1.1. Game Play Screen

Picture 35: Gameplay screen

This is the main game play screen, player has indicators to follow.

In the picture 36, the explanation of the systems used in the game for the player.

Picture 36: Instructions Screen

Power Bar shows throwing speed of horseshoe, to throw horseshoe player should hold the Space button. It fills Power Bar with green color, as much as hold it has more power.

Score shows how many points player has at the moment. Scoring system will be explained deeper in the development chapter..

Horseshoe Number is basically shows how many horseshoes left.

Target Field is a helping indicator for player; it marks the bar with a red cross which is hit by horseshoe.

Horizontal Speed is a feature for aiming. When it grows on the right side, it gives horizontal speed to +X direction (to the right in the picture). When it grows on the left side, it gives horizontal speed to –X direction (to the left in the picture). This change automatically independent of player control. This system explained deeper in the development chapter.

Changing Points is a feature for scoring system. It changes automatically in the range of 1 and 20. When horseshoe hits to the bar, player gets this score. This system explained deeper in the development chapter.

2. Development

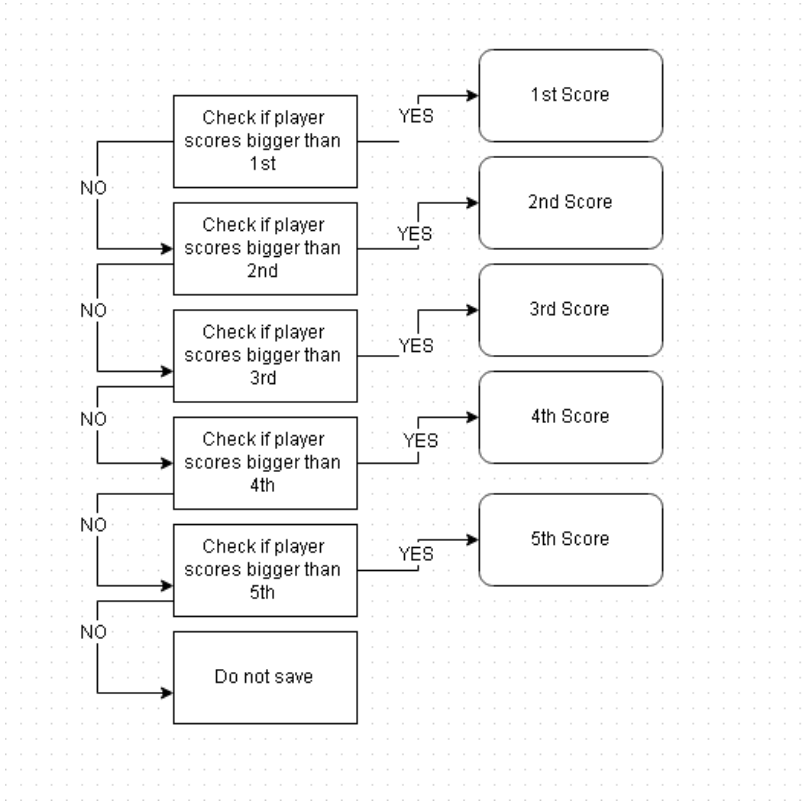
2.1. Scoreboard

Scoreboard lists the top 5 best score in Texas Horseshoe Game.

Picture 37 is a view from scoreboard screen.

Picture 37: Score board screen

Script works in this system: When the game ends, last score gets saved then script compares the players score with top 5 score starting from the 1st best to the 5th best. If player's score is in this range, it puts inside top 5 and sort it. So the last previous 5th best score disappears. An example, let's say player gets 80 points, according to picture x, player's rank will be 4th. So "qwerty – 32 points" record will be changed with "aaaa- 52 points" and player's new score will be saved to 4th rank.



