BSc (Hons) Computer Game Applications Development

CMP 404: APPLIED GAME TECHNOLOGIES

Joseph Roper
UNIVERSITY OF ABERTAY

Contents

Introduction

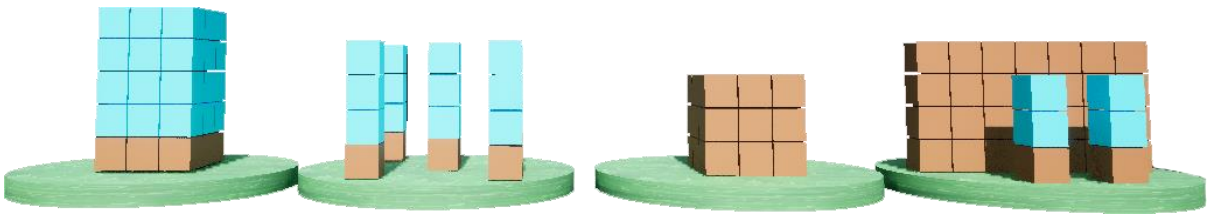
Boom Blox AR is a fan-made sequel to the Wii game Boom Blox[1], it is a physics-based augmented reality game where the player spawns a tower made of blox with their phone, projected into the real world. The player is tasked with throwing balls to knock down towers. If the player breaks all the gem blox in the level, they win and can move on to the next level. However, if they run out of balls to throw, they lose and have the option to retry.

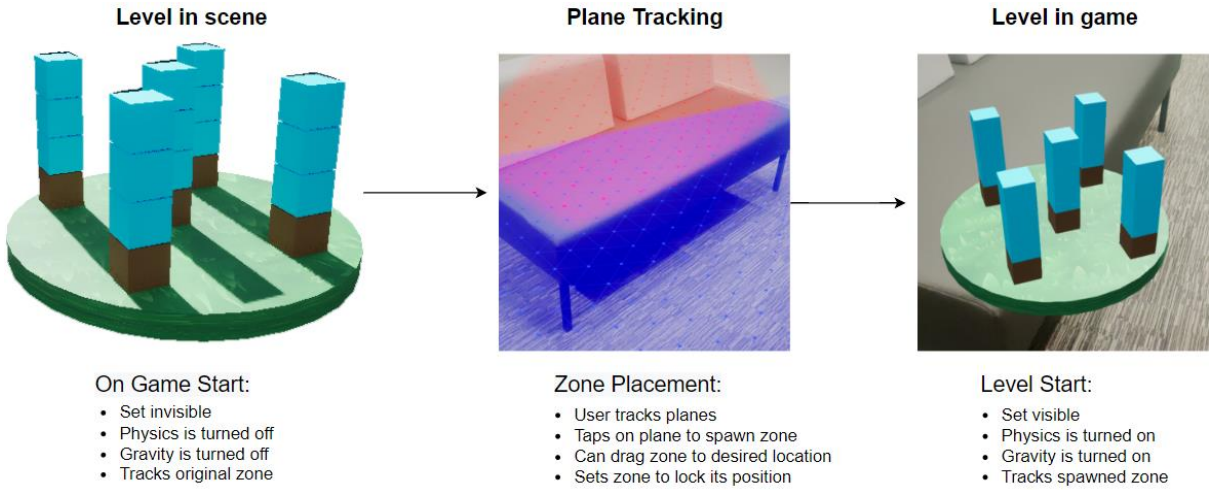
Application Guide

Open the application, Press play, track planes, tap on plane to spawn the play zone, you can move the zone by dragging it (make sure zone is flat), set the zone to start playing the game, level 1 will spawn on the zone, tap on the screen to throw a ball, the aim is to destroy all the gem blox(blue cubes) before running out of balls.

Features & Operation

The main aspect of the project is centred around blox as it's used for the main gameplay loop, the blox towers are created in the editor atop a zone which acts as a level. Levels are created in the editor and not spawned from code to allow for rapid prototyping of level designs because the levels can be visualised while being created instead of being co-ordinates in code.

