Mani Shankar Cheekati

Team16_bl.en.u4cse21046_cgv_report

Report Checking

Computer Graphics Sec A and C

Amrita Vishwa Vidyapeetham

Document Details

Submission ID

trn:oid:::1:3096920009

Submission Date

Nov 30, 2024, 1:05 PM GMT+5:30

Download Date

Nov 30, 2024, 1:26 PM GMT+5:30

File Name

Team16_bl.en.u4cse21061_cgv_Finalreport.pdf

File Size

402.1 KB

5 Pages

3,000 Words

16,723 Characters

2% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

1 Not Cited or Quoted 1%
Matches with neither in-text citation nor quotation marks

1 Missing Quotations 1%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

1% Internet sources
2% Publications
1% Submitted works (Student Papers)

Match Groups

- **1** Not Cited or Quoted 1%
  Matches with neither in-text citation nor quotation marks
- **1** Missing Quotations 1%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

1% Internet sources
2% Publications
1% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1 Student papers

Amrita Vishwa Vidyapeetham 1%

2 Publication

K.U. Akkshay, B. Sreevidya. "Development and Performance Analysis of an AI bas... 1%

Suika Game: Unity-Driven Gameplay Enhanced with Flask-Based Machine Learning Integration

Cheekati Mani Shankar, Gontla Venkat Sujan, Hothur Ranga Neha, Hothur Shreyaa,
Dr.Sajitha Krishnan
Department of Computer Science and Engineering,
Amrita School of Computing, Bengaluru, Amrita Vishwa Vidyapeetham, India
bl.en.u4cse21046@bl.students.amrita.edu, bl.en.u4cse21061@bl.students.amrita.edu,
bl.en.u4cse21068@bl.students.amrita.edu, bl.en.u4cse21069@bl.students.amrita.edu, k_sajitha@blr.amrita.edu

***Abstract*— Suika Game is a fruit-merging puzzle game designed and developed using Unity game engine. Fruit sprites in the game involve dynamic gameplay functions where players merge the fruits to achieve higher score incorporating fun and strategy into player. Unity engine was used for its excellent 2D game development features that offered easy solution for graphic display, physics-based interactions, and game controlling mechanism. The outline of proposed approach includes creation of Fruit-Merging Puzzle Game to be designed and implemented in Unity game engine platform to explain the modern features of computer graphics and visualization. In this game, the primary gameplay involves reacting two same type of fruit objects, and animating interactions using & 2D physics of Unity game engine. Dynamic sprite management system is applied to ensure the sprite is properly displayed on different resolutions of the screen with use of 2D colliders and rigid bodies to achieve accurate collision detection for the game, Physics simulation is the other core area of the project as the project utilize Unity physics for fruit motion, including effects of gravity and inertial forces. Machine learning was incorporated in the game to increase player engagement by categorizing player level as 'Beginner', 'Amateur', 'Intermediate', 'Advanced' through the use of a Support Vector Machine (SVM). The trained model proposed here allows making real-time predictions since it is integrated with a Flask web application that processes the inputs sent by Unity and returns the skills level classification.**

***Keywords*— Fruit-Merging Puzzle Game, Unity Game Engine, 2D Physics Simulation, Game Development, Flask, Machine Learning, Support Vector Machine**

I. INTRODUCTION

Easy to play and market hassles are the primary reasons casual puzzle games have always been a major staple in the gaming market. Of these, the fruit-merging puzzle genre has received considerable attention because of the combination of strategic elements and fun. These games are based on a mechanism where the same fruit on a limited matrix must be combined to make a better fruit and the goal is to get the highest score possible. This gameplay involves a combination of strategic thinking, timely decision making and spatial intelligence, something that will interest eplayer of every age.

Creation of such games requires effective game engine that will be able to meet a variety of technological demands including physics, responsiveness, and compatibility. This is a project that has been developed using Unity, one of the commonest game engines in the current market. Being universal, Unity offers a vast number of tools specifically designed for 2D game creation. The second dimension is utilized to depict the real collision, effects of inertia and movement of the fluid object that are important aspects in the game engine. Its animation tools can help produce nice looking transitions and effects and the particle system, adds a nice depth and vibrancy to the games feedback. Scripting in Unity through C# is used to implement the game play and some of the minor basic graphics like state machines, object control lifetimes and user interactions making the game smoothly responsive.

