Game Programming

Virtual Reality

Virtual reality games try to simulate reality using tracking and 3d near-eye display. Virtual reality Virtual Reality games are split into three categories:\

1. Non-immersive VR such as video games
2. Semi-immersive VR such as Simulation Games or Simulators
3. Fully immersive games such as those that require a headset or head mount display.

Augmented Reality

AR is integrating infographics with real-world objects. Real World element differentiates AR from VR:

1. Marker - based: uses Digital anchor which is using GPS to monitor your location in the real world
2. Markerless: an AR solution that lets you overlay digital content onto a real scene without using markers. It allows free movement.
3. Location - based: the technology that doesn't require any physical markers or identifiers to work. Unlike the marker-based augmented reality brands in merchandise and advertising, which typically connects to the products or shop locations, location-based technology doesn't require such connection.

So why would we AR and VR?

1. Digital transformation: the digital transformation empowered human development and advancements in technologies
2. Technology changed our lives :o

Industrial & manufacturing
Healthcare
Engineering
Education
AR VR
Retail
Military
Marketing & advertising
Emergency response
Entertainment

We use AR VR for things like:

1. Healthcare
2. Education
3. Military
4. Emergency response
5. Entertainment
6. Engineering
7. Retail
8. Marketing & Advertising
9. Industrial and engineering

Extended Reality

XR or extended reality includes AR, VR, MR and everything in between (I hate this course).

Principles of Game Design

1. Player Empathy: A good game designer should always keep the player in head. One of the hardest things for a designer to do is to keep his mouth shut while someone plays his game. Player empathy not only help you create good gameplay, but also lets you identify and eliminate problems during the design phase rather than during production.
2. Feedback: feedback is what distinguishes a game from every other form of entertainment. It's the interactivity that makes our games unique. Every input the player makes in the game should give him a discernable response. No input should go unanswered.
3. Grounding the player: the player should always know where he is in the game and why he is doing what he is doing. If a player has an overall map in his head, it encourages him to come back to the game again and again until he is done. Throughout the game, as the player wrestle with the problem in front of him, he should always have some idea of how this single step fits into the longer path that will eventually lead to success.
4. The Moment-to-Moment experience: The positive side of creating a good moment-to-moment experience is giving the player a constant stream of interesting choices that have significant outcomes. The more verbs you can give a player, the more you allow him to do. It is the doing that is at heart of good game play and a positive moment-to-moment experience. In general, automate all the set-up tasks the player might find boring. All the set-up tasks or all prerequisite tasks the player may need, get the computer to do it for them. In addition, you should make the game entertaining moment to moment. Keep it interesting! Give the player a lot to do but make sure that what they are doing is fun!
5. Immersion: Immersion is what happens when you make the moment-to-moment experience so compelling that the player is drawn completely into the game and the real world disappears. The concept of the continuous dream is when you bathe the player in a constant stream of images that pull him into your world, and you avoid mistakes that makes him lose concentration into the game. If you break the dream, you lose the immersion.
6. Writing: good writing is invisible. Bad writing draws attention to itself and instantly destroys the players sense of immersion. Writing well is hard. People spend a lifetime learning how to do it. If you have never given writing much thought before don't write. You should bring in someone else to do it.

7. Design with Limits: Designer often forget that building a game is a software development project. It has a cost and a schedule, and its ultimate success or failure hinges not just on good gameplay on time, on budget, with technical features that work, and without crashing the player's machine. The game should provide the features it promised, in a timely manner, and that can work on most computers or machines.
8. Removing Impediments: game designers should avoid issues that can make the player waste time. Avoid disk swapping (switching CDs to continue the game), make short loading times, avoid game interruptions, saving the game to backup progress, avoid bugs and housekeeping (idk what that is).
9. Interface design: make things as simple as simple but not too simple. Make things as simple as possible, but no simpler. (Albert Einstein). The design must convey the essence of the game. It must take into account the possible actions that can be taken, and what should happen at each moment. The best way to make sure something is easy to use is to make it impossible to do otherwise at any moment. Let the player do what he wants without having to think about it.
10. The Start-up Scene: Upon joining the game, the user should get into the game right away. The game should load a saved game, it should offer a tutorial or practice area, it should allow for options/menu to edit some preferences, and it should replay an opening movie.
11. Customizable Control: it would give the player as much control over the interface as possible. Keep in mind that different things are important to different players and the options screen should explain what each option does.
12. Cheat Codes: include as many cheat codes as possible. In addition, let the player decide what is fun for themselves. Keep in mind that the cheats need to be tested as well. If you can't preserve the design integrity while allowing the players to cheat, let him cheat anyways.
13. Tutorials or Practice Mode: a tutorial gives the player hands-on experience without endangering them.
14. Structure and Progression: A game should be easy to learn but difficult to master. If you don't get the player involved in the first 15 minutes you will lose them. Through the process you must listen to your tester!
15. Take Care of your Player: Give the player the information he needs. Offer several level of difficulties while also protecting newbies and reduce player paranoia (offer rewarded risk).

So how do you design a game?

1. Create an Integrated Whole

2. Think about the element of the economy of design. Keep things simple but not too simple that bores the player.
3. Find a source for your ideas:
    1. Ig you are designing something of interest to you, the ideas will come naturally. The game should be about something you are interested in, and it should be a genre you are familiar with.
    2. Brainstorm your ideas! It is the best way to get a lot of ideas. Don't talk about your ideas too soon! Think about originality and if they are feasible to implement or not.

Game Components:

The main components of the game are input, audio and visuals. Somethings to avoid...

1. The Visual Disaster: making the visuals too distorted that it distorts the users (ex Drake of the 99 Dragons)
2. The Audio Disaster: Don't make the audio or sounds of the game too low or high that it either bores the user or disorients them (ex Revolution X).
3. The Input Disaster: don't make inputting or controlling the player difficult (ex PacMan 3D).

The basic building blocks of a game are:

1. Images
2. Polygons
3. Lights & Materials...

Components of a Game:

1. 2D
2. 3D
3. Physics
4. Input
5. Music & Sound
6. Input
7. AI
8. Networking
9. Level of Detail
10. AI Updates
11. Framerate Lock
12. Billboarding

13. Genre-Specific