Fact and Fiction – The Virtual World

The Past, Present and the Future of Computer Gaming

Shivansh Srivastava
3rd Year Undergraduate
BITS Pilani, India
Email - shivansh.bits@gmail.com

July, 2012

***Abstract*—** This work reviews several aspects of the ever-increasing field of Computer Games. First, the Evolution of Computer Games is discussed. Two different classes of Principles by two great Pioneers of the field, interested in cognitive impact of Game Design, are reviewed. The study is a methodological framework, in which results & effects are compared, inferred, interpreted & generalized. The study takes on a unified research paradigm & methodologies to lay out the underlying principles of Gameplay, pursuing & achieving goals, consistency & fairness of gameplay, process of game designing. It takes a retrospect of the Interaction Model, analyzing the perspectives, comparing the Physical & Temporal Dimensions. The study also discusses the Game Business & the shape of the Industry. Finally, the Future of Game Development is concluded, deducing with respect to World of Mass Development (WMD) & HTML5.

***Keywords*—** Game Engines, Graphics, Game Design, Gameplay, Game Business, WMD.

I. Introduction

The title of my Research Paper might be a bit deceiving, but what I intend to showcase here has a larger realm. Computer Game design and production is a fast paced, hit-driven, technology-based field, where Software is extensively improving, increasingly dependent on the high RAM & Graphic Cards; and & the list of awe-inspiring & stunning features to incorporate is ever increasing. Computer games have a growing necessity of being recreational, & while novelty is good, what really matters is an enjoyable & an engaging experience. A sequel can be a hit while a strikingly original game may not. The following paper discusses gaming in the eras gone by, the gradual evolution & upcoming of different Game engines & development tools, the near Realistic games of the 21st Century, and what the future has in store for Computer Games Developers & Enthusiasts.

II. Methodology

A. *Emperical Research*

The data showcased in this paper has been taken from a Racing game project I worked on. The models used in the Racing game were designed in AutoDesk MAYA, which were exported as DirectX objects & integrated in the game, with XNA 3.0 Framework, using Microsoft Visual Studio 2008. The Physics & the Math of the development have descriptively been explained.

B. *Secondary Research*

The books I used for my Research were O'Reilly – Learning XNA 3.0, Packt Publishing 3D Graphics with XNA Game Studio 4.0 & websites include www.riemers.net & www.emunix.emich.edu/ and Top 10 Usability Recommendations by Nokia Series 40 Game Study[1].

III. Data Used in the Paper

A. *Abbreviations and Acronyms*

XNA - **X**NA is **N**ot an **A**cronym - XNA Framework is based on the native implementation of .NET Compact Framework for Xbox 360 development and .NET Framework on Windows. Its versions include Game Studio 2.0, 3.0, 3.1 & 4.0 (primarily used for Game Development on Windows Phone 7 platform (including 3D hardware acceleration), framework hardware profiles, configurable effects, built-in state objects, graphics device scalars and orientation, cross-platform and multi-touch input, microphone input and buffered audio playback, and Visual Studio 2010 integration.[2]

AI – **A**rtificial **I**ntelligence – It refers to techniques used in computer and video games to produce the illusion of intelligence in the behavior of non-player characters. It may also refer to a broad set of algorithms that also include techniques from control theory, robotics, computer graphics and computer science in general.

SDK – **S**oftware **D**evelopment **K**it – It is a set of software development tools that allows for the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar platform. It may be something as simple as an Application Programming Interface (API) in the form of some files to interface to a particular programming language or include sophisticated hardware to communicate with a certain Embedded System.

B. Units

Metrics Systems here used are mainly in terms of pixels (px) of a screen. In case of Equations, SI (MKS) Metric System is used.

C. Equations

The Physics & Math of a Game is what defines the Virtual Reality. Taking the contemporary example of EA Sports FIFA, the reasons needed to know the physics equations are:

- Making a ball bounce (Co-efficient of restitution)

$$C_R = (v_b - v_a) / (u_a - u_b) \quad (1)$$

- Make a ball move in a direction with 'x' Newtons of force
- Distance travelled by a ball in give time period 't'

$$s = v*t + x_o \quad (2)$$

$$s = u*t + (0.5)*a*t^2 \quad (3)$$

- Make something have more or less friction so it will slide at a different speed.