In addition, Unity supports cross-platform well enough, therefore the playability remains unhampered and does not depend on the device. Adaptable UI interface and resolution management are included to allow the game to run successfully for different screen resolutions and thus expand the reach of the game. Unity also as a development platform is very flexible and to it more complex graphics and visual effects and a clean user interface can be incorporated making the game attractive.

To support this, the following game project has illustrated how Unity has been effectively used to design a difficult fruit-merge type of puzzle game. The consequence is a complex and perishable operation crossing game design introduced in real-time, within an interactive framework. Through this project, one can illustrate that Unity can deliver progressive and qualitative gameplay in accord with the contemporary player expectations. The special focus is made on the development process which reveals the technical advantages of Unity and gives a clue to young developers on how to build games combining art with science.

The fruit-merging puzzle was designed as an edutainment game that guarantees the consumer's active involvement as well as strategic thinking and aesthetic experience, all thanks to Unity powerful game creation platform advantages. Main features like objects merging, grid controlling, and dynamic obstacles give a rather challenging but fun experience. To build on this experience even more, machine learning was developed as an added component. An intelligent agent, namely that of Support Vector Machine (SVM), employed with gameplay data identifies the skill levels of a player. The real-time skills-level prediction was integrated into a Flask based web application that Unity would make use of to send player data to the pipeline. He also incorporated this integration so that it helps the player with feedback of their performance and adjust the game based on what a player is capable of. This integration not only adds a personal touch to game performance by providing statistics into a player's performance but also create opportunities for further adjustments of difficulty and adaptive aspects of the game. The successful implementation of this feature proves that it is possible to combine the game mechanics with AI-based personalisation.

II. LITERATURE SURVEY

S. Singh et al. [1] uses a real-world platform namely the Unity Game Engine, a leading platform for developing or designing games to demonstrate its application of creating fun, engaging and aesthetic games. The study incorporates Unity's Resource Store, scriptable object, physics engine as some of the tools that enhance the development of 2D and 3D games. M. Johansen et al. [2] incorporates VGDL with Unity's Machine Learning Agents that is majorly focused to train AI agents on its own. By employing reinforcement learning, they seek to build intelligent behaviours for games with a efficiency which increases as it acquires experience on the players' actions. Incorporation of VGDL with Unity means that game environments and objectives can be easily described and thus permit iteration when testing AI. Speaking of the efficient outcomes of game development. R. Dewan et al. [3] indicates the value of contemporary search engine programmes including Elasticsearch and Apache Lucene. compelling games. The study emphasizes Unity's diverse tools, such as the Unity Asset Store, scriptable objects, and physics engine, to simplify the game development process for both 2D and 3D applications. B. Sreevidya et al. [4] introduces an AI agent that employs reinforcement learning strategy of Q-learning as well as the deep Q-learning or DQN to enhance the game's mechanical features. Training takes place inside of Unity using Unity ML Agents Toolkit which allows for reinforcement learning of an agent's behaviours in complex environments such as RTS games. M. Long et al. [5] examines the effects that local latency has on both competitiveness and user satisfaction in games. The work provides the latency influence simulation in current game engines such as Unity and Unreal and describes the acceptable latency thresholds in different game categories.

S. Bakkes et al. [6], enhanced the learning efficiency of the reinforcement learning algorithms by storing and reusing previous experiences. In this context, evaluation of the proposed system proves that RL algorithms are effective in solving complex problems in the context of gaming environment. A. Meliones et al. [7] was tasked with studying the implementation of simple adaptive artificial intelligence (AI) systems into video games designed with the Unity engine and proposed a set of ideas or approach for deploying adaptive AI techniques for improving game play environment.

Y. Liu et al. [8] employs data analytics and machine learning to model mobile gamer behaviours, utilizing tools like TensorFlow and Unity's analytics SDK. It is concentrated on the usage of player preferences to customize game mechanics for increased play time and everyday revenues. H. Li. [9] leverages Unity ML-Agents to simulate soccer gameplay and train AI agents in tactical decision-making. PPO, an acronym for Proximal Policy Optimization is among the reinforcement learning frameworks incorporated in the system and ideal for building realistic AI players. M. Sindhu et al. [10] integrates AI-driven mechanisms, such as pathfinding and dynamic difficulty adjustment, into Unity-based 2D games. Using Unity's NavMesh and custom scripts, it focuses on enhancing gameplay interactivity and engagement.

