Leap Report

L·E·A·P Leadership. Education. Achievement. Personal Growth.

LEAP Report – Competition Engagement (HS)
Competitive Event Leadership Experiences

Video Game Design
Competitive Event

7135
Participant/Team ID#

The Student Leadership Challenge*		Leadership Experiences
Practices	**Behaviors**	
✓ Model the Way	Follow through on promises and commitments	All team members came together to create a visual appealing game that was fun and challenging. All differences were set aside and we were all able to discuss to come to a common goal.
	Set a personal example through actions	
	Align others with principles and standards	
	Seek feedback about impact of actions	
	Make sure teammates support common values	
	Talk about values and principles	
✓ Inspire a Shared Vision	Look ahead and communicate future ideas	The team worked together to teach new techniques and spread knowledge about the Video Game Design field. All team members ideas were properly utilized in the creation of the game.
	Describe ideal capabilities	
	Talk about how future could be improved	
	Be upbeat and positive	
	Communicate purpose and meaning	
	Show others how their interests can be realized	
✓ Challenge the Process	Challenge current skills and abilities	The team evenly divided the workload and made game files accessible to all using repository servers. New advanced techniques were learned to create the challenging virtual reality aspect of the game.
	Break projects into smaller do-able portions	
	Search for innovative ways to improve	
	Ask "What can we learn?"	
	Take initiative in experimenting	
	Help others try out new ideas	
✓ Enable Others to Act	Foster cooperative relationships with others	All team members got along great. All team members utilized our main communication method of online messaging. All team members were given equal opportunities to participate in the Video Game Design process.
	Actively listen to diverse viewpoints	
	Treat others with respect	
	Support the decisions other people make	
	Give people freedom and choice	
	Provide leadership opportunities for others	
✓ Encourage the Heart	Praise people	Several team celebrations took place after school to celebrate major milestones in the creation of this year's Video Game.
	Encourage others	
	Express appreciation for people's contributions	
	Publicly recognize alignment with values	
	Celebrate accomplishments	
	Creatively recognize people's contributions	

L·E·A·P Leadership. Education. Achievement. Personal Growth.

LEAP Report – Competition Engagement (HS)
General Leadership Experiences

Video Game Design
Competitive Event

7135
Participant/Team ID#

Leadership Categories	Leadership Experiences
✓ Leadership Roles	Team members took leadership roles to help distribute the work down the pipeline. Also leaders were in charge of making sure that communication was present in their branch. Leaders were also responsible for making sure that work done by team members was efficient and compatible with the game engine software.
✓ Community Service/ Volunteer Experiences	This year several sessions were held to help the Hart County Middle school TSA chapter with their Video Game challenge. Middle school had less skill in this field so the High School team taught them valuable lessons that will help their design in middle school and in the future when they get to high school. We also taught a class at the 2019 TSA Fall Leadership Conference on Jekyll Island covering Video Game Design. Team members also mentored fellow students to teach them more about the field of Video Game Design. Members taught 3D Modeling, Music production, and Coding.
✓ Leadership Development	All team members were active participants of the chapter and attended all chapter meetings. All team members took part in chapter fund-raisers to help raise money for the 2020 state conference.
✓ College Career Planning	The team practiced industry standard techniques prepare ourselves for college level video game design. Also the team used industry standard software to create complex systems and games.

The Student Leadership Challenge* Practices and Behaviors	
Model the Way	Follow through on promises and commitments
	Set a personal example through actions
	Align others with principles and standards
	Seek feedback about impact of actions
	Make sure teammates support common values
	Talk about values and principles
Inspire a Shared Vision	Look ahead and communicate future ideas
	Describe ideal capabilities
	Talk about how future could be improved
	Be upbeat and positive
	Communicate purpose and meaning
	Show others how their interests can be realized
Challenge the Process	Challenge current skills and abilities
	Break projects into smaller do-able portions
	Search for innovative ways to improve
	Ask "What can we learn?"
	Take initiative in experimenting
	Help others try out new ideas
Enable Others to Act	Foster cooperative relationships with others
	Actively listen to diverse viewpoints
	Treat others with respect
	Support the decisions other people make
	Give people freedom and choice
	Provide leadership opportunities for others
Encourage the Heart	Praise people
	Encourage others
	Express appreciation for people's contributions
	Publicly recognize alignment with values
	Celebrate accomplishments
	Creatively recognize people's contributions

ARCHERY 2020

THE PATH TO TOKYO

Video Game Design Portfolio

Athens, GA

2020

Team ID: 7135

Download Link:

https://drive.google.com/open?id=1VSGDIt8_4bq0SCPN01PpW2_xPURu2CF2

Table of Contents

Description and Purpose of the Game

Archery 2020: The Path To Tokyo is intended for virtual reality headsets (specifically the Oculus Rift and the Oculus Quest); however, we also created a non-VR version playable on any Windows PC. Competitive archery is an exciting sport in the 2020 Olympics, targeted towards an audience that is looking for a fun, challenging, and educational virtual reality experience.

The game has two basic modes: Campaign and Practice. The Practice game mode allows the player to select a location, then remain there for as many rounds of archery as desired. Alternatively, the Campaign game mode takes the player across several countries, beginning in the United States, then to Russia, Japan, and finally ending up at the archery event in the 2020 Tokyo Games. This is the "the Path to Tokyo." The idea being an emphasis on the various countries that participate in the event, while at the same time giving the player the educational experience of shooting a virtual bow in a VR archery competition.

There are currently four locations for our virtual archers to travel on "the Path to Tokyo." The United States is a farm scene, where young professional archers might begin their careers. Russia offers a glimpse into a post-Soviet city, where an archer might rise to maturity. Finally arriving in Japan, the archer is now training in the finer arts of balance and harmony with the bow in a night-time Japanese zen garden. Now with the skills required, the archer at last is prepared for the world class archery competition at the Tokyo Games, where the crowd reacts to every miss, and every bull's eye.

As we have seen with students at our school, many of our play-testers have very little prior experience shooting a real bow. Using a virtual reality environment with the Oculus Rift or the wireless Oculus Quest, *Archery 2020* allows these students to have a safe virtual experience with a bow - how to knock the arrow, aim at the target, pull the string, and the reward of maybe hitting that bull's eye.

Controls and How to Play

Non-VR	VR	Function
Right Mouse Button	Right Grip Trigger	Grab Arrow while looking at Quiver
Left Mouse Button	Right Index Finger Trigger	Hold to Draw Bow, Release to Fire Arrow
Spacebar	Controller Button B	Close-Up Target View
Enter	Controller Button X	Retrieve Arrows / Next Level (Campaign)
“R” Key	Push in Left Thumbstick	Restart (only in Practice Mode)
	Push in Right Thumbstick	Recenter VR View
	Left Oculus Button	Toggle Help UI Overlay *

*In non-VR version, most instructions are posted on in-game help screen

** To change game settings, such as Graphics Quality and Resolution, right click on .exe then hold down the left shift key while clicking Open from the drop down selection.

Team Self-Evaluation

[Chandler]https://docs.google.com/spreadsheets/d/1mjnnex1Risf5rc6Vgyd9ZHzaNJMu7TbuYlS4OvpJl7g/edit#gid=0https://docs.google.com/spreadsheets/d/1mjnnex1Risf5rc6Vgyd9ZHzaNJMu7TbuYlS4OvpJl7g/edit#gid=0

We worked as a team to complete our video game. With one month in school before video game rules were announced, we started brainstorming. We did not give specific jobs but we aligned ourselves with our specialties. We had a lot of ideas because the sporting event hosts so many events. We came up with an archery game. After a while of development, we decided to create multiple maps to add to the diversity of the game and express the diversity of the world. CJ, AJ, and I worked on 3D modeling, while CJ and Sean worked on coding and all Unity configuration, and Carter worked on music.

I created several models during fall break. Later, I learned to improve my texturing. I created several of the models for the game. We all had to participate during our personal time to reach our goals, and we all aligned ourselves with the common goals of the video game design. We all worked on the notebook and spreadsheets. All of the main designers have made models which have been used in the game, and all of us have taught someone something to help them get better at video game design.

[AJ]

This year was certainly a different experience for me with the video game. This year the whole team saw a better diverse field of ideas towards the game we decided to make. With new software the collaboration aspect of making video games was much more utilized. Proper communication through a discord chat was implemented. Also this year we created a GitHub repository server to host the files. This allowed team members to work whenever and wherever they wanted to. I think this made working on the game more accessible to us and we saw an even work distribution across the board.

At the beginning of this year we struggled to come to a decision on the event inside of the olympics that we were going to choose to make. Because of technical limitations we were very skeptical about how we were going to make an archery game. For my job as a modeler I had to constantly practice keeping my models very low spec because we were going to be running the game on a new wireless VR platform. (Oculus Quest). My experience working with a team to find the way that we were going to efficiently make this game was challenging, time consuming, and overall fun.

The modelling workflow this year was much more efficient than last year. With the use of the repository server the modelers were able to import our models straight into the game without going through someone who had the game files. This made the process easier because with every model at least once it had to be retopologized to fix normals and other shading issues.

Because of the modellers having access to the game files we could easily do the fixing ourselves instead of going through a middle man.

Hundreds of man hours went into creating the visual aspects of the game. (Models and Graphics). This was especially a challenge for me to balance sports, school, and TSA. And with this being a team project I did not move at my own pace I moved at the pace of my team. I learned new techniques for modelling such as Dynamic Geometry Displacement, Sculpting, Texture Painting, Texture Baking, and Photogrammetry which I plan to use for models in the future of this game. I'm really excited to see the future of this game and its place among other Virtual Reality games.

[Sean Pak]

Personally, I greatly enjoyed being able to work together with my fellow teammates in the production of Archery 2020: The Path to Tokyo. Before I joined the team, I was always aiming to create my own video game by my own efforts with my own work; however, I would fail to reach my goal each time because of the fact that I was on my own.

