VR MOTION PLATFORM DEVELOPMENT COURSES

Unit 2

Unity Game Design Document

TOPICS

- Game Design
  - Game Design Document, Game Level Design
  - Game Plan, Design, Implement Game
  - Sound, Graphics Design, Game Story
- Game Scripting and Animation
- Dialogue System
  - Live2D
  - Fungus

GAME DESIGN TOPICS

- Structure of Game Design Document and Other Considerations.
- Character and Object Design.
- User Interface Design.
- Level Design.
- Types of Prototype for Games.
- Criteria of Good Prototyping.

GAME DESIGN DOCUMENT

- We have already introduced the one-page game proposal.
- This is the simplest form of design document, and is a good starting point to test interest.
- Before extending a game design, you should consider the different objectives for the game design document, e.g.
  - **To describe the game for development**
  - **As part of a project management plan**
  - To help to secure a publisher deal (if there is not yet a contract between publisher and developer)
  - As part of the marketing materials for consumers

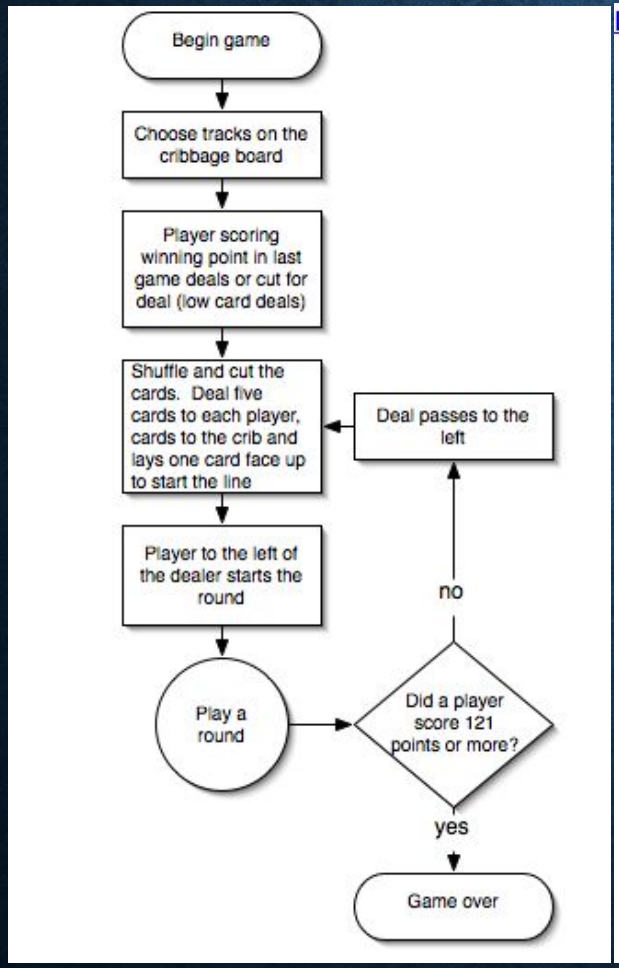
GAME DESIGN DOCUMENT STRUCTURE

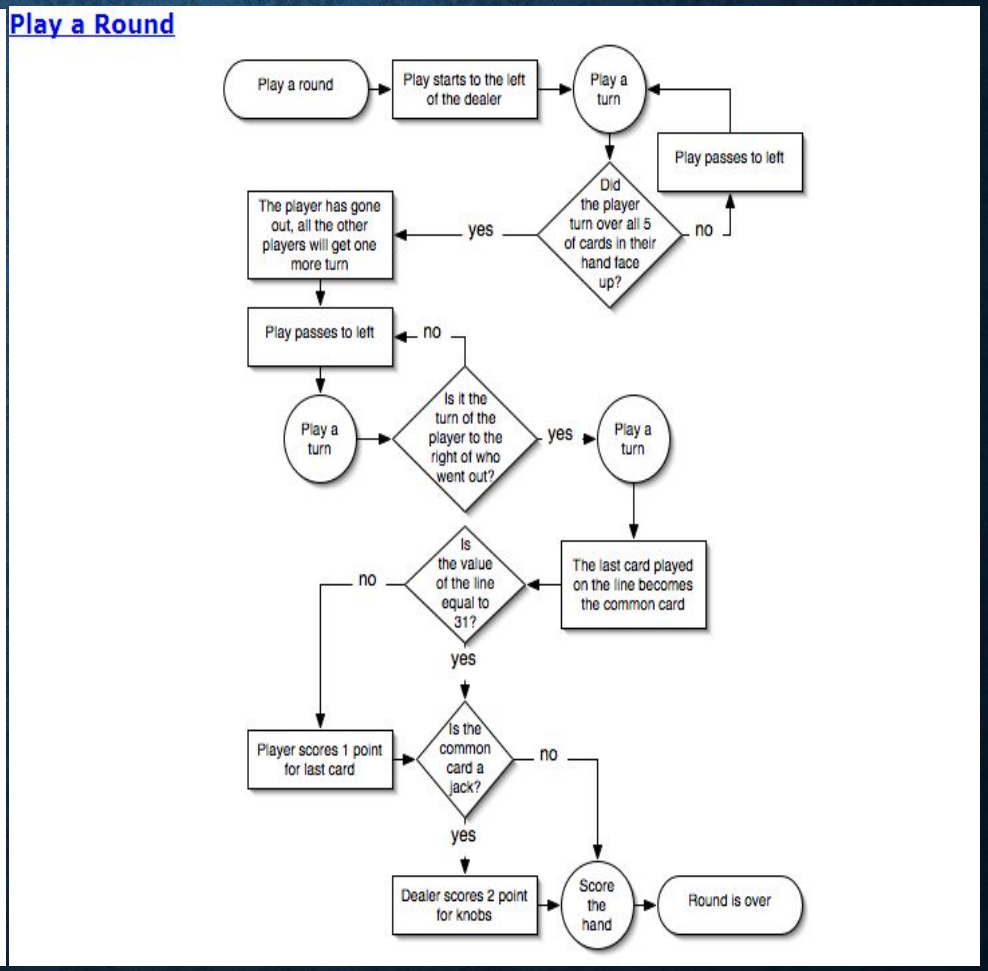
- The game design documents prepared by different developers **may vary in the content and structure** because different developers may have different emphasis in different aspects in different games.
- A typical game design document may include:
  - **Introduction**
    - A summary of the rest of the document
    - After writing the rest of the document, pick up the highlights to give a brief overview in this section.
    - State the goals of the game clearly
    - One important function is to let those who do not have time to read the entire document will at least get a reasonably accurate overview of what the game is about.

GAME DESIGN DOCUMENT STRUCTURE (CONT')

