



DESIGN OF FRUIT GAME USING UNITY 3D

Bandaru Mamatha

PG student, Department of CSE, Jawaharlal Nehru Technological University college of Engineering (Autonomous)
Ananthapuramu

Abstract: The goal of this straightforward single-player platformer game is to collect all the fruits in order to succeed. Taking that many fruits as one can is the main goal of this game. It is structured with the use of visual and aural elements, interactive items, and periodic updates to the display per second in accordance with the simulated situation. One element of computer games is the presence of music to boost emotional feelings that might activate the emotions so that it can be utilised as an alternative system for the disabled persons. This helps player understand the game's plot.

A straightforward GUI is offered for effortless gameplay, which the client won't find challenging to utilise or comprehend. The creation of this game made use of various photos, to talk about the creation of a series of design features in the form of unified packages, which have been discovered to significantly enhance fruit accessibility in video games. Concentrate on using Unity's built-in components to create additional behaviours using the C# scripting API and incorporating third-party components, such as 3D models, that are offered in the Unity asset store.

This fruits game system was created using the Unity software. Users of Unity can develop 2D and 3D games and experiences, and the engine provides a main script API in C# for both the Unity editor and the games themselves. Drag-and-drop capability is also available. It offers a comprehensive toolkit for making a game, containing protocols for graphics and audio, and it necessitates the least amount of external application use when working on a project.

Game designing and playing provide motivation like learn how to systematically deal with complexity and time management. This can boost the mind, feel stress relief, and may improve skills academically, social skills so that they built better relationships with others. It is also has been used as an exercise to improving and strengthening muscles. It provides good vision and improves the brain's performance.

Keywords: sprites, assets, prefabs

I. INTRODUCTION

There is no longer a set age group or consumer that the gaming sector targets. Thanks to the development of mobile gaming and improvements in the hardware needed to play these games, gaming has become a feasible form of recreation for users of all ages all backgrounds. As a result of this transition to the mainstream, the industry's revenues rose, reaching \$25.1 billion in 2010 from \$9.5 dollars in the United States in 2007 and 11.7 billion in 2008.

With improvements in technology, like hardware, graphics, and faster processors, the gaming business has expanded and improved. Modern video games are becoming increasingly resource-intensive, especially PC-based ones. Devoted players are among those who purchase strong PCs to keep up with the newest titles.

1.1 Gaming Categories

With time, video gaming evolved to include a range of platforms. Evolution has led to the emergence of a vast ecosystem of mobile games, more potent consoles, and PC games that are multi-player and more complex.

1.1.1 PC Gaming

On a computer, you can play video games using the keyboard, mouse, joystick, or gamepad. These games stand out for their increased input, processing, and output capacities as well as the absence of a central controlling entity. In the middle of the 1990's, PC games briefly lost market share to consoles, but thanks to digital distribution, they come back. Online and offline gaming are also options.

1.1.2 Video game console

A machine that used play video games is a video game console. A visual image that originates from the console is used to display the game. There are many different sorts of consoles, including handheld, micro, specialised, and home game consoles. The console has evolved into a single device that performs multiple functions including set-top box, web browser, CD and DVD player, and others.

1.1.3 Gaming on mobile

A game that is playable on a mobile device, as in a mobile, tablet, calculator, or smart watch, is referred to as a mobile game. Smartphones may download mobile games through an app store, whereas tablets come with them already installed. You can play games on the device or in the cloud.

1.2 Background

The project's goal is to create a fruit game, and the game itself should be an endless runner in which the player collects fruits in a basket to earn points.

1.3 Motivation

The main goal of the project is to build a continuous pattern of fruits by accelerating the playtime of a fruit game using a unity 3d. Fruits will appear in a top-down direction.

1.4 Problem Statement

The main issue is fruit control, which makes it difficult for the system to operate dynamically.

1.5 Objective

The proposed project is to build a 3D map and control the fruit with keys, must capture the fruit earn points. When the player leaves the fruits, the game is over. A straight forward GUI is offered for effortless gameplay, which the user won't find challenging to utilise or comprehend. For fruit collection, the player needs to be swift, agile and in charge of the basket.

