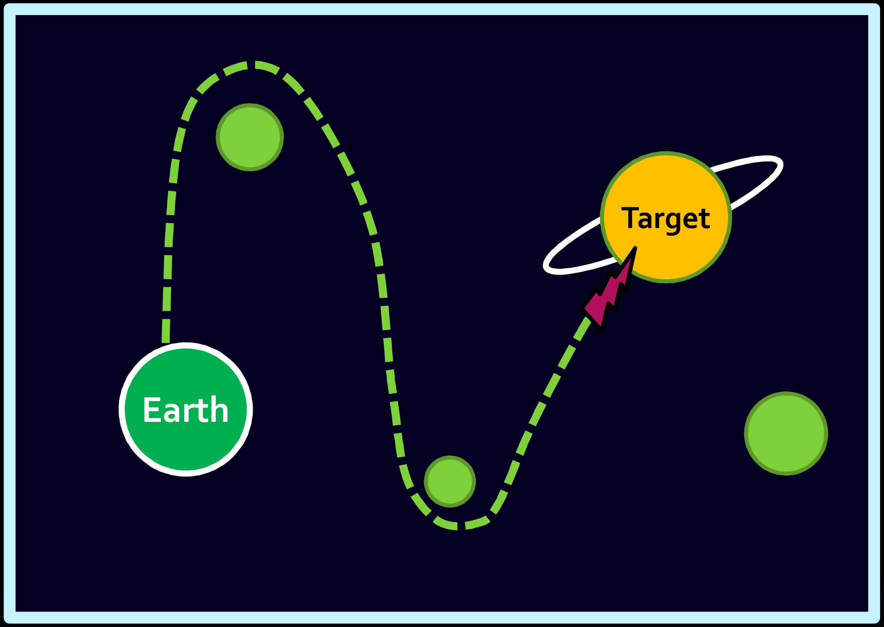
Rocket Launcher

- Game starts with general game options:
  - Instructions & controls.
- Gameplay is broken down into the following stages:
  - Adjust rocket speed and angle.
  - Perform launch simulation.
  - Player can choose to replay or return to main menu.



Level Design

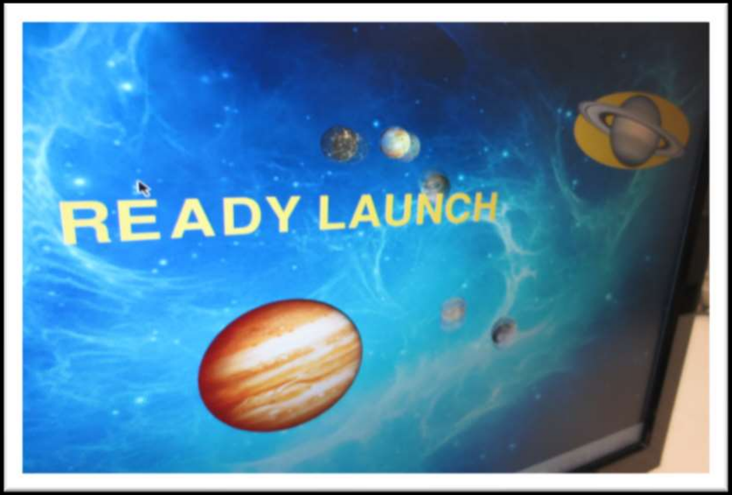
- Each level is made up of a starting planet (Earth), a highlighted target planet, and several intermediate planets.



- Players are given a chance to observe the planets' movements before launching.

Gameplay Outline

- 10 levels total.
- When players click on "Launch", the rocket takes off at the specified angle and speed.



- As the trajectory approaches the planets in the field, the movement is affected by the equation for universal gravitation:

$$F_g = G\frac{m_1 m_2}{d^2}$$

Score breakdown

- Score is awarded as a sum of time and proximity factors:
  - Each 50 ms of travel time adds 1 point to the overall score.
  - Traveling within 1000 km of a planet increases score by 10 for every 50 ms spent close to planet.
- Level is cleared if target planet is reached, and score is over 100 points.
  - Stars are awarded for every 25 points above 100, to a maximum of three stars.