Without teammates who were keeping me responsible for my work, I would never have finished a project by myself within a set time frame. Without teammates who filled in the gaps I could not cover, I wouldn't be able to overcome the various pitfalls and holes that my skills would never be able to fill. Without teammates who were just as enthusiastic as I about video game design, I would never have understood how amazing it is to be able to work as a team.

A lot of my efforts this year were focused towards developing a lot of the backend or unseen aspects of the game. Some of the C# scripts that I developed included working on the scoring mechanism, the controls for loading and firing the bow, as well as the menu UI.

The scoring mechanism was my own creation, birthed from research on the distance formula in relation to the formula for a circle's radius. This formula allowed the game to recognize the collision point of an arrow with any target and calculate score/distance from the center with an accuracy to approximately three digits. The formula is shown below:

$$Score = \frac{\sqrt{(targetX - arrowX)^2 + (targetY - arrowY)^2}}{targetScale} \times 20$$

Next, in collaboration with one of my mentors, I devised a control scheme for both VR and non-VR environments that function on the same premise but with different inputs. The VR controls were based on object collision with the hand sensors whereas the non-VR controls were based on the mouse and camera positioning.

Arrows would be placed into the player's hand/bow using colliders/raycasting respectively. Next, the power of the arrow's display position would be modified based on a formula that calculated the distance the bowstring was pulled back in relation to the predetermined power ratio. The ratio was as follows:

$$Pullback\ Distance = \frac{\sqrt{(powerFactor \times elapsedTime)} - maxPower}{5000}$$

The third major element that I worked on and designed was the main menu UI elements. A lot of time was dedicated towards appropriate animations that covered the following requirements:

- Looked aesthetically pleasing and intriguing to the eye
- Flowed smoothly without any choppy motion or sudden changes
- Scaled with the size of the screen dynamically

Although the task seemed simple at first, designing the current menu took weeks to implement due to the trial-and-error based method of ensuring that the UI elements followed all three of the above stated requirements.

All in all, this experience in being able to work together with fellow students who are passionate in video game design was overall a greatly positive one. Not only have I learned a lot in this experience, I have also learned the importance of having a good team that is capable of filling in the missing aspects of your skillset.

[CJ]

This year Video Game Design presented a lot of new challenges and experiences. Overall it was a great year getting to work with the team to overcome all the roadblocks we hit throughout the year. Our team used specialization, organization and communication to get our game done this year. It was interesting to use GitHub for the first time this year so the team could work on the game wherever and combine work with others all in one place. It made our workflow run much smoother this year and allowed for less conflict when trying to make changes to the game. Our team used discord as a way of communication so everyone knew what they were supposed to be working on at all times and if there were any problems it was a great place to ask questions.

I learned a lot about modeling and texturing this year as we created our archery game. It was difficult to keep the models fairly low poly without taking away the looks of the model but, as the year went on our team got better and better at creating models that we could use in VR. Lots of time was spent creating the aesthetics of the game and the team did a great job working together to make a final product that looked good. Overall this year taught me a lot about what my limitations as a modeler were and how I can surpass that to become an even better

modeler. I am excited to see where our team takes the game next and how we become better in our fields that we specialize in.

[Erik Blomberg]

I worked primarily on coding and unity development. This year's game was a challenge to beat last year's game in both creativity and Technical aspects. We released it on both the quest, rift, and a 'pancake' monitor version. I worked on a couple more niche items such as a code that generated people for the stands as well as a Virtual reality UI system that both did not make it into the game. Things that did include a lot of bug testing and a jumbotron.

Unfortunately, I didn't get to work as much on this game as I did in previous years due to high hours I work at my job and rigorous college classes with Kennesaw State and Toccoa Falls. Although, I still stay up to date with how everything is run and how it works together.

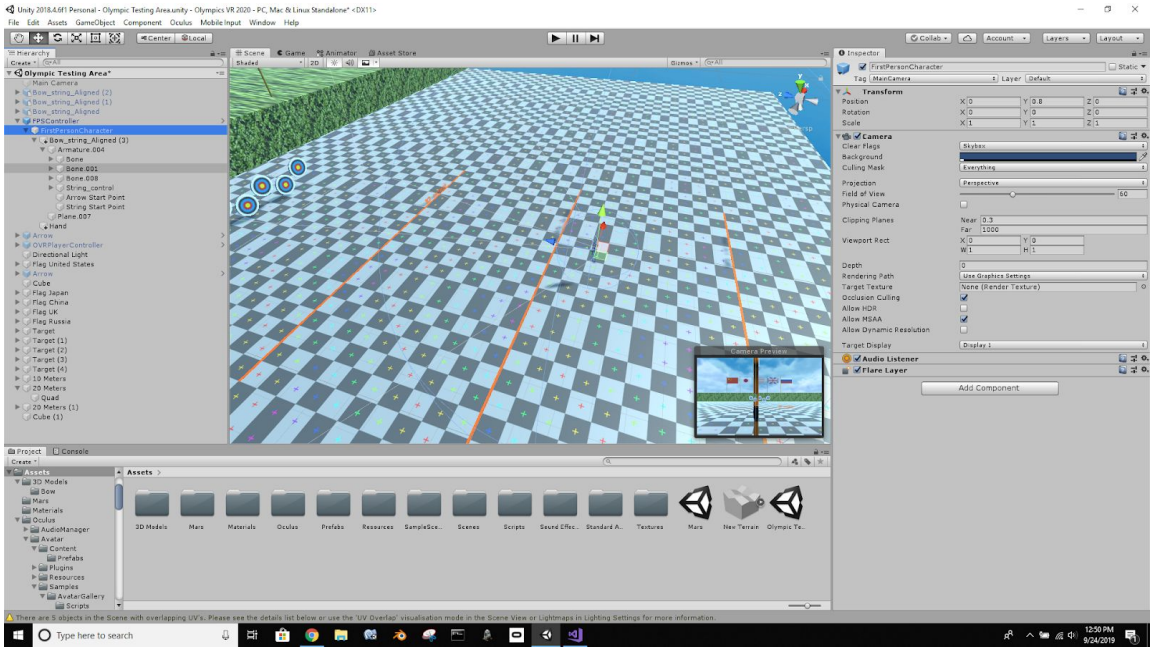
[Carter Reed]

I served as an audio design mentor for the team, and I wrote some of the music like the ambience for the Japanese themed level and worked together with my mentees to create the intro theme.

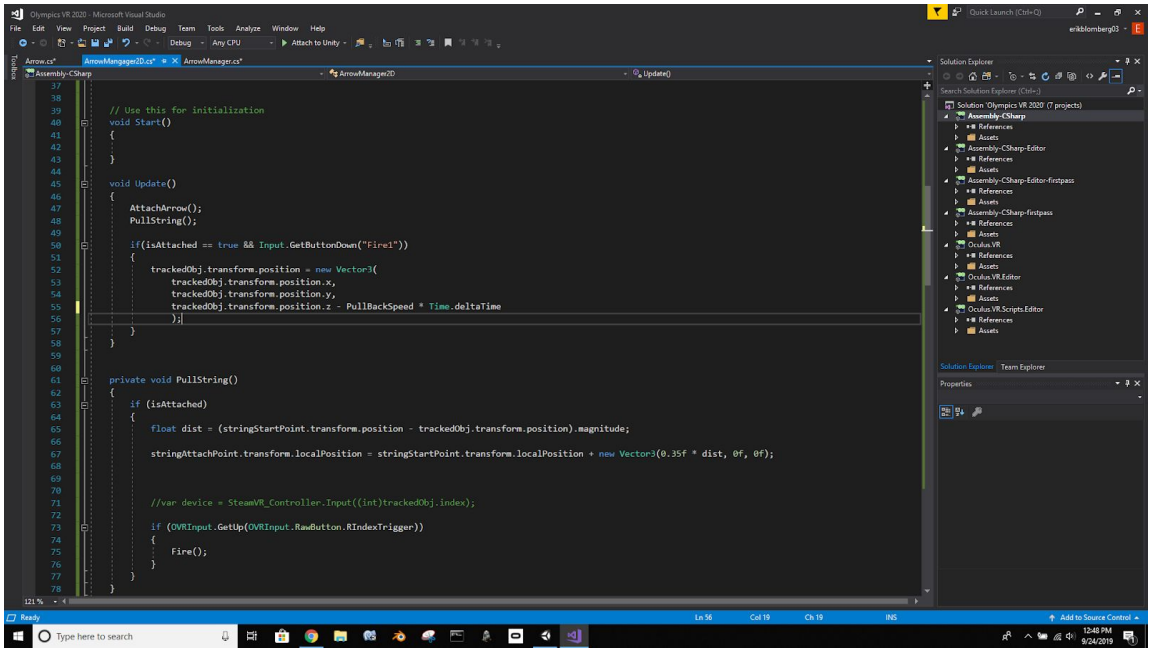
This year, we learned how to use an online music-creating software called "SoundTrap" which is a browser based DAW (Digital Audio Workstation) in order to create more electronic/dubstep sounding music that would fit the theme of a digitally advanced city like Tokyo. This project gave us an environment to learn more about MIDI software as well as various clipping and remixing techniques. The intro theme was created solely using electronic instruments in order to preserve the futuristic feel that most people would imagine when referring to Tokyo.

At the beginning, I began tutoring my mentees on various techniques and insights on how to create audibly aesthetic music. Soon, they were able to take off on their own and they created an entire slew of various musical bits on their own without my direct assistance.

