Cube Catcher 2D Endless Runner Game

Mirza Ahmad Shayer
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
mirza.ahmad.shayer@g.bracu.ac.bd

Sushana Islam Mim
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
sushana.islam.mim@g.bracu.ac.bd

Md. Abrar Ur Rahman
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
md.abrar.ur.rahman@g.bracu.ac.bd

Salman Sadat Nur
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
salman.sadat.nur@g.bracu.ac.bd

Dr. Md. Ashraful Alam
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
ashraful.alam@bracu.ac.bd

***Abstract*—In today's modern era - video gaming is a very popular hobby. The earliest iterations of gaming are very basic in nature. Platform games are often considered the most popular and is often the most well-known genre. During the 1980's to 1990's this genre was at the peak of gaming. During those times all games were only available in the arcade. In the present era very powerful yet portable gaming machines have become the common way to play video games. Usually personal computers, gaming consoles and handhelds. Cloud and multiplayer gaming are quickly gaining traction. Nonetheless, the gaming industry would not have been catapulted to the heights of success it enjoys today without the popularity of platform games in the previous era. As such to this day many platform games are still being made. The most popular platform games - Mario, Sonic and Donkey Kong are still going strong even after 30 years. Endless runner games are also another popular genre. In the mobile devices games such as Temple Run and Subway-Surfer are very popular. People enjoy platform and endless games very much. Cube Catcher is one such endless platform game that is based off of those prior successful games. It pays homage to a time of simpler game design. The game design principles are discussed in detail with all the necessary information - design concept, assets, controls and work preparations. The development process is highlighted and elaborated on. The concepts and planning followed by the programming and development. The unity engine is also discussed with reasons for using it. The testing and optimization phases are looked upon. Testing parameters, and optimization improvements are discussed. Finally, concluding remarks followed by future work discussion- updates with more optimization, assets and features.**

***Index Terms*—Cube-Catcher, 2D Platform, Endless Runner, Unity Engine, Game Design**

I. Introduction

In the present times 2D video games usually mean games with flat graphics. There are only two dimensions - x and y. It basically means players can only move up, down and side to side. Some games also restrict side to side movement - in particular endless runners. 2D video games can provide enriching entertainment despite being the most simple games to develop. With the rise of mobile devices - platformers, endless runners and various other types of video games are becoming widely popular and highly accessible to people of all ages. During the 1980's and 1990's there has been a rise in video games. During those times games were not as accessible and not everyone played video games.Games were available in a fixed location at an arcade. Teenagers would go to arcade machines and play the games. According to Koyama, the arcade machines took coins and allowed users to play video games. If the player lost - another coin had to be inserted to play again. If the player kept winning - the could continue to play. During those times space invaders - a space shooter was the first ever video game to achieve prominent success and fame after Pong [1]. Again according to Kocurek, During those era's video games could only earn a decent profit thanks to the coin drop functionality. As teenagers would play more video games, more coins would be bought and more money transferred. If teenagers lost games more often, many were willing to buy consecutive coins, thus a micro coin economy would begin. It did bring both initiative and controversy, some games became more difficult and more coins were spent to beat them. Many people complained and some even protested for this as well [2].

In the modern era, this is no longer the case. Thanks to more powerful consoles and gaming devices people can buy. Arcades were no longer popular places and by the 2000s, almost everyone had at least one gaming device in the house. During the 2000s there was a huge technological shift and devices became more consumer friendly. Personal Computers - PCs started to become more popular and available. In the 2000s there was also a rise in gaming consoles such as playstation and xbox. According to Kirriemuir, gaming consoles have become very successful and are now set to have great marketing value. Consoles are cheaper than PCs and thus more people can buy them albeit at the cost of other work related functionalities. The cheaper prices mean more teenagers and even young adults aged 10 and over could also purchase them. Handhelds became more popular in the early 2005s and mobile began it's success journey from 2010's and onwards. All these facilitated the game development scene making more popular

video games and more jobs for the gaming industry [3]. All these factors have increased the production of various video games in particular platformers and endless loop games. As such Cube Catcher 2D wishes to relive those simpler times and bring back a good dose of nostalgia with the usage of a modern gaming engine called Unity. The games premise is a simple 2d platform like endless runner, where the character has to catch as many cubes as possible and dodge obstacle. One hit from the obstacle and the game is over. The higher the score the better. The prior works of platform and endless runner game design is discussed in the literature review, the design and principles are elaborated with necessary details in section 3. In section 4 the development process is explained with all phases. Testing and development is summerized with issues, resolutions and further improvements to the game in section 5. Lastly, section 6 gives concluding remarks followed by future updates in section 7.