II. Literature Survey

The primary purpose of the literature review is to investigate the project's past in order to uncover defects in the current system and provide directions for addressing unresolved problems. The subsequent part discusses the project's history and describes the obstacles and flaws that lead to the project's remedies and efforts.

A networked game featuring olfactory system was the subject of a QoE (Quality of Experience) assessment of participant fairness by Y. Ishibashis, S. Hoshino, Qi Zeng, N. Fukushima, and S. Sugawara. The evaluation will look into how fairness is affected by how it takes for such a player to scent something. Two players play a fruit-harvesting game in a 3D virtual environment in the networked game. As a result, as an illustration, fairness is seldom compromised when the delay is less than approximately 500 milliseconds.

M. Joselli, F. Binder, E. Clua, and E. Soluri, "Mindninja: A Mind Action Game's Idea, Development, and Evaluation Based on EEGs" claims that commercially available. Brain Computer Interfaces (BCI) are becoming more widely available and affordable. It develops a brand-new model of game interaction. This study suggests a ground-breaking framework and architecture for creating apps for both BCI and traditional interfaces. This article serves as a proof-of-concept, showcasing the skills used in designing and building a game prototype that uses EEG waves among the player's inputs. Equal to Fruit Ninja in style, MindNinja is an additional input, this game differs from the majority of BCI games. With both a group of people, this game was tested and evaluated, and the findings in terms of improving the subject's attention were encouraging.

According to "Fancy Fruits – An Augmented Reality Application for Special Needs Education" by S.C. Steinhacusser, A. Riedmann, M. Haller, S. Oberdorfer, K. Bucher, and M. E. Latoschik, AR has enormous promise for Special Needs Education by allowing a connection between the actual and virtual worlds (SNE). Now, a learning tool called Elegant Fruits has been developed to teach children with disabilities about the nutrients found in local fruits and vegetables. The programme includes augmented reality elements based on markers that connect the physical environment to virtual data. An on-site examination was conducted to evaluate the application. Eleven children with mental disabilities took part in the study. The results showed that participants had a blast. The results revealed that participants had a blast. The findings also supported the kid-friendly design of the app.

III. System Requirements

These are the minimal specifications needed to operate the Unity Editor. Depending on size of a project, actual results and graphics quality may differ.

3.1 Hardware Specifications:

System	Minimum Requirements
Operating System Version	Windows 11, 64 – bit.
CPU	Intel(R) Core (TM) i5-8265U CPU @ 1.60 GHZ
Graphics API	DX11 capable GPU
Processor	I5
RAM	8.00 GB
Hard disk	SSD

Table 3.1 – Hardware Requirements

3.2 Software Specifications:

System	Minimum Requirements
Software Version	Unity 2020.1.6f1
Compiler Version	Microsoft visual studio 15.9.42

Table 3.2 – Software Requirements

IV. Software Environment

A standalone programme called the Unity Hub makes it easier to search, download, and manage Unity projects and installs. Additionally, versions of the Editor that are already installed on a computer can be manually added to the hub.

- By managing your Unity account and Editor licenses through the Hub.
- Create a project, link the project to the Unity Editor's default version, and control how many Editor versions are installed.
- Launch several Unity editions from the Project view.
- Without starting the Editor, manage and choose Project build targets.
- Utilize two instances of Unity simultaneously. Keep in mind that you should only have one Editor instance active at a time in order to avoid local conflicts and other strange situations.
- Adding elements to the Editor installations that already exist. Find and add extras (such particular platform support, Visual Studio, offline documentation, and standard Assets) whenever you download a version of the Editor through Unity Hub, whether during installation or later.
- To expedite the construction of common project kinds, use project templates. When creating a new Project template offer preset values for typical settings. The options for a target game level of visual fidelity are presented in groups by project templates.

4.2 Simulation Environment

Downloading the Unity engine and development environment is the primary need for using Unity to create content. The programmes that allow Visual Studio to incorporate Unity programming, as well as optional modules for delivering to different platforms, are all available for download in addition to the main engine. Navigate to this page to install Unity. Click on – Choose you Unity + Download when you arrive.