$$f = u*M*g \quad (4)$$

Where, u = Coefficient of Friction

M = Mass of the object

g = Acceleration due to Gravity

- Collision responses - What direction will the ball be directed when hit. In such cases, Kinetic Energy & Linear Momentum will be conserved.

$$M_1*V_{i1} + M_2*Vi_2 = M_1*V_{f1} + M_2*V_{f2} \quad (5)$$

$$M_1*V_{i1}^2 + M_2*Vi_2^2 = M_1*V_{f1}^2 + M_2*V_{f2}^2 \quad (6)$$

- Gravity – Force between two objects

$$\text{Force, } F = G*(M_1*M_2)/ D^2 \quad (7)$$

Where, G = Gravitational Constant

D = Distance between the two objects

Game development is as data driven as art production. Math is like technique, the more the better, as long as you can apply it.

- Arithmetic
- Linear Algebra (with geometric interpretations)

$$\text{Distance travelled} = \text{sqrt}(a^2 + b^2 + c^2) \quad (8)$$

- Calculus
- Combinatory (randomized levels)
- Probability (balancing)
- Statistics (Bayesian esp. for AI)

IV. SCOPE OF RESEARCH

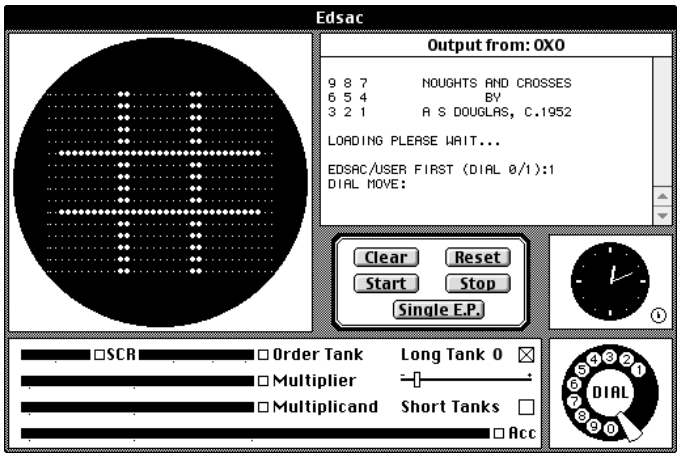
The world of Computer Gaming & its Development is vast; hence the study focuses on its History, Evolution, Design Principles, Game Design Process, Implementation of Physics, Types of Challenges, Game Business and its Future Expectations.

V. CONCEPTUAL FRAMEWORK

A. History of Computer Gaming& Industry

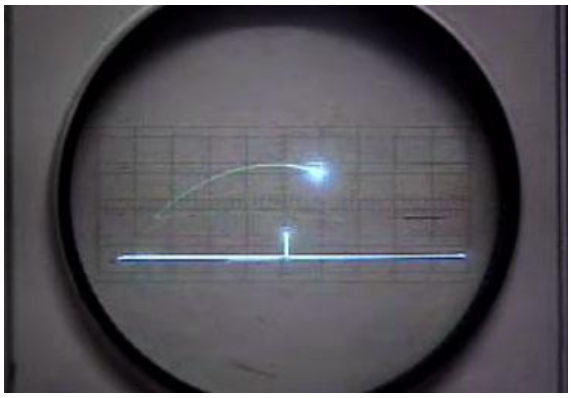
1) First "Games"

"OXO" (TicTacToe) was a computer game written for the EDSAC computer in 1952, an implementation of the game known as Noughts and Crosses in the UK, or tic-tac-toe in the United States. It was written by Alexander S. Douglas as an illustration for his Ph.D. thesis on human-computer interaction for the University of Cambridge. OXO was the first digital graphical game to run on a computer.[3]



"Tennis for Two" was a game developed in 1958 on an analog computer, which simulated a game of tennis or ping pong on an oscilloscope[4], created by American physicist William Higinbotham, it was one of the first electronic games to use a graphical display. He took an oscilloscope, what would be the first monitor, and a simple analog computer and put together a simple ping pong simulation using the computer to talk to the oscilloscope and put a bouncing dot of light and a bar which it bounced off of. The oscilloscope had been around for a little while but the computer was similar to the kinds which were used to coordinate the millions of bombs

dropped in World War II. Also, in order to program the physics of the game he used the current and much used protocol of missile trajectory plotting.