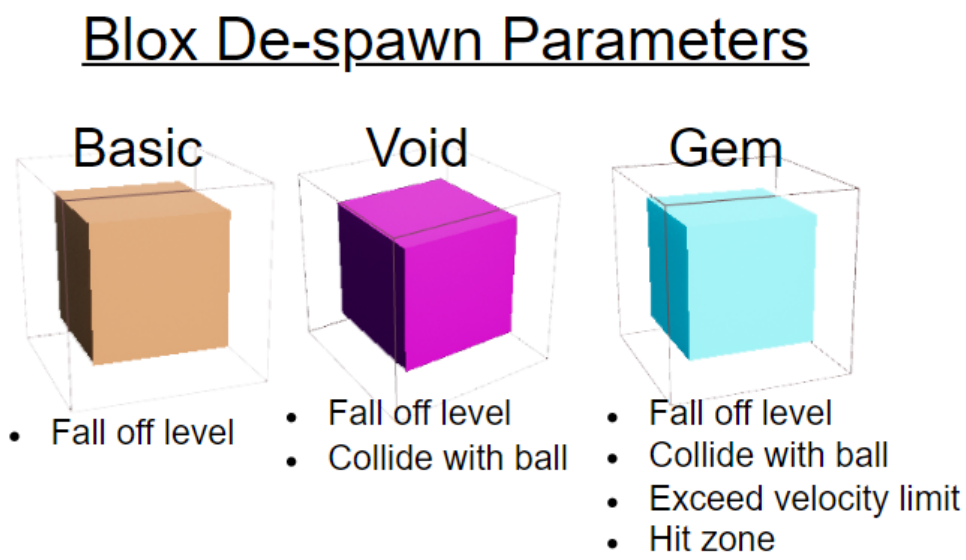




The diagram above shows the states of the blox when they are constructed, they start as being not interactable in the game, however, once the player spawns a zone, the SpawnLevel function is called in the CustomGameMode. This function takes in the number of the level that wants to be spawned as a parameter, if the active level number is not equal to this, the blox in play are sent back to their original position in the editor and track their levels zone using the BloxActor function MoveActorToOrigin which makes it not interactable. It then gets the blox contained in the level

[1] *Boom Blox* (2008)

wanting to be spawned and calls the SetUpLevel function which makes the blox interactable as stated in stage 3 of the diagram.



The basic blox behaves like wood, it can collide with actors using its applied physics body.

The void blox acts like a balloon and pops on impact with the ball.

The gem blox is the scoring system of the game, the goal of every level is to destroy all the gem blox contained within it. When a gem blox is destroyed, it accesses the game mode and increases the active LevelGemBlox hit counter by 1.

The fall-off level de-spawn is calculated in the blox actors tick function. It uses the zone's location and checks if the blox Z position is lower than the zones.

The collide with ball de-spawn is achieved by having a trigger-sphere attached to the ball, and a trigger-box to the blox. When the blox is in play, it checks if the trigger sphere overlaps with its box. If true, the blox de-spawns instantly. This allows for interesting levels to be made which require the player to utilize this mechanic to beat the level.

