Project Title : An Aesthetic 2d Plat-Former based game with an Astute AI and its Quality Evaluation

Group Members :

1.Tejas Deshmukh
2.Gouresh Kochare
3.Preet Gandhi

GAME DEVELOPMENT

Literature Review

#	IEEE Paper	Orientation	Author	Idea that we took
1	Measuring Quality of Indie Game developed using Unity Framework (Base paper)	Game coding	Mateo Bosnjak & Tihomir Orehovacki	Basic player movement & Quality Evaluation
2	Researching on AI Path-finding Algorithm in the Game Development	Game coding	Yanyan Cao	Implementation of A* Path find
3	Character Design as Bridging Tools of Ideological Message in Game	Game design	Tubagus Zufri, Dodi Hilman, Wahyudi Pratama	Designing of any game object
4	Drawing Equipments with Adobe Illustrator	Game design	Daniel Tofan	Using different tools to make animation sprites

GAME DEVELOPMENT

Game Design

- How the game looks like
- What the game will include
- What is the game all about

- ❑ Scope
- ❑ Technology Stack
- ❑ Benefits to environment
- ❑ Theme
- ❑ Applications
- ❑ Future Scope

- ➢ Preparing a questionnaire for quality evaluation

- ➢ Animations via Adobe Photoshop/Illustrator

Game Coding

- How the game will work
- The logic of the game
- Bug fixes

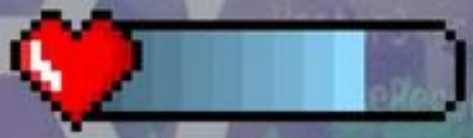
- ❑ C# sharp Scripts
- ❑ Game Assets
- ❑ Giving a meaning to animation

- ➢ Coding via Unity Game Engine
- ➢ Inbuilt Unity Physics Engine

GAME DESIGN

- A 2D Plat former Game
- Player on an odyssey
- Number of puzzles
- Number of levels with an increasing complexity
- Enemies and pitfalls
- Score system
- Health and die system
- Path finding mechanism



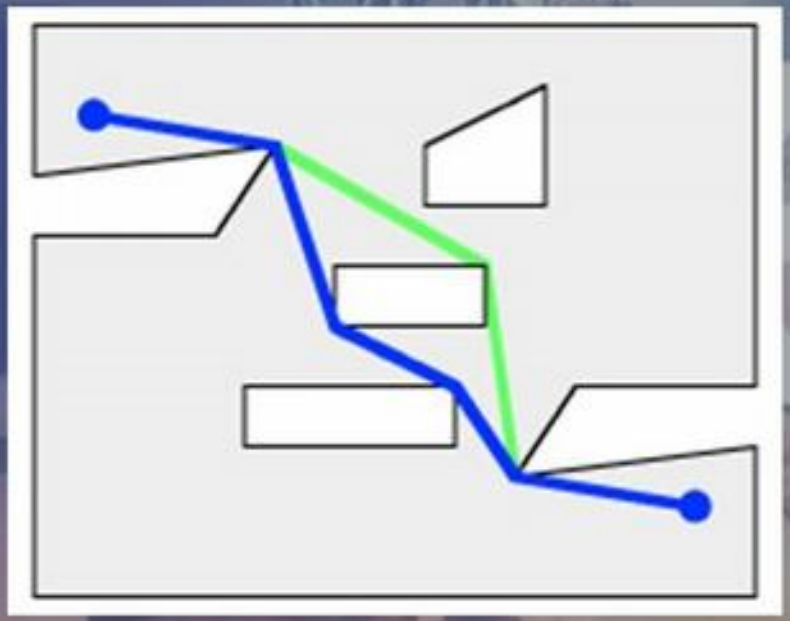
GAME DESIGN

Scope

- Basic Player & Enemy Movement
- Implementation of A* path finding algorithm of AI
- Score, Health, Die system
- Various Puzzles and Level advancement logic
- Making original animation, sounds and music
- Designing a Brief Questionnaire for Feedback and continuous improvements

GAME DESIGN

A* Path Finding AI algorithm



GAME DESIGN

Score, Health, Die & Traps, Puzzles

GAME DESIGN

Making Pixel Art Animations via Adobe Photoshop

Sprite Sheet

Shoot

Idle

Roll under

Climb

Double Jump

Run

Layer & Timer Components

GAME CODING


Scripts
PlayerMovement.cs
EnemyMovement.cs
Projectile.cs
DamageDealer.cs
AIPath.cs
Spawner.cs
Score.cs
Health.cs
Level.cs
Analytics.cs
Sfx.cs
BackgroundScroller.cs
Objects
Player
Enemy
Projectile
Pitfall
SceneLoader
AudioManager
ParticleSystem
Canvas
GameCamera
Background

