ECS657U: Multi-Platform Game Development Coursework

Game Name:
Square Dasher Bob

Part I: Design and Prototype

SECTION 1: HIGH LEVEL CONCEPTS

Concept statement

The game is 2D platform game based solely on moving and jumping to avoid obstacles and reach the goal of each level. It will progressively get harder each level, introducing newer and higher obstacles after each level. It should be fast-paced, basic and simple to understand while providing a challenging experience testing both attention and response times.

Genre

The genre of this game is primarily action-based. Action-based games focus on physical challenges requiring quick thinking and good response times. Unlike other genres, there is not a real time investment required. They are simple to understand but hard to master, providing a challenge of skillset rather than endurance.

Target audience

The target audience will be mainly teenagers/young adults. As with a lot of 2D platformer games (and actions games in general), it mainly focuses on quick reaction, which is more appealing for younger, generally more hyperactive audiences.

SECTION 2: DESIGN

Player experience & POV

The player should be able to tell the simplicity of the game through the graphics of the used. The graphics should feel appealing, however, only containing entities that are involved as a function in the level rather than a bombardment of unnecessary background details that can lead to confusion/overwhelming the player. The start point to the end point of each level should be obvious, in terms of aims. even without the support of instructions. The components of the levels should stand out from the background elements making it clear of the focus. Users should feel comfortable for the first levels but eventually reach a stage where they have trouble succeeding.

Users will encounter a spectrum level of difficulty that will be of an appropriate skill-level balance ratio. When they reach a stage where they begin to struggle to reach the following level, is it expected for players to feel slightly frustrated. However, this frustration also comes with the motivation to succeed into the next level, becoming a fuel for higher levels of concentration. At this stage, the player's combination of frustration and motivations increases the satisfaction when a level is succeeded. The game will be designed for many deaths with quick respawns, keeping the players' focus on retrying rather than giving up. It is expected that with each death, they do reach a step closer to succeed which will make the victory much more rewarding. Players are expected to feel relieved and excited after completing levels they struggled on and the feeling of accomplishment becomes the fuel for continuing the challenge until they reach the end of the game.

During the end of the game, this will obviously be the highest peak of skill-level balance. Players are expected to die a few times at the least. Players should feel frustration and determination to finish the game. The end level should be relatively obvious in comparison to the other stages e.g. boss stage. The style of the game should still be consistent throughout and if possible, the special stage should try to implement all the obstacles that were introduced through the game as a nice form of summary for the player. After completing the level, it should send them to a "Congratulations" screen which will confirm that the game has been completed. As it is a fast-paced game that challenges your physical aspects, it is expected that after the player has finished their "victory dance" they will consider retrying the game in hopes for few deaths or speed runs.

Visual and audio style

In terms of visual, I intend to mimic the cartoon-like style that usual 2D platform games work on e.g. Super Mario Bros, with the exception of possibly less pixelated depending on the free assets available on the Asset Store which would be appropriate for the game (if I cannot find an appropriate free asset for the game, I will opt for creating simple models to represent game object models). This means that the terrain will most likely be made off grass and stone (especially considering the backstory of the game) which gives an indication of forests and possibly caves. It would be a good addition to have some trees in the background for visual aesthetics and a background of a forest/cave depending on the where I decide that the bandit camp will be situated.

The visual effects will support the experience by having a simplistic design sprite that have appealing but do not stray from the main focus of the game which will be based on simply traversing through the map avoiding obstacles until the goal has been reached. Of course, this decision was also based on the limited creative control on a 2D plain as well as depending on the free assets for game object models.

The art style that suggests the style of the game will be based around popular 2D platform games such as Super Mario Bros, Terraria, etc. however the level of detail will, as mentioned before, depend on the free assets that are available.

In terms of music, I'm expecting to some retro-style music to go with the platformer. It should be non-intrusive, relaxing and essentially just to add another dimension to the player experience. It should have different soundtracks depending on the situation, for example, if we were to create a boss level, the soundtrack should be more intense and give an impression of immediacy rather than in a full jump zone in the forest where there is no need of urgency.

