Mobile Game Development Using Unity

Ethan La, Department of Computer Science

Mentor: Dr. Markus Eger

CPP Student RSCA Conference 2023

Abstract

This project delves into the technical intricacies of creating a mobile game using the Unity game engine. Focusing on the design and implementation phases, I plan to showcase the developmental journey and game development pipeline. The use of Unity as the game engine and the mobile platform underscores the contemporary relevance of this project.

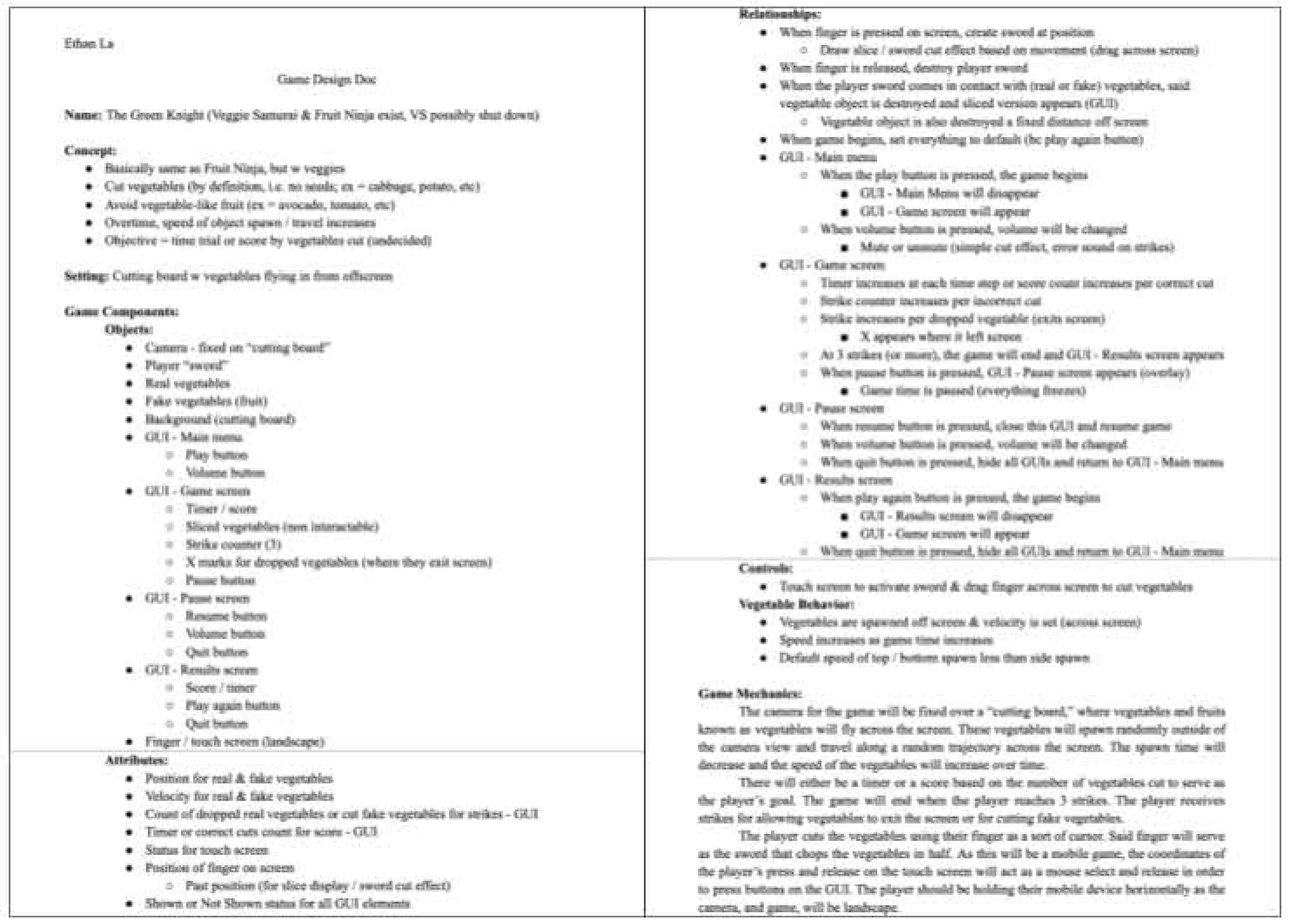
I cover the typical game development workflow, beginning with the design phase, where aspects such as setting, genre, and mechanics are defined. Transitioning into the implementation phase using Unity, the project aims to culminate in a fully playable mobile game. While the time constraint limits a direct demonstration of the post-launch maintenance phase, the importance of this phase in the game development pipeline is not overlooked.