GAME DESIGN

Technology Stack

Unity Game Engine 18/19

Adobe Photoshop/ Illustrator

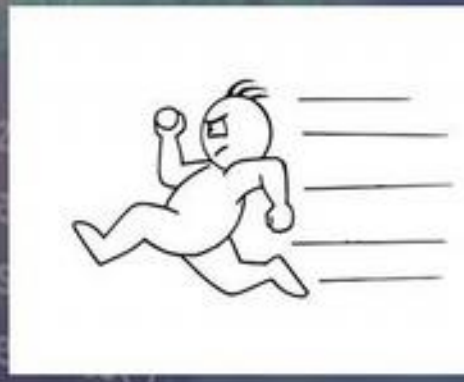
C# Programming Language

GAME DESIGN

Benefits to Society & Environment

- Stress Buster
- Logical thinking
- Boosting thinking process
- Puzzle solving capabilities
- Sense of accomplishment
- Hand-eye co-ordination
- Helps socializing

- Simulation of a Quagmire
- Making way through dangerous situations
- Enhances split-second decision making
- Boosts auditory perception



GAME DESIGN

Applications

Learning AID in software engineering

Entertainment

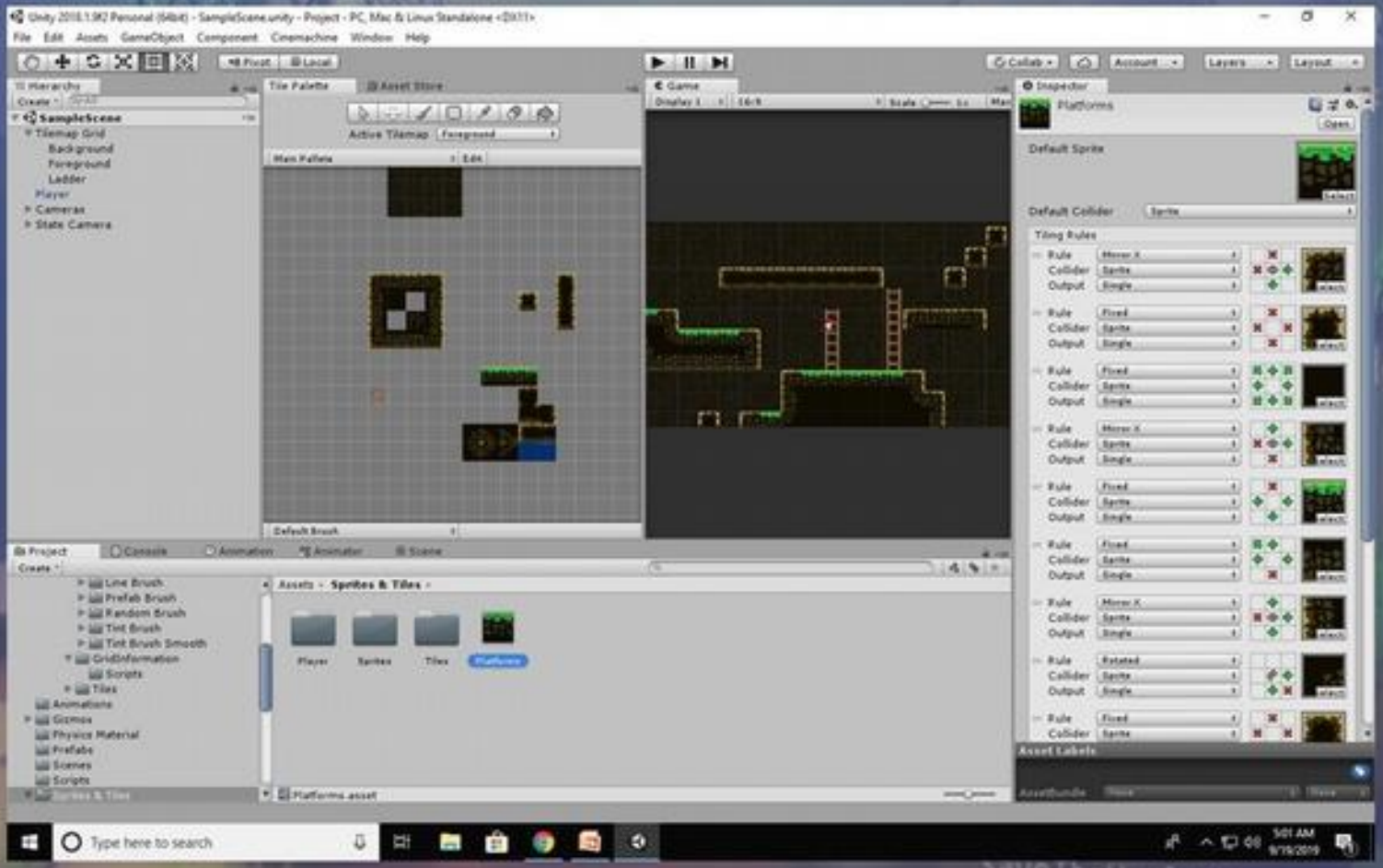