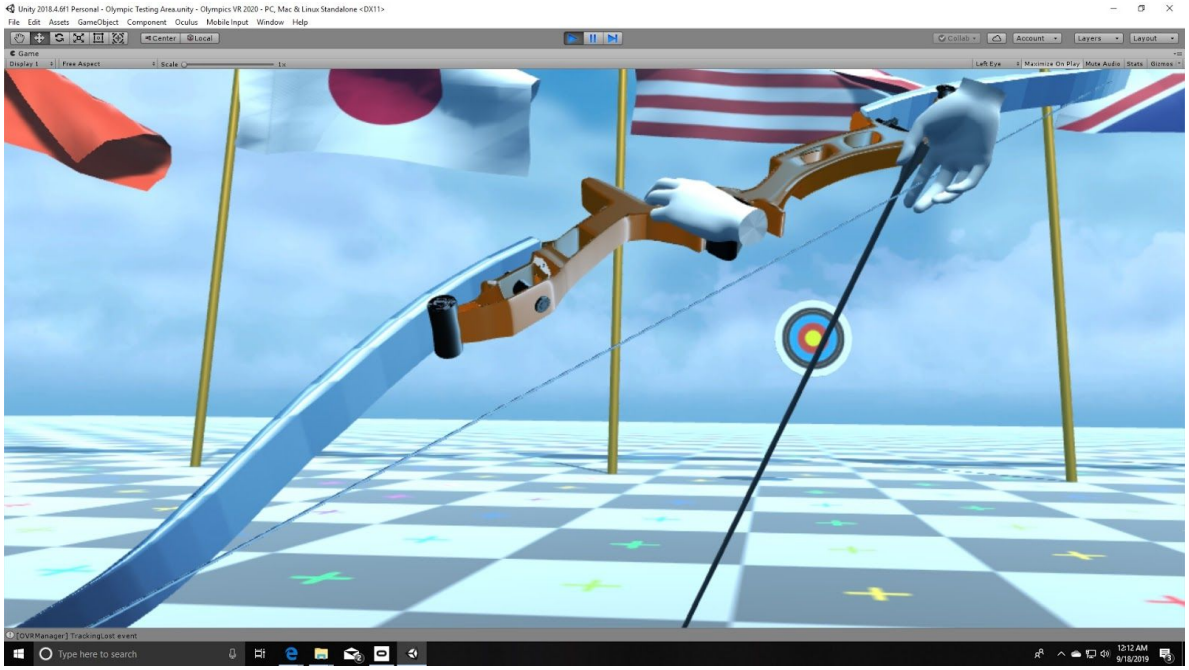
Screen Captures of Development



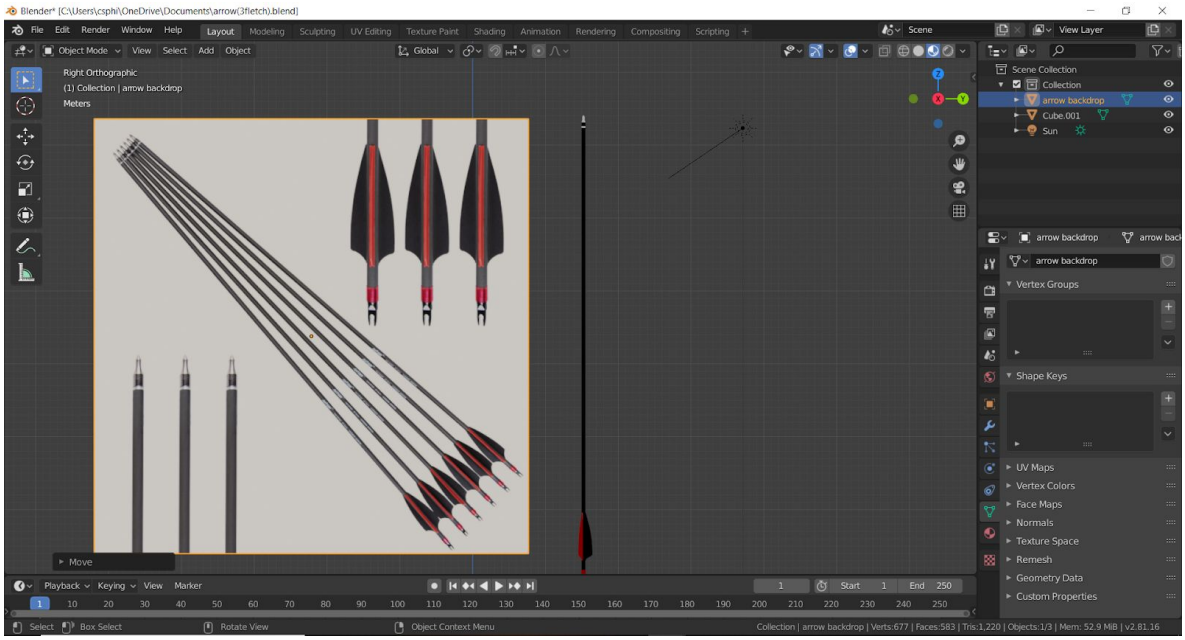
Creating the first test scene in Unity with distances and targets.



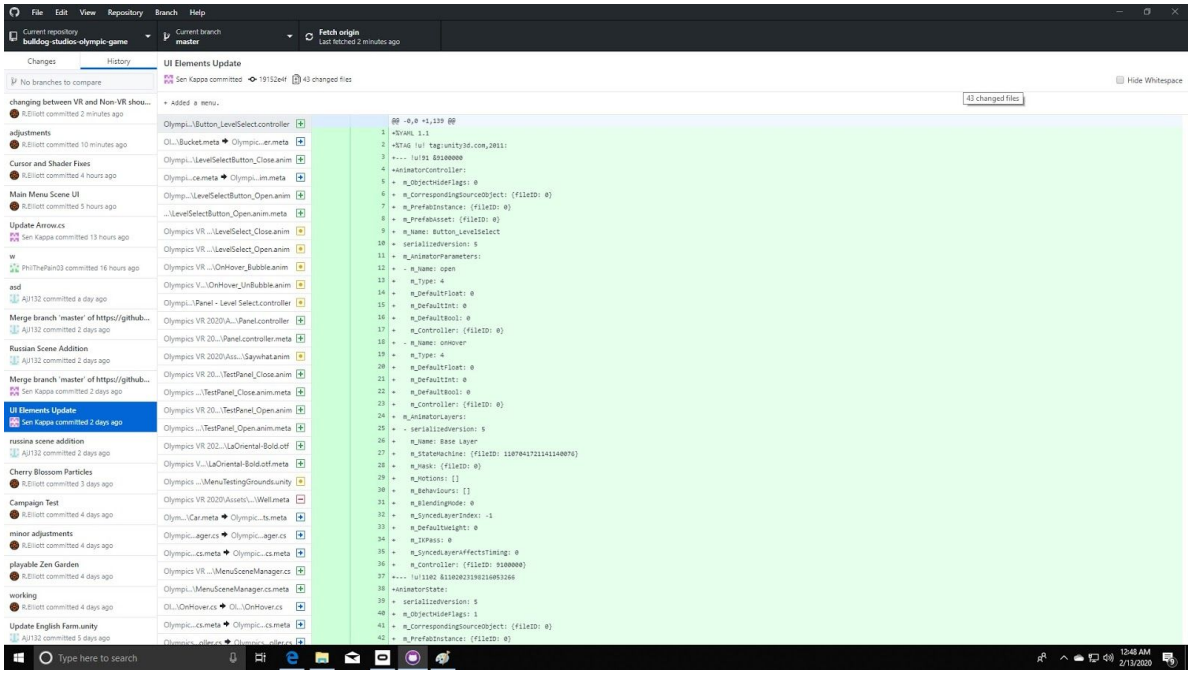
ArrowManger.cs is one of the C# scripts we created for the bow mechanic using Visual Studio and a lot of testing.



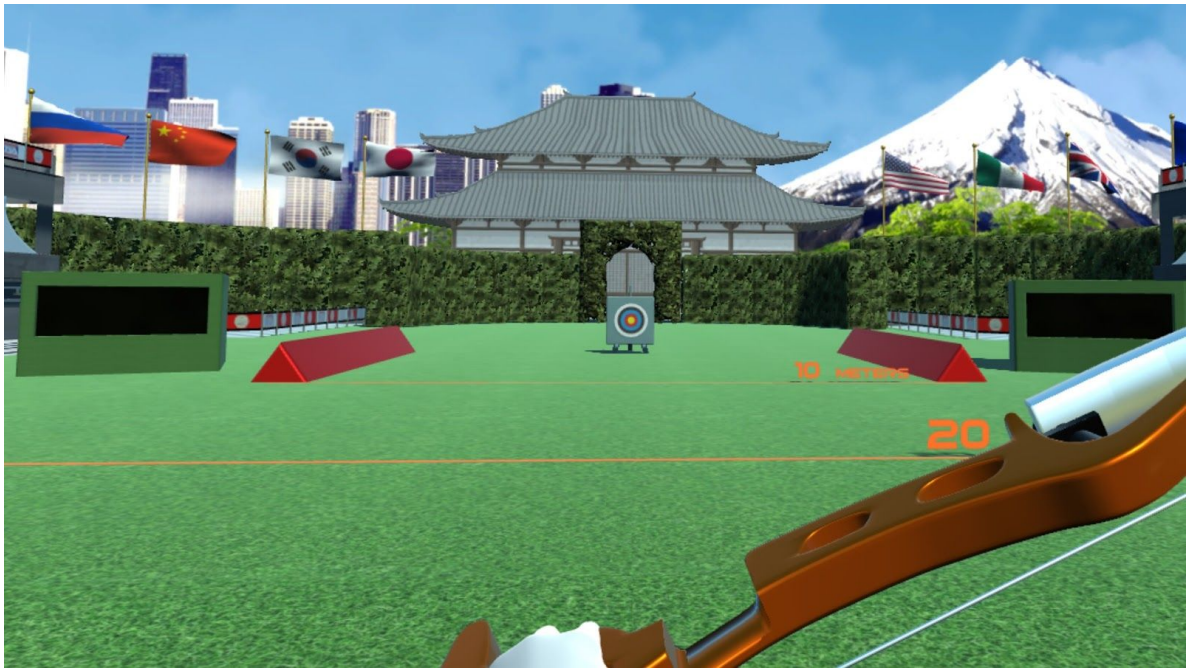
First testing scene used to test the bow mechanics in VR and non-VR.