Figure 4.1 - Downloading the Unity

4.3 Creating a First Project

Both 2D and 3D games can be created in Unity. Every unity game begins as a Project via way of the Startup screen. The screen below comes when you run Unity after it has been freshly installed.

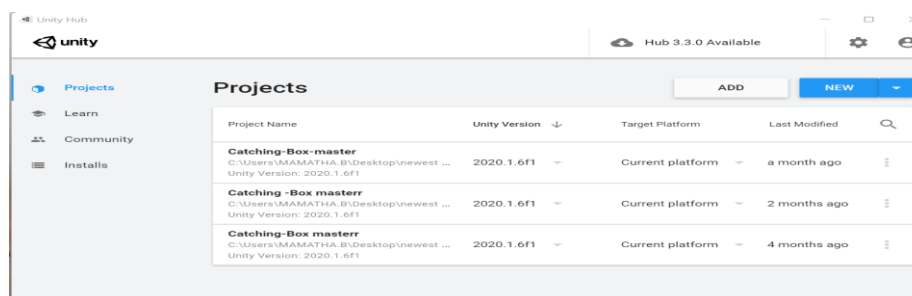


Figure 4.2 Unity window

The New icon is visible on the in the upper-right corner of the window as shown above. The project setup screen will appear after clicking on the icon.

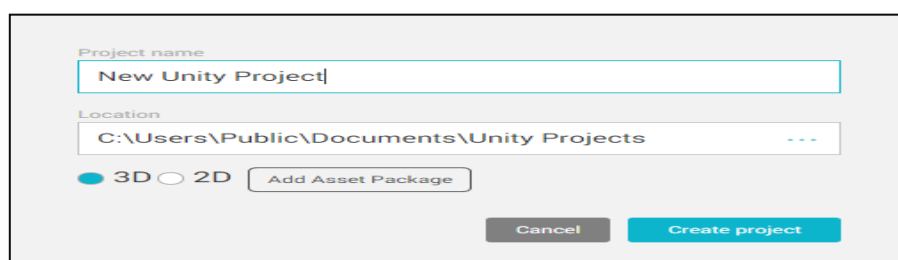


Figure 4.3 Creating a project

Here, you can give the name as “Catching-box-master” and specify its type to 2D then place where it will be saved and add any existing assets after click on create project. Depending on the project type, pre-added assets, and computer speed, this could take some time.

4.4 Exploring the Engine

The following screen shows after Unity has opened and a new project is created.

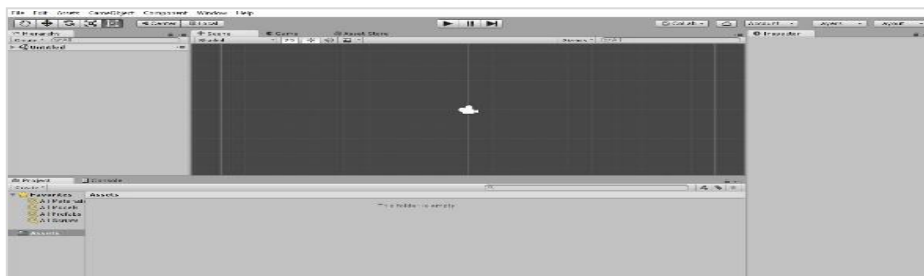


Figure 4.4 – Unity Engine

In this window five main regions are visible:

- Scene view
- Hierarchy window
- Project window
- Inspector window
- Status bar
- Game view

V. Implementation

Change the standalone resolution (1024*768) on the Game tab to affect how unity generates and displays an app. If not, click + (add) to modify width and height, pick Fixed Resolution, then click ok. Standalone describes Player settings.

Create Scenes, Scripts, Sprite folders. Unity uses scenes to organise material. They’re game or app as sets. Unity uses scenes. Your work should be saved as a scene (asset extension for Unity). If you press Ctrl+S and name the asset, it will appear in Assets file. Select Create-> Scene and enter a scene name.