Scheme 1: Score board workflow

```
else
if (ScoreShow.Scoreall > PlayerPrefs.GetInt("2ndScore"))
{

PlayerPrefs.SetInt("5thScore", PlayerPrefs.GetInt("4thScore"));
PlayerPrefs.SetString("5thPlayer", PlayerPrefs.GetString("4thPlayer"));

PlayerPrefs.SetInt("4thScore", PlayerPrefs.GetInt("3thScore"));
PlayerPrefs.SetString("4thPlayer", PlayerPrefs.GetString("3thPlayer"));

PlayerPrefs.SetInt("3rdScore", PlayerPrefs.GetInt("2ndScore"));
PlayerPrefs.SetString("3rdPlayer", PlayerPrefs.GetString("2ndPlayer"));

PlayerPrefs.SetInt("2ndScore", ScoreShow.Scoreall);
PlayerPrefs.SetString("2ndPlayer", PlayerName.text);
Player2nd.text=PlayerName.text;
IsChanged=true;
}
else
if (ScoreShow.Scoreall > PlayerPrefs.GetInt("3rdScore"))
{

PlayerPrefs.SetInt("5thScore", PlayerPrefs.GetInt("4thScore"));
PlayerPrefs.SetString("5thPlayer", PlayerPrefs.GetString("4thPlayer"));

PlayerPrefs.SetInt("4thScore", PlayerPrefs.GetInt("3thScore"));
PlayerPrefs.SetString("4thPlayer", PlayerPrefs.GetString("3thPlayer"));

PlayerPrefs.SetInt("3rdScore", ScoreShow.Scoreall);
PlayerPrefs.SetString("3rdPlayer", PlayerName.text);
Player3rd.text=PlayerName.text;
IsChanged=true;
}
else
if (ScoreShow.Scoreall > PlayerPrefs.GetInt("4thScore"))
{
```

Example Code 1: Score keeping system

2.2. Highscore Saving Scene

A game view from the scene in the picture 38.

Picture 38: High score saving scene

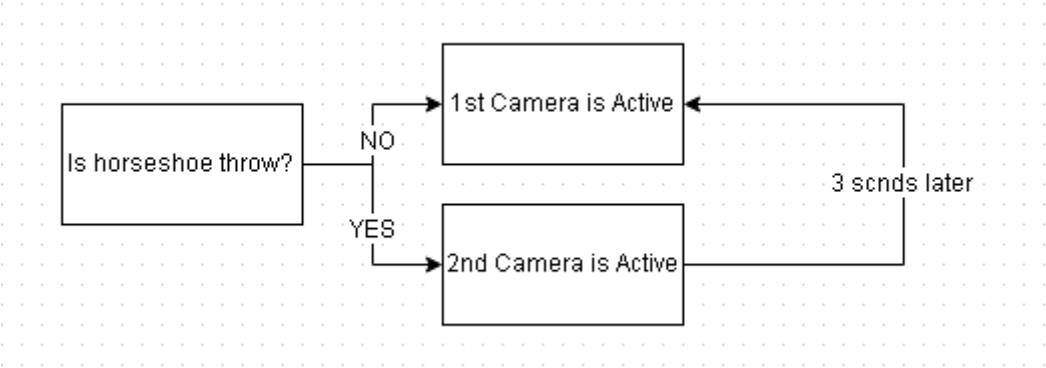
When the game ends, this scene appears. Related with the game's story, sheriff wants player's name. Score is shown to player and player is supposed to write his name into the text field. After this scene player is able to see his rank in the top 5 list.

```
var PlayerNameInput: UI.Text;


function Start () {
Screen.showCursor=true;
}

function Update () {


ScoreTable.PlayerName=PlayerNameInput;


if(Input.GetKeyDown (KeyCode.Return))

 {
//Compare();
Application.LoadLevel("HighScore");

 }

}
```

Example Code 2: High score saving system

2.3. Following Camera System

When player throws the horseshoe, 2nd camera activates and it follows the horseshoe until the destination approximately 3 seconds. So, 3 seconds later, 2nd camera deactivates itself and activates the 1st camera. 2nd camera gets it's velocity depending on the horseshoe velocity. It explained in the scheme 2.



Scheme 2: Following camera system workflow

```
function Follow()

{
FollowScnds+=1;


if (!LevelUp)
{
if(FollowScnds==3)
{
Cam.SetActive(true);
CamAbove.SetActive(false);
 Reload.ReloadPermit=true;
}}

if (LevelUp)
{
if(FollowScnds==4)                // at 2nd and 3rd level bars move further so the time is 1 scnd longer
{
Cam.SetActive(true);
CamAbove.SetActive(false);
 Reload.ReloadPermit=true;

}
}
```

Example Code 3: Following camera system

2.4. Bar Hit & Marking The Field

When the horseshoe touches the bar, a Boolean changes to true, a kind of switch. On the bar red cross appears which means it has been hit. This Boolean helps many features in the game such as level pass and triggering the other Booleans.