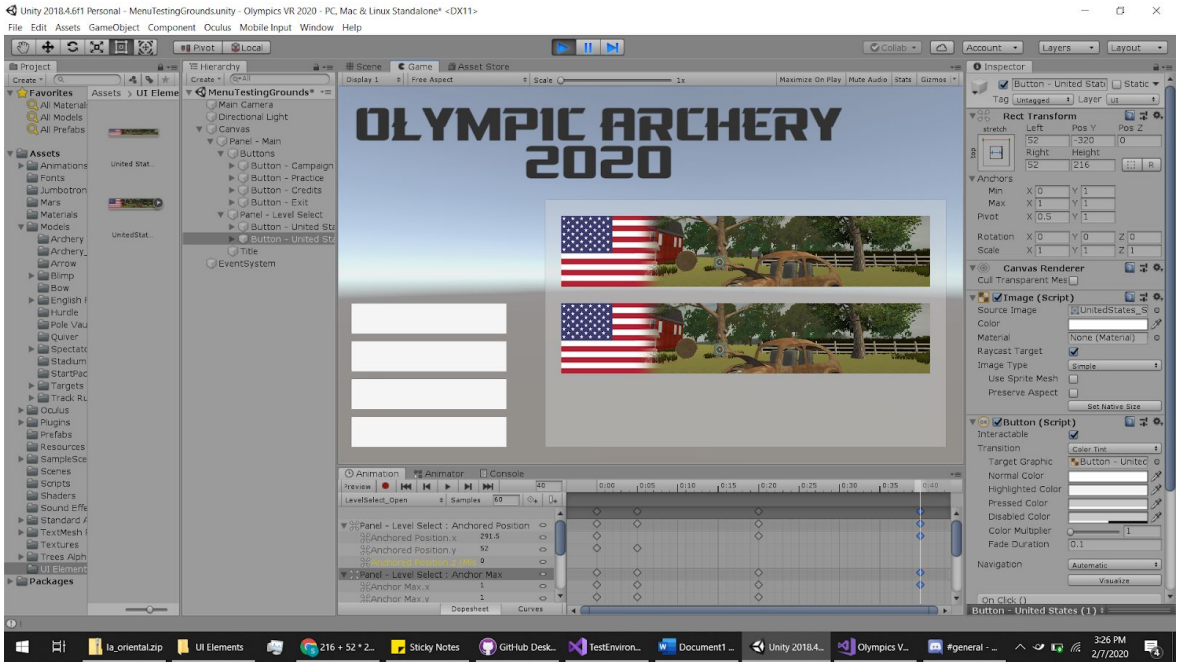
Modeling the arrow.



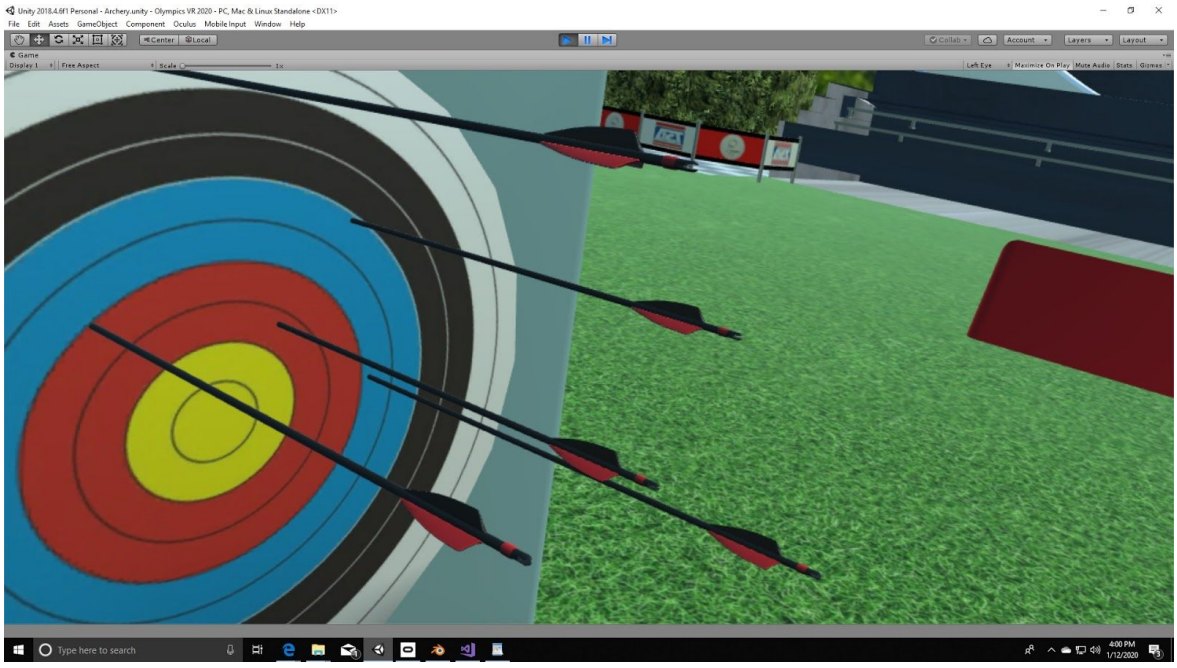
The team used a GitHub repository to store and update versions of the game while in development.



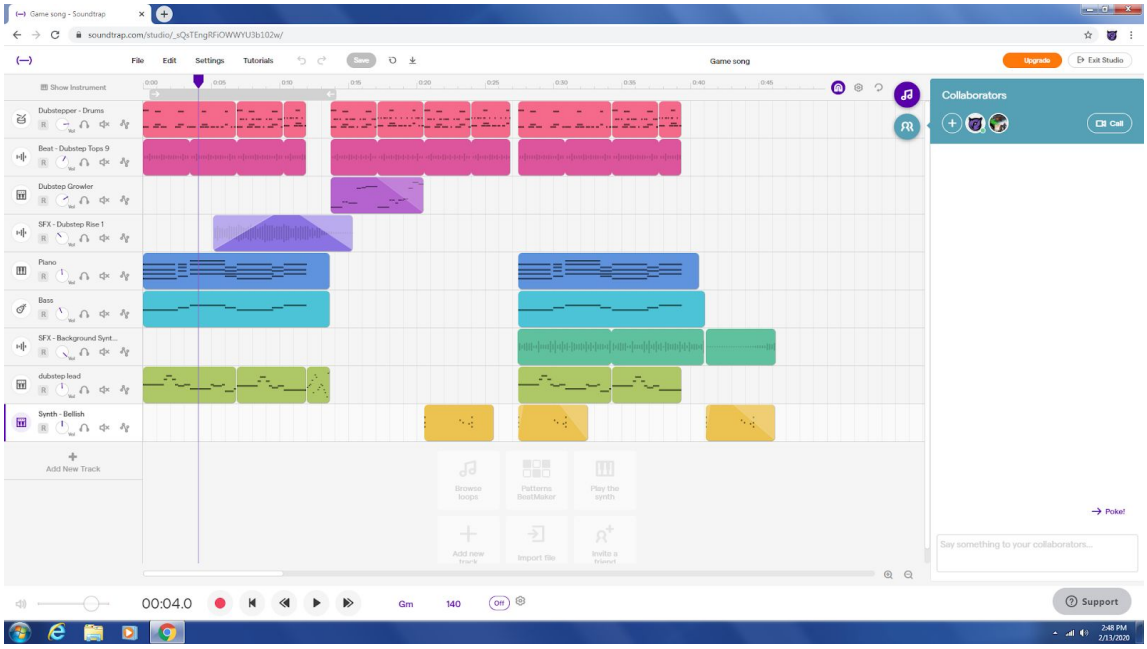
Testing out the VR bow in the archery arena.



Animating the main menu, using Unity legacy animation and C#. Work-in-progress title changed.

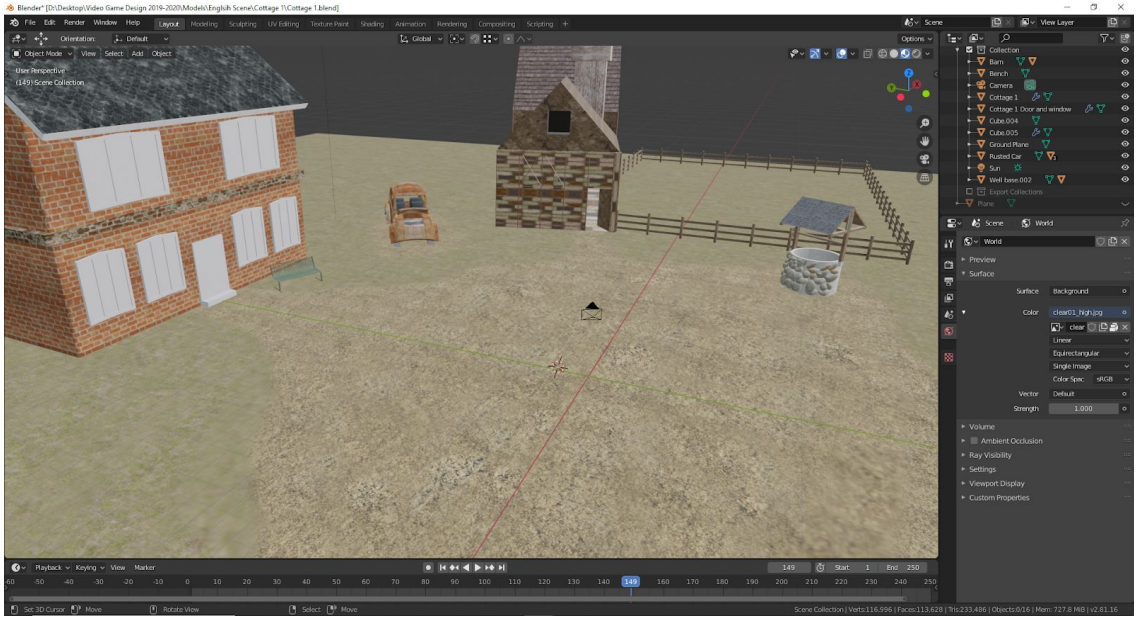


Arrows will sink into the target or any other objects in the scene at a random depth.