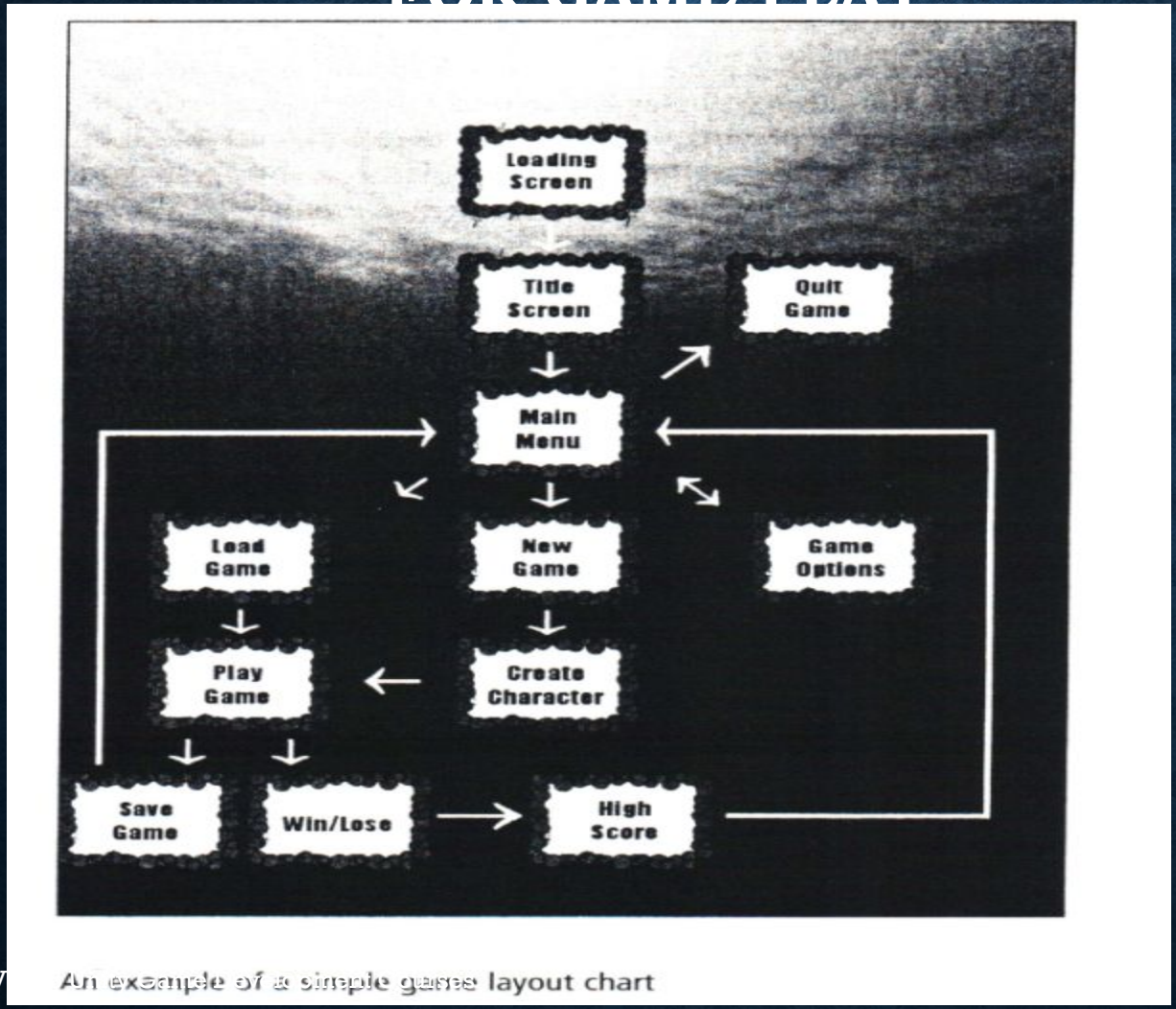
- **Game Overview**
  - The game overview gives the background story and the basic game play description.
  - A flow chart to describe the game play may be useful.
  - It is derived from the proposal document, but there should be more detail.
  - Avoid information overflow, refer to other sections in the document where appropriate
    - e.g. When introducing a character in the overview, refer to the character section for a full description
    - Don't include complex rules or tables etc. Such detail belongs to later parts in the document.
  - May include the game requirements here

GAME DESIGN DOCUMENT STRUCTURE (CONT') – A SAMPLE FLOW CHART FOR GAME PLAY





GAME DESIGN DOCUMENT STRUCTURE (CONT') – ANOTHER SAMPLE FLOW CHART FOR GAME PLAY



An example of a simple game layout chart

INV

GAME DESIGN DOCUMENT STRUCTURE (CONT')

- **The overview should answer these questions:**
  - What is the game about?
  - Where and when does the game take place?
  - What do I control as a character?
  - How many characters do I control?
  - What is my character's objective (mission)?
  - How will my character achieve the objective?
  - What threats does my character face?
  - What competition does my character face?
  - What help can my character find?

GAME DESIGN DOCUMENT STRUCTURE (CONT')

- **The Game World**
  - Game Environment
    - Should provide a descriptive summary of the game world. e.g.:
      - **The criminal underworld of Perez City occupies an industrial wasteland between the magnificent high-rises of the financial sector and the burnt-out slums of 3rd district…**
      - **The Pacific Rim Rally circuit takes in eight dramatic stages from the bleached-white sands of Hawaii to the scattered gravel volcanoes of Sumatra…**
    - These introductions should be compelling as well as descriptive.

GAME DESIGN DOCUMENT STRUCTURE (CONT')

- Show sample game environment (free-hand sketch is acceptable)

The main character stands in the center of an environment sketch.

GAME DESIGN DOCUMENT STRUCTURE (CONT')

- Levels
  - Some games have very large environments with many areas rather than single levels, you still need to describe the different areas in each environment.
  - In some games, location is not relevant, e.g. board games, you should still try to describe the visual environment of the game if possible.
  - The scale of each location should be indicated (e.g. 1km square, 2 blocks of a city)
  - Level Design will be discussed in detail later.

GAME DESIGN DOCUMENT STRUCTURE (CONT')

- Progression
  - Describe how the player progresses through the levels e.g.
    - By reaching a target position
    - By defeating all the enemies
    - By achieving other conditions (describe them)
  - Can simply walk from area to area

GAME DESIGN DOCUMENT STRUCTURE (CONT')

- **Characters**
  - Player character(s)
  - Non-player characters (NPCs)
  - Character attributes
  - Character abilities & development
- **Objects**
  - Interactive world elements (e.g. doors, chairs)
  - Key Items (e.g. weapons)
  - Inventory Items (e.g. food)

Character and Object design will be discussed in detail later.

GAME DESIGN DOCUMENT STRUCTURE (CONT')

- **User Interface (UI)**
  - On screen display –OSD (Heads Up Display – HUD)
  - Camera / Point of View
  - Controller usage
  - UI design will be discussed in detail later

- **Play Modes**
  - Single Player / Multiplayer ?
  - Single Player: What does the player do after he wins the game? How much time on average you think a player will spend on playing the game?
  - Multiplayer : Specify the maximum number of players who can play the game simultaneously. Describe how the game will be presented to different players (split screen, single screen or different screens? Play over LAN, bluetooth or Internet? Competitor or co-operation?)

GAME DESIGN DOCUMENT STRUCTURE (CONT')

- **Play Modes (cont')**
  - For each mode, state if there will be a way to save the game in progress.
  - MMORPGs (Massive Multiplayer Online Role Playing Games) have many specific design considerations.

- **Audio-Visual Requirements**
  - Models/graphics for the game world, characters and objects
  - List of animations for the above
  - Sound effects, vocals & text (& localisation)
  - List of sound files for the above
  - Music requirements and style

GAME DESIGN DOCUMENT STRUCTURE (CONT')

- **Technical Considerations**
  - Platform / hardware specifications
  - Programming Languages and Development Platform
  - 2D/3D Graphics Rendering engine
  - Artificial Intelligent (AI)
  - Networking support
  - Game Engines
- **Appendix**
  - Fine details of previous sections (if necessary)
  - References (if you use any external reference materials)

OTHER DOCUMENTATIONS WHICH MAY BE INCLUDED IN THE APPENDIX

- The following Project Management documentation can be directly linked to a game design:
  - Development timelines (Project Plan)
  - Asset management (people/hardware/software)
  - Budget Estimation
  - Development Team
- When the target audiences are publishers or consumers, you may also include:
  - Unique Selling Points (USPs)
  - Market research / Competitor analysis
  - Prototype

LEVEL DESIGN OVERVIEW

- A Level can be a mission, stage, map.
- Players play games and want to be challenged. Challenge should always come in the form of testing the players' skills at the core gameplay.
- Like a good television show or book, the game must maintain a player's interest : story, conflict, acquisition of new assets, display of new art, increase in difficulty.
- They are used to keep the player interested and looking forward to the next level.
- One boring level can be the course of death of a game, players don't like playing and buying the same game twice. Each level should introduce some variations.

LEVEL DESIGN BUILDING BLOCKS

The basic elements in building a level are :

1. Concept
2. Environment to exist in
3. Beginning
4. Ending
5. Goal
6. Challenge to be overcome by the player
7. Reward when the goal is achieved
8. Way of handling failure - Punishment

Example: The level building block of Tetris

Concept: Find a place for the blocks or lose the level.

Environment: The active play area to the left of the game data.

Beginning: The player starts with an empty screen and a score of 0.

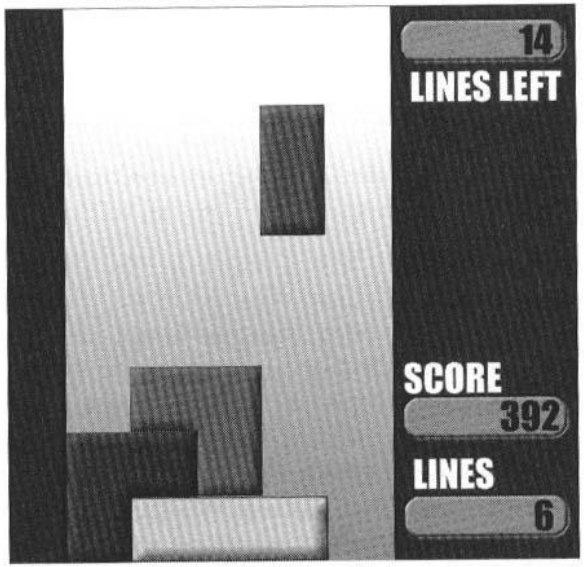
Ending: The level is over when the player either creates the correct number of vertical lines (success) or the blocks pile up to the top of the screen (failure).