The Scene view has set of navigation controls to move around efficiently,

- a) Scene gizmo
- b) Move, Orbit and Zoom tools
- c) Center tool

Scenes can be loaded into the editor by double clicking them while the action is not running. Loading an unsaved scene prompts a save or discard. For a simple game use only one scene, but for a complicated game, use one scene every level with its own environments, characters, obstacles, decorations and UI. In this project three scenes are used.

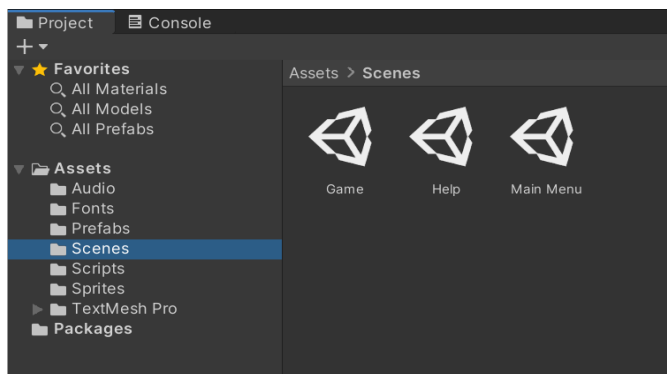


Figure 5.1 Scenes Folder

To generate a sprite in unity, save a PNG or JPG image and drag it into Unity sprite creation.

- Dragging an Asset from outside Unity.
- This picture Asset becomes a texture.
- Drag the texture into the scene hierarchy to create a GameObject the use of a Sprite Renderer.
- This game’s sprite renderer makes use of that substance.

Opened asset folder has Sprites. Currently the Sprite name is apple.

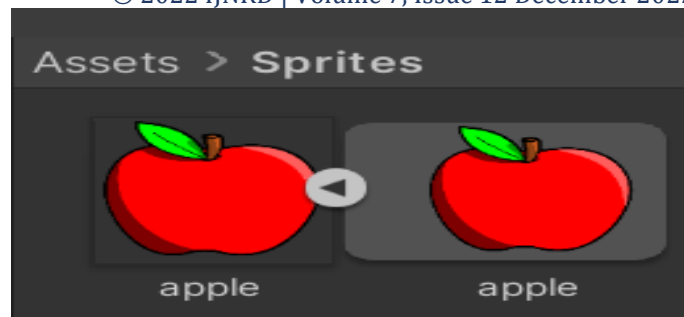


Figure 5.2 Sprite Creator

Sprites are 2D objects with texture graphics. Drag the picture in sources further into stage structure after that to create a new GameObject. Image texture type is Sprite (2D and UI). See the image in view. In 2D mode, Unity employs sprites. Because they have no Z- width, sprites look paper- thin in 3D space, sprites face the camera diagonally.

Sprites in Unity are created using textures. A new GameObject is created using this texture, and it is then assigned a Tile Ray tracer. This gives our GameObject a texture and on-screen characteristics. Unity automatically produces a GameObject using the Sprite Renderer component when generating a Sprite (GameObject > 2D Object > Sprite).

Imported sprites can be changed in several ways. Find a toolbar in the top left corner of the engine's UI.



Figure 5.3 Asset Toolbar

- The Hand tool moves without harming items.
- You can move things around the video game by using the Move tool.
- The Rotate tool turns items in the middle along the z-axis.
- Upward facing Scaling tool, this utility scales things along specific axes.
- Rect tool combines Move and Scale but it is inaccurate. It organises UI elements better.

Unity scenes represents 3D GameObjects. Unity must 'flatten' a view for a two-dimensional screen. Each GameObject has a transform component and additional components. Component attributes define GameObject behavior depending on the object type, you add different GameObject components. Here Catcher uses Transform, Sprite Renderer, Colliders and Script.

Transform is GameObject's most important component. Every scene element has a change that specifies where it is in the virtual world or its parent, how it is rotated, and how big it is. A GameObject always has a Transform component, it can't be removed or created without one. In 3D, transforms are manipulated along the X, Y, and Z axes. In 2D, only X and Y. Unity axes are red, green and blue colors.