Despite the topic's familiarity, the presentation underscores its educational value for computer science students, especially those aspiring to enter the gaming industry. As a result, the project aligns with my major and covers multiple areas of interest. Beyond its relevance to gaming, the outlined development process holds broader applicability and could serve as a valuable learning resource for creating applications outside the realm of game development.

Pre-Production Phase

The initiation of the game development pipeline takes place in the pre-production phase, or the design phase. During this stage, game developers make critical decisions regarding various aspects of their prospective game. This culminates in a comprehensive game design document, outlining the overarching vision for the game. Importantly, this document remains dynamic, undergoing continuous modifications and updates throughout the game's development.

Personally, I found the most challenging aspect of this phase to be selecting a game concept rather than creating the design document. Despite my initial inclination towards creating a roguelike game akin to Soul Knight, my final decision was to develop a game with similarities to Fruit Ninja.



Ethan La

Game Design Doc

Name: The Green Knight (Veggie Samurai & Fruit Ninja exist, VS possibly shut down)

Concept:

- Basically same as Fruit Ninja, but w veggies
- Cut vegetables (by definition, i.e. no seeds; ex = cabbage, potato, etc)
- Avoid vegetable-like fruit (ex = avocado, tomato, etc)
- Overtime, speed of object spawn / travel increases
- Objective = time trial or score by vegetables cut (undecided)

Setting: Cutting board w vegetables flying in from offscreen

Game Components:

Objects:

- Camera - fixed on "cutting board"
- Player "sword"
- Real vegetables
- Fake vegetables (fruit)
- Background (cutting board)
- GUI - Main menu
    - Play button
    - Volume button
- GUI - Game screen
    - Timer / score
    - Sliced vegetables (non interactable)
    - Strike counter (3)
    - X marks for dropped vegetables (where they exit screen)
    - Pause button
- GUI - Pause screen
    - Resume button
    - Volume button
    - Quit button
- GUI - Results screen
    - Score / timer
    - Play again button
    - Quit button
- Finger / touch screen (landscape)

Attributes:

- Position for real & fake vegetables
- Velocity for real & fake vegetables
- Count of dropped real vegetables or cut fake vegetables for strikes - GUI
- Timer or correct cuts count for score - GUI
- Status for touch screen
- Position of finger on screen
    - Past position (for slice display / sword cut effect)
- Shown or Not Shown status for all GUI elements

Relationships:

- When finger is pressed on screen, create sword at position
    - Draw slice / sword cut effect based on movement (drag across screen)
- When finger is released, destroy player sword
- When the player sword comes in contact with (real or fake) vegetables, said vegetable object is destroyed and sliced version appears (GUI)
    - Vegetable object is also destroyed a fixed distance off screen
- When game begins, set everything to default (bc play again button)
- GUI - Main menu
    - When the play button is pressed, the game begins
        - GUI - Main Menu will disappear
        - GUI - Game screen will appear
    - When volume button is pressed, volume will be changed
        - Mute or unmute (simple cut effect, error sound on strikes)
- GUI - Game screen
    - Timer increases at each time step or score count increases per correct cut
    - Strike counter increases per incorrect cut
    - Strike increases per dropped vegetable (exits screen)
        - X appears where it left screen
    - At 3 strikes (or more), the game will end and GUI - Results screen appears
    - When pause button is pressed, GUI - Pause screen appears (overlay)
        - Game time is paused (everything freezes)
- GUI - Pause screen
    - When resume button is pressed, close this GUI and resume game
    - When volume button is pressed, volume will be changed
    - When quit button is pressed, hide all GUIs and return to GUI - Main menu
- GUI - Results screen
    - When play again button is pressed, the game begins
        - GUI - Results screen will disappear
        - GUI - Game screen will appear
    - When quit button is pressed, hide all GUIs and return to GUI - Main menu

Controls:

- Touch screen to activate sword & drag finger across screen to cut vegetables

Vegetable Behavior:

- Vegetables are spawned off screen & velocity is set (across screen)
- Speed increases as game time increases
- Default speed of top / bottom spawn less than side spawn

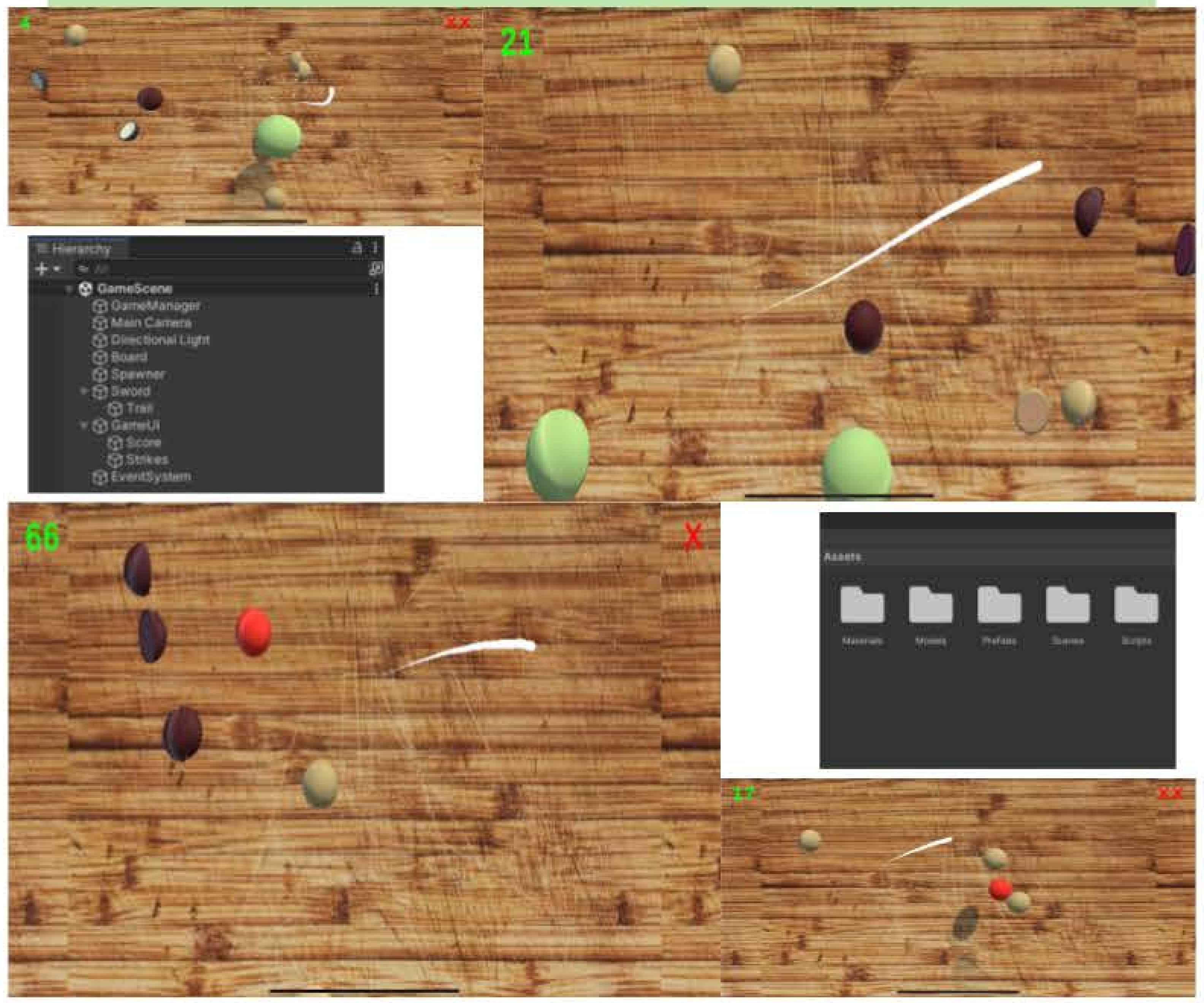
Game Mechanics:

The camera for the game will be fixed over a "cutting board," where vegetables and fruits known as vegetables will fly across the screen. These vegetables will spawn randomly outside of the camera view and travel along a random trajectory across the screen. The spawn time will decrease and the speed of the vegetables will increase over time.