2) *First Generation Games*

The 70's saw a massive outbreak of ideas & implementation. One of the pioneers of such innovations was Ralph Baer. The idea began to form when he decided that he wanted to change the passive nature of the television into something more active, something that people could interact with. One day, while waiting for a tardy colleague in a New York City Bus Terminal, Baer sketched out his ideas for a TV based entertainment system. He later fleshed out his ideas in a four page paper. In the paper, he outlined his ideas for a low-cost system that would attach to a TV and display different games through the TV. His invention, the Light Gun, became the first video game in history to be filed for patent.

3) *Birth of Commercial Games*

In 1972, Nolan Bushnell started a company called Atari, & published a game by the name of Pong, which was hard wired into a wooden cabinet made for the sole purpose of playing Pong. It went on sale for $1,200 per unit.

4) *Arrival of Color*

The next revolution was the advent of color. In 1974, a Japanese company known as Namco bought out the Japanese division of Atari and thereby instantly made a name for themselves in video games. In 1979, they develop what would be the first color video game. Up until this point, all the color on video games was faked by overlays on certain parts of the screen to give the appearance of color. The game was called Galaxian and it was rampantly successful.[5]

5) *Subsequent Evolution*

In 1974, a company by the name Kee released the game Tank, the first game to use ROM. Tank featured far more advanced graphics than Pong in that it featured graphic memory which allowed for a higher degree of detail on screen. Also, in 1974, Atari launched the first racing game (Trak 10) & maze chase game (Gotcha).

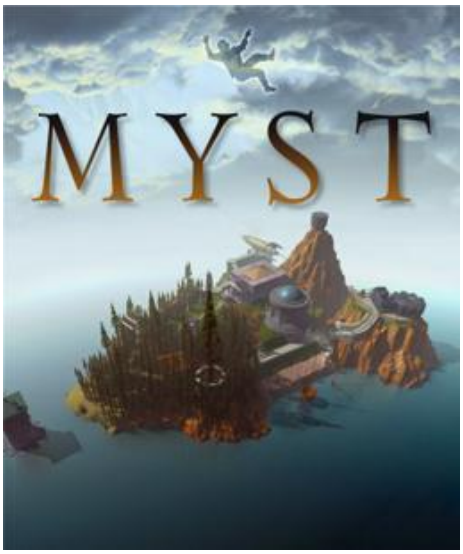
In 1980, a designer by the name Moru Iwatani developed a game that would be Pac-man. This name soon became in household use and was the first video game to be popular enough to warrant merchandising. There were Pac-man cereals, a Pac-man album, and even a Pac-man TV show.[6]

6) *Console Wars*

In 1990, Nintendo released Super Mario 3 - all-time best-seller & PC's and Consoles are major game platforms.[7] Electronic Arts acquired other game publishers. In 1991, Nintendo launched Super-NES (16 bit) & soon in 1992, PC gaming explodes as Nintendo has $7 billion in sales ($4.7B in US). It made higher profits than all U.S. movie and TV studios combined.

In 1994, Atari shipped Jaguar (64 bit), becoming one of the costliest consoles at around $700 (>$100/game).

Soon, DOOM, a first person shooter game was released by id Software and MYST, a graphic adventure video game, was released by Cyan Worlds, becoming the All time biggest selling PC game until 2002.[8]



7) *32 Bit Wars*

In 1995, Sony introduced Playstation (32-bit) & Microsoft released Window 95, including the Game SDK – DirectX, bringing major game performance to Windows.[9]

8) *Era of Playstations*

Playstation was launched in U.S. in the September of 1995, processing 300,000 polygons/sec at 30MIPS processor, with 4MB RAM & 2MB VRAM.[10]

9) *Pre-Modern Gaming Era*

1996 saw Nintendo introduced Ultra 6, with multi-player gaming going commercial via modem and internet and network companies. It was 1997 that saw 3D acceleration starting to standardize on 3D-FX & games start to assume 3D acceleration. Pentium II's enhanced to 200Mhz.[11]

In the September of 1996, Nintendo 64 is launched in US with 93.75 MHz & 64 Bit CPU with a 64-bit MIPS co-processor & over 500,000,000 - 16-bit operations/sec. It consisted of a built-in Pixel Drawing Processor (RDP) with 4.5MB RAM & 150,000 polygons/sec.[12]

In 1999, the Maximum Score for Pac-Man was achieved, when Billy Mitchell achieved the highest possible score for Pac-Man, as he completed every board and winded up with a score of 3,333,360.[13]

In 2000, development moves from PC to consoles as Playstation II is launched. Diablo II sold 1 million units in 1 week & SIMS sold 2.3 million units ($95M). In 2001, Nintendo launched the Gamecube & Xbox was launched by Microsoft.