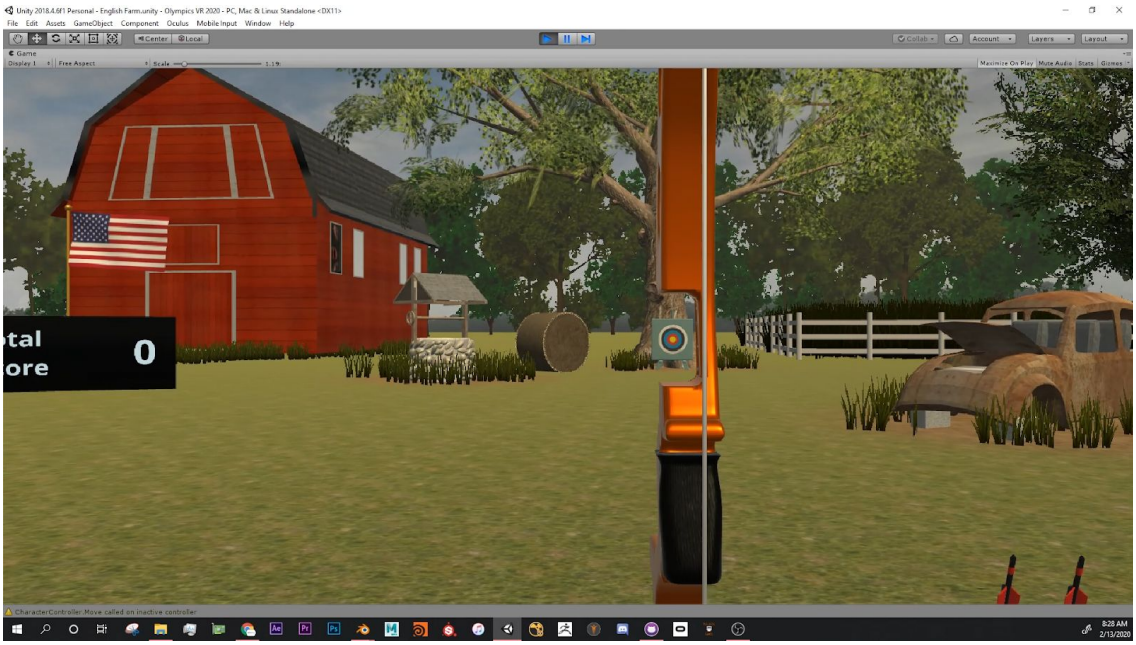


Making the main theme music in *Soundtrap*, a browser based “DAW” (Digital Audio Workstation).

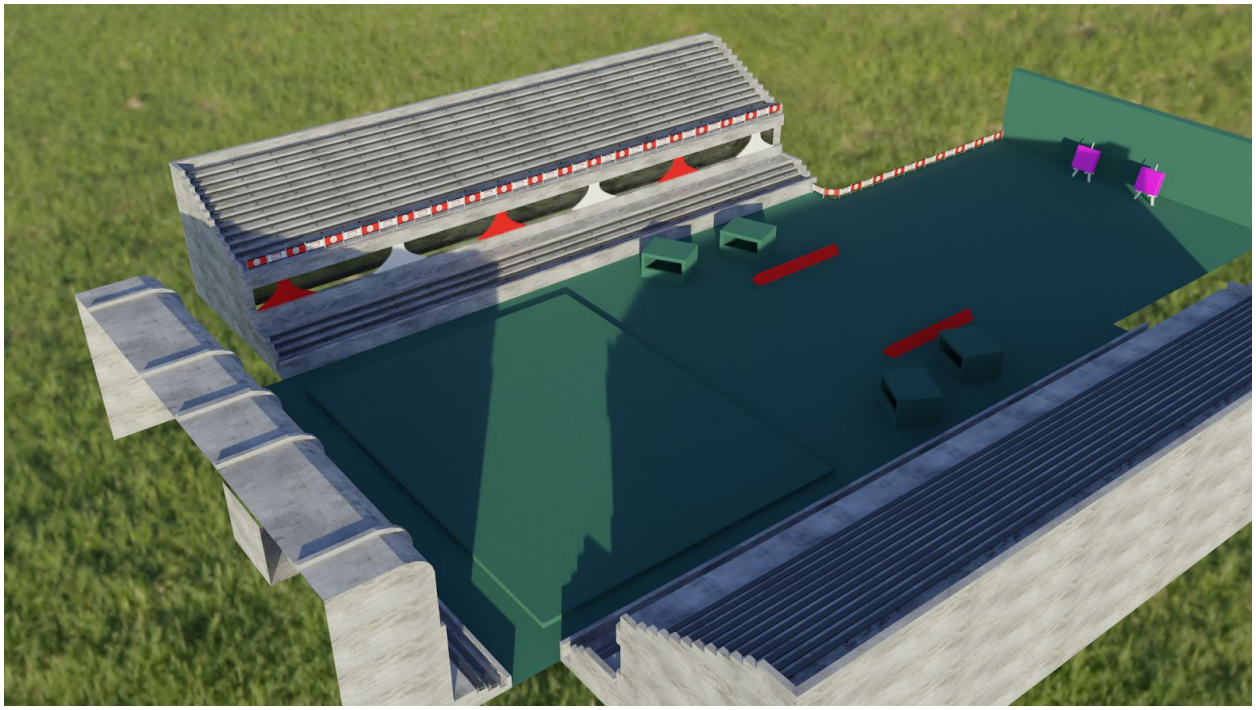
Before the scoreboards were completed, we used quads to display the score.



The original ‘english cottage scene’ in blender before it was exported to Unity

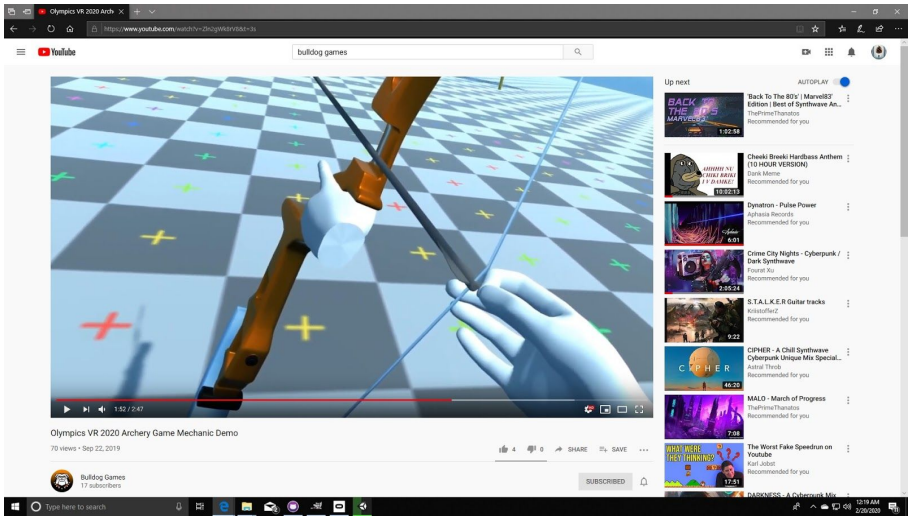


Modeling the farm scene in blender and moving it into the Unity game engine. This process became our workflow and will allow us to easily continue development on additional scenes and locations in the future.

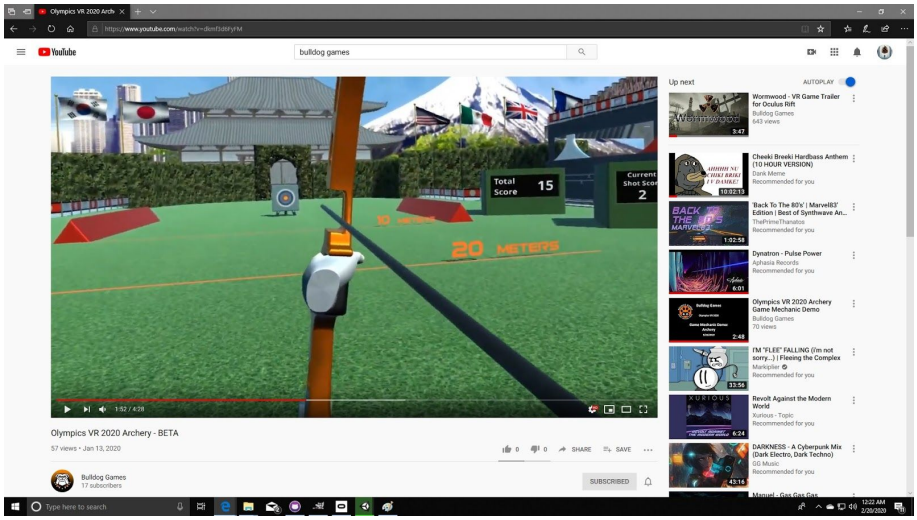


A rendered version of the Stadium in Blender

Development Videos on YouTube



https://youtu.be/Zln2gWk8rV8



https://youtu.be/dkmf3d6FyFM

Screen Captures of Final Product


USA
Total Score
0
Current
Shot Score
0


Russia
Total Score
0
Current
Shot Score
0


Japan


High Score 150
Total Score 0
Tokyo Games
2020
20 METERS

Hardware and Software Used

Hardware and Software Used

Unity

Wormwood was created on Windows PCs using the Unity game engine version 5.6.0f3. The personal edition of Unity is available for free at www.unity3d.com.