Goal: Create a number of lines that meet the target requirement for success.

Challenge: The speed of descent, type of blocks, and number of lines needed.

Reward: The player moves to the next level, or receives a brief animated sequence.

Failure: The game ends and must be started from the beginning.



The level building blocks as seen in a simple puzzle game.

WHAT MAKES A LEVEL FUN

- *Ergonomics* – No learning by death (of player character)
- *Flow* – Keep the player moving
- *Rhythm* – Create a roller coaster rather than a highway
- *Wow Factor* – The water cooler moments
- *Difficulty* – Should not be unreasonably difficult, let the player wins (eventually)

SOME RULES TO LEVEL DESIGN

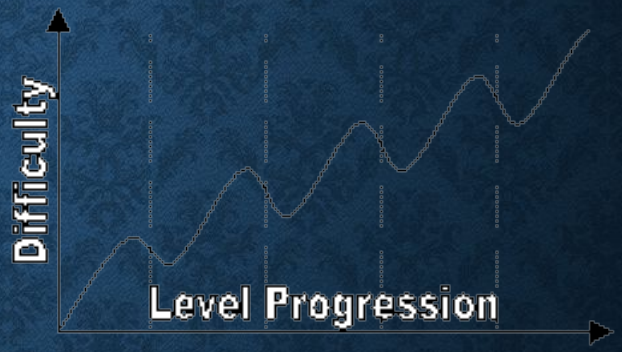
- Maintain the vision (the core idea of the Game Design).
- Learn the design palette (all the elements in your game design).
- Have fun while you work (convey the joy that you experience to the players).
- A level will not be better than what you can imagine.
- If there's no difference between 2 levels, what's the point?
- Cater for different playing styles and abilities.
- Reward player imagination and efforts.
- Pay attention to level pacing.

The Typical Pacing Curve in a Level

SOME RULES TO LEVEL DESIGN (CONT')

- Expose assets to players (Assets are the game's candy).
- Challenge the player.

The Difficulty Curve of the Game



The Difficulty Curve in a Modular Game

- See through the player's eyes (put yourself in the player's position)
- Fulfill players' expectations

SOME RULES TO LEVEL DESIGN (CONT')

- Balance the difficulty for the median skill level.

- Know the players' tricks.
- Learn what players bring to each level (in terms of weapons etc).
- Be the enemy (you should also think from the AI's perspective).
- Test play.
- Take the time to make it better (Iteration and refinement).

CHARACTER DESIGN

- Open this section with an in-depth profile of the key player characters, e.g.:
  - Jack Travis trained as a boxer in 50s Chicago…
  - The Lotus Elise is a small, light and agile sports car, ideal for junior racers…

- Next, describe any other key characters that are not controlled by the player (Non-player characters NPC).

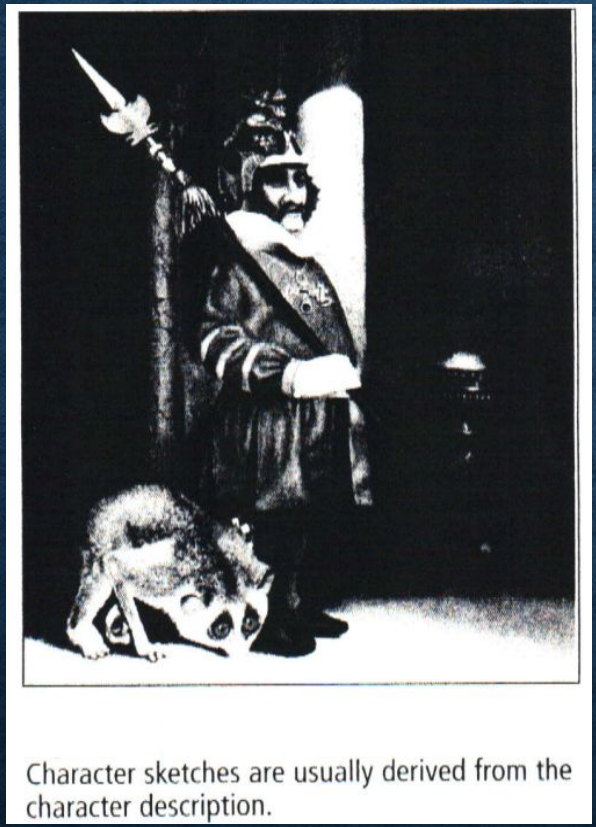
CHARACTER DESIGN - CHARACTER DESCRIPTION

- A character description is a short explanation of the character. It is usually the first step in creating a character design.

- For example,

  Malden Stonebridge – Malden is a guard in the queen's palace. He is a dwarf and was chosen as a guard for his strength. He never leaves his post and is very loyal to the queen. Malden wears a red uniform with a white neck ruffle and gloves. His head wears a dragon-crested helmet. He carries an ornate spear as a weapon. A creature that looks like an overgrown rodent with large luminous eyes accompanies Malden on his rounds.

CHARACTER DESIGN - CHARACTER DESCRIPTION (CONT')



Character sketches are usually derived from the character description.

CHARACTER DESIGN - CHARACTER ATTRIBUTES

- Characters in the game will have particular attributes depending on the game genre:
  - Speed, acceleration… (Driving)
  - Strength, Skill, Morale… (War)
  - Age, weight, fighting style… (Fighting)
- The attributes needed for characters in the game should be identified.
  - You should give actual values for the key characters.
- Choosing a good attribute set is important in balancing a game.

CHARACTER DESIGN - CHARACTER ABILITIES & DEVELOPMENT

- Characters may have inborn abilities that they can use in the game. These should be listed.
- Character development describes any change in character attributes or abilities due to experience / objects gained in the game.
- Abilities and development can be very varied:
  - A wizard starts with a limited set of spells, and acquires new ones with experience and by reading books. List the basic & full set of spells.
  - A football team may have an initial set of 'special' passes and shots. They gain more with experience and funding. List the basic set of moves and the full range of new ones.

CHARACTER DESIGN - CHARACTER ABILITIES & DEVELOPMENT (CONT')

- Ensure you give all the required details for a character development. How much experience, what objects …etc are required.
- It is a good idea to give an overview of the character development in this section and then provide detailed tables in an appendix to the document.
  - For the wizard example, provide a selection of spells and descriptions in this section and a fully detailed table of spells, requirements, effects etc. in the appendix.

OBJECT DESIGN

- Objects may range from a door to buildings - anything with a **distinct game-play function**.
- Categorise your objects into different types:
  - **Key Items**, e.g. the weapons in a FPS, the main building types in an RTS. These are central to the game-play, and need to be described carefully.
  - **Inventory Items**, e.g. key, bullet, food, power-up, shield etc. If there are many, give a short list here and a full list in an appendix.
  - **Interactive environment elements**, e.g. moving spikes, special platforms, ramps etc. **Don't include things that animate but have no special purpose.**

OBJECT DESIGN - OBJECT ATTRIBUTES

- Objects usually have attributes just as characters do. These need to be detailed along with the object description.
- Some examples are:
  - Strength, power requirement etc. for a building in an RTS.
  - Range, bullet type, size for weapons in a FPS game
  - Health gained for food / med-kit
  - Damage coursed by spikes / toxic waste

MANAGING DETAILS IN CHARACTER & OBJECT DESIGN

- The character and object sections of a design document can contain a great deal of information.
- A full design needs detail, but it should still be readable. Some tips:
  - Move large lists or tables of detail to appendix.
  - If necessary only detail a few example characters / objects in the main document – choose examples from early in the game.
  - For examples,
    - A list of 30 spells, with difficulty level and detailed effects. Some individual spells may need further tables of detail.
    - A list of 25 cars with different engines, brakes and suspension specifications, and available upgrades.
    - For a fighting game, a list of all the moves for each character split into categories (e.g. high punch, low block), each move is also given attributes. Then a combat table stating the outcomes of all the combinations of attack / defence.

USER INTERFACE - OVERVIEW

- The game's interface is the ***language*** you use to speak to the game.

- Games are usually made up of hundreds of ***decisions***: the big ones can be achieved by ***making*** many little ones. The player makes a lot of decisions all the time. The interface for making them should be **as invisible as possible**.

- A main function of an interface is to let player ***make decisions much more easier and quicker***, especially the most commonly used functions.