II. Literature Review

2D video games are often made with very simple game play loops, objects and stories. However, with the changing landscape of gaming - where players want to see and hear interesting stories, wish to see higher graphical fidelity and enjoy good game mechanics. As such, games had to step up in every corner, games began to have more interesting stories in graphically detailed cut-scenes, very engaging game play and other stuff as well, like unlockables, collectables, different characters, downloadable content, multiplayer and more. Gaming has now become a favourite hobby for people of all ages. According to Prot, McDonald, Anderson and Gentile, it has become a very well liked albeit pervasive past time. Although, various other issues have sparked debates between various parties with valid concerns. Some people become addicted to video games and many companies try to make games addictive so people will play longer leading to health problems. Some negative aspects were found with children who play for too long, like aggressive behaviour, eye sight problems and finger pains. Though there are many positive aspects of video games such as more creative minds are those of people who play video games that challenge their imaginations. Also people have faster reaction times to those who play platformers or endless runners. Many people also bond over video games, forming lifelong friendships. Overall, games have both positive and negative side effects [4].

There are many peculiar types of video game design that are currently available. As time goes on newer video game designs come to the limelight. Some games often have a huge studio behind the production with lots of people working on the project for years. These games have a lot of assets in high graphical fidelity. However, many games on smaller scales can also do well, as many people prefer different niche varieties. Many minimalist style games have also gotten very popular over the recent years. Many players also want to play games to simply have fun or pass the time. Many large games have campaigns and many objectives that may become tedious. Some players simply want very minimal game with a very simple objective. According to Fouché, minimalist designed games have become a rather good option more many players who want simpler games to play to pass the time. A calm relaxing experience that can be delivered with a minimalist design. Minimalist game play often have a very unique design, relaxed pacing and a very good calming story. There is not much information available on this topic, though more and more people are describing such experiences [5]. 2D platformers and endless runners are also often used with educative purposes. In recent years many games have been incorporated into education and learning. Many games are also made with the intention of teaching people various topics. According to Oprean, Gould, Riedel and Larsen, most platformers often have too simple mechanics that are incapable of facilitating higher learning, thus to mitigate this issue the 2D platformer has added more features such as solving puzzles, finding items and finishing quests etc. Often mystery platformers, have gameplay integrated with different kinds of puzzles. Dexterity centered mechanics around changing environments gives the player a unique experience. These mechanics guide players to use wit and critical thinking to solve the problems [6].

Many game genres are growing in rapid popularity. Various types like shooters, racing and open world adventure games. In 2015's mobile games began to take a large chunk of game development. Popular titles such as Angry Birds, Traffic Racer, Subway Surfers and Temple Run. According to Rosyid, Hasanah, Fathurrozi and Akbar, a good game should provide sufficient entertainment and is balanced in fairness to difficulty for the player to give the best possible experience. A procedural content generation is introduced to ensure the difficulty of the game is fairly distributed among players. Similar to flappy bird - another popular mobile game. Obstacles will come to the player at a consistent rate. Often times classification algorithms are often used in many games today to streamline the game development process [7]. Most often in endless runners, the pace of the game ultimately depends on the player. Some games allow the pace to be set while others are of fixed pace and only get more faster as the player progresses. Some players enjoy a slow pace while others a faster one. Some games can change pace according to the players wishes. As per Misra, Segura and Arif, in many games the number of objects spawned increases and the score multiplier decreases. The pace is changed and it becomes more challenging the longer the player survives and at one point becomes rather frustrating to play than fun. Game performance and player interactions are also changed. In some cases it made the player happy as more challenge is welcomed but others were displeased and stopped playing. Game pace should be based off of player mobility [8].

Often times, many games also use automata and machine learning to increase the games functionality and provide players with an adequate and unforgettable experience. In

recent times, most often many games have far more advanced artificial intelligence to ensure the game is organized to player pace, movement and ease of playability. According to Nendya and Redono, endless runner games have collectibles and many obstacles spawn for the player to dodge. Here a finite state machine is utilized to ensure the game never gets boring and works perfectly well without error. Often times this is also used in many different types of endless games like horror, adventure or action. Many players have reported to be very satisfied when finite machines are often used in games. The cost though is complexity in setting the finite machine around the game. Most common issue is state collision. Nonetheless, finite machines are still used as many players are very satisfied with the game mechanics [9].

Taking all above works into account, cube catcher 2D will ensure to use the minimalist design with a simple game loop of catching cubes and dodging obstacles. The pace of the game gets faster as the more time the player survives. The pacing is slow at first but gets faster and faster and time goes on. Changing of pace is gradual.

III. Game Design and Principles

In this section the basic design of the game, followed by the in game assets and the controls are elaborated on. Furthermore, other details of the unity engine are discussed as well. Why the unity engine is used and why a 2D game made is also shed light on. The current trends on the gaming markets as well along with some more market details as well will be shared.

A. Basic Game Design

The title of the game is ”Cube Catcher 2D”. The title represents the game itself but only gives partial clues. In the game, the basic premise for the character is to catch the cubes for points. There are also many obstacles in the game the player has to dodge. The more the number of cubes caught the higher the score. The more the score the better the player is at playing the game. The design concept - consists of four parts - game-play loop, character, background and in game camera view. All are discussed in detail below.