Microsoft Visual Studio and MonoDevelop

All code was written in C# using Microsoft Visual Studio and MonoDevelop, the integrated development environment supplied with Unity.

Blender

The 3d models were created in Blender, an open source 3d modeling program available for free at www.blender.org.

Audacity

Sound effects were edited in Audacity, an open source audio editing program available for free at www.audacityteam.org.

MuseScore

Original music score and sound effects were created in MuseScore, available for free at www.musescore.org.

Photoshop

Logos and object textures were created in Photoshop 2017(provided by the school system).

adobe.com/products/photoshop.html

Github Desktop Client

Unity files were stored on a repository server made available to the whole team.

https://desktop.github.com/

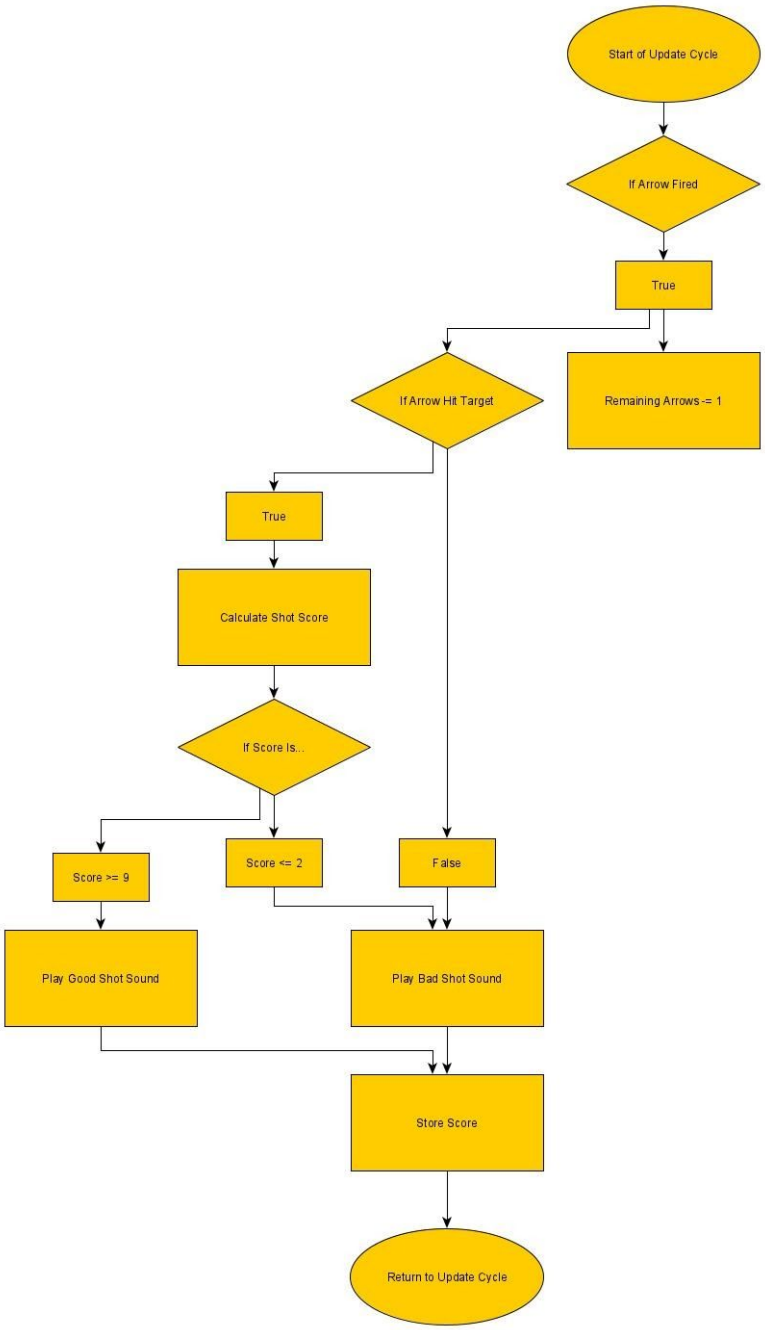
Discord

Discord was the main communication service used by the team.

https://discordapp.com/download

Code Examples and Flowcharts

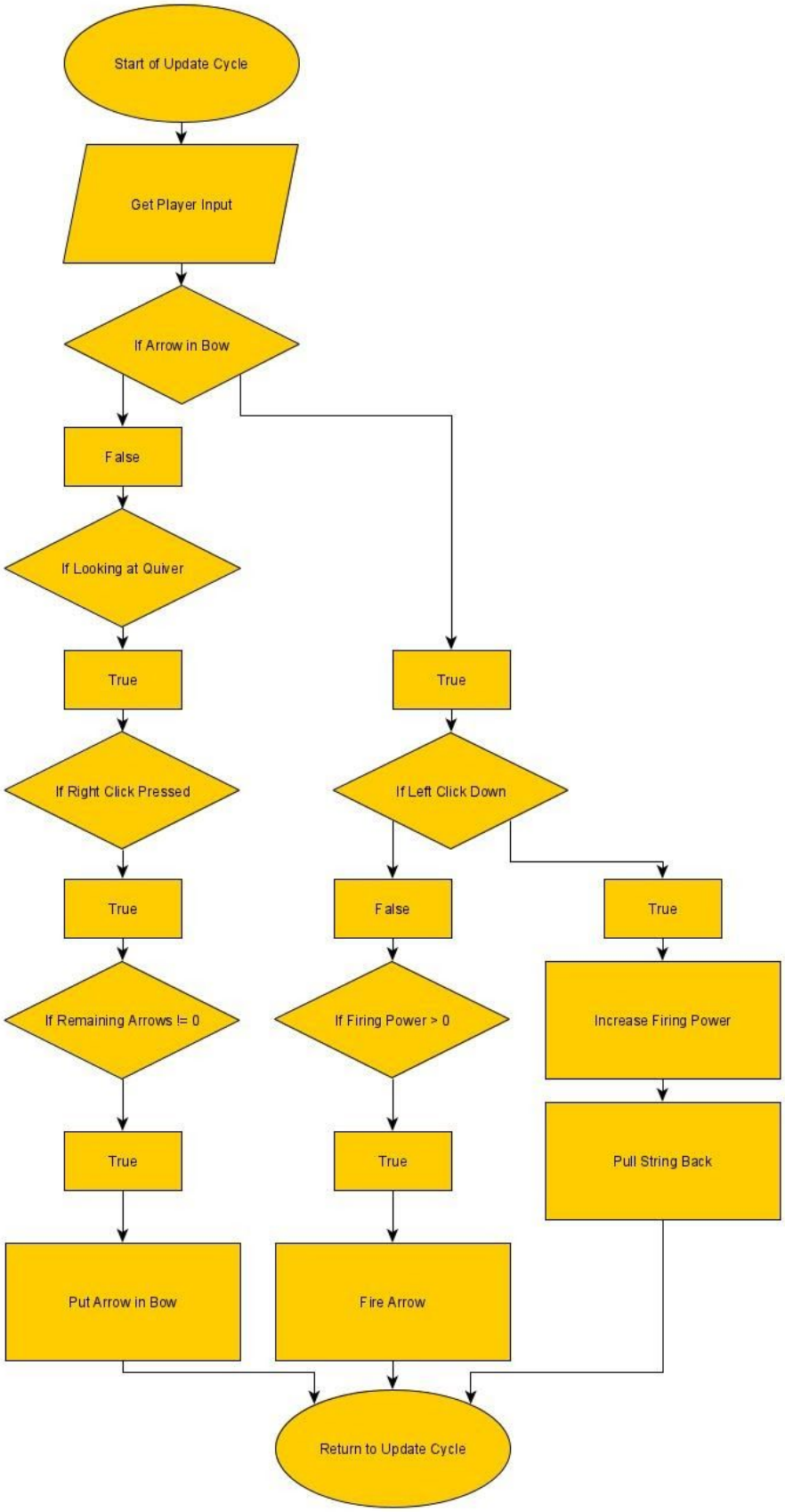
Arrow Code:



```
using UnityEngine;
using System.Collections;
public class Arrow : MonoBehaviour
{
    public AudioSource arrowImpact;
    public Rigidbody ArrowRB;
    public float depthOfImpact = 0.10f;
    public Vector3 collidercenter;
    private BoxCollider col;
    private bool isAttached = false;
    private bool isFired = false;
    public int currentShotScore;
    private GameObject PlayerArcher;
    private float randomDepth;
    void Start()
    {
        ArrowRB = GetComponent<Rigidbody>();
        col = GetComponent<BoxCollider>();
        collidercenter = col.center;
        PlayerArcher = GameObject.FindWithTag("Player");
    }
    void OnTriggerStay(Collider other)
    {
        if (other.gameObject.tag == "ArrowStart")
        {
            AttachArrow();
        }
    }
    void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.tag == "ArrowStart")
        {
            AttachArrow();
        }
    }
    void Update()
    {
        // 5f represents normal arrow flight conditions
        if (isFired && transform.GetComponent<Rigidbody>().velocity.magnitude > 5f)
        {
            transform.LookAt(transform.position +
transform.GetComponent<Rigidbody>().velocity);
        }
        // destroy all fired arrows in the scene and clear score
        if ((Input.GetButtonDown("RetrieveArrows") ||
OVRInput.GetDown(OVRInput.Button.PrimaryThumbstick) || Input.GetKeyDown(KeyCode.R)) &&
isFired)
        {
            Destroy(gameObject);
            //GameVariables.TargetScore = 0;
        }
    }
    public void Fired()
    {
        isFired = true;
        randomDepth = Random.Range(12.0f, 25.0f);
        col.center = new Vector3(0, 0, randomDepth);
        col.size = new Vector3(0.35f, 0.35f, 10.86177f);
        //GameVariables.CurrentShotScore = 0;
    }
    private void AttachArrow()
    {
        //var device = SteamVR_Controller.Input((int)ArrowManager.Instance.trackedObj.index);
        if (!isAttached && OVRInput.Get(OVRInput.RawButton.RIndexTrigger))
        {
            ArrowManager.Instance.AttachBowToArrow();
```

```
            isAttached = true;
        }
    }
    void OnCollisionEnter(Collision collision)
    {
        if (collision.collider.tag == "ArrowHit" && isFired == true)
        {
            ArrowRB.velocity = new Vector3(0, 0, 0);
            ArrowRB.isKinematic = true;
            arrowImpact.Play();
            // moves arrow a few inches inside the target
            transform.Translate(depthOfImpact * Vector3.forward);
            transform.Rotate(0, 0, 0, Space.World);
            //See CalculateScore method for more details
            float temp = CalculateScore (collision);
            Debug.Log(temp.ToString());
            GameVariables.TargetScore += 10 - Mathf.FloorToInt(temp);
            // calculates current shot score
            GameVariables.CurrentShotScore = 10 - Mathf.FloorToInt(temp);
            PlayerArcher.GetComponent<ArcheryComp>().CrowdReaction();
            //PlayerArcher.GetComponent<ArcheryComp>().BuzzerEnd();
        }
        if (isFired == true)
        {
            PlayerArcher.GetComponent<ArcheryComp>().BuzzerEnd();
        }
    }
    private float CalculateScore(Collision collision)
    {
        // The Square Root of ((Target.Position.X - CollisionPoint.X) ^ 2 + (Target.Position.Y
- CollisionPoint.Y) ^ 2)
        // Multiplied by 2 and by 10 in order to use radius instead of diameter and scale the
points to base ten, respectively.
        // Finally, divided by the local scale of the Target in order to make it dynamic even
when changing its scale.
        Debug.Log(collision.transform.position.x + ", " + collision.transform.position.y);
        Debug.Log(collision.GetContact(0).point.x + "," + collision.GetContact(0).point.y);
        Debug.Log(collision.transform.localScale.x);
        return Mathf.Sqrt(Mathf.Pow(collision.transform.position.x -
collision.GetContact(0).point.x, 2) + Mathf.Pow(collision.transform.position.y -
collision.GetContact(0).point.y, 2)) * 20 / collision.transform.lossyScale.x;
    }
}
```