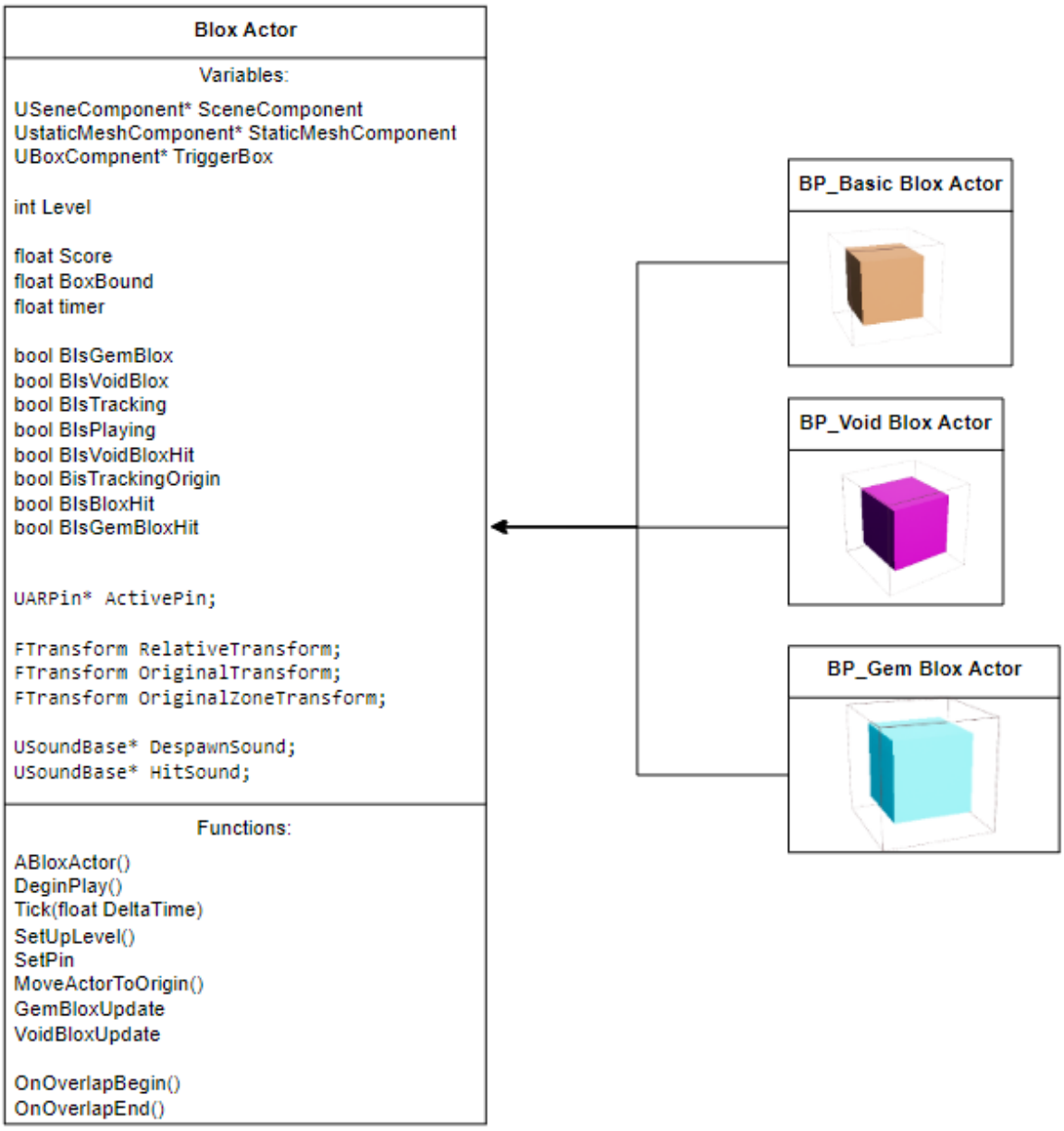
The exceed velocity limit de-spawn is achieved by checking the blox velocity in the tick function, if it exceeds a set limit it de-spawns. This allows the player to knock other blox into the gem blox to de-spawn them.

The hit zone de-spawn uses the blox trigger box to check for overlap between the zones trigger box, if true the blox de-spawns after a set time.

When a blox is “de-spawned” it calls the MoveActorToOrigin function which basically reverts it back to stage one non- interactable in the game.