1) Game-play Loop: The basic loop is when the game is started, the player is greeted with a simple play menu. After player clicks on the menu the game begins. The game is very simple - simply dodge the obstacles and catch the cubes coming the characters way. The more cubes the character catches, the higher the score. If the character touches any of the obstacles the game is over and a ’Game Over’ screen will be shown with the high score. Player can choose to go back to the menu or play again starting from zero. According to Guardiola, the basic work of the developer is to create an experience - in particular game play that is engaging and fun to approach. The potential of the game play loop depends upon the design elements that make the game fun to play and how the player interacts with the system makes the experience memorable. A developer must ensure the loop is exciting else the game will become stale and all fun will be lost [10]. Below shows a flowchart of the simple game-play loop.

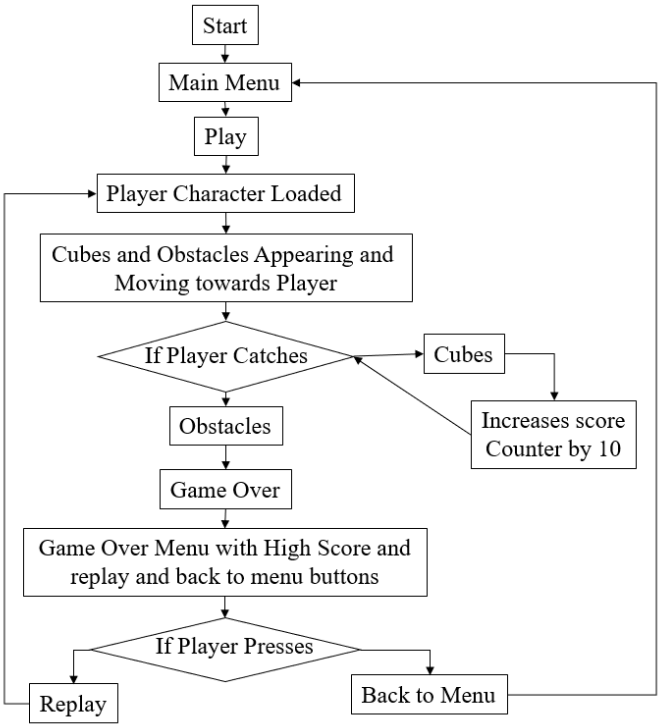

Fig. 1. game-play Loop Flowchart

As per Fig 1, the game-play loop is elaborated and integrated into the game nicely.

2) Character: As with all games there is a main character. The player controls the character and must catch the on coming cubes. Simultaneously, many dangerous obstacles will approach the character as well. The character must dodge all obstacles, else it will be game over. Below shows the first stages of the character and then the final stage of the character.

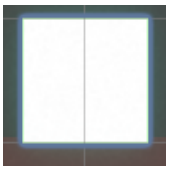

Fig. 2. Character in the beginning stages

Fig. 3. Character in the final stages

As per Fig 2 and Fig 3, the character design stages are shown. According to Poon, characters that showcase realism, good design, social connection and a good story to back the characters motivations makes the characters very marketable and popular. Even if the design is simple, the game play loop can be utilized to give the character meaning and motivation. If the game play loop is fun and engaging, the player can also form a connection to the character. Some games utilize story, others game-play and there are some rare ones that the design itself makes the character iconic [11].

3) Background: In 2D video games the backgrounds are very important. As the character will be going through the game in a single plane, the background will always be visible. As such, it is imperative to give the game a very well designed background that conveys the beauty of the world the game takes place in. According to Garver, Adamo and Dib, the games visuals carries the tone of the game. If the visual style is attractive, beautiful and very relaxing, it generates the most positive tone of serenity and peace for the player. Most often in visual style color and form are very important. Bright colors give off a vibrant vibe while more darker colors give off a feeling of uneasiness or cold environment. Color has a significant impact in video game backgrounds. Most often green, orange, yellow and red when used in the right context can give well tuned effects that attract and engage the player. Complementary colors can crate a wonderful picture. It is best for developers to utilize all aspects of colors when designing video games [12]. Below shows an image of the background used in the game.

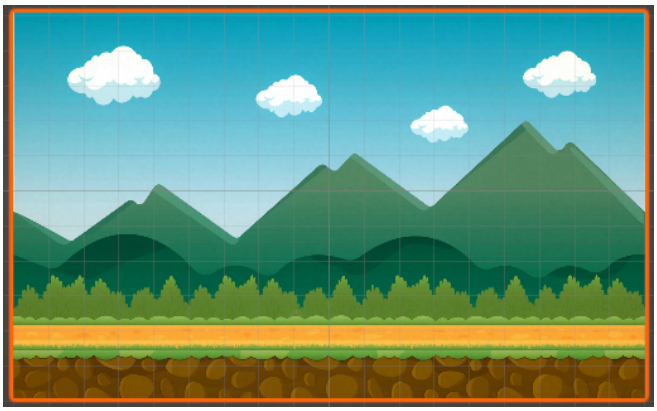

Fig. 4. Game Background

The vibrant colors as per Fig 4 give off a rich and attractive atmosphere, while maintaining a good vibe and serenity.