To create Music

IncrediBox or MusicLab, etc

Recording original sounds over the mic

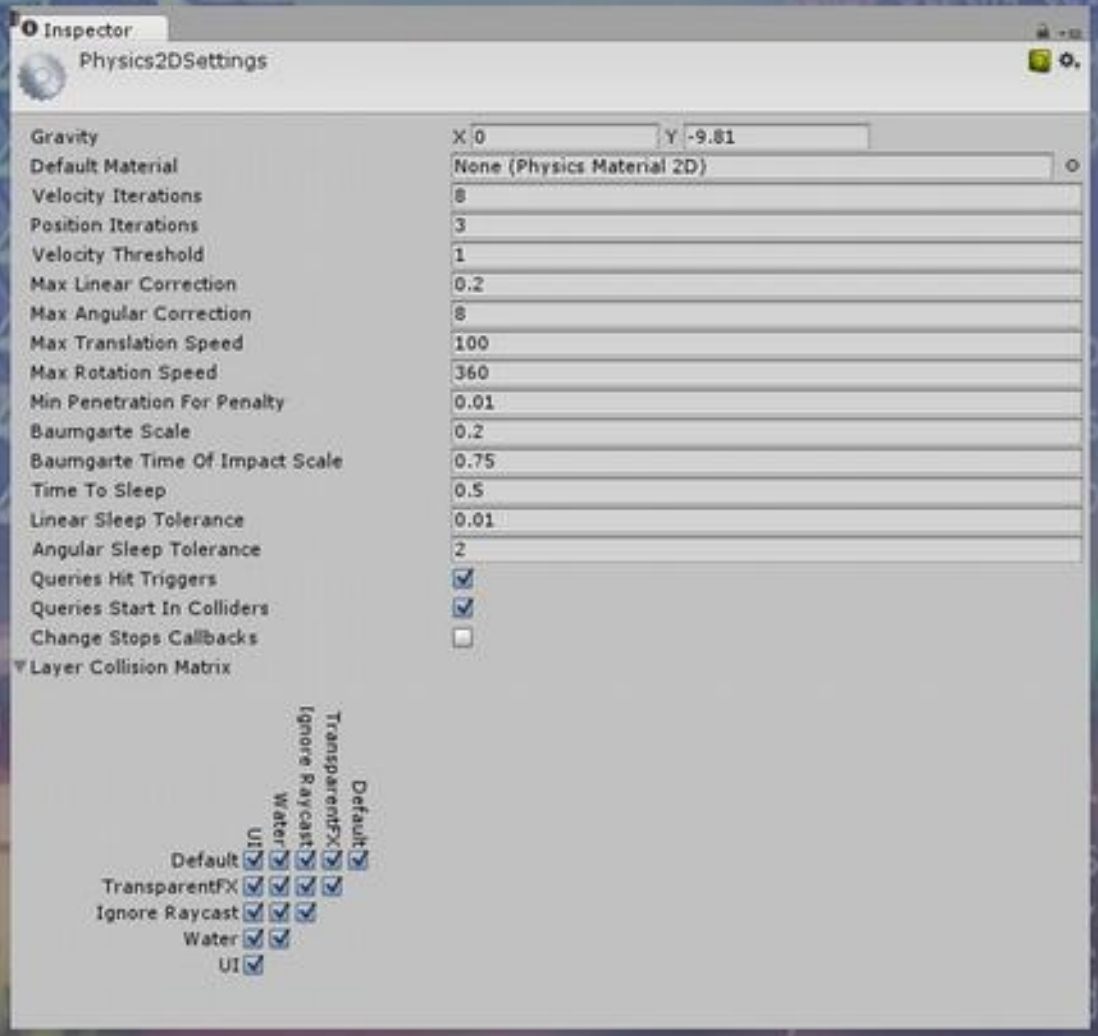
GAME CODING

Tile Maps



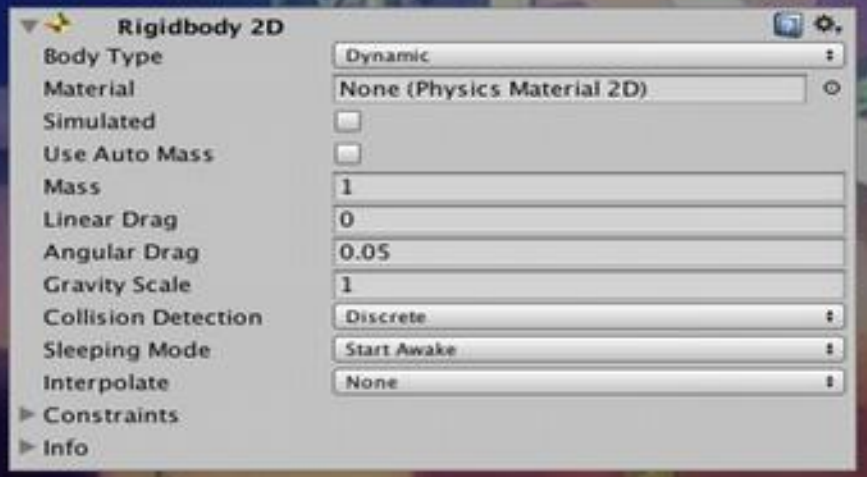
GAME CODING

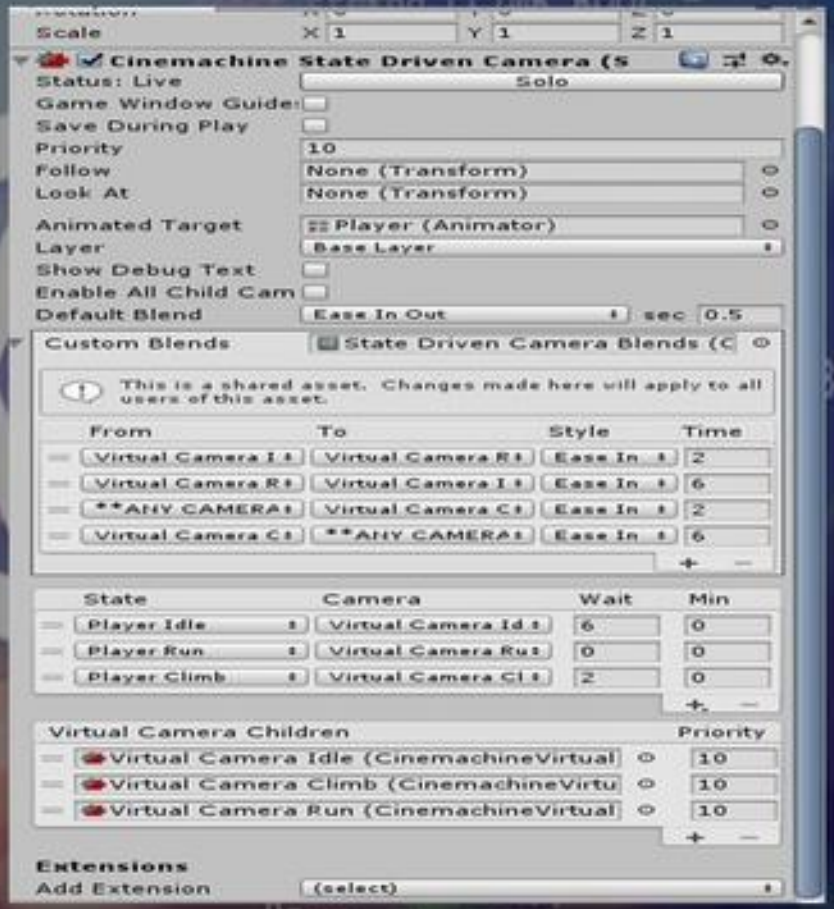
Physics 2D Engine & Layer Collision Matrix