III. PROPOSED METHODOLOGY

The process of creating the fruit-merging puzzle game was systemized and build in stages to utilize main platform, the Unity game engine. This was in response to the various stages that the methodology was made up of, each tackling a different aspect of game development, without causing a disruption in the game's development process. The methodology therefore involves the incorporation of gameplay as well as machine learning systems. Several inputs relating to a player is in controller, game mechanics, and objects, whereby the collision detection, score, and simple user interface feedback are the outputs. At the same time, raw data is obtained for the purpose of training the ML model and real-time predictions are given using the same trained model. Unity sends information to the backend service to improve the game sessions with custom analytics and live updates

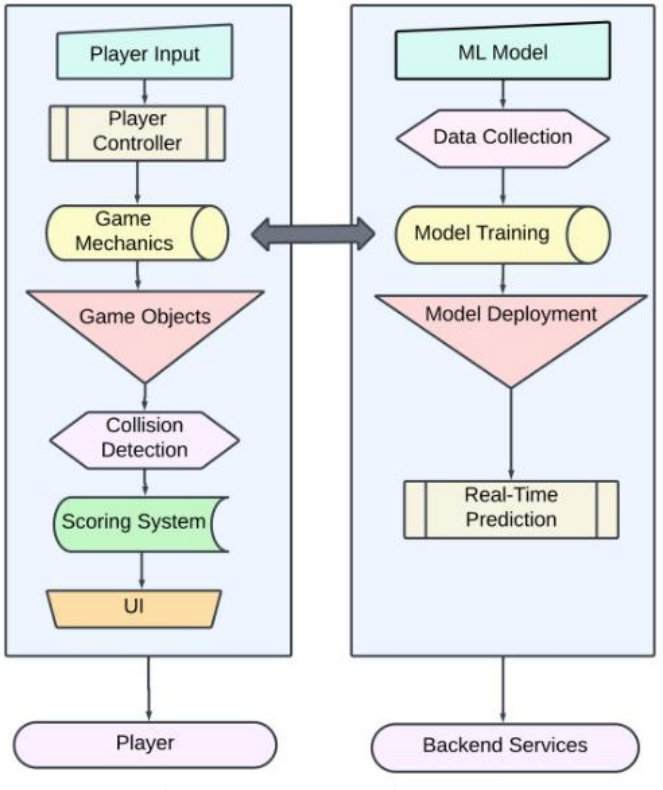

Fig.1. Flowchart of the Game Process

1. **Design and Planning:** The initial stage of proposed methodology was dedicated to the game design and planning, which outlines the essential rules of the game. The goal was to design a grid-based puzzling game where similar fruit create a more complex fruit, and increase the level of interest of the player in decision making. To capture the high level of detail of the gameplay, a detailed flowchart was created to describe the gameplay and the grid format, the interaction of the players, and the rating system. Also, the work on the user interface (UI) and the user experience (UX) presented a notion of clear and convenient controls, the possibility of navigation, and real-time updates.

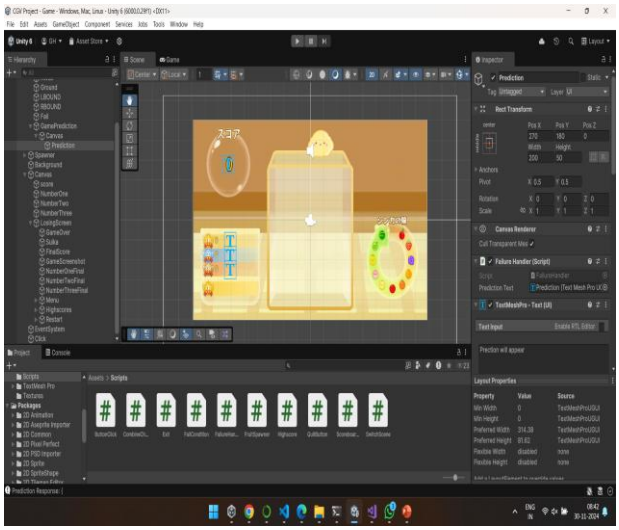

Fig.2. Unity Setup for Game Development

2. **Game Logic:** The game logic is one of the principal objectives to spatially organize fruit objects and their movements, a two-plane grid pattern was established. Sprite management tools that are available at Unity were adopted for rendering fruit objects in detail. To facilitate collision detection and response, collision detection and responses included Unity's 2D colliders and rigid bodes for merging identical fruits. A scoring scheme was established to update scores electronically based on score merges for player accomplishment.