4) In Game Camera View: When developing video games it is very important to ensure the camera of the game is in the right position. This task is rather difficult to do in 3D video games due to three axis being present. In 2D games it is less hectic, but still somewhat challenging. The camera must be in the right place and ensure enough visibility so that the player can see what is going on. If the camera is off it will affect the game-play and the player will be unable to enjoy the game. According to Naftis, Tsatiris and Karpouzis, camera placement in video games can greatly affect game-play and in turn the satisfaction or dissatisfaction of the player. The virtual camera model must be in a good range so that players can see all elements and are given enough time to react to any oncoming object. Different players prefer different camera angles, which is why large video games often have changing camera modes. In 2D platformers, the camera is often far away so the player can see the full environment and is in a fixed position. 2D platformers do not require shifting camera angles [13]. Below shows the in game camera view with the start menu.

Fig. 5. In Game Camera View With Start Menu

As per Fig 5, the camera of the game is at the perfect angle as it allows the player to see everything and the player is given enough time to react as obstacles and cubes approach the character.

B. In Game Assets

Cube Catcher 2D contains many assets that make up the game. In video games assets make up the bulk of the game. Game assets are objects that are present in the game environment. At the beginning stages any and all assets are simple 3D or 2D shapes. As development continues the assets begin to take shape. Assets are shaped up, colors are added and finally the location of the asset in the environment is put in place. This game is no exception. Here there are some assets for all parts. The player character and background are already discussed. Now the cube and obstacle assets are elaborated. Below the cube and obstacle assets at the beginning of the development are shown.

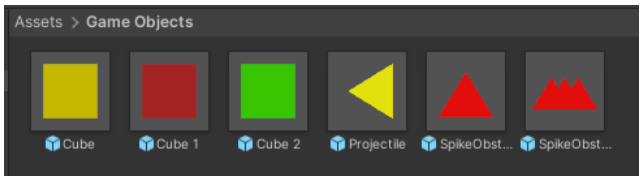

Fig. 6. Cube and Obstacle Assets at the Beginning

According to Fig 6, these basic shapes are utilized as the obstacles and cubes. Spiked and projectile obstacles are

utilized. Simple shapes for cubes are used. Later during development these obstacles and cubes are given sprites and more better shapes. The cubes can be caught by the player while the obstacles have to be dodged.

C. Controls

All video games need to take inputs from the player so that the character can perform movements to the players wish in accordance to instructions and objectives of the game. Controls for games must be easy for players to understand. If the controls become to difficult, the player experience will be stinted. According to Seal, the form and structure of the controls are tantamount to the enjoyment of the game. Terms such as maneuverability, control response sensitivity, and control-display ratio are often alluded to in the Journals of Human Factors Engineering.

Other design variables are important

- Ease of use
- Comfort
- Perceived reliability
- Control response
- Control shape

Controls also decide weather the games are comfortable for casual players to play or require long term playing to achieve some form of mastery. The greater the comfort of the control scheme the more comfortable the players get when playing the game. For a successful video game, a good control scheme is very important [14]. Below shows the game controls used for Cube Catcher 2D.

Fig. 7. Keyboard Mapping

As per Fig 7, here a very simple control scheme is used to ensure player can move the character. The left and right arrow keys are use to move the character left and right. The space-bar is used as the jump button. pressing the button will make the character jump to avoid obstacles and also to catch cubes. This simple control scheme ensures players do not need excessive instruction in playing this simple game.

D. Why Unity Engine and Not Others

Before beginning development, there were some decisions on which game engine that would be utilized to build the game from scratch.

The most popular game engines for making 2D games are given below.

1) Unity Engine
2) Unreal Engine
3) Godot Engine
4) Pygame Python Engine extension
5) GameMaker Engine

The most two popular ones are Unity and Unreal. According to Abramowicz and Borczuk, for comparing the two engines five factors are considered.

- Frames per second
- CPU usage
- RAM
- GPU memory

Unity achieved a better average frame rate. Unreal Engine required more RAM and GPU resources. Analyzing CPU load values revealed that on the first system, Unity demanded less CPU usage. However, on the second system, Unreal Engine used over 10 percentage points less CPU. Unity requires fewer computer resources, although in some cases, Unreal Engine may demand fewer CPU resources. So unity is less resource intensive compared to unreal [15].

Taking all the above factors into consideration the final choice in game engine became unity. Unreal is far too complex and requires more time due to a more sophisticated UI. The other ones - Godot is online and many are not that familiar with that engine - even though many games are made from it. Pygame in python would also be a popular choice but it is entire code based and will take more time. GameMaker is also another online engine and is rather popular but not many games are made with it all that much. So with all factors considered unity would allow the development process of the game to be very smooth and straight forward. The UI of unity is also very user friendly, simple and organized very well. Below shows an image of Unity UI.