GOALS OF UI DESIGN

- The three main goals are as follows:
  - Interaction: it allows the player to interact with the game. The first and foremost purpose of the interface is to translate data between the game logic and the player. Without the interface, you can't play the game. The interface tells the player what's happening in the game and lets him or her respond accordingly.
  - Information: it displays information about the setting, characters, objects, and events in the game. This information allows players to make decisions. **Images incorporated into the game often display information more efficiently and effectively than text.** This can be done through symbolic representations.
  - Entertainment: visual element adds value to the playing experience. They make the game world richer.

COMPONENTS OF UI

- The UI of a game has three components:
  - The On Screen Display – OSD (Heads Up Display – HUD)
  - The camera / viewpoint.
  - The controller interface
- You should justify your choice of OSD and compare it to other games.
- However, do not simply copy an OSD design, especially if your game is innovative.

ON SCREEN DISPLAY (OSD)

- The OSD describes the things shown in front of the main view of the scene.
- You **must** draw a picture of your OSD, including:
  - Cursor / Target
  - Critical information, e.g. health, speed, no. of life.. etc.
  - Control and status icons
- This can be combined with a mock-up showing the camera view.

CAMERA / POINT OF VIEW

- This is the choice of camera view in a game.
- Some possibilities are:
  - First person - you see the scene through a character's eyes, e.g. First Person Shooting games
  - Third person – the camera focuses on a main character in the environment, e.g. Third Person Shooting games
  - God-view – the camera looks down on a scene from far away, e.g. Rome II: Total War
- Many genres have a 'standard' point of view .
- Ensure you have a good reason to use a non-standard point of view.

CONTROLLER INTERFACE

- List the different controller methods the game can use.
- A sample controller layout may also be given.
- With a mouse / icon method, combine this section with the OSD.
- Identify any unusual control decisions.
- You must justify a complex control scheme.

CONTROLLER INTERFACE –CONT'

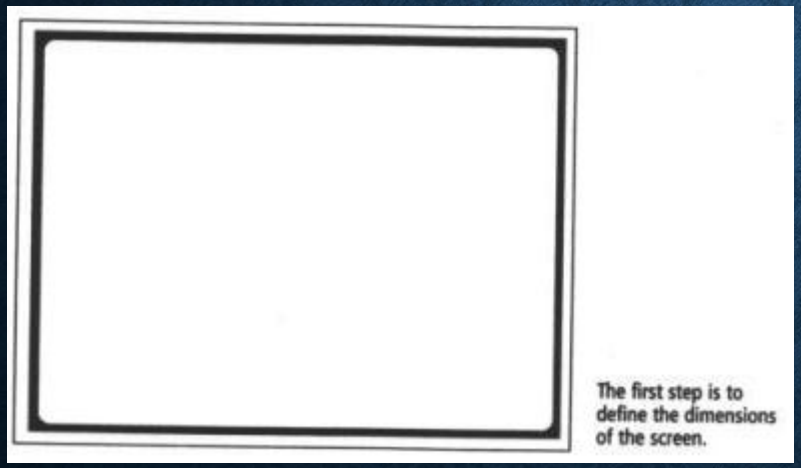
- The following is a simple specification on the controls of a car racing game (using keyboard).

up arrow - accelerate
down arrow - brake/reverse
left arrow - steer left
right arrow - steer right
space bar - emergency brake

CONTROLLER INTERFACE –CONT'

- Common Input Devices
  - Joystick / gamepad
  - Mouse and Keyboard
  - Steering wheel
  - Gun
  - Musical Instrument
  - Sport Item
  - Touch screen
  - Motion Sensing Device

AN EXAMPLE OF UI DESIGN – GRAPHIC DISPLAY FOR A RACING GAME

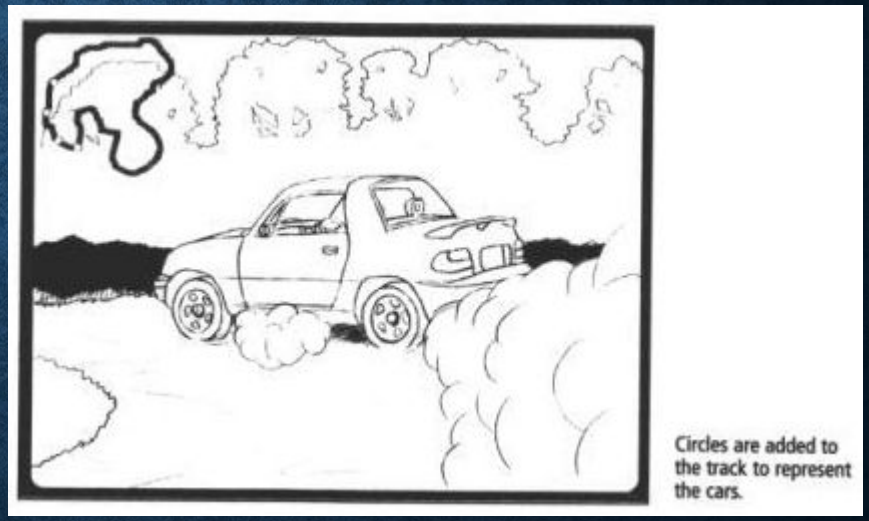


The first step is to define the dimensions of the screen.

Add a sketch of the game.

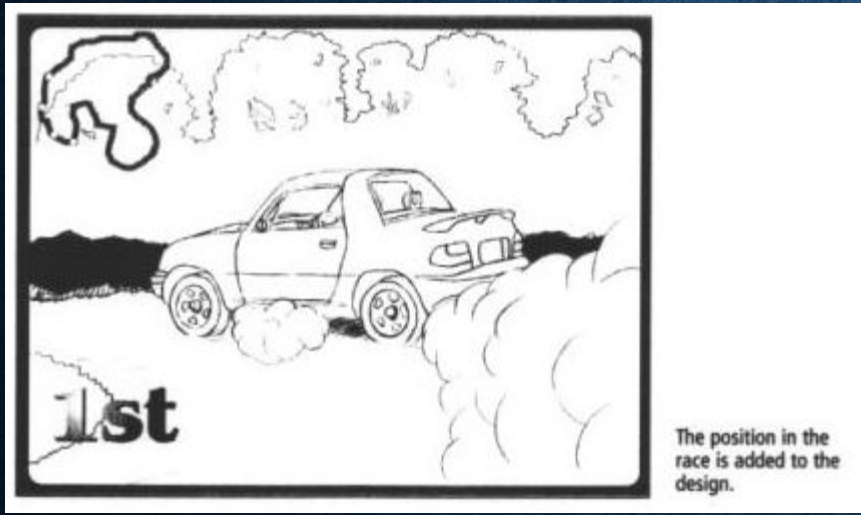
AN EXAMPLE OF UI DESIGN – GRAPHIC DISPLAY FOR A RACING GAME (CONT')

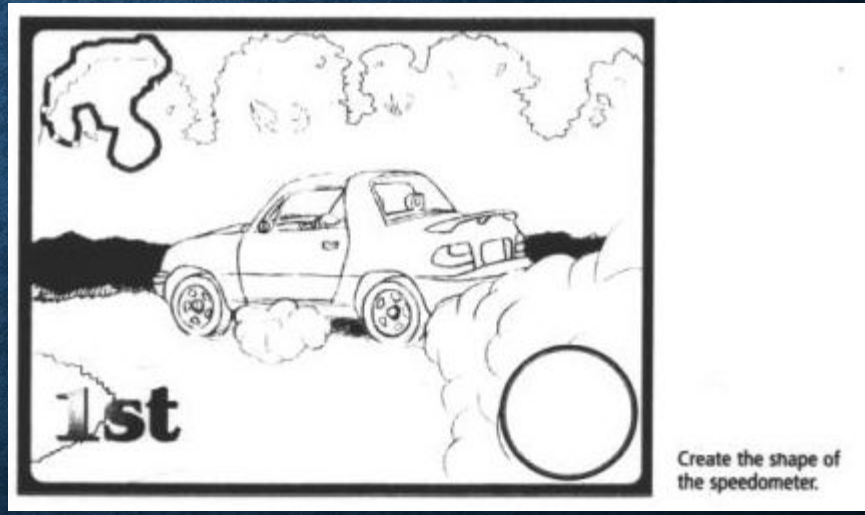
A miniature map of the track is added.



Circles are added to the track to represent the cars.