In terms of sound effects, behaviours such a movement (not entirely sure), jumping, death and reaching the goal would be the basic sound effects the game will have. The music should be significantly lower than the sounds so that the sound effects can be heard without any issues. Having a toggleable sound and music button that can be accessed at any point in the game would be extremely useful and having independent scrollers would solve the problem of sound to music ratio.

The audio effects will support the experience by providing relaxing music that can help the player focus as well as general enjoyment from the tune. When the sounds are triggered it can as an audio helper as second output response from the computer. In some cases, the music acts as indicator of trouble/disturbances notifying the player of cautiousness. The extra sound dimension also makes the player more immersed in the experience as it's a step closer from the virtual world to reality.

Game world fiction

The player's avatar, known as "Bob", lives in a village in a remote countryside. Bob's possessions were robbed from his home while he was away on a visit to see an old friend. The player finds out through a local villager that a group of bandits had raided not only his, but other his other fellow villagers' possessions. Bob journeys on through the forests in search of the bandits in order to reclaim the possessions that he and his fellow villagers once had. However, he is one while his enemies are several therefore his needs to be proceed with caution in order to secretly receive what was once his.

Core gameplay

For player to player avatar commands, it will be based on the player's keyboard input. These commands would generally come under movement and behaviours of the player avatar. Unity3D allows you to use the commands using a script that is added to the entity that you wish to interact with. In this case, if we wished to interact with the player, we would need a Rigidbody component added to our player avatar as well as a script to allow us to interact with the void Update() method that essentially acts as a loop for each frame allowing code to be constantly detected for each frame such as methods that detect keyboard inputs and behave according to the correlating behaviour assigned to it. For example: pressing the left arrow key may cause the velocity of the player avatar to move to the left, forming a basic gameplay command. In this case, the player's avatar will need to have movement for left and right as well as jumping command and potentially a climb command (using either up arrow or jump) for climbing up ladder platforms.

For player avatar to other game objects, it will mainly be based on the collision, either through solid collision (where the player cannot pass through) or through entering the collision zone (where the player can go through but triggers an event in doing so). Both these methods of interaction between game objects can be used to invoke a method that can allow certain actions to be performed. Leaving the collision zone can also invoke a method allowing game objects to interact depending on their relative positions. In this case, most of interactions between the private avatar and the other game objects will be based on collision (either through script code or through the Unity3D interface functions) as majority of game objects will be static.

Note: components inside the game object themselves act as the formation of gameplay. The player stopped at the collision acts as a behaviour of gameplay without an associated script to invoke a method. Other components such as animations, Rigidbody, spriteRenderer are essential for the creation of gameplay and these components do interact with not only game object to player avatar but game object to game object.

Lastly, game objects can have AI (Artificial Intelligence) functions such as enemies (with the help of a script) that can mimic the interactions between a player avatar with other game objects. AIs can really help formulate an immersive experience for the player as it gives the illusion of "real" players and usually associates with increased difficulty making the challenge much more rewarding. In this case, we can create AI-type enemies that can detect when the player is within its radius (using the BoxCollision component again) and attack the player. In this case, the boss that I wish to implement will have a projectile attack that can attack the player according to the position it was currently at. It will use a delay function so that it does not repetitively throw projectiles at the frame rate (since I will be using the update() function to check whether it's within the radius), giving the player an opportunity to move about and hopefully avoid getting bombarded with projectiles.

It's engaging because it means that, especially with AI, that an element of tactic/planning is involved in order to smoothly progress. There is also the element of curiosity, if game objects are to trigger events when you are in reach of them (e.g. a secret passageway) then players would invest different game objects in order to see what would happen if they did happen to interact with other game objects. In this case, we will have enemies with animations and potentially some AI functions and hopefully the addition of some triggerable features.