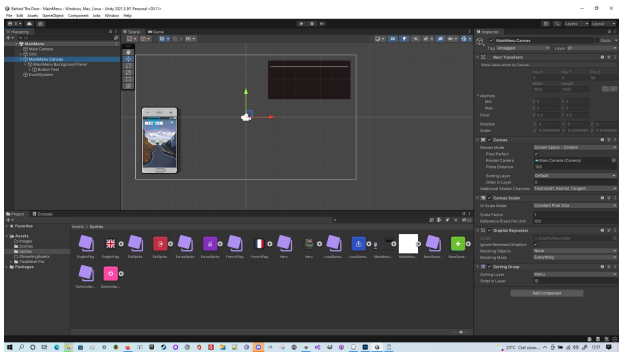

Fig. 8. Unity UI

As per Fig 8, the UI shows all visible aspects very clearly. Center of the UI is the work space - on the left all the structures of the game files, on the right all selected file details, components and features of selected object. On top all file options, settings and extra stuff. On the bottom is the folders that contain game extra stuff like shading, rendering, lighting etc. In the bottom space new folders, assets, objects and script files can be added or deleted. All those can have components added or can be customized according to the developers will.

IV. Development Process

A. *Game Concept and Planning*

Before diving into development, it was crucial to establish a clear concept and plan for Cube Catcher. This involved defining the game's genre, target audience, core mechanics, and level design.

Cube Catcher is a 2D platformer game aimed at players who enjoy casual gaming experiences. The game mechanics revolve around catching cubes of various types to score points while avoiding hazardous obstacles. Players can freely move their character, jump, and interact with cubes within each level. The game features an endless loop structure where players can continuously catch cubes to increase their score. Different types of cubes offer varying points, adding depth to the game-play.

Level design focuses on creating diverse environments filled with cubes and hazards. Each level presents unique challenges and opportunities for players to test their skills and improve their scores. The game aims to provide a satisfying and addictive experience for players, encouraging them to strive for higher scores with each play-through.

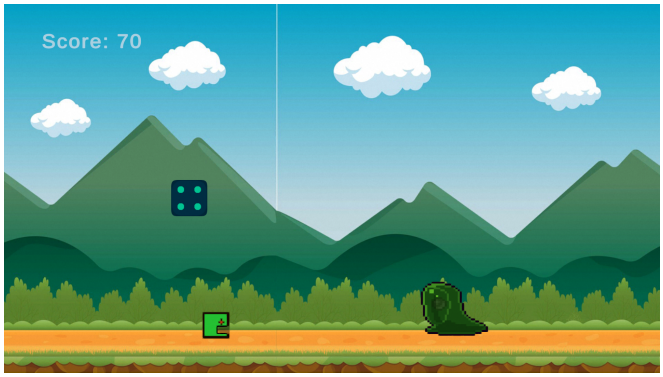

Fig. 9. Cube Catcher 2D Game

As per Fig 9, this is how the game is played and what it looks like. The player character tries to avoid the monster and tries to catch the dice like cube.

B. *Tools and Technologies Used*

Unity was chosen as the primary development tool for Cube Catcher due to its versatility and robust features for 2D game development. Unity provides a user-friendly interface and powerful tools that streamline the development process, making it an ideal choice for both beginners and experienced developers.

C Sharp was the primary programming language used for scripting within Unity. C Sharp offers a balance of simplicity and flexibility, allowing for efficient development of game-play mechanics, user interface elements, and other features.

Adobe Photoshop and Gimp were used for creating graphics assets such as character sprites, cube designs, background elements, and UI components. These tools provided the flexibility and precision needed to design visually appealing assets that complemented the game's aesthetic.

Google drive was utilized for version control, enabling collaborative development and tracking changes to the project over time. Version control was essential for managing the project's code-base, assets, and other resources, ensuring smooth teamwork and efficient project management.

C. *Advantages of Unity*

The advantages of the Unity Engine is given below.

1) Cross-Platform Deployment: Unity's cross-platform support enables Cube Catcher to reach a broad audience across various devices, including PC, mobile, and console platforms, maximizing accessibility and potential player engagement.
2) Asset Store Integration: Unity's Asset Store provides a vast repository of assets, plugins, and tools that expedite development and enhance project capabilities. Leveraging the Asset Store's resources enabled the development team to accelerate prototyping and streamline workflow processes.
3) Community and Documentation: Unity boasts a vibrant community of developers and extensive documentation resources that offer valuable support and guidance throughout the development journey. Access to forums, tutorials, and online communities facilitated problem-solving and knowledge sharing, empowering developers to overcome challenges and achieve their goals.
4) Visual Editor and Prototyping Tools: Unity's visual editor and prototyping tools enable rapid iteration and experimentation, allowing developers to iterate on game mechanics, level designs, and visual elements in real-time. The visual editor's intuitive interface empowers designers to translate their creative vision into interactive experiences with ease and efficiency.
5) Performance Optimization Features: Unity provides built-in tools and features for optimizing game performance, including asset compression, occlusion culling, and performance profiling. These optimization techniques ensure smooth game-play experiences across diverse hardware configurations, enhancing player satisfaction and retention.

6) Extensibility and Customization: Unity's extensible architecture enables developers to customize and extend the engine's functionality to meet specific project requirements. Integration with third-party tools, plugins, and frameworks expands Unity's capabilities, enabling developers to implement advanced features and achieve desired outcomes effectively.
7) Real-Time Collaboration: Unity Collaborate facilitates real-time collaboration among team members, allowing developers to work concurrently on the same project and synchronize changes seamlessly. This collaborative workflow promotes productivity, minimizes conflicts, and accelerates project development, ensuring timely delivery of high-quality results.
8) Has a very easy to use code-base and is very user friendly for many developers.