AN EXAMPLE OF UI DESIGN – GRAPHIC DISPLAY FOR A RACING GAME (CONT')





AN EXAMPLE OF UI DESIGN – GRAPHIC DISPLAY FOR A RACING GAME (CONT')

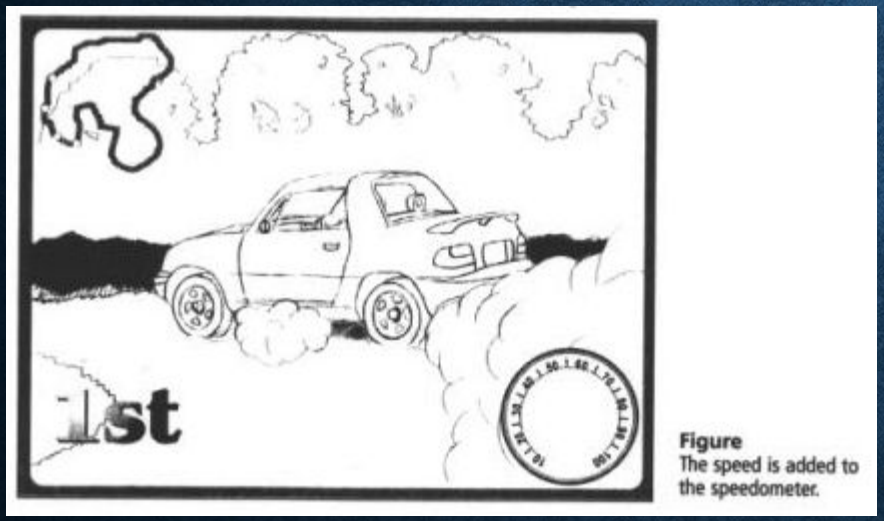
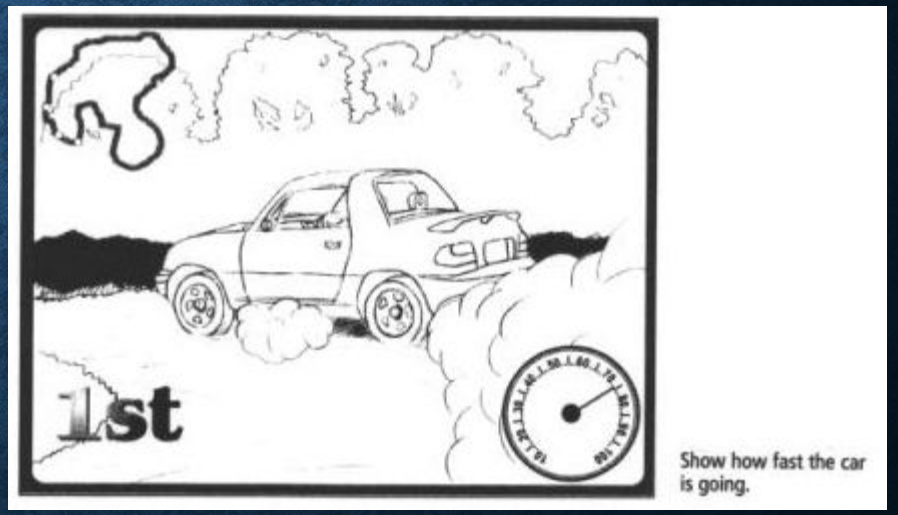


Figure
The speed is added to the speedometer.



Show how fast the car is going.

OTHER CONSIDERATIONS IN UI DESIGN

- Hardware limitation: video card
- Screen size: different resolution
- UI customization
- Localization: Text as graphic?

WHAT MAKES A GOOD INTERFACE?

- ***Simplicity***: Build an interface that is easy to figure out and doesn't require a complicated instruction manual. Information should be accessed and responded to without a lot of intermediate steps, commands, or keystrokes. The information is on the screen when the player needs it. Players shouldn't have trouble to figure out how to use the keyboard, mouse, joystick, or other tool to continue playing the game.

- ***Consistency***: Player should be able to do the things in the same way everywhere in the game; and to find vital information in the same place all the time. Using a logic in the layout that allows players to anticipate where information can be found and how commands can be executed in different parts of the game.

- ***Immersion***: preserve the illusion of immersion in the game world is to make elements of the interface as parts of the game world.

GAME INTERFACE DESIGN – A FINAL CONCLUSION

- Interface defines gameplay: interface has been explained to be the interaction with the virtual world. Actually the interface **MEANS** the game to most players and it defines how a game is experienced. If an interface is easy, perceptive, and fun, the game will benefit greatly.

- Simplify the interface and use the minimal number of keys to do maximum number of actions.

- **The supreme interface will never be noticed: If the player simply plays the game without noticing the interface, your interface design is a success. If the player notices the interface and must struggle to master it, then you have failed.**

PROTOTYPING - OVERVIEW

- A prototype is a way to see how the game actually plays before too much time or money is spent on developing the game.
- Prototyping the game offers a way to play the game early in development and determine if the play is fun.
- Prototyping can be applied to level, mission or playfield design.
- Prototyping can be a way to test programming issues early in development.

PROTOTYPING – OVERVIEW (CONT')

- 4 common kinds of prototypes are
  - **Paper prototypes**
  - **Storyboards and flowcharts**
  - **Interactive prototypes**
  - **Fly-through prototypes**
- The fourth kind of prototype – the fly-through prototype, is used for marketing.
- Not all the 4 kinds of prototypes are necessary to test every design and it is not necessary to prototype the whole design.
- Only the most important gameplay elements should be prototyped. (e.g. The combat in a role-playing game or the availability of resources in a real-time strategy game)

PAPER PROTOTYPE

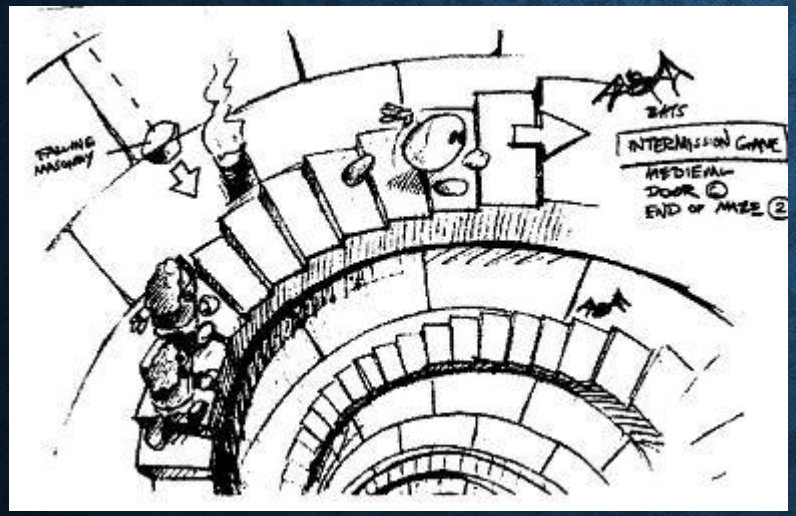
- Paper prototype is a method of making a playable version of the game using non-electronic means such as a grid board and/or a deck of specialized cards, or modifying an existing board or card game.
- A paper prototype is a physical game that can be played either in small subsections or as a whole of the game to be developed.
- A drawback of paper prototypes is that continuation in action (dynamic game) is difficult to be captured in paper version.
- That means Action Games are difficult to be prototyped using paper.
- Games in 3D is also difficult to be prototyped using paper.
- On the other hand, static games such as card games, chess game or mah-jong (that is the games with very less ongoing movement) can easily be prototyped on paper for early testing.

STORYBOARD AND FLOWCHART PROTOTYPE

- Storyboards are useful in games with story-based such as Role Playing games games.
- Each storyboard shows one visual representation of the game in action. Arrows can be used to show the flow of action.
- A flowchart is a simple diagram that shows an activity step-by-step using a few symbols.
- When creating storyboards, the designer should also create a flowchart of the game's action to show the team how everything ties together.
- A good use of flowchart prototyping is for designing adventure games.