In the Inspector, type "Rigidbody" under AddComponent. Click the first component to reveal Rigidbody 2D.

Rigidbody makes GameObjects physics controlled. A Rigidbody's appearance and feel are altered by applying forces as opposed to changing the Transform Component. Transform can't receive forces or torque, but Rigidbody can. Adding forces/torque to the Rigidbody changes the transform component's position and rotation. GameObjects must have Rigidbody before the physics engine can affect them.

By default, gravity causes GameObjects to fall vertically. Set Gravity Scale=0 to avoid this. Playing the game won't change because the GameObject has no physics yet. Scripts will solve our problem.

It directly modifies GameObject position. If a player presses a key, add it to the position. Now make the player react properly to boundaries and GameObjects. GameObjects real-time physics capabilities. Forces, gravity, mass, drag and momentum are included.

Attach a Rect Transform by clicking AddComponent and searching for Rect. Rect Transform attaches components to GameObjects. Many new fields have opened now that it's attached.

Rect Transform is Transform's 2D layout counterpart. Rect Transform is a rectangle that can include UI elements. If a Rect Transform's parent is another Rect Transform, the child can specify its location and size relative to the parent.

2D Colliders define a GameObject's shape for physical collisions. An unseen collider can be any shape. Click Inspector's Add Component and type "Circle Collider". Click the first component to reveal Circle Collider 2D attaches to a GameObject.

CircleCollider2D is 2D collider. The collider has a defined position and radius in a Sprite's local coordinate space.

BoxCollider is a cuboid-shaped collision primitive.

Edge Collider is a 2D collider. The Collider's shape is defined by a freedom edge made of line segments that fits the Sprite graphic precisely.

A coroutine spreads tasks across frames. It's a technique that can pause execution and return control to Unity, then begin the next frame. Calling a method runs it to completion and returns control and optional return values. Any method action must occur inside a single frame update. Unity resumes a coroutine after a yield by default. WaitForSeconds delays. Unity separates collisions from the sprite and calculates them separately.

Every component is a GameObject, and a scene could have hundreds interacting. If Unity introduced collisions each and every GameObject would be impracticable that the engine can determine them all and provide a simple "apple" for the player collide with.

A child GameObject moves, rotates, and scales much like its parent. Every object has many children but only one parent. There is a hierarchy in the parent-child connections in transform. In a hierarchy, the root item is at the top. To make a parent, drag a GameObject from Hierarchy view onto another. The two GameObjects are now parent and child.

Unity on windows automatically installs Visual Studio, creates a script, and adds a new asset. Double-click the name for now. The script and IDE should open. MonoBehaviour derived script name it contains classes and methods. Start and Update don't have return types.

Update goals:

- Check user input.
- Read input instructions for user points.
- Change the transform's position values based on speed and direction.

The Start method executes once at the GameObject's initial scene frame. After, Start and Update method executes every frame. Normally, Unity games operate at 60 FPS, this translates to 60 calls to the Update method each second.

GameObject behaviour controls associated components. Unity components are scripted. It triggers game events, adjusts component characteristics over time and responds to user input.

Unity should compile if you save this code with changing any other methods. When the bottom-right right icon compiles. Drag the Asset script onto the GameObject.

Unity UI is a game and app UI toolkit. It leverages components and the Game View to arrange, position, and style UIs.

Prefabs are blueprint of a GameObject. Prefabs are copies of GameObjects that can be reproduced and added to a scene after it was created. Dynamically generating GameObjects with prefabs. Drag a to create a prefab, drag a feature of the app as from visual level into the funds.

In a game, instantiation and destruction are crucial. Instantiate means create, GUI pieces are disappeared, and scenes load constantly. Knowing how properly dispose of and retrieve unwanted items are crucial. Attach a script to a food named object. Drag prefabs into the GameObject's slot.

Canvas should contain all UI components. A Canvas GameObject must be the parent of all UI components. A canvas is automatically generated when an UI design component, like an image is created if one doesn't already exist in the scene. Canvas child UI element is created. Scene View shows Canvas as a rectangle. This allows positioning UI items easier without needing the Game View. Canvas Messaging system uses EventSystem.