In summary, Unity's comprehensive tool-set, cross-platform support, and vibrant ecosystem played instrumental roles in Cube Catcher's development journey. The synergy of Unity with other tools and technologies empowered the development team to create a captivating and polished 2D platformer game that delights players with its engaging game-play and charming aesthetics.

D. Enemy and Player Character Design

Player Character:

Fig. 10. Player Character Model

Again as per Fig 10, the character is shown on the left and on the right the sprite animation after catching a cube. The character moves and every time it catches a cube the animation plays. If it hits the obstacle its game over. The more cubes it catches the higher the score. Game continues until player hits the monster.

Conceptualization and Characterization:

The initial conceptualization phase focused on defining the visual style, personality, and characteristics of each character to ensure they aligned with the game's theme and narrative.

Enemy Design:

Enemy characters in Cube Catcher were designed to provide engaging challenges and obstacles for players to overcome. Each enemy character was meticulously crafted to possess distinct visual traits, behaviors, and attack patterns that complemented the game's mechanics and level design.

- Visual Traits: Enemy characters feature vibrant colors, dynamic shapes, and expressive animations that make them visually distinct and memorable. The design of enemy sprites emphasizes clarity and readability to ensure players can easily identify and react to different enemy types during game-play.

- Behaviors and Attack Patterns: Enemy characters exhibit a variety of behaviors and attack patterns that pose unique challenges for players. Some enemies may move in predictable patterns, while others may pursue the player aggressively or utilize ranged attacks. The diversity of enemy behaviors adds depth and complexity to the game-play experience, requiring players to strategize and adapt their approach accordingly.

As per the points above the enemies is designed along with the behaviors.

Fig. 11. Enemy Character models

As per the Fig 11, it shows a bat enemy coming at the player. The sprites show the animation of the bat as it is flying by flapping its wings. The color of this enemy is purple as it can be clearly seen visible when contrasted to the bright background. Again, contrasting colors play a vital role in allowing the player to see the oncoming threat.

Fig. 12. Enemy Character models

As per the Fig 12, it shows a slime enemy that will come at the player. The slime enemy is slightly harder to see when taking the background to context. Here it done to provide a sense of challenge. The player must be vigilant and dodge the slime enemy by jumping above. The pace of the slime enemy is rather slow so player has enough time to react.

Player Character Design:
The player sprite character in Cube Catcher was designed to

be expressive, versatile, and relatable, serving as the player's avatar and conduit for interaction within the game world. The design of the player character underwent several iterations to refine its appearance, animations, and abilities to ensure a satisfying and immersive player experience.

- Visual Appearance: The player sprite character features a charming and whimsical design that reflects the game's lighthearted tone and aesthetic. Character sprites are carefully crafted to convey personality and emotion through subtle facial expressions, body language, and movement.

- Animation and Movement: The animation of the player sprite character is fluid, responsive, and intuitive, ensuring smooth and natural movement during gameplay. Animations for actions such as running, jumping, and interacting with objects are meticulously crafted to be visually appealing and responsive to player input, enhancing the overall sense of immersion and control.

As per the points above the main character the player controls in designed and animated. The simplicity ensures the player can understand the character and the animations make the character fun to play with in game.

Integration and Implementation:
Once the design of enemy and player sprite characters was finalized, they were integrated into the game engine and implemented within the game world. Character sprites were imported into the Unity engine and integrated into the game's scenes, where they were animated, programmed, and assigned behaviors and interactions according to their design specifications.

1) Unity Integration: Enemy and player sprite characters were imported into Unity as sprite assets and integrated into the game's hierarchy and scene structure. Sprite animations, collision detection, and physics interactions were programmed and configured using Unity's built-in tools and scripting capabilities.

2) Behavior Programming: Enemy characters were programmed with attack patterns using Unity's scripting language, C Sharp. Player character controls and interactions were also programmed using C Sharp, allowing players to move, jump, and interact with the game environment using keyboard, mouse inputs.

3) Animation Rigging: Sprite animations for enemy and player characters were rigged and animated using Unity's animation tools, allowing for smooth and responsive movement and transitions between animation states. Animation curves, blending, and transitions were fine-tuned to ensure seamless and natural movement during game-play.

As per the points above the game is integrated and implemented in the simulation.

V. Testing and Optimization

As per traditional game development, usually within the intermediate phase, after the plannings are concluded and the coding of the game has begun. After creating an initial prototype, the prototype is tested with beta players. Depending upon the reviews, opinions and liking of the beta testers the game is then changed as per the reviews. Many games often go through multiple beta testing phases. Bigger the studio, more beta testing can take place - although there should be balance between beta testing and finishing deadlines.

A. Basic Game Cycle

Below shows a basic game development cycle for a 2D game.