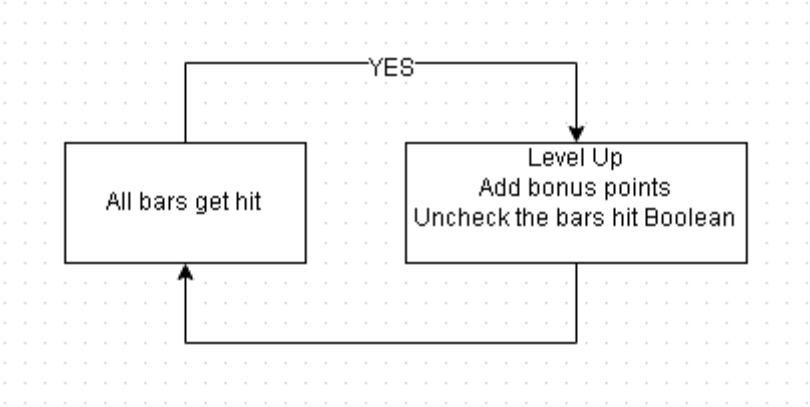
Picture 39: Marking field

```
var CamAbove: GameObject;


function OnCollisionEnter (collision : Collision)
{


if(collision.gameObject.tag == "Horseshoe")
{

    Level1Mission.Bar1Hit=true;
//      CamAbove.speed.z=5;

}

}
```

Example Code 4: Bar hit & marking field

2.5. Level System

Player can achieve a level only when all bars are hit.



Scheme 3: Level System Workflow

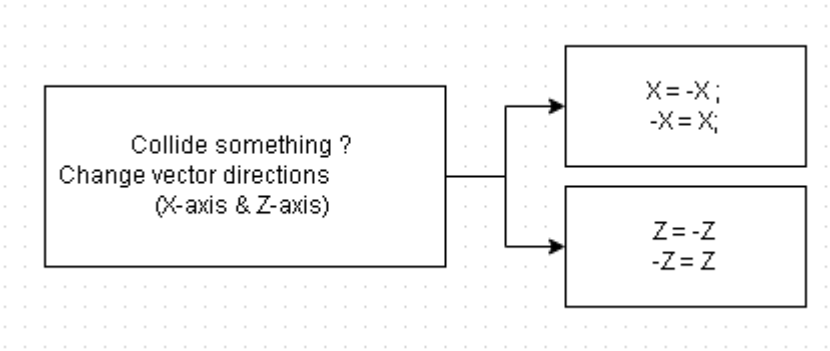
Texas Horseshoe Game has 3 levels. 1st level is the starting level. In the 2nd level, the distance between player gets longer. In the 3rd level, bars' heights get half.

```
function LevelUp2() {
Throw.DeactiveObjectt=true;       // Deactive the object , which is Reloaded horseshoe from previous level
 Reload.ReloadPermit=false;       // While leveling up, player cant reload
yield WaitForSeconds(3);

Bar1.transform.position.z+=50;     // Bars get further
Bar2.transform.position.z+=50;
Bar3.transform.position.z+=50;
Bar4.transform.position.z+=50;
Bar5.transform.position.z+=50;
Bar6.transform.position.z+=50;

ScoreShow.Scoreall+=(9-Reload.Ammo)*10;    // Left horseshoe number multiply by 10 and adds to score..
                                           //ex: 5 horseshoe left, 50 points to score

LuckyAmmo.LuckyShoe-=1;                   // 2nc charger
IsLevelUp2=true;                          // Level ip bools
Reload.Level2=true;
Reload.Level1=false;
Reload.Ammo=1;
GUIAmmo.ammo=9;                           // Horseshoes starts from 9 again
CameraFollow.LevelUp=true;
audio.PlayOneShot(OhYeah, 0.8);
GUIBarHit.IsLevelUp=true;
//Reload.ReloadPermit=true;
yield WaitForSeconds(1.0);
audio.PlayOneShot(Claps, 0.8);
GUIBarHit.IsLevelUp=false;

Throw.PowerBarShowEmpty=true;
}
```

Example Code 5: Level System

2.6. Moving Grass ball

Grassballs are moving around not so randomly. They have acceleration until they collide with a game object in the game world. They change the direction which explained in the scheme.