The launch of Sony Playstation 2 on May 4, 2000 in Japan became the best-selling console of all time. By Feb 2011, 1.52 billion PS2 titles had been sold since launch. Hardware used by it was a 128 Bit 300MHz processor, 3 Special purpose 150 MHz co-processors, 32MB DRAM: 3.2 GB/sec, DVD & CD drive, MPEG2 hardware, a Dual Shock 2 analog controller; processing at 66M polygons/sec geometry – 16M polygons/sec for curved. Evidently, Software development became tough.[14]

10) *Modern Gaming Era*

November of 2001 saw the arrival of Direct X API, with Pentium IV 733 Mhz, a Custom 3-D 300Mhz GPU, 64MB Ram, 6.4 GB/sec, an 8GB hard drive; with sleek performance of 150 million transformed, 100+ million polygons per second sustained performance (shaded, textured), 300 million micropolygons/particles per second, full-scene anti-aliasing, 1920x1080 Maximum Resolution with HDTV support.

With arrival of more than ever PC games, in 2003, SIMS continued to grow, becoming the best-selling PC game of all time. WarCraft III, UT 2003 & GTA also took the market by a storm. Second Life and There.com are launched. EA bulked $2.5B in 2003.[15]

2004 became a year of sequels. Big Gaming firms increased their profits by launching SIMS 2, Halo 2, Half-life 2 & Doom.

2005 saw World of Warcraft take the whole Gaming Planet by storm, with 4 Million Subscribers at $700M/year subscriptions.[16]

11) *Post Modern Gaming Era*

November 2005 saw the arrival of Microsoft's XBox 360, introducing an entirely new dimension to Gaming experience. The hardware included a Custom IBM PowerPC CPU with 3 symmetrical cores - 3.2 GHz each, with 2 threads/core, VMX-128 vector unit/core, Custom ATI Graphics Processor with 10MB DRAM, processing 500 million triangles/sec, 16 gigasamples/sec, 48 billion shader operations/sec, supporting 16:9, 720p or 1080i – HD output, 512 MB of 700MHz GDDR3 RAM – unified memory architecture, 22.4 GB/s interface bus bandwidth, 256 GB/s memory bandwith to EDRAM & 21.6 GB/s front-side bus. It had an Overall system floating-point of 1 TeraFlop.

Playstation 3 influenced the market equally, with in-built Graphics by Nvidia, 550 Mhz GPU at 1.8 TeraFlops, 100 billion shader operations/sec, 51 billion dot products/sec. It consisted of 512MB RAM split between the CPU and graphics, with 512KB L2 cache & 7 AltiVec vector processing units.[17]

The PlayStation 4 is expected to sport a 1.5 TB hard disk drive, which allows users to store a large number of games. It will have USB 3.0 ports along with an HDMI connect port. The Sony PlayStation 4 will have full 3D support with 4K2K compatibility, and supports 3D Blu-Ray.

B. ***Principles of Game Design***

Games are a part of lives that have a story, one which is interactive & demands participation. Computer Games are a completely new form of entertainment, with completely new worlds to play in, allowing players to take on a new persona.

A game is a representation of physical objects – real or imaginary, such as a Terrain, Buildings (exterior and interior – walls, floors, etc.), Game objects (furniture, balls, fluids, weapons, vehicles, etc.), Animate objects (player, opponents, cars, etc), Providing dynamics to world, Physics, Behavior with Artificial Intelligence (AI), Supporting interaction, Graphics, Audio, such as dynamic sound, music & speech, and Networking.

ENGINEERING: TRADITIONAL VS. GAME

Traditional engineering	Game engineering
• High precision • Realism	• High speed • Low memory • No spiking in resources • Scalability • Believability • Control • Low Cost Development

True for graphics, physics, AI, audio, etc.

1) *Good Gameplay*

Challenges

Interactivity

Feedback about position relative to goals

Interesting choices required to achieve goals

Consistency and fairness

Avoid repetition

2) *Pursuing and Achieving goals*

There is always something to achieve & the Character is always achieving something. Often, there are three levels of goals with rewards-

Long-term goal - "I can conquer the world."

Medium-term goal - "I can take over a city."

Short-term goal - "I can win a battle."