1) Plan the game with all the ideas, concepts, details and begin sketching the ideas into the drawing board. Ensure the main idea of the game is built and all other factors are considered.
2) Choose the best engine for the job in this case the unity engine is chosen due to numerous factors and ease of use. The UI is also very understandable.
3) Begin coding the prototype, and fix any early game bugs. Also ensure game runs smoothly and does not crash or show any errors. Also ensure game does not freeze after launching. Often times many games freeze due to unforeseen issues or unaddressed bugs.
4) After a successful prototype is made, it is sent to beta players to test the game. Beta testers can then give reviews on the game and these can be taken into account to change all the required settings. Change of pace, character, design or even environment - all are acceptable as these are the early stages of the prototype.
5) After changes are made to the prototype it is beta tested again to see for any issues or problems that might occur. Once again beta tester reviews are taken into account. Once everything is improved - the next stage can come to fruition.
6) The game gets into the final stages of finalizing assets, optimizing all functions and ensure it is presentable to the customers. Also games can utilize advertisements, billboards, buildings, or even digital libraries.

As stated above, this approach is followed to ensure the game is developed to the customers tastes and ensures maximum profitability. If the game fails to meet basic requirements then there is a chance the game may suffer in sales. To ensure good sales all initiatives must be followed.

B. Computer Configuration

Below the configuration of the computer used to build and run the game in the Unity Engine is given. All details regarding the configuration.

TABLE I
COMPUTER CONFIGURATION

CPU	Ryzen 5 5600G
GPU	Nvidea RTX 3050
RAM	16GB 3200Mhz
Operating System	Windows 10 64bit

The Ryzen 5 5600G is a 6-core and 12-thread processor, and it has AMD Radeon Integrated Graphics. It allows for the unity engine to be run smoothly. Even though Unity does not use too much ram, having 16GB at 3200Mhz means unity processes can compile easily even if the game was a heavy duty one. The operating system of Windows 10 is very well optimized to run Unity at a good pace. Usually Unity is well optimized for all operating including Windows, Mac and Linux. Unity 22.03 build is very stable, has connection to the asset store and has an active community with many services available. Finally, the Graphics Card is the Nvidia RTX 3050, with its powerful RT cores can be utilized to give ray tracing effects to the game. Although in case of Cube Catcher 2D ray tracing was not utilized as the process of integration is rather long, difficult and often leads to game crashing errors unless it is properly integrated into the game system.

C. Player Ratings As Per Developers

Below the ratings are given as per the developers of the game. The ratings are given based upon the game-loop, game-play, controls, atmosphere, background, characters, enemies and obstacles and menu design. Out of 5 all ratings are taken.

TABLE II
OVERALL GAME COMPONENTS RATINGS

Criteria	Dev1	Dev2	Dev3	Dev4
Game-loop	5	5	5	5
Game-play	5	4	4	5
Controls	5	5	5	5
Atmosphere	5	5	5	4
Background	5	4	5	5
Characters	5	4	4	5
Enemies	5	5	5	5
Obstacles	4	5	4	5
Menu Design	5	4	4	5

As per Table II above, the game overall is good on all the given aspects. Some improvements can be made in the future. For now the game is fine to play and the design is very eye-catching.

D. Playable on Other Consoles?

Currently Cube Catcher 2D is playable on Personal Computers (PC). However, since the unity engine is also supported in Xbox and Playstation consoles, it might be possible to play the game in those consoles as well. It may also be playable on handheld consoles like Nintendo Switch or Steam Deck. The game will be developed later for Android and Iphone. Currently Unity is not very compatible with smartphone devices due to their size limitations. For now it is playable on PC. In the future it may be playable on all platforms. In the current market PC games are the most in demand therefore Cube Catcher 2D has been made for PC.

The game might be available in digital libraries once more updates are added and the game is improved upon. For now the game runs on PC. Steam and Epic games stores will be the main targets as these two are the most popular and bought from. With the large customer base, the game will be able to be sold a lot of copies digitally - all that in the future. For now the focus is on a small scale. After the android and iphone versions are released, the target focus will be AppStore, PlayStore and Iphone store. All of this may happen in the future of development if the game succeeds in sales and meeting expectations.

VI. CONCLUSION

In conclusion, the future of cube catcher video games holds immense potential for innovation, creativity, and diverse game-play experiences. Whether through virtual reality, augmented reality, multiplayer modes, or narrative-driven adventures, cube catching games are poised to offer players unique and engaging experiences.

These games could become more immersive and interactive with advancements in technology, such as virtual reality and augmented reality, allowing players to physically engage with the game world. Puzzle-based mechanics, physics simulations, and dynamic environments will add depth and challenge to game-play, ensuring that each session feels fresh and rewarding.

Moreover, customization options and cross-platform integration will empower players to tailor their experiences and seamlessly transition between devices. Artificial intelligence integration will enhance both single-player and multiplayer modes, providing dynamic challenges and companionship.

Overall, the future of cube catcher video games promises to be exciting, offering a wide range of experiences for players to enjoy and explore. Whether you're a casual gamer looking for some fun or a dedicated player seeking complex challenges, there will be something for everyone in the world of cube catching.

