**What is a game** give player freedom of control, random events, more dynamic (movies or books: linear experience, a fairly direct mapping between viewer and creator) **Game Genres** Traditional (computerized card games/board games), Adventure (no action; text/graphical), Action (superset of many other genres, involves fast-paced combat and movements), Platformer (involves a character running and jumping in a playing field), RPG (characters need to explore the

world & have specific skill sets, also with character development), MMORPG (thousands of players simultaneously in a virtual world; intensive networking considerations in development), Racing/Sports (requires realistic graphics & gameplay experience), Education (design for a purpose) **Others** FPS, Survival, Battle Royale, MOBA, Turn-based strategy, Realtime strategy, Simulation, Puzzle games, Casual games

**Skillset** be creative, be able to collaborate & communicate, have a business/entrepreneurial mindset, be aware of the latest technologies **Hard skills** programming, math, physics, art & design, modelling, animation, sound/music design & composition, computer graphics, cinematography

**Development methodology:** Code-and-fix: aka firefighting, little/no planning, results in many technical problems, may only work for small team size & short

development cycles; Waterfall: processes well planned, set up accurate schedule & concrete milestones, works when nothing unexpected happens; Agile: plan for short periods of time, goals can change from time to time, more flexible

**Game Design** no formal or well-established map to follow, evolves over time, not a set of principles, but an activity **Basic Elements** Aesthetics, Story, Mechanics, Technology **Bartle's Taxonomy of Player Types** Achievers, Explorers, Socializers, Killers **Playtesting** get people to play your game and to check if they experience exactly what you have designed (key questions: Why, Who, Where, What and How)

**Frame** All updates are made in a buffer first and then contents of the buffer are copied to the screen; Unity default frame rate = 50fps; fixed timestep = 1 / FixedUpdate() frame rate; maximum allowed time step: limits the amount of time Unity will spend processing physics during a given frame update; **Game Loop** A collection of frames form a game loop (each iteration of the game loop is a frame); real-time games mostly update at 30-60 fps **Game Objects** anything in the world that needs to be updated and/or drawn every frame

**Audio** 5.1 surround system (Back/Surround Left, Back/Surround Right, Front Left, Front Right, Centre, Subwoofer), 7.1 surround system (Surround Back Left, Surround Back Right, Surround Left, Surround Right, Front Left, Front Right, Centre) analog to digital: sampling + quantization **Audio format** PCM: uncompressed raw data, .wav, sample rate 44.1kHz for CD quality, 8kHz for human voice, quantization 8/16/24 bit per sample; MP3: MPEG2 Layer 3, lossy compression by reducing audio parts not sensitive to human, bit rates: 32kbit/s acceptable for only speech, 96kbit/s acceptable for speech/low-quality streaming, 128 or 160kbit/s mid-range quality, 192kbit/s common high quality, 320kbit/s highest quality possible; WMA: audio data compression technology developed by Microsoft in 1999, lossy, encodes up to 48kHz with two discrete channels; MIDI;**Image** BMP: Raw data with header, simple and direct operation, Header with information: XL - # of pixels in x-direction, YL - # of pixels in y-direction, bpp - bit per pixel, etc. Each pixel has 16 colors & uses 4 bits, may have compression; RAW: Original image pixel data, for professional digital photographers, can adjust brightness, contrast, sharpen, white balance, noise reduction, sensor dust removal, etc. S-RAW: about 25% pixel count, half file size; M-RAW: about 55-60% pixel count, 2/3 file size; GIF: relies on lossless LZW compression; Takes advantage of repetition of data, encoding defines codes to build up dictionary, decoding recovers codes based on the dictionary, maximum 256 colors (8 bits per pixel), not good for photos; PNG: Lossless data compression DEFLATE, not limited to 256 colors (palette, grayscale or RGB); 8-byte signature followed by chunks, PLTE for palette, list of colors; IDAT for image data; IEND for marking the image end;

JPEG: Good for natural image and photo, digital camera photo native format, for 640x480 24-bit image, can set different compression rate; M-JPEG: encode each video frame or interlaced field of a digital video sequence by JPEG images, Sound is often uncompressed PCM or low compressed ADPCM; MPEG: High compression by exploiting both intra-frame and interframe correlation: I (Intra, or Index) pic - only intra-frame coded; P (Predicted) pic - with reference to a previous picture (I or P), same quality as I picture; B (Bidirectional) pic: high compression based on interpolation, need both a past and a future picture

**2D** refresh rate: the number of times a screen can be redrawn in one second; framebuffer: RAM used to store the entire image to be displayed on a screen, each element is a pixel; Frame rate: the rate that the game/GPU can output a frame; screen tearing: the display shows part of two different frames at the same time; Double-buffering: Front buffer (monitor shows what is inside this buffer) + Back buffer (where we are current drawing to), buffer swap takes place once the back buffer is filled/during VBLANK; Triple-buffering: 1 front + 2 back; back1 & 2 swap when 1 is filled, back2 & front swap during VBLANK; **Sprites** a 2D visual object in a game world that can be drawn using a single image on any frame; Painter's Algorithm: Bottom-most sprites are rendered first, top-most sprites are rendered

last; sprite sheet: holding multiple sprites in a single image file; **Texture** an image that is used to modify the appearance of a surface; Wrapping: Tile/Repeat/Wrap, Mirror, Clamp, Border; **Tile maps** Each square references the sprite (tile) that should appear in that location (A tile set contains all the tiles & all tiles are placed into a single sprite sheet)

**3D** Cylindrical coordinates: (height, radius, yaw); Spherical coordinates: (radius, pitch, yaw); Homogenous coordinates: (x, y, 1) for 2D, (x, y, z, 1) for 3D **Quaternion** rotating theta degrees around axis (ax, ay, az): q=(ax\*sin(theta/2), ay\*sin(theta/2), az\*sin(theta/2), cos(theta/2)); Position space: Model, World, View;

Camera projection: Orthographic, Perspective; Light: Ambient, Point, Spot, Directional, Area; Object rendering: Opaque, Transparent, Translucent; Surface representations: Spline, Subdivision; Shading modes: Flat (once per triangle), Gourand (once per vertex), Phong (once per pixel); Phong reflection model: Ambient, Diffuse, Specular

**Collision** The Separating Axis Theorem: Two convex shapes do not intersect if and only if there is an axis (the separating axis) along which the projection of the two shapes do not overlap; Collision Detection: Instantaneous, Continuous

**Networking** Network LOD: For each client, the server adaptively adjusts the update frame rate for the other clients according to their physical distance; MMORPG: standalone servers, using a set of large servers to act as game servers, each server plays a single function (database, gameplay, community), fault tolerance, load distribution; Zone: logically a region of game happening on servers, usually mapped to a scene in the gameplay, a software region that the players communicate directly (in the same memory block); Dynamic Zoning: A zone is divided into several groups, a group is a collection of players in the scene

(population), all players in the same group run the same process, players from different groups communicate between processes; groups can be created in, deleted from or moved across servers for load balancing purpose

**Camera** field of view: specifies the amount of the world that is visible using an angle subtended at the camera, too large FOV leads to the fisheye effect; setups: fixed, first-person, follow (third-person), cutscene

**Optimization** bitwise operation, int arithmetic, move frequently used values out of loops

**AI** Finite State Machines, Artificial Neural Networks, Game Tree (Minimax Algorithm, Alpha-Beta Pruning); Pathfinding (Path Nodes/Navigation Mesh)