GAME CODING

State Camera : Cinemachine & other components



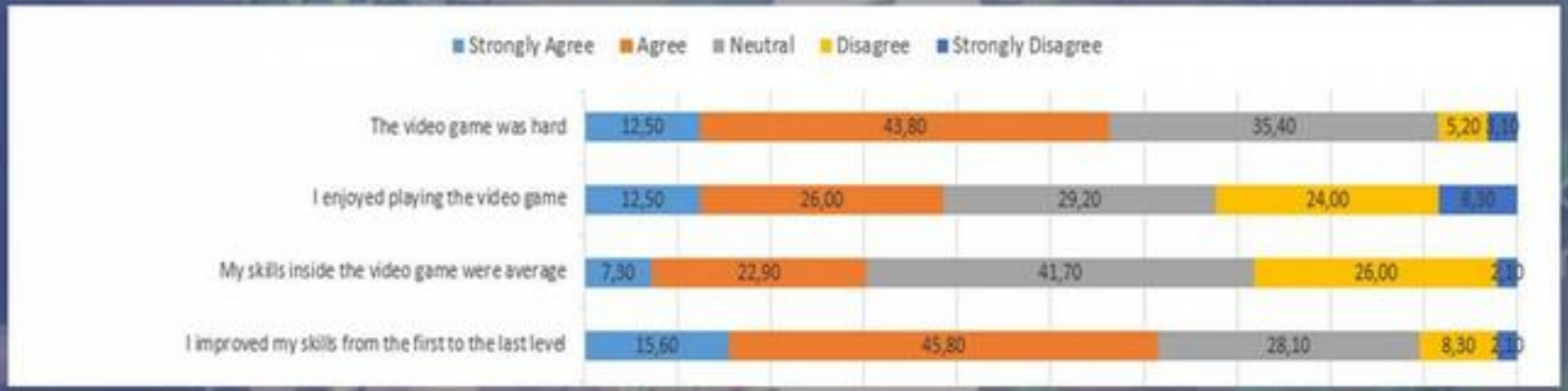


And Many other Components and concepts!

Virtual State Cameras

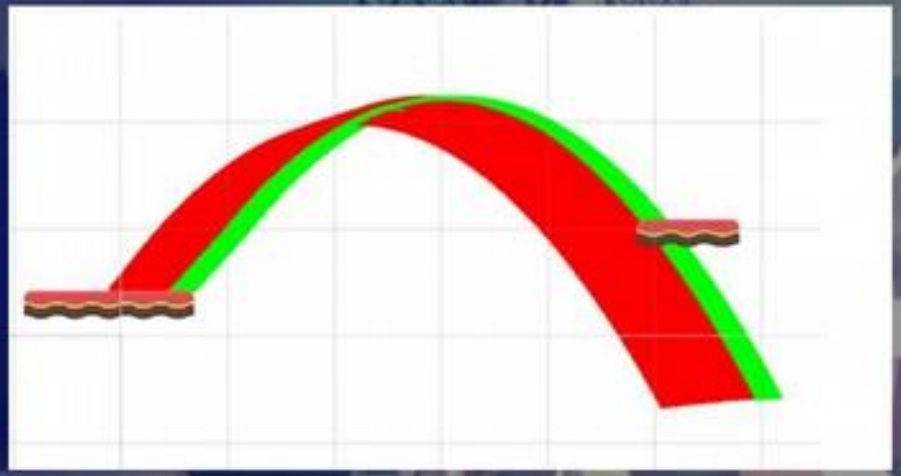
GAME DEVELOPMENT

Designing a Questionnaire



GAME DEVELOPMENT

Future Scope



An AI attached to player may help us classify the levels according to their level of complexity and also let us know if it is even possible to complete the level and if yes, know how hard the level actually is which has been designed by us.

TABLE I

A SUMMARY OF THE DATA COLLECTED FOR EACH JUMP TRIAL.

screen ID	Trajectory Type	#Jumps	#Successes	%Success
0	Simple	156	136	0.872
1	Simple	131	80	0.611
2	Simple	158	114	0.722
3	Simple	159	104	0.654
4	Simple	153	33	0.216
5	Reentrant	150	53	0.353
6	Reentrant	139	76	0.547
7	Reentrant	134	43	0.321
8	Reentrant	148	32	0.216
9	Trivial	157	139	0.885
10	Falling	151	142	0.940
11	Falling	156	143	0.917
12	Trivial	157	156	0.994
13	Falling	140	128	0.914
14	Falling	141	93	0.660
15	Simple	131	64	0.489


Thank you
[SerializeField] float runSpeed
[SerializeField] float jumpSpeed
[SerializeField] float climbSpeed
const string CLIMB_BOOL =
string JUMP_TRIGGER =
string LADDER_TAG =
Vector3()
SpriteRenderer spriteRenderer;
Animator animator;
MoveHorizontally();
ClimbLadders();
ProcessJump();
SaveTheWorld();