VII. FUTURE WORK

Video games involving cube catching could take various forms, each offering unique gameplay experiences. Here's a glimpse into potential future developments:

1) Virtual Reality (VR) Cube Catcher: With the advancement of VR technology, cube-catching games could become incredibly immersive. Players might use motion controllers to grab, toss, or manipulate cubes in a virtual environment. The sensation of catching cubes in mid-air could feel remarkably real, enhancing the gameplay experience.
2) Augmented Reality (AR) Cube Catcher: AR games overlay virtual elements onto the real world, creating interactive experiences. In AR cube catcher games, players might see cubes appearing in their physical environment through their smartphones or AR glasses. They could then move around to catch them, fostering physical activity and exploration.
3) Puzzle-based Cube Catcher: Imagine a game where catching cubes isn't just about speed or reflexes, but also about solving puzzles. Cubes might have different properties or powers, requiring players to strategize on how to catch and use them effectively to progress through levels.
4) Multiplayer Cube Catcher: These games could involve cooperative or competitive multiplayer modes. In cooperative modes, players work together to catch cubes, possibly requiring teamwork and coordination. Competitive modes could pit players against each other to catch the most cubes within a time limit or to fulfill certain objectives.
5) Narrative-driven Cube Catcher: Some games might integrate cube-catching mechanics into a larger narrative. Players could embark on a journey where catching specific cubes unlocks parts of the story or reveals hidden secrets. The cubes themselves could have lore attached to them, adding depth to the game world.
6) Physics-based Cube Catcher: Physics simulations could play a significant role in cube catcher games. Players might need to consider the trajectory, velocity, and even the weight of cubes when catching them. This could lead to challenging yet rewarding gameplay as players master the physics mechanics.
7) Customization and Personalization: Future cube catcher games could offer extensive customization options. Players might be able to customize their avatars, design their own cubes, or create unique environments. This personalization could enhance player engagement and foster a sense of ownership over the game experience.
8) Cross-platform Integration: With the growing trend of cross-platform gaming, cube catcher games could allow players to seamlessly switch between different devices. Players might start a game session on their console, continue on their smartphone while on the go, and then resume on their PC without losing progress.
9) Dynamic Environments: Cube catcher games could feature dynamic environments where the surroundings change over time. This could introduce new challenges, such as obstacles that appear or disappear, altering the gameplay dynamics and keeping it fresh and exciting.
10) Artificial Intelligence (AI) Integration: Advanced AI systems could be incorporated to make cube catching games more dynamic and challenging. AI-controlled cubes could adapt to the player's skill level, providing an increasingly challenging experience. Additionally, AI companions or opponents could enhance single-player and multiplayer modes, respectively.

These are just a few possibilities, but the future of cube catcher video games is undoubtedly filled with innovation and excitement.

REFERENCES

[1] Y. Koyama, "Arcade games (1): From elemecha to video games: The birth of space invaders and the establishment of the arcade game industry," pp. 15–25, 06 2023.
[2] C. Kocurek, "Coin-drop capitalism: Economic lessons from the video game arcade," pp. 189–208, 01 2012.
[3] J. Kirriemuir, "The relevance of video games and gaming consoles to the higher and further education learning experience," 04 2002.
[4] S. Prot, K. McDonald, C. Anderson, and D. Gentile, "Video games: Good, bad, or other?" *Pediatric clinics of North America*, vol. 59, pp. 647–58, viii, 06 2012.
[5] M. Fouché, "Player experience of minimalist video game design: Case study of indie horror iron lung," *Games and Culture*, 03 2024.
[6] D. Oprean, H. Gould, N. Riedel, and S. Larsen, "Collect that coin: Efficacy testing of platformer game mechanics with adult learners," *European Conference on Games Based Learning*, vol. 17, pp. 459–466, 09 2023.
[7] H. Ar Rosyid, H. Hasanah, M. Fathurrozi, and M. Akbar, "Block-based approach to observe game content space in endless-runner game," pp. 273–278, 10 2019.
[8] M. Misra, E. Segura, and A. Arif, "Exploring the pace of an endless runner game in stationary and mobile settings," pp. 543–550, 10 2019.
[9] M. Nendya and D. Redono, "Pocong rush: Endless runner game based on finite state machine," *JOINCS (Journal of Informatics, Network, and Computer Science)*, vol. 5, pp. 14–20, 04 2022.
[10] E. Guardiola, "The gameplay loop: a player activity model for game design and analysis," 11 2016.
[11] S. Poon, *A Contagious Thrill: Identifying Experiential Factors for 2D Graphic Art Appeal Through Character Design for Video Games*, 09 2020.
[12] S. Garver, N. Adamo, and H. Dib, "The impact of visual style on user experience in games," *EAI Endorsed Transactions on Game-Based Learning*, vol. 4, p. 153535, 01 2018.
[13] M. Naftis, G. Tsatiris, and K. Karpouzis, "How camera placement affects gameplay in video games," 09 2021.
[14] D. Seal, "A study of control design and video game performance," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 27, pp. 505–505, 10 1983.
[15] K. Abramowicz and P. Borczuk, "Comparative analysis of the performance of unity and unreal engine game engines in 3d games," *Journal of Computer Sciences Institute*, vol. 30, pp. 53–60, 03 2024.