For different genres of Games, there can be different Goals, such as-

Eliminate other players - Action games

Score points - Sports games

Get somewhere first - Racing games

Solve puzzles - Adventure games

Gain territory - Strategy games

Improve abilities - Role-playing games

Develop social relationships - Massively multiplayer games

Play God – Simulation

3) *Gameplay Consistency and Fairness*

There is consistency in the actions and associated outcomes for trying to achieve goals

Must be a reason for failure (or success)

Not arbitrary: Players know what to expect and can plan, e.g. - A pinball game uses "pinball" physics all the time.

Don't solve problems by unique & unlikely actions.

Don't break suspension of disbelief.

No "dead man" walking.

Fairness - Player should think they have a fair chance – game balance.

Can still be plot twists, but must be explainable.

4) *Gameplay Sins*

Poor production - Break the suspension of reality

Linear plot/gameplay - Player's actions don't affect how the plot progresses.

Micromanagement - Player is forced to perform menial tasks & AI should take care of all the obvious choices.

Repetition - Player must do same action over and over again, Player must sit through same cut scenes every time they play, have to replay 90% of level to fight boss, etc.

Doesn't track user's learning curve - Should start easy and get harder as game progresses.

Poor game balance - Same strategy always works & Trial and error is not fun Gameplay.

Not enough variety - Same graphics, objects, monsters, level design, sounds.

Awkward user interface

Limited feedback - Player is confused about goals, Player is confused about current progress to goals, no maps or help.

Inconsistency in story - There are no compelling and consistent goals for the player.

Dead and you don't even know it.

5) ***Shigeru Miyamoto Design Principles***

Shigeru Miyamoto is mainly known for his work at the video game production company Nintendo, where he created some of the most successful video game franchises of all time, including *Mario*, *Donkey Kong*, *The Legend of Zelda*, *Star Fox*, *F-Zero*, and *Pikmin*.[18][19] The Principles are-

- Start with a simple concept
- "running, climbing, jumping"
- Design around the computer's limitations
- Character wears dungarees so easier to see arms move
- Wears a hat because don't have to have hair
- Has mustache because couldn't draw nose and mouth
- Minimize the player's confusion
- What to do should be clear without consulting a manual
- The importance of play testing
- Incorporate a smooth learning curve
- Accommodate all skill levels

6) ***Sid Meier Design Principles***

Sidney Meier is a Canadian-American programmer and designer of several popular computer strategy games, most notably *Civilization*. He has won accolades for his contributions to the computer games industry. Meier is a Director of Creative Development for computer game developer Firaxis Games.[20] The Principles are-

- Player should have fun, not designer, programmer, or computer.
- Begin your game with a great first few minutes.
- Great game-play is a stream of interesting decisions that the player must resolve.
- The inverted pyramid of decision making lets the users make a few decisions to deal with first, and then let them multiply until the player is totally engrossed.
- Put the player in his dreams, where he/she is the hero.

7) ***Nokia Series 40 Game Study: Top 10 Usability Recommendations*** **[1]**

- Provide a Clear Menu Structure
- Simplicity Is Key -If two solutions are equally valid, use the simpler.
- Provide Help When Needed
- Be Relentlessly Consistent - Use the mother tongue of the user; Be consistent with the phone's UI, with game industry conventions, and within the game itself.
- Don't Waste the User's Time
- Use Natural Controls
- Enable Save and Pause - Provide a simple save-game feature. Have the game auto-save when the user presses the red phone button - use the destroyApp() method to do this.
- Conform to Real-World Expectations – e.g. when jumping or throwing objects, the flight path should be predictable. There must be no invisible barriers that the player cannot pass or holes that he cannot reach. Do not end the game arbitrarily. Implement a realistic physics model, if relevant (e.g. racing games).
- Go Easy on the Sound
- Implement a High Scores List

C. ***Game Design Process***

1) ***Idea for a Game***

Most games begin with a single idea. Idea can revolve around - A character [James Bond], Gameplay/Genre [A twitch FPS, an RTS game], a sport [Football, Baseball, Ice Hockey], a story/quest/goal [A time-travel adventure], a new technology [Motion capture of pro basketball players]. Along the same line, idea may be original, old, or hybrid – like SIMS, Civilization, FIFA

2) ***Inspiration***

For inspiration, mix existing ideas from other games. Steal ideas (but not characters) from other media, books, movies, comics, etc. Market Research - surveys, focus groups, sampling, case study, critical theory, etc. can prove helpful. Take a contemporary idea and make it better. Better technology - graphics, sound, AI, can be implemented. Better story should be devised in a different environment.