Canvas Render Mode option can render in screen or world space.

Screen Space – Overlay renders UI elements over the scene. Canvas instantly adjusts to screen size and resolution.

In Screen Space – Camera mode, the canvas is positioned before a Camera. This camera renders UI elements hence Camera parameters affect UI look. The Camera's Field of View controls the amount of perspective distortion if the Camera is set to Perspective. Canvas dynamically adjusts to screen size, resolution, and camera frustum.

Canvas behaves like any other object in World Space mode. Widgets display in front of or behind other objects based on 3D location, and the canvas size can be manually modified using its Rect Transform. For world-facing UIs, this is helpful. It can be adjusted to ignore back – facing Graphics and 2D or 3D items in front of it. If this element's processing is forced to the front or back of Raycasting, a manual priority can be set.

Canvas Scaler controls UI element scale and pixel density, Scaling Canvas affects font sizes and image borders.

Hierarchy displays all GameObjects in a scene, such as models, cameras, and prefabs. Right-click Scene Hierarchy and select Create UI Button to add a button. Unity creates one and places it in the Canvas. A button in the UI needs functionality. Canvas size is independent of camera size in Overlay rendering mode. This script only calls the specified method when attached to an empty GameObject. Click the + icon on the bottom tab to add an entry. This entry defines which object button presses affect and which script function is called due to system for recording button presses. By listing them, you can activate multiple functions.

TextMeshPro is Textual answer from Unity. It takes the place of Unity's Text Mesh and UI text. TextMeshPro uses custom colors and powerful Text Rendering algorithms to enhance visual quality and give users flexibility in text styling and texturing. TextMeshPro increases text style and design with letter, word, line, section width, aligned text, ties, over 30 Rich Text elements, Multi typeface & Sprites, Custom styles.

Unity can import AIFF, WAV, and Ogg audio files in the project panel. Importing an audio file creates an Audio Clip that can be used in a script. The Audio Clip reference page explains audio import options. Unity supports tracker modules for music, which use short audio sample as "instruments" Unity can directly record from the computer's microphones using a script.

AudioListener is a microphone. It plays any Audio source in the scene through the computer speakers. Attach listener to Main Camera within a Reverb Zone, all audible sounds are reverberated. Listeners can apply Audio Effects to all audible sounds in the scene.

EventSystem sends application events based on keyboard, mouse, touch, or custom input. EventSystem components send events. Adding an EventSystem component to a GameObject does not expose much functionality because it is designed to manage and facilitate communication between EventSystem modules. This subsystem controls all eventing elements. It coordinates the active Input Module and the selected GameObject. Each 'Update' the Event System receives a call and determines which Input Module to use.

VI. Experimental Results

Open the Game tab and click on Play button in the toolbar to start the game.

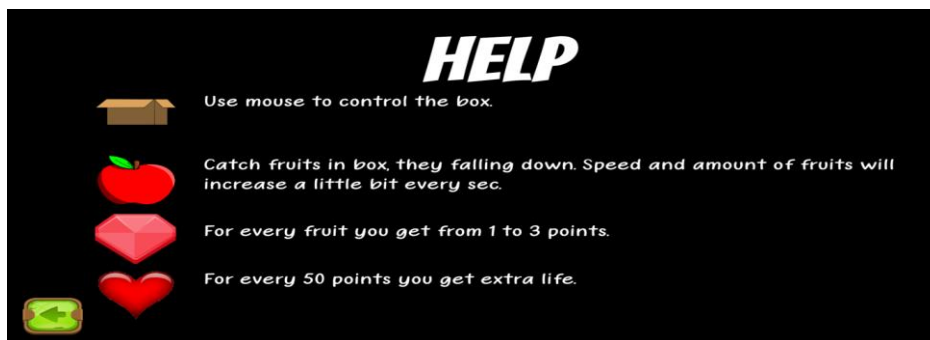


Move about by pressing the arrow keys. Simply press the Play button once again to end the game. Even change the pace in real-time so you don't constantly have to restart it.