Bow Controls:

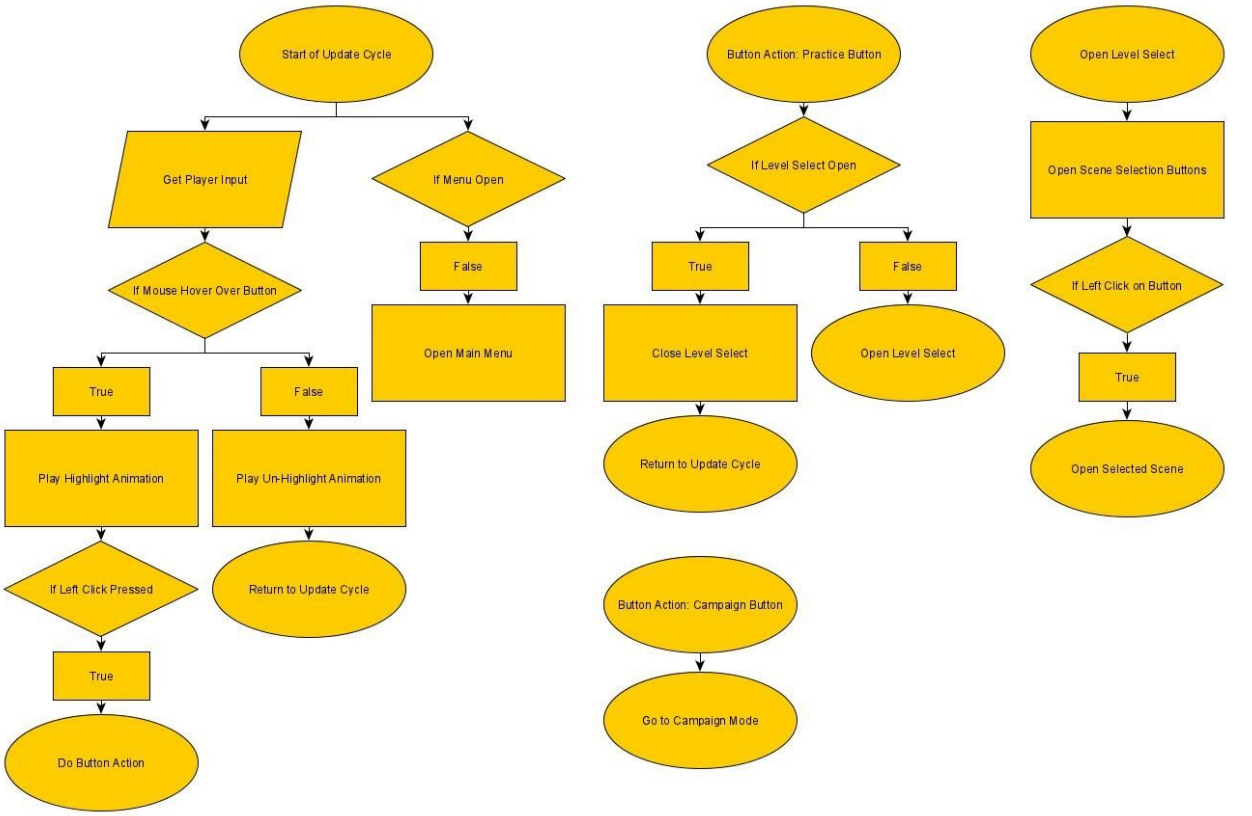


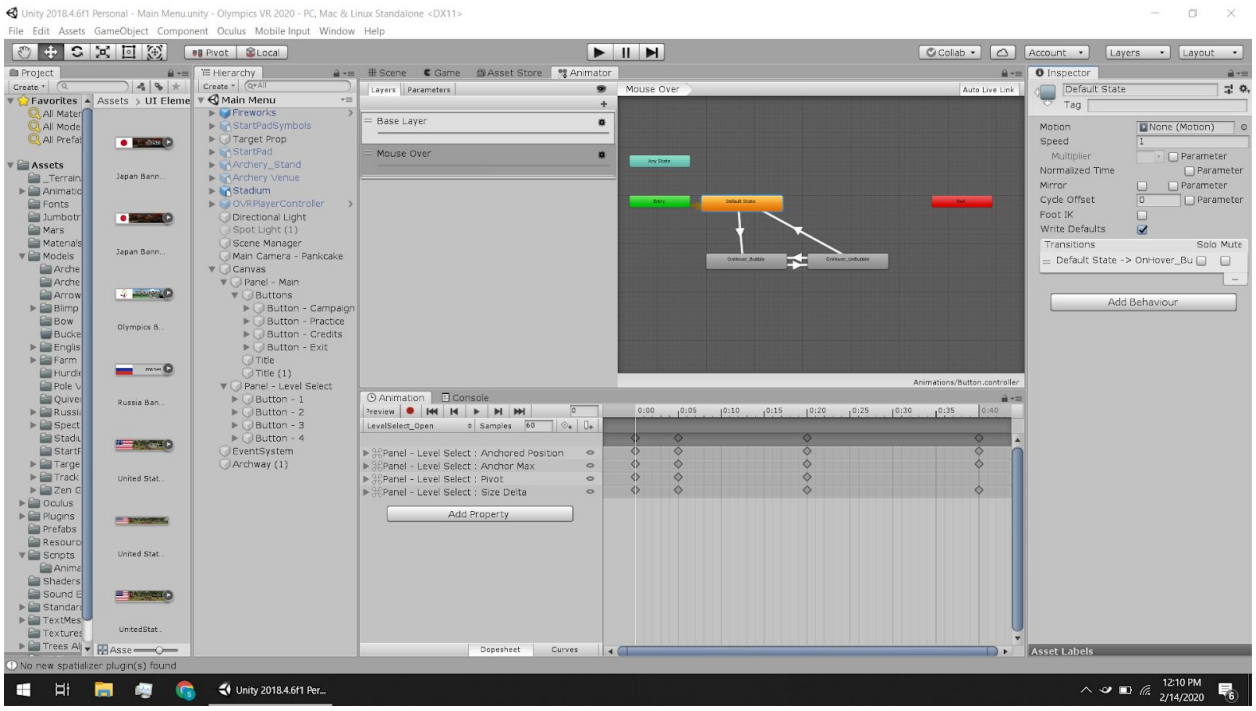
```
using UnityEngine;
using System.Collections;
public class ArrowManager2 : MonoBehaviour
{
    public static ArrowManager2 Instance;
    public GameObject testCamera;
    private GameObject currentArrow;
    public GameObject stringAttachPoint;
    public GameObject arrowStartPoint;
    public GameObject stringStartPoint;
    public GameObject arrowPrefab;
    public AudioSource arrowFired;
    public bool isAttached = false;
    public float Strength;
    public float CurrentStrength;
    public float MaxStrength;
    public Transform camTransform;
    public float shakeAmount = 0.7f;
    Private Vector3 originalPos;
    void Awake()
    {
        if (Instance == null)
            Instance = this;
        camTransform = transform;
    }
    void OnDestroy()
    {
        if (Instance == this)
            Instance = null;
    }
    void OnEnable()
    {
        originalPos = camTransform.localPosition;
    }
    void Update()
    {
        int layerMask = 1 << 10;
        RaycastHit hit;
        if (Physics.Raycast(transform.position, transform.TransformDirection(Vector3.right *
-1), out hit, Mathf.Infinity, layerMask) && hit.collider.tag == "Quiver")
        {
            Debug.DrawRay(transform.position, transform.TransformDirection(Vector3.right * -1)
* hit.distance, Color.yellow);
            Debug.Log("Did Hit");

            GameVariables.LookingAtQuiver = true;
        }
        else
        {
            GameVariables.LookingAtQuiver = false;
        }
        AttachArrow();
        PullString();
    }
    private void PullString()
    {
        if (isAttached == true)
        {
            if (Input.GetMouseButton(0))
            {
                if (CurrentStrength + Time.deltaTime < MaxStrength)
                {
                    CurrentStrength += Strength * Time.deltaTime;
                }
                else
                {
```

```
                    CurrentStrength = MaxStrength;
                }
                Debug.Log(MaxStrength - CurrentStrength);
                if (MaxStrength - CurrentStrength < 0)
                {
                    stringAttachPoint.transform.localPosition -= new Vector3(0, 0, 0);
                }
                else
                {
                    stringAttachPoint.transform.localPosition -= new
Vector3(-Mathf.Sqrt(MaxStrength - CurrentStrength) / 5000, 0, 0);
                }
                testCamera.GetComponent<Camera>().fieldOfView = 60 - ((CurrentStrength /
MaxStrength) * 20);
                camTransform.localPosition = originalPos + Random.insideUnitSphere *
shakeAmount;
            }
            else if (CurrentStrength != 0)
            {
                Fire(CurrentStrength);
                CurrentStrength = 0;
                stringAttachPoint.transform.localPosition =
stringStartPoint.transform.localPosition;
                testCamera.GetComponent<Camera>().fieldOfView = 60;
                camTransform.localPosition = originalPos;
            }
        }
    }
    private void Fire(float strength)
    {
        currentArrow.transform.parent = null;
        currentArrow.GetComponent<Arrow>().Fired();
        currentArrow.transform.GetChild(0).gameObject.SetActive(true);
        Rigidbody r = currentArrow.GetComponent<Rigidbody>();
        r.velocity = currentArrow.transform.forward * strength;
        r.useGravity = true;
        currentArrow.GetComponent<Collider>().isTrigger = false;
        stringAttachPoint.transform.position = stringStartPoint.transform.position;
        currentArrow = null;
        isAttached = false;
        arrowFired.Play();
        GameVariables.RemainingArrows -= 1;
        GameVariables.hasPlayedCrowdCheer = false;
        GameVariables.hasPlayedCrowdAhhh = false;
        GameVariables.ArrowInHand = false;
    }
    private void AttachArrow()
    {
        if (currentArrow == null && Input.GetMouseButton(1) && GameVariables.RemainingArrows >
0 && !isAttached && GameVariables.LookingAtQuiver)
        {
            currentArrow = Instantiate(arrowPrefab);
            currentArrow.SetActive(true);
            currentArrow.transform.localPosition = new Vector3(0f, 0f, .47f);
            currentArrow.transform.localRotation = Quaternion.identity;
            AttachBowToArrow();
            GameVariables.ArrowInHand = true;
        }
    }
    public void AttachBowToArrow()
    {
        currentArrow.transform.parent = stringAttachPoint.transform;
        currentArrow.transform.position = arrowStartPoint.transform.position;
        currentArrow.transform.rotation = arrowStartPoint.transform.rotation;
        isAttached = true;
    }
```