3) ***Interactivity is the Raison d'être of Computer Games***

Ask "What is the player going to do?" - This question comes before all others.

Do not get sidetracked with story, character, core mechanics, artwork or ANYTHING else until you know the answer to this question.

4) ***The Player's Role***

Who is the player trying to be - Critical for representational/realistic games

In single game may have multiple roles/multiple modes – In Football – manager, coach, player

If you can't describe it clearly, it will be confusing for the player

5) ***Interaction Model***

As an avatar - A single character or object that represents the play, Player's actions are limited to the avatar's location

Omnipresence (by not necessarily omniscience) - Player can act in many or all places in the world, Chess is an obvious example

6) ***Perspectives***

First-person - Doom, Half-Life, halo, Call of Duty

Third-person – Max Payne, Resident Evil 4

Side scrolling - Sonic

Aerial – isometric/top-down – HAWX, FIFA 12

Context sensitive - Resident Evil

DIMENSIONS: PHYSICAL VS. TEMPORAL

The Physical Dimension	The Temporal Dimension
• Dimensionality - 2D, 3D, 4D (multiple 3D spaces) • Scale - How big is the virtual world, How big are things relative to each other • Boundaries - What happens at the edge of the virtual world	• *Is time meaningful* - Does the passage of time itself change the game • *Real time* or turn based • *Variable time* – eg.- in The Sims, time speeds up while people sleep. • *Anomalous time* - Time goes faster for some things than others. • Can the player adjust time – flight simulators and RTS game

7) ***Types of Challenges***

Physical Challenges - Speed and reaction time (twitch games), Accuracy and precision (steering and shooting), Timing and rhythm (dance games), Learning special moves (fighting games)

Races – achieving something first

Logical challenges (puzzles) - Should be based on an underlying principle, Trial-and-error solution is a sign of bad design,

Exploration Challenges - Locked doors and traps, Mazes and illogical spaces, Teleporters

Conflict - Strategy, tactics, Logistics, Survival and reduction of enemy forces, Defending vulnerable items or units, Stealth

Economic Challenges - Accumulating wealth or points, Efficient Manufacturing, Achieving balance or stability in a system, Caring for living things within a system

Conceptual Challenges - Understanding something new, Deduction, observation, interpretation, Detective games offer conceptual challenges

8) ***Positive Feedback***

Positive Feedback is an achievement that makes subsequent achievements easier, like taking an opponent's piece in chess. Without positive feedback, it is too easy to get stalemate. It must be controlled to avoid giving the lead player too much advantage. Examples of Positive Feedback include getting ahead in a race, more likely to get power-ups or special scores; in Monopoly – get houses, more likely to get even more money.

9) ***Negative Feedback***

Negative Feedback increases the impact of chance - if chance is fair, it helps as much as hurts. Define victory in non-numeric ways - Chess is not won by taking away the most pieces. Increase the difficulty level as feedback kicks in, which happens in role-playing games. Examples of Negative Feedback – while getting ahead in a race, user is more likely to lose, due to Drafting Car Racing.

D. ***Game Physics***

1) ***Why Physics?***

Some games don't need any physics. Games based on the real world should look realistic, implying realistic action and reaction. Complex Games need more physics- such as sliding through a turn in a racecar, running and jumping off the edge of a cliff. [21]

For Newtonian physics with Rigid bodies, equations (1), (2) & so on can be applied.

2) ***Position and Velocity***

Modeling the movement of objects with velocity - Where is an object at any time t? (Assume our metric is pixels)

Taking equation (2) into account -

player_x(t) = t * x_velocity + x_initial

player_y(t) = t * y_velocity + y_initial

Hence, for Computation, equation would be comprehended as -

player_x=player_x + x_velocity

player_y=player_y + y_velocity

Where, x_velocity & y_velocity are the vectors in the X & Y direction, travelled by the player from the Original Position.

3) ***Acceleration***

For Computation, equation would be written as -

x_velocity=x_velocity + x_acceleration

y_velocity=y_velocity + y_acceleration

Changing acceleration can be computed as:

x_acceleration = cos (Θ) * acceleration

y_acceleration = sin (Θ) * acceleration

Acceleration comes into account in case of Racing Games such as Need for Speed Series.