3. **Physics Simulation:** In the physics simulation stage, 2D physics engine of Unity was used to make simulations of fruit movements and actions. These included implementations of effects, such as gravity, inertia, and inline collision responses to facilitate on the dynamic of the game play. Real physical properties were carefully fine-tuned from a real-world physics simulation to provide responsive control and fun.

4. **Machine Learning:** A Support Vector Machine model was performed to categorise players into performance levels which included: Beginner, Amateur, Intermediate, Advanced and Professional. The trained model was saved in a format also known as pickle that allowed it to be saved, loaded easily. To incorporate this machine learning functionality into the game, the model was integrated using Flask web application. First, Unity transmits player actions information to Flask in real-time, then Flask analyses the information to obtain predictions. The conclusions drawn as far as the approximate skill level of a given player are concerned are then passed back to Unity and presented on the game environment interface. This smooth incorporation of Machine learning further empowers the gaming world by offering the capability of player feedback that can lead to detailed insight into a player's ability and thereby opens the door to true adaptive difficulty and added features.

IV. RESULTS AND ANALYSIS

The introduction of the fruit merging puzzle game as well as the addition of finishing moves effectively eradicate the problem of a dull and simple game interface because of added interactivity. The manoeuvrability of the game design includes the means of control, configurable barriers, and other elements that are essential to the player, influence during the game while keeping a fair balance. This section discusses the results obtained after the implementation of the developed features and carries out a discussion on efficacy and disadvantages of above features and future possibilities.

1. **Game Functionality and Player Experience**:

The main mechanisms work fine, letting people merge identical fruits in a limited board with the help of dynamic constraints. The control scheme guarantees simple and natural interactions to avoid any hitches, to allow the players to sort fruits effectively. New challenges of the gameplay appear in barriers and grids, merging actions and their position should be strategized by players. The scoring mechanism is also interesting, which gives the players more incentive to play so they can have higher levels of merges.

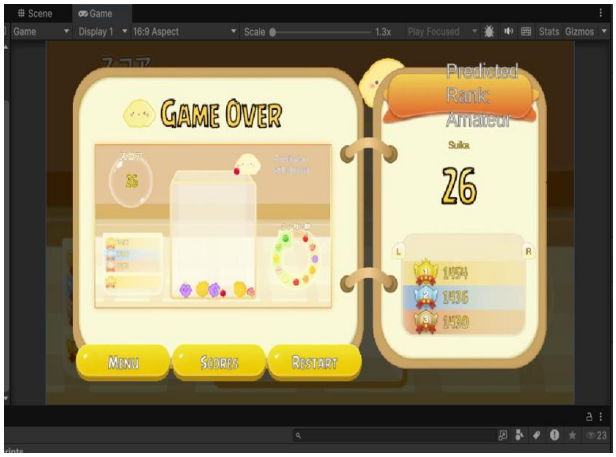

Fig.3. Output of Game with Scorecard and Rank Classification

2. **Support Vector Matrix Model Analysis:**

The result shows that the implemented mechanics effectively provide an engaging experience. The qualitative responses indicated high work engagement since the role's activities requires timely decision-making coupled with the nature of grid relations. This produces competition whilst staying fun due to the equal amount of focus placed on reward (scoring) and penalty (grid limitations).

Fig.4. represents the classification report derived from the trained SVM is over all accuracy of the proposed model including high value of precision and recall of most classes. However, there are notable insights relevant to the game's functionality

High Performance for Most Classes: There is also high accuracy and reliability of the classification in case of "Amateur," "Beginner," "Intermediate," "Pro" levels.

Weak Performance for 'Advanced': The precision for "Advanced" is about 1.00 and recall is 0.18 that indicate the high precision but very low recall which means the model rarely makes a mistake to classify non-Advanced players as Advanced players but at the same time it can miss on the actual Advanced players. It could lead to some players categorized here being left out in the gameplay activities more often.

Balanced Weighted Averages: This comes down to the fact that the averaged over the dataset F1-score of 0.83 thus taking into consideration the class frequency signifies relatively high net model performance provoking equally high gross performance of a majority of the players.