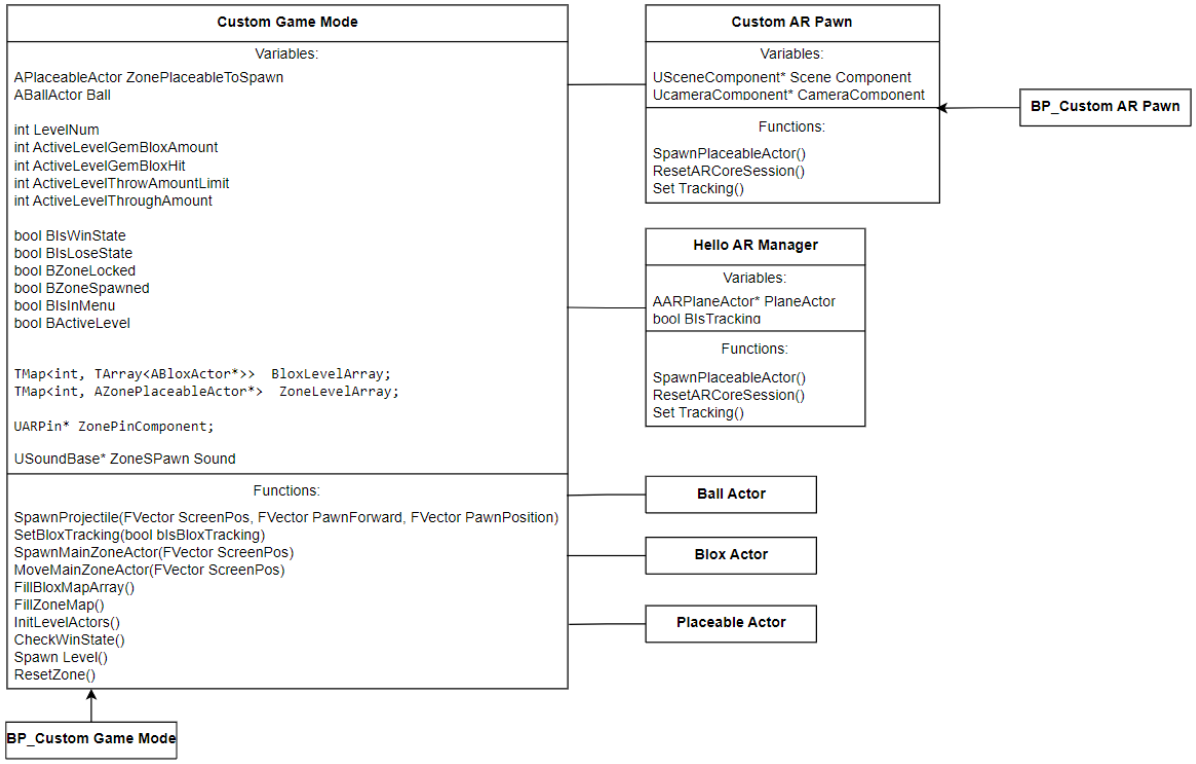
Class Diagrams

Blox Actor



The blox actor is a default class which is inherited by the different blox types, data only blueprints are created which inherit from the blox actor class to create new blox. They are each given different materials and set values, data-only blueprints are used so that the actor can be represented in the editor and can be easily resized for the creation of levels, in the future the materials will be further expanded to be more aesthetically pleasing similar to the original game. Different sounds can be easily applied allowing for a more dynamic soundscape when playing the game.

Custom Game Mode



The custom game mode acts as the brain of the application. It holds most of the functionality for the game and collects the level data within the scene such as the blox and zone and stores it so that it can be accessed for spawning levels.

On BeginPlay the levels are filled into 2D maps, the blox are filled into a map containing an integer representing the level the blox are associated with and a blox actor array to be filled with the levels blox. The type is ABloxActor so that any subclass including the basic, gem and void blox can be contained within the map. The second map contains the levels zones and is used to fill the blox with position data so that they can track their original position.

```
TMap<int, TArray<ABloxActor*>>  BloxLevelArray;
TMap<int, AZonePlaceableActor*>  ZoneLevelArray;
```

The maps are filled using the GetAllActorsOfClass function within Unreal. Maps were used to store the scene actors because the function is computationally expensive, and since the blox and zones will consistently be accessed throughout the game, reusing that function could cause delay based on the hardware of the user's phone and the game should be playable on most devices, even budget ones without the best processors.

After the maps are filled the level actors are initialised, the blox relative transform to the zone in its level is calculated and the blox is set to tracking. The zone's gem blox amount is then increased based on if the blox is of type Gem.

The CustomGameMode handles the plane tracking and ARCore session allowing the user to spawn the zone, change its position by dragging, lock its position and reset the plane tracking allowing the user to reset the zone.

The CustomGameModes tick called every frame checks the win state to see if all the gem blox in the level have been destroyed. If true, the player wins and can proceed to the next level. However, if the player runs out of balls to throw for that level without destroying all the gem blox they lose and are given the option to retry.

CustomARPawn

The custom AR pawn controls the user input, allowing the player to interact with the game. This works by getting user input and calling functions from the CustomGameMode. Currently, it recognises touch and hold movement from the player, depending on which state of play the game is in, touch output behaves differently. On start touch is required to spawn the active zone, if the zone is spawned the user can hold their finger and drag the zone to a new position. Once the zone position is locked the user will then be able to spawn balls on touch.