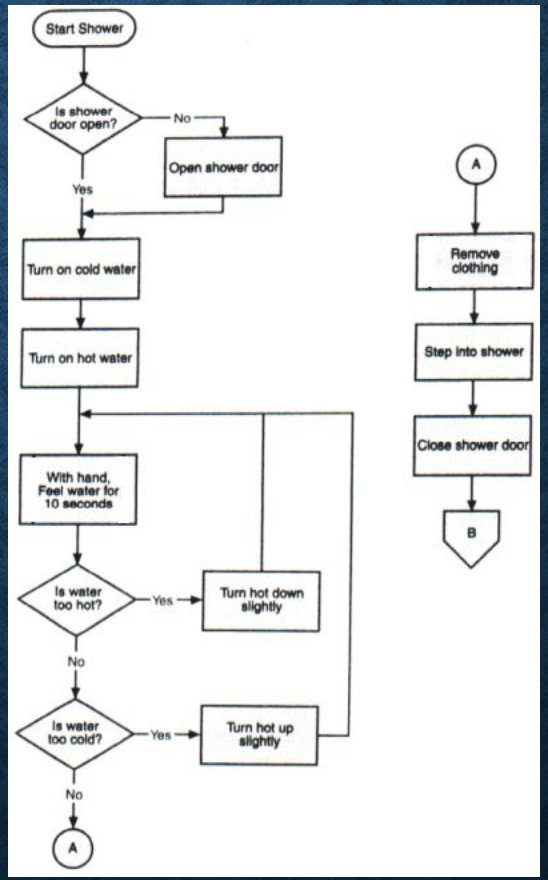
STORYBOARD AND FLOWCHART PROTOTYPES – CONT'

- A sample story board



STORYBOARD AND FLOWCHART PROTOTYPES – CONT'

- A sample flowchart to take shower



Interactive Prototype

- An Interactive Prototype can be created using a commercially available software/game engine (such as game maker).
- It is a fundamental for the designers to test their design assumptions using a simple model of the gameplay elements.
- This can be a first demonstration to the company boss, the publisher, the marketing and sales team.
- However, this kind of prototype cannot be used for testing complex gameplay mechanics and complicated interactions.
- Interactive Prototype is suitable for developing games with a lot of actions (e.g. Fighting games, Shooting games, Sports games ..etc)

Fly-Through Prototype

- It is often just a short non-interactive movie in which a camera views through a level or follows the main character as he/she performs a number of game actions (just like a movie trailer).
- It can also be interactive by letting a player explore a level or experiment with the game activities.
- This prototype uses finished artwork and audio to show a complete level as it will appear in the game.
- The objective is to let the audience view some of the important characters, see what a level will look like, and play around with some of the gameplay elements.
- One possible purpose of fly-through is for an independent developer to show to publisher as part of contract negotiations.
- Another purpose is to act as a promotion material for the marketing department to retailers and customers.

VIDEO GAME DESIGN

- Game Definition and classification
- Game Elements
- Design Stages and roles
- Game Idea and concept
- Gameplay
- Core Mechanics and balancing
- Game World
- Character Design
- Storytelling
- User Interface and Prototyping

VIDEO GAMES BREADTH OF VIDEO GAMES

- User Interface/User Experience (UI/UX)
- Computer Programming/Software
- Math and Physics
- Art
- Engineering and Networks
- Sales and Marketing
- Business and many more fields

STORYTELLING ROLES

- Creative Writer
  - Conduct story writing and partner with story designer
  - Requirement : passionate, strong communicator, collaborative, Team Work
- Interactive Story Designer
  - Generate core gameplay concepts and partner with creative writer
  - Requirement : Innovation, Player-centric Approach, Writing skills
- Lead Game Designer
  - Create Basic concepts of game and responsible for the design and implementation of the gameplay
  - Requirement : Creativity + Innovation, Team Work, Idea Formulation

STORYTELLING

- Texts
- Cut-Scenes
- Sound Effects / Musics
- Interaction
- Voice-Over
- Dialogues
- Art/Graphics
- Point of View

GENERATION AI

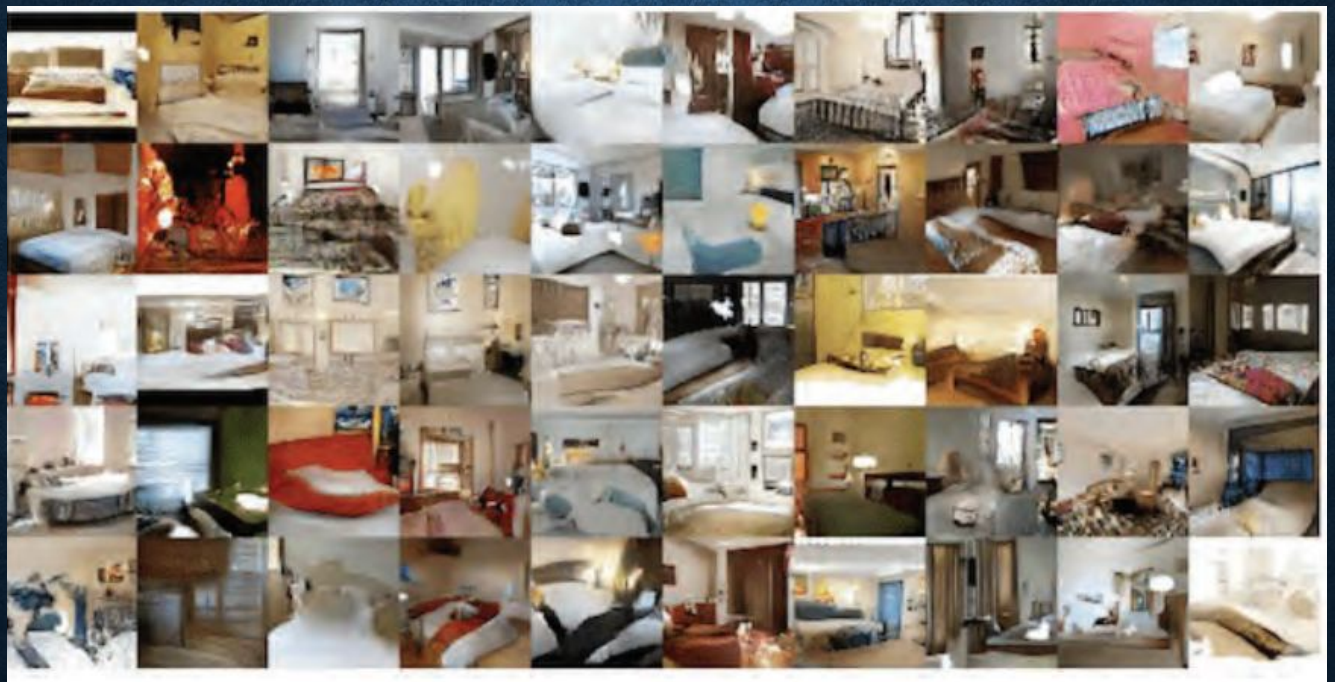


Fig 7: Example of GAN-Generated Photographs of Bedrooms (Source: Arxiv)

GENERATION AI

"a lovely cat running in the desert in Van Gogh style, trending art."

Why should we care?

Could be a model of imagination.

Similar techniques could be used to generate any number of things (e.g. neural data).

It's *cool!*

GENERATION AI

"Batman eating pizza
in a diner"

How does it work?

It's complicated...
but here's the high-level idea.

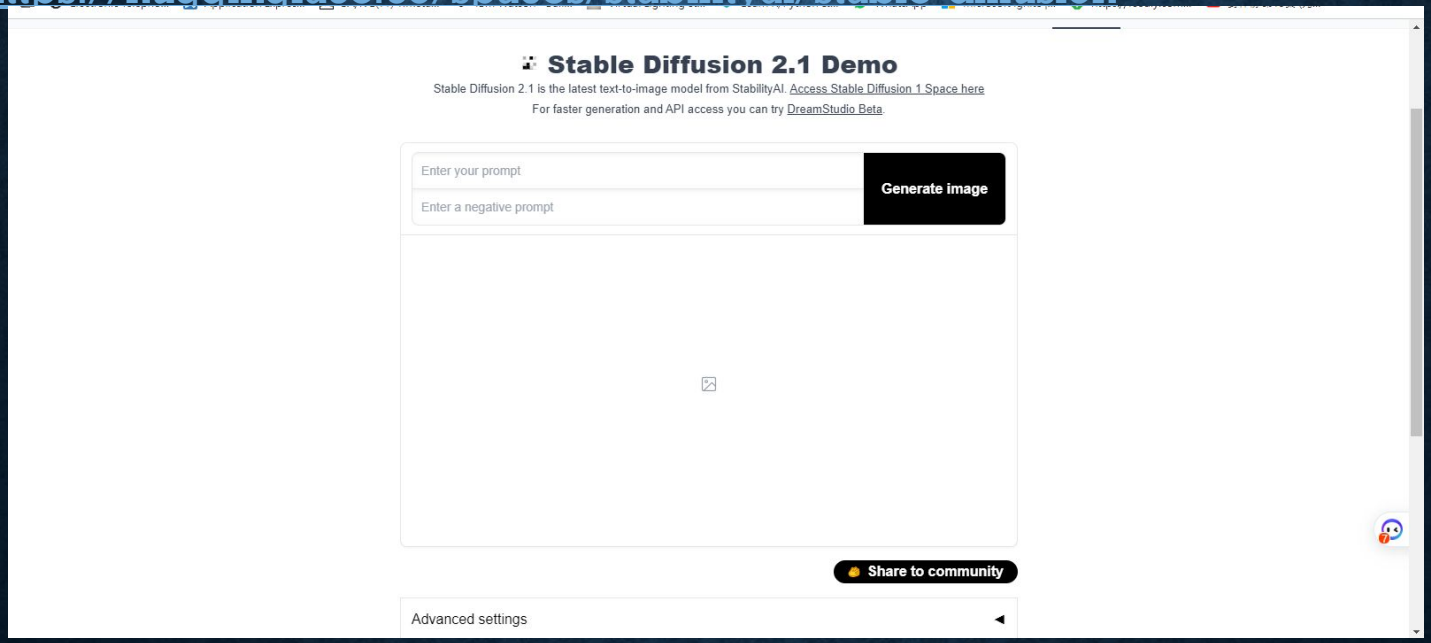
GENERATION AI

Cartoon with StableDiffusion + Cartoon

https://www.reddit.com/r/StableDiffusion/comments/xcjj7u/sd_img2img_after_effects_i_generated_2_images_and/