Classification Report:

	precision	recall	f1-score	support
Advanced	1.00	0.18	0.30	281
Amateur	0.85	1.00	0.92	271
Beginner	0.79	1.00	0.88	275
Intermediate	0.90	1.00	0.95	261
Pro	0.76	1.00	0.86	259
accuracy			0.83	1347
macro avg	0.86	0.84	0.78	1347
weighted avg	0.86	0.83	0.78	1347

Fig.4.Classification Report of Trained Support Vector Machine

The basic aim is to obtain the highest scores and a secondary goal given by the constraint of the available grid space make for a highly exciting competition between short term strategies and strategic planning. This balanced design makes game more user-friendly and challenging at a same time and we can all agree that more challenging programs have a higher replay value.

V. Conclusion

The development of this fruit-merging puzzle game show that Unity has a good ability to make a fun and visually polished game. Thus, by utilizing Unity 2D physics engine, animation and cross-platform compatibility features, game attains both – high realism, interactivity, and visual appeal. This project demonstrates the value of ordered progress in developing an essentially realistic physics-based simulation with complex, evolving dynamics and engaging, visually appealing rich UI/UX. Therefore, even though the game in its current state is enjoyable and easy to play, the described additions sketch the space for further growth. This work also provides a clear illustration of how Unity can be a viable platform for developing casual games and at the same time set the base for future research in casual games.

VI. Acknowledgement

We would like to express our sincere gratitude to Amrita Vishwa Vidyapeetham for the infrastructure and support provided towards the completion of this project

REFERENCES

[1] S. Singh and A. Kaur, "Game Development using Unity Game Engine," 2022 3rd International Conference on Computing, Analytics and Networks (ICAN), Rajpura, Punjab, India, 2022, pp. 1-6, doi: 10.1109/ICAN56228.2022.10007155.

[2] M. Johansen, M. Pichlmair and S. Risi, "Video Game Description Language Environment for Unity Machine Learning Agents," 2019 IEEE Conference on Games (CoG), London, UK, 2019, pp. 1-8, doi: 10.1109/CIG.2019.8848072.

[3] R. Dewan, R. Polishetty, N. Jagadam, M. K. Goyal, K. K. Ravulakollu and B. Sharan, "Significance of State-of-Art Search Engine in Game Development," 2023 10th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2023, pp. 1134-1139.

[4] K. U. Akkshay and B. Sreevidya, "Development and Performance Analysis of an AI based Agent to Play Computer Games using Reinforcement Learning Techniques," 2023 IEEE 3rd Mysore Sub Section International Conference (MysuruCon), HASSAN, India, 2023, pp. 1-8, doi: 10.1109/MysuruCon59703.2023.10397011.

[5] M. Long and C. Gutwin, “Characterizing and modeling the effects of local latency on game performance and experience,” in Proceedings of ACM CHI Play, New York, NY, USA, 2018, pp. 285–297.

[6] 6 S. Bakkes, P. Spronck and J. van den Herik, "Rapid and Reliable Adaptation of Video Game AI," in IEEE Transactions on Computational Intelligence and AI in Games, vol. 1, no. 2, pp. 93-104, June 2009, doi: 10.1109/TCIAIG.2009.2029084.

[7] A. Meliones and I. Plas, "Developing video games with elementary adaptive artificial intelligence in unity: An intelligent systems approach," 2017 Intelligent Systems Conference (IntelliSys), London, UK, 2017, pp. 104-111, doi: 10.1109/IntelliSys.2017.8324230.

[8] Y. Liu, H. Dar and R. Sharp, "Mobile Gamer Modelling and Game Performance Preference Measurement," 2020 IEEE Conference on Games (CoG), Osaka, Japan, 2020, pp. 632-635, doi: 10.1109/CoG47356.2020.9231860.

[9] H. Li, "Design and Implement of Soccer Player AI Training System using Unity ML-Agents," CIBDA 2022; 3rd International Conference on Computer Information and Big Data Applications, Wuhan, China, 2022, pp. 1-4.
.

[10] R. M. Sindhu, L. S. P. Annabel and G. Monisha, "Development of a 2D Game using Artificial Intelligence in Unity," 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2022, pp. 1031-1037, doi:10.1109/ICOEI53556.2022.9776750.