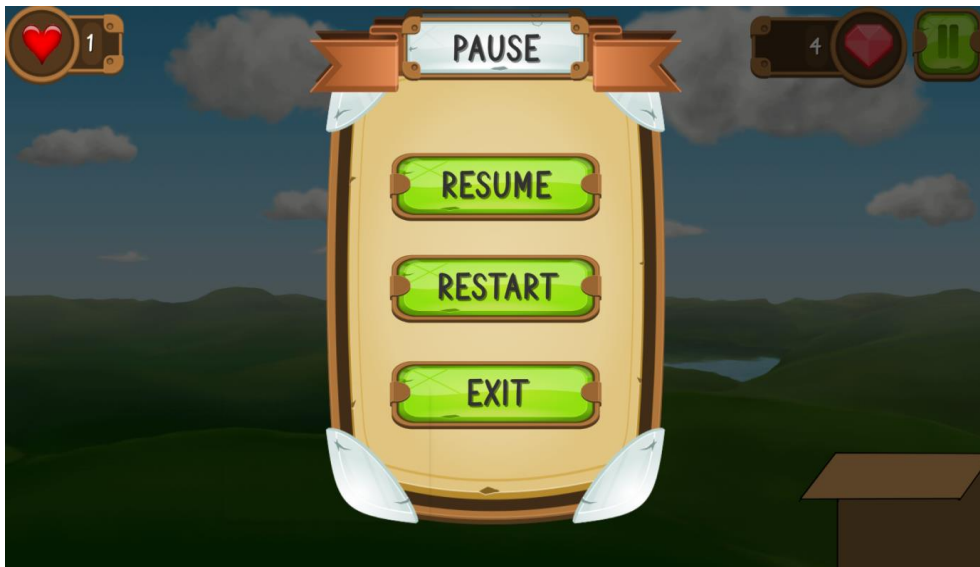
Click on the Play button it shows where the game takes place.



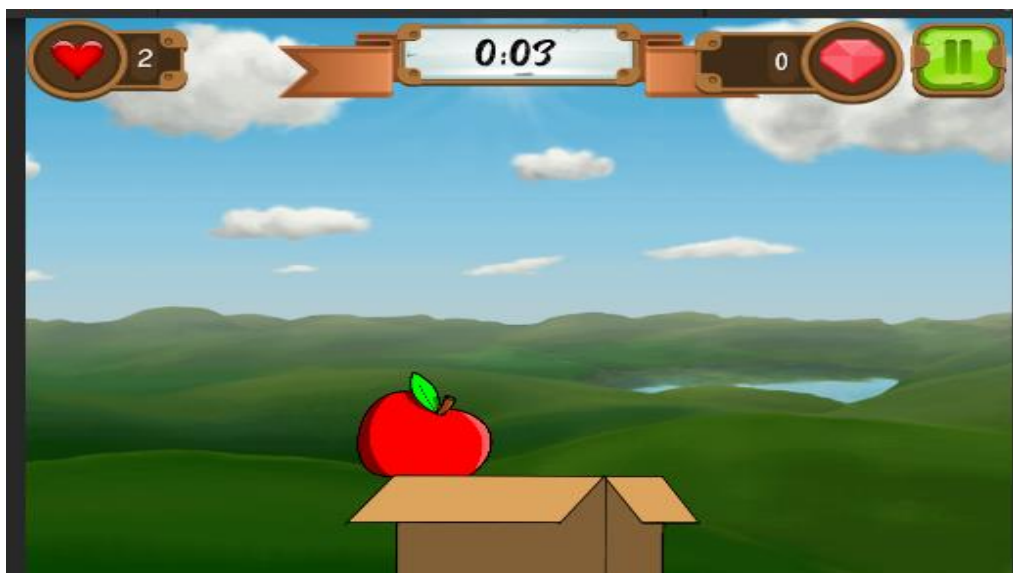
When you click on the HELP and then return to click on the Backward arrow direction symbol, it displays game instructions.