Scheme 4: Moving grass ball workflow

```
var forceAmount : Vector3;  // The force applying to the grassball


function Update () {

rigidbody.AddForce(forceAmount); // Applying the force to the object

}
function OnCollisionEnter (grassball : Collision) {


        forceAmount=-forceAmount;   // When grassball hits anyobject (except Horseshoe),
                                    //  force gets negative or positive, aim is moving back

    }
```

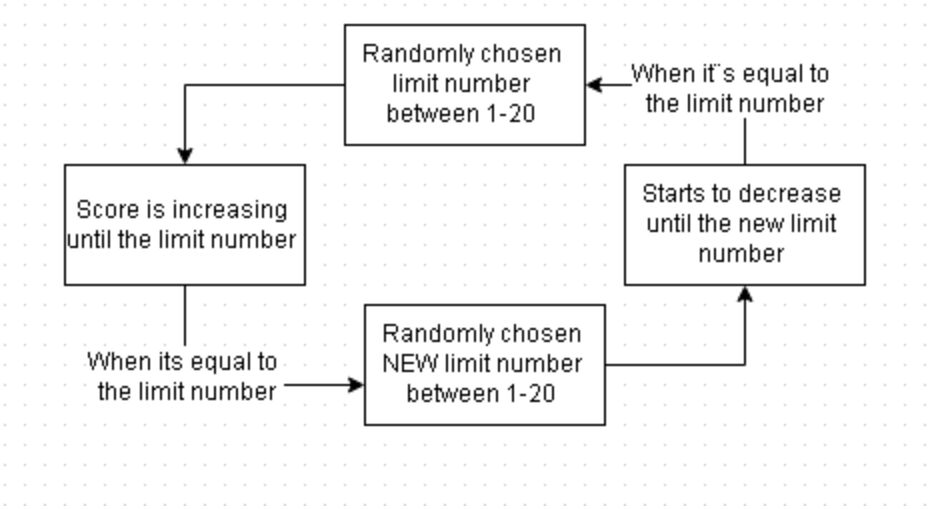
Example Code 6: Moving grass ball

2.7. Scoring System

When a horseshoe touches to a bar, player earns points according to changing score.

It goes up or goes down until the limit number between 1-20. When it comes to that number, it starts to decrease if it was increasing, increase if it was decreasing. A green arrow appears when score is increasing and a red arrow symbolizes decreasing as seen on the picture 40 and 41.

Picture 40-41: Scoring system



Scheme 5: Scoring system workflow

```
function Limit()
{
LimitNumberTemp=LimitNumber;            // Previous random number
LimitNumber=Random.Range(1,19);         // Randomizing the limit number


}
```

```
function Score()         // Score changing function
                        // Score changes randomly the number between 1-20 and it goes one up or on down.
{
if (GoingUp)
{
if(ScoreChange>LimitNumber)
{
Limit();
}
ScoreChange+=1;
if (ScoreChange==LimitNumber)
{
GoingUp=false;
GoingDown=true;
UpArrow.SetActive(false);      // Green and Red arrows .. active and false
DownArrow.SetActive(true);
Limit();
}
}

if (GoingDown)
{
if(ScoreChange<LimitNumber)
{
Limit();
}
ScoreChange-=1;
if(ScoreChange==LimitNumber)
{
GoingDown=false;
GoingUp=true;
UpArrow.SetActive(true);
DownArrow.SetActive(false);
Limit();
}
```

Example Code 7-8 : Scoring System

2.8. Throwing

Player holds the space button to give horseshoe speed velocity to the direction of Z-axis in the game world. When player release the button “throwing” process launches and this process triggers also many other Booleans in the game mechanic.

```
    if(Input.GetKeyUp (KeyCode.Space) && IsFlying==false )
    {

 CamAbove.SetActive(true);  // 2nd camera starts
 Cam.SetActive(false);  // 1st camera stops
 IsPowerBarUse=false;   // PowerBar cant change by pressing Space

CameraFollow.FollowScnds=0;   // 2nd camera timer starts

   IsFlying=true;
   IsCollided=false;
   rigidbody.isKinematic = false;
   transform.parent = null;
   rigidbody.velocity = transform.TransformDirection(speed);
   animation.CrossFade("Throw", 0.2f);


    if(PowerBarShowEmpty)      // PowerBar gets empty for next turn
    {
    FuncPowerBarEmpty();
    PowerBarShowEmpty=false;
    }

     }
```

Example Code 9: Throwing System

2.9. Main Menu Scene

Main menu is another important part in games. A simple menu consists a button group of “Start Game” and “Quit Game”. In pic x. an example of Texas Horseshoe game project.

Picture 42: Main menu scene

Since multiplayer feature is not ready the button is non-clickable.

```
10
11 function ScoreBoard()
12 {
13 Application.LoadLevel("HighScore");
14 }
15
16
17 function LevelStart()
18
19 {
20
21
22 Application.LoadLevel("Instructions");
23
24
25 }
26
27 function Quit()
28 {
29 Application.Quit();
30 }
31
```

Example Code 10: Main menu scene