There will either be a timer or a score based on the number of vegetables cut to serve as the player's goal. The game will end when the player reaches 3 strikes. The player receives strikes for allowing vegetables to exit the screen or for cutting fake vegetables.

The player cuts the vegetables using their finger as a sort of cursor. Said finger will serve as the sword that chops the vegetables in half. As this will be a mobile game, the coordinates of the player's press and release on the touch screen will act as a mouse select and release in order to press buttons on the GUI. The player should be holding their mobile device horizontally as the camera, and game, will be landscape.

Prototype



Production Phase

In the production phase, the focus shifts to turning concepts into a tangible game. This stage is commonly referred to as the implementation phase, where game developers invest a substantial amount of time and effort. To break down this extensive phase, three key milestones are typically identified: the playable prototype, the beta version, and the finished product.

The initial milestone involves creating a basic version of the game with core mechanics implemented. The playable prototype often lacks polish and sophisticated graphics but provides a foundation for further development. The beta version is an advanced iteration of the game, where improvements are made based on feedback from the playable prototype. The game is refined, bugs are fixed, and additional features may be introduced. The finished product represents the fully developed and polished game ready for public release. It incorporates all planned features, has undergone rigorous testing, and meets the quality standards set by the development team.

As of now, I have successfully achieved the first milestone, the playable prototype. This can be observed through the above screenshots. Utilizing various features of the Unity Engine, such as Materials, Scripts, and Prefabs, a modified version of Fruit Ninja has been created – tentatively named "The Green Knight." The game introduces a unique twist by focusing on cutting botanically-defined vegetables, like cabbages and onions, while challenging players to avoid cutting fruits that resemble vegetables, such as avocados and tomatoes. Scoring is based on correctly identifying and cutting real vegetables, with three incorrect cuts resulting in a game over.

Post-Production

Finally, there is the post-production phase, which is commonly referred to as the maintenance phase, though its scope extends beyond mere upkeep. The actual game launch occurs during this phase, marking the completion of the project. Simultaneously, considerations for marketing and distribution come to the forefront, laying the foundation for the game's future success. True maintenance activities commence only after a successful launch. Said activities consist of developers releasing patches which contain additional content or bug fixes. This is particularly crucial for live service games. Many mobile games adopt a free distribution model, heightening the importance of effective monetization strategies, such as microtransactions.

In regards to this project, completing "The Green Knight" poses challenges in both the launch and maintenance aspects. Opting for the App Store as my preferred platform introduces complexities, given its stringent application release checks. In addition, the intricacies of updating live service applications and games remain unknown to me. Finally, I have no serious expectations for the success nor monetization of my game, as Fruit Ninja already exists.

Conclusion

Although I have only completed the prototype as of now, I have learned a lot about the steps necessary to create a success game, mobile or otherwise. In addition, creating a mobile game is both extremely similar and completely different from creating a console or computer. Of course, this is because one must consider the touch screen aspect of mobile devices. However, many touch screen controls, such as taps or joysticks, can also be replicated with other I/O devices. For "The Green Knight," swiping across the screen is the same as holding down the mouse button and moving it across a mouse pad. The Unity Engine is definitely convenient and efficient at creating cross platform games. My final realization is that, in spite of this project's title, the three phases of 'game design' can actually be applied to creating applications completely unrelated to games.

Future Work

Once again, only a rough prototype of my game has been created so far. Therefore, there is still much to do before "The Green Knight" can be considered complete and ready to publish.

First, I must complete the beta version of the game by refining the game as a whole. In particular, the art and vegetables need significant work. Also, a more attractive GUI and more game modes would greatly improve player experience. Of course, bugs will need to be fixed when they appear.

For the finished product, I simply must improve on the beta version of the game. In addition, adding possible customization would make my game theoretically marketable.

During the post-production phase, my only concern is launching the game as the App Store's checks are quite rigorous, which I mentioned above. In regards to maintenance, I currently do not plan to update "The Green Knight" after launch. Perhaps learning more about how to do so will spark my interest. However, I do plan to do more game development in the future.

Acknowledgments

This entire project would not have been possible without the involvement of Dr. Markus Eger and Dr. Claudia Garcia-Des Lauriers. Even if they do not remember doing so, I would like to express my gratitude to them both for inspiring me to realize this project. Thank you as well, Won!