4) ***Gravity***

Taking equation (3) & (7) into account -

$v(t) = 0.5*g*t^2$

$g = 9.8\ m/s^2$

For Computation, equation would be written as – (from equation (7))

x_velocity = x_velocity + 0

y_velocity = y_velocity + gravity

Gravity comes into account in case of Aerial games, like HAWX, Flight Simulator, or Space based Games.

5) ***Friction***

Friction *(Equation (4))* comes into account in case of Complex Racing Games; examples include Need for Speed Series, Billiards (Pool), FIFA.

6) ***Momentum and Energy***

Conservation of Kinetic Energy & Linear Momentum will come in case of an occurrence of a Collision. Equation (5) & (6) can be used, accordingly.

E. ***Game Business***

1) ***Shape of the Industry***

Hardware - Sony, Nintendo, Intel, IBM, Microsoft

Software – Electronic Arts, Activision, Sony, Microsoft, UbiSoft, THQ, Vivendi, Warner Bros.

Internet - Sales, updates, multiplayer versions of games, massively multiplayer games.

2) ***A Hit-Driven, Entertainment Business***

The interactive entertainment business is ENTERTAINMENT - It is NOT a packaged goods business. Games generate emotional responses, and are designed to fulfill fantasies, provide escape from reality, and stimulate the senses.

3) ***Top 10 Best-Selling Games of 2011***[22]

1. *Battlefield 3* (360, PS3, PC) Electronic Arts –2 million
2. *Batman: Arkham City* (360, PS3) Warner Bros. Interactive – 1.5 million
3. *NBA 2K12* (360, PS3, PSP, Wii, PS2, PC) Take Two Interactive
4. *Rage* (360, PS3, PC) Bethesda Softworks – 550k
5. *Just Dance 3* (Wii, 360) Ubisoft
6. *Dark Souls* (PS3, 360) Namco Bandai Games
7. *Madden NFL 12* (360, PS3, Wii, PS2, PSP) Electronic Arts
8. *Forza Motorsport 4* (360) Microsoft
9. *Gears of War 3* (360) Microsoft
10. *FIFA Soccer 12* (360, PS3, Wii, PSP, PS2, 3DS) Electronic Arts

F. ***Game Releases***

1) ***Alpha*** **-** When all the features are in, but not all bugs are out.

2) ***Beta-*** Development team believes all the bugs are out and No new features except ones to eliminate huge problems.

3) ***Release*** **-** No new features and Everything is done.

G. ***The Future of Game Development***

1) ***World of Mass Development (WMD)***

New platform for games creation that allows Developers to submit ideas to an active gaming community, raise the funds needed to develop them, get continual feedback from community team members that can play work-in-progress builds, use the WMD Portal to promote their project, get help from other developers, and ultimately launch their game to an eagerly-awaiting audience. It allows players to browse available projects and join any they are interested in, download and play regular builds of the game, participate in discussion & polls, speak directly to the developers, and eventually earn money back based on their contribution when the game is released.[23][24]

Project CARS is the first of those titles to use the WMD development system and represents the 'Ultimate Driver Journey'.[25]

- *Franchise Mode* allows you to carve out a personalized career starting in the Karting world and then progressing on to whichever motorsport specialization you prefer including Rally, Touring Cars, Open-Wheel, GT, Le Mans, and many more!
- *Play CO-OP* with a friend as Driver/Co-Driver
- *Full Team Management* - Have a large number of friends? Create, manage, and compete together!
- Experience the excitement of *Pit Stops* like you've never seen before.

- *10+ Game Modes* covering every form of motorsport
- *Cloud-Based Social Network* - allows you to connect with friends, compare times & scores, compete and challenge each other, and share content
- *User-Generated Content* – Create your own liveries, decals, tuning setups, and even events! Then share them with the world – either for free, in-game credits or even real money!
- *Pushing To The Limits* – Advanced physics, lighting, and AI

2) *Low level and API-free programming*

The limiting factor on PC is the performance overhead of the 3D API (mainly DirectX) while on consoles, game developers can use low level code to process more triangles than on PC. More render calls allow more creativity freedom for game designers. The solution would be to have a low level access to PC graphics hardware (direct-to-metal programming). Either way, it looks as though DirectX's future as the primary gateway to PC graphics hardware is no longer 100 per cent assured, especially when it comes to cutting edge graphics. Maybe there's an opportunity for an API like OpenGL- thanks to extensions, hardware vendors could offer new OpenGL extensions to have low level access to the GPU.[26]