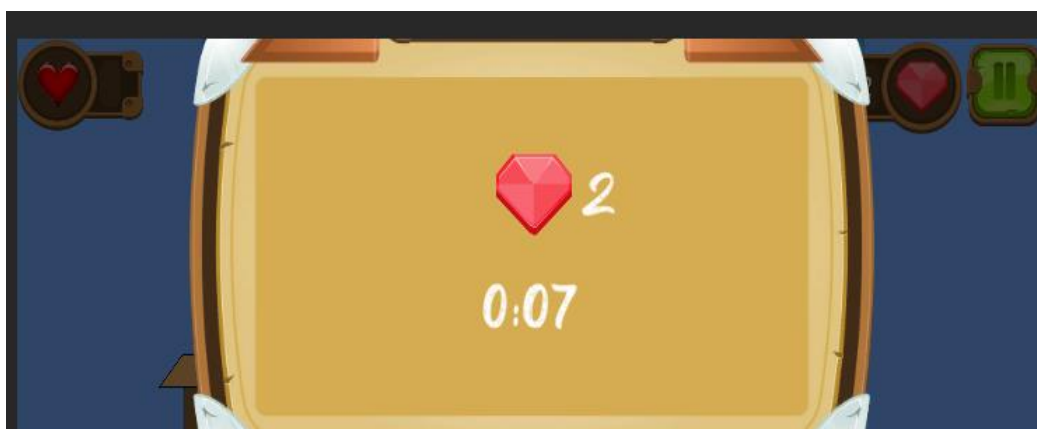
If click on the PAUSE button it shows a menu with resume, restart, and exit buttons.



If the user playing the game it looks like



When the user loses the game



VII. Conclusion

This project attempts to make fruit game using Unity. Game engine is complex and powerful, .vxe integrates model and script design to create a game.

Three scenes and six scripts were written to make the game integrated. Game shows life, diamonds, points and a timer. In Help, game instructions are given. The Main menu scenario help users interact. Add controls to each GameObject.

Fruit game had all features. The article showed how to develop a 3D game with Unity and scripting. The game design focuses on how to generate more fruits based on player abilities using a basket although it considers life and score.

References

- [1] Guangcai Liu, Wenwu Zhuang, Weiwei Li. The research of the fruit schedules and management mechanism in foreign and enlightenment to our country. *Transportation Enterprise Management*, 2008, 2:70-71.
- [2] Fengyun Cao, Jianwei Zhang. Research on the queue model and delays of airport schedule. *Journal of Transportation Engineering and Information*. 2010, 8 (4): 77-80.
- [3] Lei Zhao. Research on the effectiveness and fairness of the slot allocation mechanism when the plane into the port. *The 4th China intelligent transportation conference proceedings*, 2008.
- [4] Vossen T, Ball M O. Slot trading opportunities in collaborative ground delay programs. *Transportation Science*,2006,40(1): 29-43.
- [5] Y. Vikrant Vaze. Competition and congestion in National Aviation System: Multi-agent, Multi-stakeholder Approaches for Evaluation and Mitigation. Ph.D. Thesis Operations Research Center, MIT, Cambridge, MA, 2011.
- [6] Hanif D. Sherali, Justin M. Hill, etc. Integrating Slot Exchange, Safety, Capability, and Equity Mechanism within an Airspace Flow Program. *Transportation Science*,2011,45 (2) :271-284.
- [7] Ruguo Fan, Minchun Li. *Game theory*. Hubei: Wuhan University Press, 2006.
- [8] Tao Huang. *Game tutorial*. Beijing: Capital Economic and Trade University Press, 2012.
- [9] Fayek D, Kesidis G, Vannelli A. Hierarchical Virtual Paths Allocation in Large-Scale ATM Networks Using Noncooperative Game Models. In *Proc. Canadian Conference on Broadband Research (CCBR)*,1999.
- [10] Fei Wang. Research on slot allocation models and algorithms in-ground holding policy. Nanjing: Nanjing University of Aeronautics and Astronautics, 2010.
- [11] Wenjuan Dai. The research on network resource allocation is based on the Stackelberg game. Wuhan: Huazhong University of Science and Technology, 2008.