The shooting mechanic works by getting the 2D screen position of where the player's finger touched and de-projecting it into the game 3D world space, it then spawns a ball actor in that position and applies a force in the direction the camera was facing when the touch was inputted by performing a ray trace. It then accesses the CustumGameMode to increase the ActiveLevelThrowAmount integer.

AR Implementation

The design exploits AR by converting a Wii game which was shackled by camera movement. While playing you must constantly move the camera for better positioning to shoot, without zoom being implemented. By converting this into AR the player can move around the level with ease to figure out how to complete it with the least amount of throws, and zoom by simply moving closer for pinpoint accuracy.

With the use of plane tracking the user gets to select their play zone, and with pin-tracking, once the play zone is set it will stay in place not interrupting gameplay by moving, so the player always knows where the zone is, if the user wishes to reset the zones' location they can through the settings menu so that if mid-game the zone gets obstructed, they can reset its position to the new desired location.

Application Improvements

To improve the application and add new gameplay elements research went into a paper which uses augmented reality to assist level creation, by scanning 3d objects it can convert them into models to be shown in-game. This could be used for Boom Blox AR by allowing users to create a tower and scan it into the game, making playable levels.[2]

By combining this with a paper documenting how to add pattern recognition to AR's scanning, the user-created tower could use colour to define the blox types in the structure. For example, the purple block would be scanned and converted into a void blox within the game. Further application of this pattern recognition could allow for card scanning which spawns new towers to play, or equipable one-use items such as a bowling ball which could be thrown causing more damage to the tower, allowing for higher scores.[3]

[2] Beever (2019)
[3] Lee (2009)

Evaluation

The system planning of the core game mechanics was well executed and is designed to allow for new blox or balls with different behaviours to be easily implemented through inheritance. The Boom Blox gameplay was implemented well in AR and could make for a fun release.

While developing this application, many issues were found and resolved. The first problem encountered was that blox would teleport back to their initial position when moving. Due to the actor simulating physics, the static mesh components' position was overriding the actor's position, therefore the static mesh's position needed to be changed.

When initially trying to store the level's blox, a 2D array containing an integer and actor pointer was created. Although, the option to populate the array within the editor would not work. When researching the problem, developers recommended using the expensive function GetAllActorsOfClass although as stated previously this option was not viable for the application.

During the audio implementation, the game's music would stop playing, this was due to Unreal's garbage collection, if too many sounds were played it would stop others and because the music played on startup, it would be the first deleted. The resolution was overriding the files' concurrency, removing it from the garbage collection.

Research

An Angry Birds AR game[4] was recently produced, matching the gameplay of the project. Whilst playing, the game expects the player to place the play area on a flat surface, as the blocks are meant to act like they fall onto the surface scanned. When the play area is not placed on a surface the immersion breaks as the blocks collide with an invisible plane leaving them suspended in the air.

To counteract this, the project spawns a playable zone holding the blox, which can fall off the zone. This allows the user to play with the zone suspended in the air without breaking immersion, meaning the game can be played in more locations.

Blocks Tower AR[5] is a recreation of the classic game Jenga for AR. It uses physics for the tower blocks like the project's blox. Whilst playing it was noted that when you lose and try to restart it requires the user to track the planes again to spawn the tower, this takes a substantial amount of time and causes annoyance whilst playing the game due to the long reset process. The project overcomes this issue by using the zone spawned for all the levels unless the user wants to reset its position. Also,

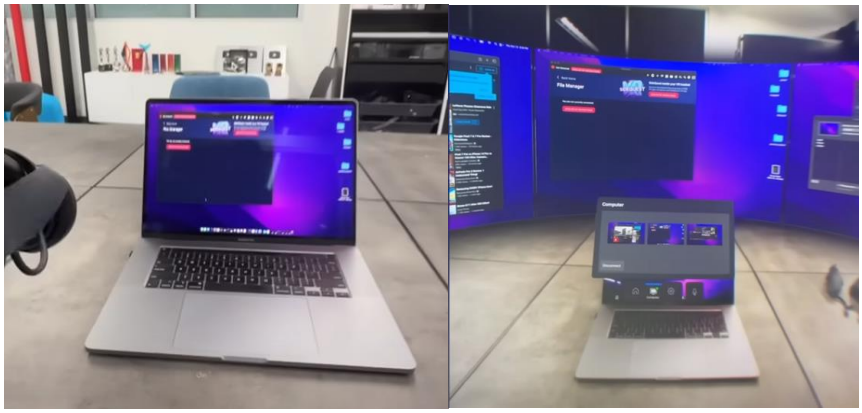
[4] *Angry birds ar: Isle of pigs* (2019)
[5] *Blocks Tower AR : Mobile Jenga* (2022)

the game's play area moves resulting in the tower falling this is a common fault with plane tracking currently and steps must be taken to resolve it.