3) *HTML5 is the Future of Social Game Development*

HTML5 is the future of social game development, especially on Smartphones because just about every phone and browser out there supports the web development platform. If a game is developed with HTML5 on one platform, developers can take that game almost anywhere that supports the language. The iPhone, Android, WebOS, Backberry 6.0, Bada and Nokia's Symbian and Meego, all support HTML5 apps. Even Windows Phone 7 will roll out support in the next year. The highlight would be WebGL, Canvas and WebSockets, which have given developers the opportunity to flaunt their creativity by manipulating images, creating 3D environments and providing real-time interaction.[27]

Acknowledgment

I owe my deepest gratitude and humblest appreciation to Prof. Matthew P. Evett, Professor, Department of Computer Science, Eastern Michigan University, whose Learning Material on Game Programming helped me elucidate various doubts in the vast realm of Game Development & Designing and inspired me to the overall success of the undertaking.

I would also like to thank my Mother for her support, both financially & emotionally.

And above all, to the Almighty God, who never cease in loving us and for the continued guidance and protection.

References

[1] Nokia Series 40 J2ME™ Game Usability Guidelines and Implementation Model. *(references), Version 1.0; June 23, 2003.*

[2] XNA Game Studio - http://en.wikipedia.org/wiki/Microsoft_XNA#XNA_Game_Studio_Express

[3] First Game - http://en.wikipedia.org/wiki/OXO

[4] First Digital game - http://en.wikipedia.org/wiki/Tennis_for_Two

[5] First Coloured Game - http://en.wikipedia.org/wiki/Galaxian

[6] First – Most popular Game - http://en.wikipedia.org/wiki/Pac-Man

[7] All time best Seller - http://en.wikipedia.org/wiki/Super_Mario_Bros._3

[8] History of Myst (1993) - http://myst5.com/mysthistory.html

[9] History of DirectX & API's - http://en.wikipedia.org/wiki/DirectX#History

[10] Emergence of Playstation 1 - http://en.wikipedia.org/wiki/PlayStation

[11] Ultra Weekend 6 - http://en.wikipedia.org/wiki/Ultra.Weekend_6

[12] Nintendo 64 - http://en.wikipedia.org/wiki/Nintendo_64

[13] Highest possible score for Pac-Man achieved - http://en.wikipedia.org/wiki/Billy_Mitchell_(electronic_sports_player)

[14] Emergence of Playstation 2 - http://en.wikipedia.org/wiki/PlayStation_2

[15] PC Games in 2003 - http://en.wikipedia.org/wiki/2003_in_video_gaming

[16] Emergence of World of Warcraft - http://en.wikipedia.org/wiki/World_of_Warcraft

[17] Emergence of Playstation 3 - http://en.wikipedia.org/wiki/PlayStation_3

[18] The Art of Game Design, by Jesse Schell - http://books.google.co.in/books?id=LP5xOYMjQKQC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false , p. 309-329.

[19] Principles of Game Design - http://ai.vancouver.wsu.edu/~wallaces/courses/slides/game-design/GameDesignPrinciples136.pdf

[20] The Art of Game Design, by Jesse Schell - http://books.google.co.in/books?id=LP5xOYMjQKQC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false , p. 171-206.

[21] Game Physics, By David H. Eberly, Ken Shoemak – http://books.google.co.in/books?id=a9SzFHPJ0mwC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false , p. 13-82

[22] Top 10 Games of 2011 - http://www.videogamesblogger.com/2011/11/15/top-10-best-selling-games-of-october-2011-usa.htm

[23] WMD - http://www.wmdportal.com/

[24] WMD-http://www.slightlymadstudios.com/world-of-mass-development/

[25] Project CARS - http://www.wmdportal.com/projects/cars/

[26] http://www.bit-tech.net/hardware/graphics/2011/03/16/farewell-to-directx/1

[27] http://blog.games.com/2010/10/14/html5-is-the-future-of-social-game-development-moblyng-ceo-cl/

[28] Extenal References – Game Development for the PC, Xbox 360 and Windows Phone 7 – Learning XNA 3.0 – Aaron Reed – by O'Reilly

[29] Extenal References – Game Development for the PC, Xbox 360 and Windows Phone 7 – Learning XNA 4.0 – Aaron Reed – by O'Reilly

[30] External References – 3D Graphics with XNA Game Studio 4.0- Sean James.

****