EXERCISE

- Open Chrome and enter https://huggingface.co/spaces/stabilityai/stable-diffusion



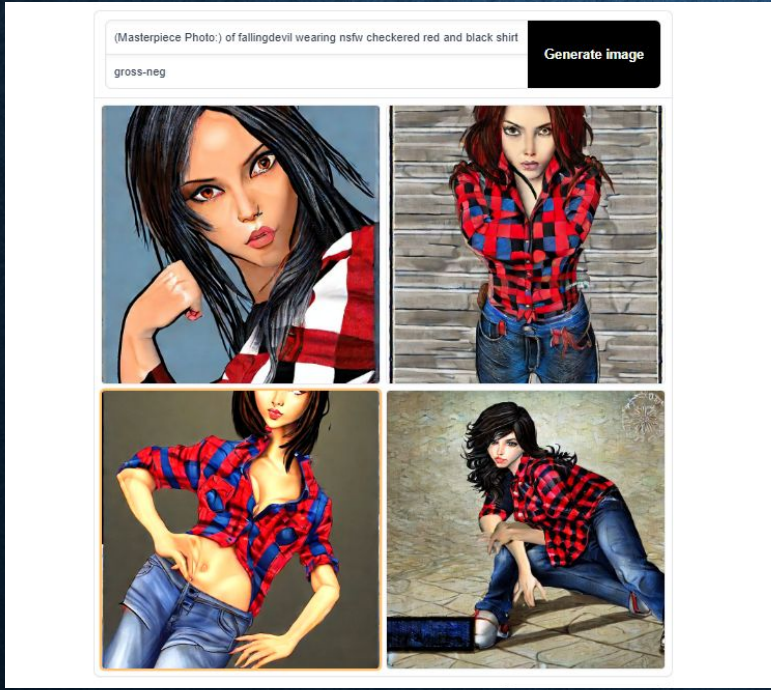
EXERCISE

- Enter the following prompt and negative prompt

Prompt	Negative prompt
(Masterpiece Photo:) of fallingdevil wearing nsfw checkered red and black shirt and blue jeans staring at the viewer,(big brown eyes) ,(checkered red and black shirt), (blue jeans),Highly Detailed,(close portrait:1.3),(Feminine:1.4),(beautiful:1.4),(attractive:1.3),calendar pose,perfectly detailed eyes,studio lighting,thematic background, (high detailed skin:1.2), 8k uhd, dslr, soft lighting, high quality, film grain, Fujifilm XT3	gross-neg

EXERCISE

- This will generate a new photo

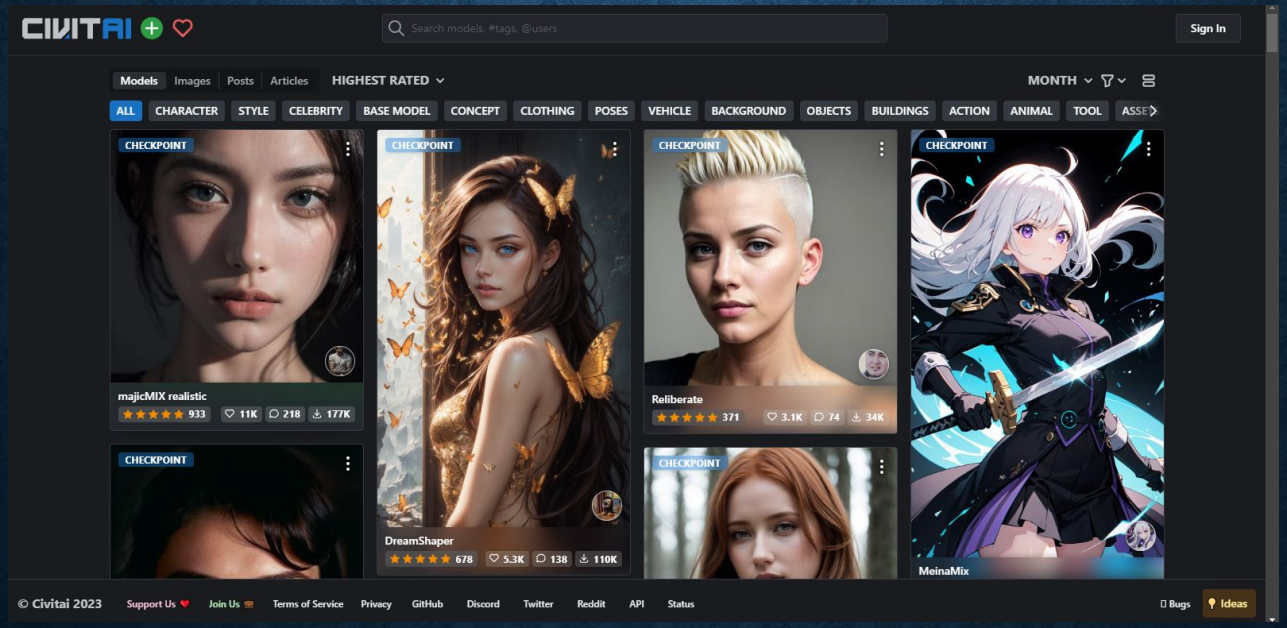


EXERCISE

Enter Prompt “A high tech solarpunk utopia in the Amazon rainforest” and Negative prompt “low quality”