<u>Objectives & progression</u>

The game is set into levels. Each level has a start position and an end position/goal to reach. Using the movement commands that will be available (i.e. jump and horizontal movement), the player will be able to transverse through platforms in the level. Upon reaching the goal of each level, the game will send the player's avatar to the next level in order to repeat the progress until the final level has been conquered in which the game should prompt a "Congratulation" screen for the player, indicating that the game has been completed with the option to go back to the main menu if they wish to repeat or close the application. Upon death via several means e.g. fall off the screen, will respawn the user back into the original level spawn location allowing the user to reattempt the level.

The short-term goal is to complete each level that has been tasked to the player. The long-term goal would be eventual completion of the game (reaching the "Congratulations" screen). These are fundamentally the most prominent goals, however, the player may assign their own goals e.g. "I want to pass this obstacle by 10 turns" or "complete the game within [insert time]", etc.

Game systems

Since the game has very basic behaviours, majority of the bulk can be handled by the standard system provided in unity. There are some extensions that may be required for certain additions under the Unity engine such as sceneManager which can be used to create a serializeField of the scene that we wish for the goal to send us to. In the case of sceneManagement, an extension of the UnityEngine, this can actually interact with the player's avatar as it we can use it to switch the scenes according to the collision between the player's avatar and the game object goal.

Interaction

Since the game will be primarily a running and jumping style platform game, the player controls are limited to arrows keys/AWSD with the exception of having space for jump instead of up arrow if it feels appropriate to do so. Each arrow key will response to it's corresponding direction which will be the way the player will interact with the player avatar.

The primary game UI will be containing a canvas to allow the menu button selections alongside toggleable buttons. The menu buttons should consist of "Start", "End Game" and potentially "Continue" which should allow you to continue from the level you were on (but restarts it). For toggleable buttons, the buttons that would be suitable include music, sound effects and return to menu main button. In the case of the return to menu main button, the button should be removed for the main menu. If I intend to put in other features such as quality, key-binding, etc. it may be worth changing the icon into a single gear icon (as oppose to music note, speaker icon and a X for exiting to the main menu). When the buttons have been toddled off, the method should change the spriteRenderer to the muted version of the icon (this only applies to both music and sound effect icons).

Therefore the buttons in the main menu will be…
Toggleable buttons: [Music] [Sound]
Buttons: ["Start"] ["Continue"] and ["Exit"]

Whereas in-game, the setup will be…
Toggleable buttons: [Music] [Sound] [Return]
Buttons: Null

And finally, at the end of the game (in the "Congratulations" scene)…
Toggleable buttons: [Music] [Sound]
Buttons: [Return]

These toggleable buttons would look best if they were in icons and to the right-hand side of the screen in a small size so that they do not intrude with the map scenery. For the normal buttons, an animation to suggest that the player is hovering over it as well as should be situated somewhere within the bottom middle of the screen.

The game object that represents the goal of the level will be act as the transistors of the scenes by providing a script that can use the sceneManagement to serializeField the value of scenes so that it can easily be accessible to change the level (the scene change) according to whether the player's avatar has collider within the radius of the game object.

For the level themselves, the player will be restricted within the "Main Camera" vision using borders to prevent over-extending. I was debating whether to have the "Main Camera" follow the player avatar and restrict the maximum and minimum x and y values (so that the player cannot see outside the borders), however, it deemed more appropriate for the player to see the full level layout and therefore I will have decided that it should be one fixed camera point to show the full layout of the level. However, having the camera following the player does have its advantages such as allowing the seek out platforms and directions to finish the level rather than just spotting it first-hand.

However, as a boss level it may be a nice addition to have two camera points for visual effect of transitioning. In this case, it will have a reduced scope of vision following the player avatar's movements until it reaches the "boss room" which will change the scope so that the full boss room will be revealed.

For background, since we're basing the story on the Bob (the player's avatar) searching the forests for the bandits and still undecided whether their camp is situated in the forest or in a cave but essentially will either have backgrounds dedicated towards trying to convey that message. At the end of the game the background should say "Congratulations" with a short message and a background with loot that the bandits have gathered implying that Bob has recovered the stolen goods. For the main menu, the animation screen would be visually appealing, however, it should be either a picture of a village or the forest, being that those are the two main highlights of his journey.