As Pin tracking is occasionally unstable in the project based on where the planes were scanned. This results in blox getting dislodged due to zone shifting. Research into plane/pin tracking is needed to resolve this. A countermeasure could be to track zone velocity, if it exceeds a certain limit, blox physics could be temporarily disabled, stopping movement. A new blox distance would be tracked every frame and applied to the blox when the zones' velocity decreased, along with the physics being turned on, resulting in the blox remaining in its position on the zone before the shift.

State of AR

AR is an innovative technology that is still growing, but with proper care and development, it could allow for the creation of immersive experiences, by overlaying digital graphics onto the physical world. For example, the Quest Pro headset has a pass-through feature that can extend your computer's display onto multiple screens without the hardware, with improvement into a lightweight design like glasses many things could be developed such as a horror game using your own home/environment around you as the level, overlaying a monster onto the player's vision.



References

Papers

Beever, L., Pop, S.R. and John, N.W., 2019, September. Assisting Serious Games Level Design with an Augmented Reality Application and Workflow. In CGVC (pp. 9-17).

Lee, S.H., Choi, J. and Park, J.I., 2009. Interactive e-learning system using pattern recognition and augmented reality. IEEE Transactions on Consumer Electronics, 55(2), pp.883-890.

Games

Boom Blox (2008) Nintendo of Europe GmbH. EA. Available at: https://www.nintendo.co.uk/Games/Wii/Boom-Blox-280682.html (Accessed: December 12, 2022).

Angry birds ar: Isle of pigs (2019). Rovio Entertainment. Available at: https://play.google.com/store/apps/details?id=com.rovio.abar (Accessed: December 12, 2022).

Blocks Tower AR : Mobile Jenga (2022). Extins Interactive. Available at: https://play.google.com/store/apps/details?id=com.KaksProduction.BlockstowerARVR&pli=1 (Accessed: December 12, 2022).

Art

Mobile Game GUI (2013). Available at: https://graphicburger.com/mobile-game-gui/ (Accessed: December 12, 2022).

Tiling Grass Texture (no date). Available at: https://myrthevantol.artstation.com/projects/q9eEnP?album_id=1415984 (Accessed: December 12, 2022).

Audio

Mothersbaugh, M. (2008) Boom Blox Ost - 01 Boom Blox main themes. Available at: https://www.youtube.com/watch?v=ec_zKke1LJw&list=PLC748A1EF3979463E&index=2 (Accessed: December 12, 2022).

Mothersbaugh, M. (2008) Boom Blox Ost - 02 - Hum Stutter. Available at: https://www.youtube.com/watch?v=mxcEJrJo3-U&list=PLC748A1EF3979463E&index=4 (Accessed: December 12, 2022).

Cheap hollow plastic remote control button press, click 1 sound effect (no date) ZapSplat. Available at: https://www.zapsplat.com/music/cheap-hollow-plastic-remote-control-button-press-click-1/ (Accessed: December 12, 2022).

Black powder poof sound effect (no date). Available at: https://www.videvo.net/sound-effect/black-powder-poof-pe1036901/236552/ (Accessed: December 12, 2022).

Cartoon mouth pop, Lip Flick 3 sound effect (no date) ZapSplat. Available at: https://www.zapsplat.com/music/cartoon-mouth-pop-lip-flick-3/ (Accessed: December 12, 2022).

Cartoon Poof - sound effect (2021). Available at: https://www.youtube.com/watch?v=UO0jQMhof6I (Accessed: December 12, 2022).

Minecraft Break Wood sound (2017). Available at: https://www.youtube.com/watch?v=E2ge0I1lUpA (Accessed: December 12, 2022).