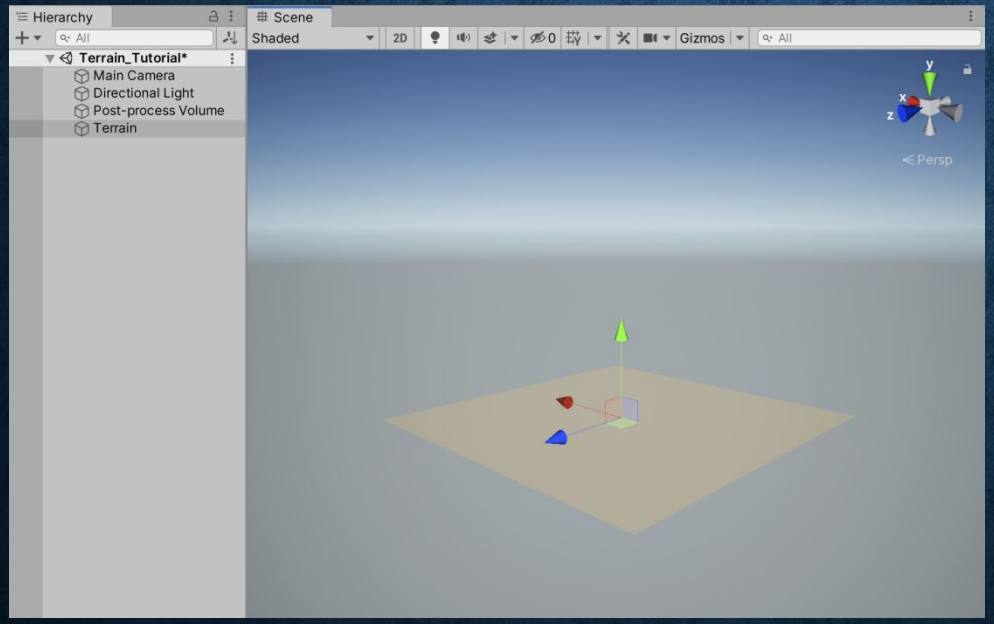
EXERCISE

- Go to https://civitai.com/ to find the prompt



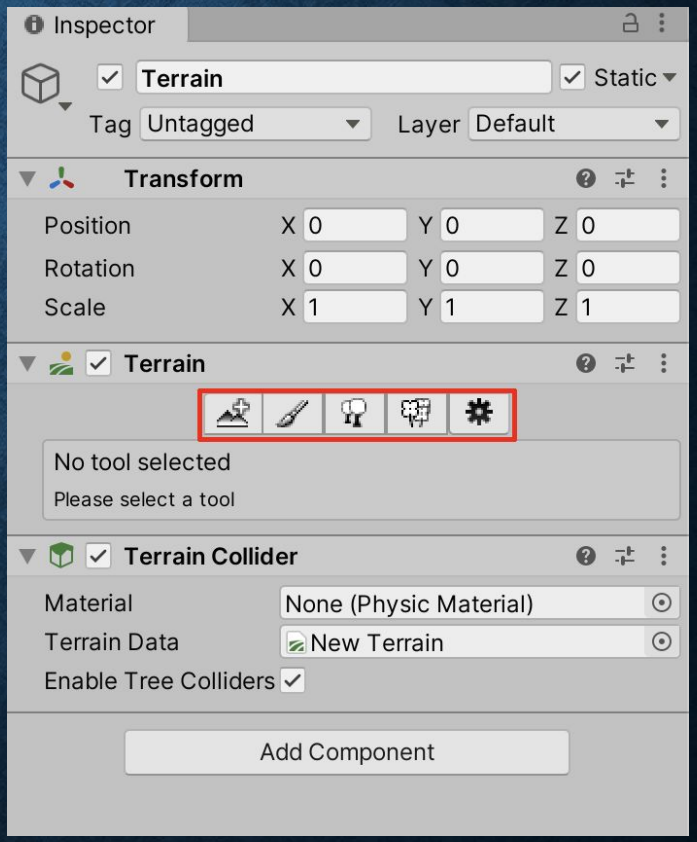
CREATE TERRAIN

- select: **GameObject** > **3D Object** > **Terrain.** This will add a new Terrain GameObject into your Scene.



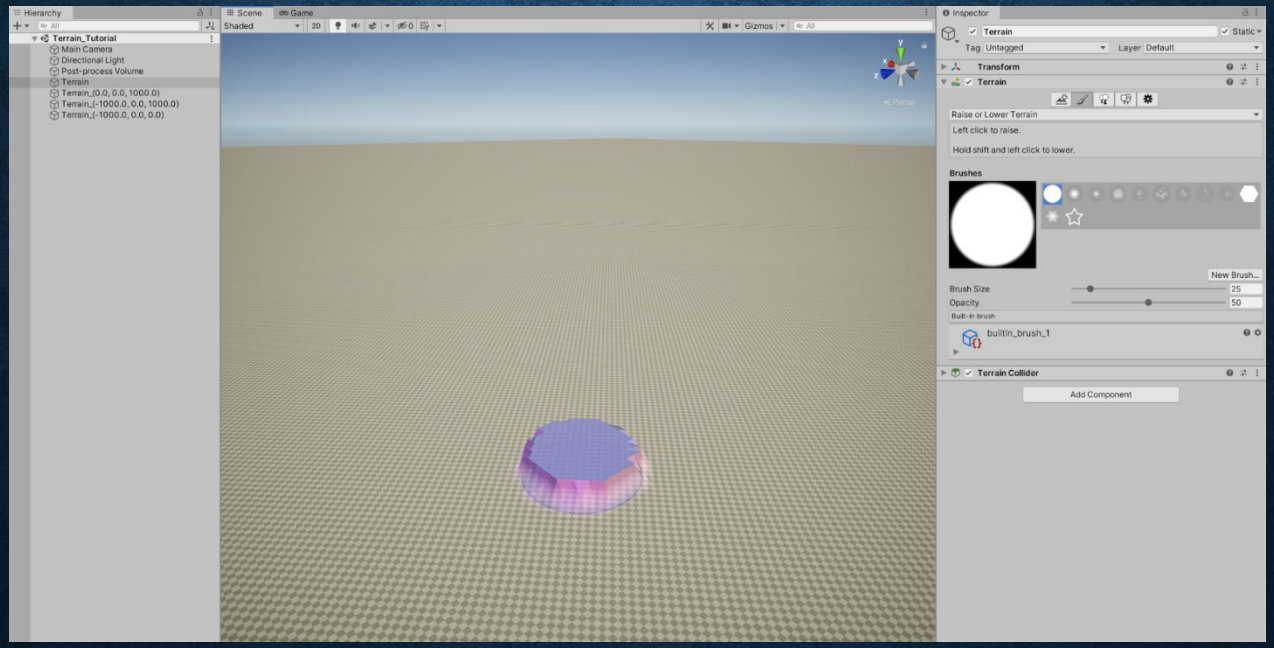
CREATE TERRAIN

- Your Terrain will be a large, flat plane. When the Terrain GameObject is selected in the Hierarchy window, the Terrain tools appear in the Inspector.



CREATE TERRAIN

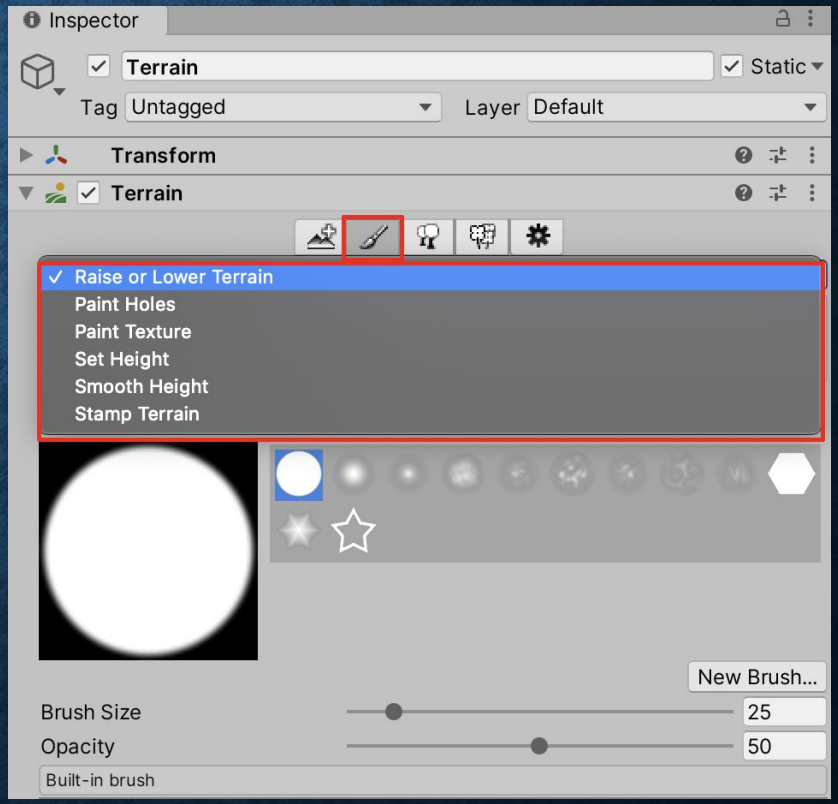
- In the Scene window, you will see a cylindrical object hover wherever the mouse cursor is positioned. This is the default brush that will enable Terrain sculpting.



https://learn.unity.com/tutorial/working-with-the-terrain-editor-1#

SCULPTING TERRAIN FEATURES

- Select the **Paint Terrain** tool, which is the second button in the Terrain toolbar. Select the dropdown menu titled: "**Raise or Lower Terrain**" to see the list of available options. This dropdown menu accesses the different painting and sculpting tools. Let's first explore the **Raise or Lower Terrain** tool option.



GROUP PROJECT

- At the end of our lesson, all of you need to create an group project in 2 people to create a virtual reality game and give an presentation to us.

1. Introduction: Briefly introduce the project and its objectives.
2. Game Concept: Discuss the concept of the game and what makes it unique. Highlight the key features that we plan to include in the game.
3. Gameplay Mechanics: Discuss the gameplay mechanics, including how the player will interact with the game world and the controls they will use.
4. Graphics and Sound Design: Discuss the visual and audio elements of the game, including the art style and the sound effects and music.
5. Technical Implementation: Discuss the technical details of how we plan to develop the game, including the software and hardware that we will use.
6. Challenges and Solutions: Discuss any challenges that we anticipate facing during the development process and our proposed solutions to overcome them.