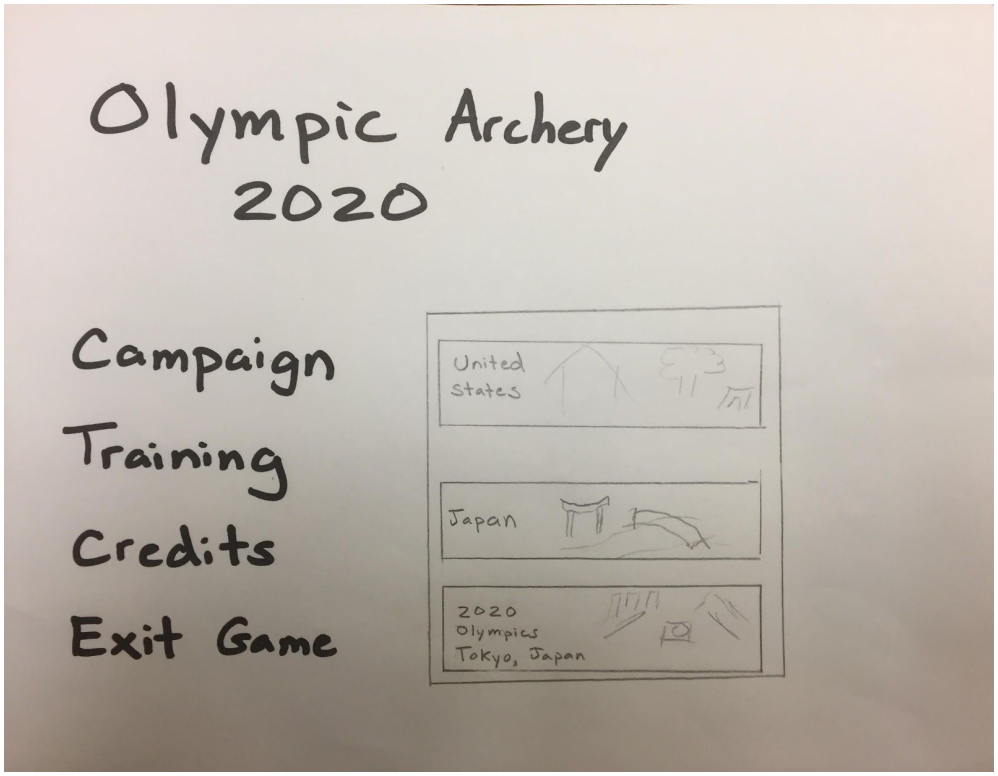
}

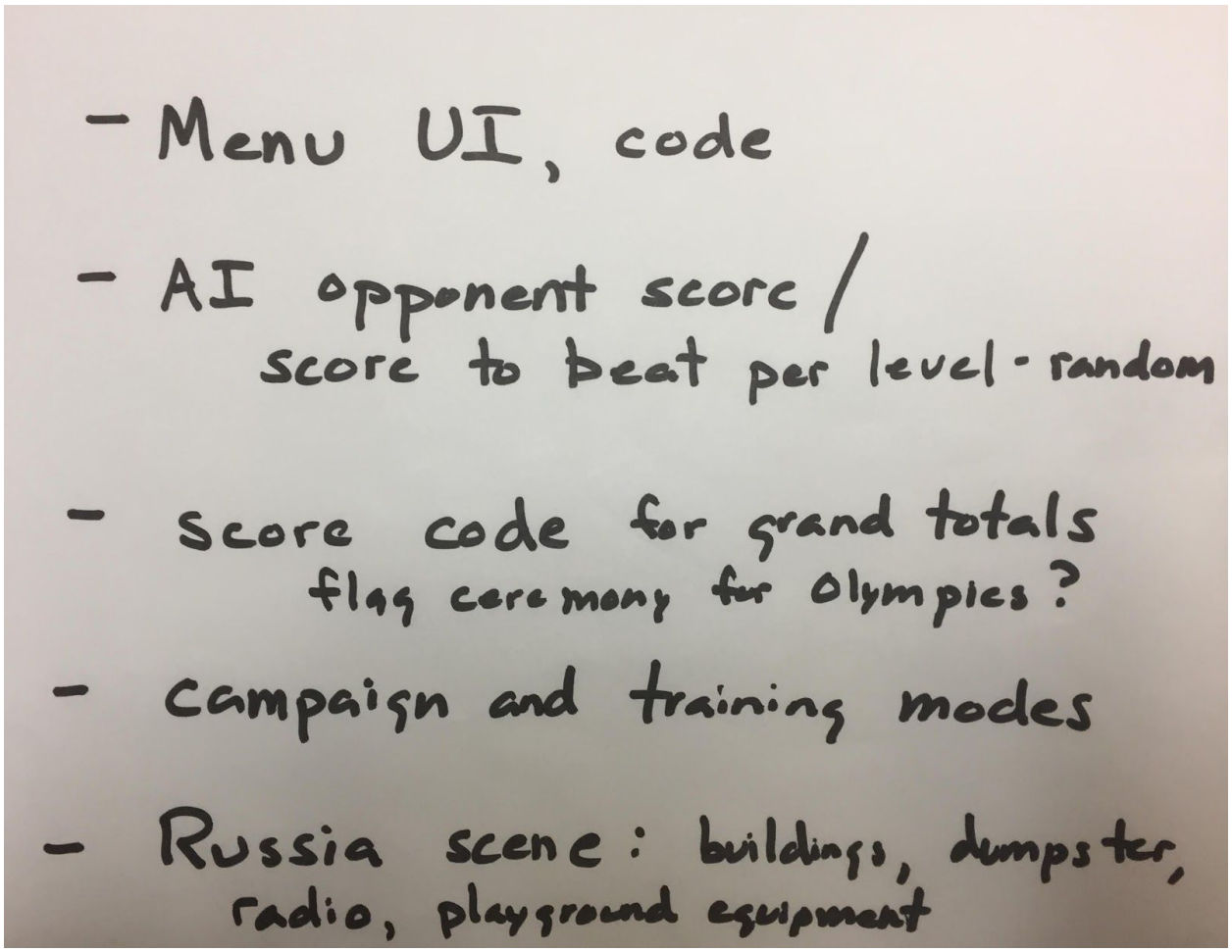
Main Menu:


Start of Update Cycle
Get Player Input
If Menu Open
If Mouse Hover Over Button
False
Open Main Menu
True
False
Play Highlight Animation
Play Un-Highlight Animation
If Left Click Pressed
Return to Update Cycle
True
Do Button Action
Button Action: Practice Button
If Level Select Open
True
False
Close Level Select
Open Level Select
Return to Update Cycle
Button Action: Campaign Button
Go to Campaign Mode
Open Level Select
Open Scene Selection Buttons
If Left Click on Button
True
Open Selected Scene


Unity 2018.4.6f1 Personal - Main Menu.unity - Olympics VR 2020 - PC, Mac & Linux Standalone <DX11>
File Edit Assets GameObject Component Oculus Mobile Input Window Help
Hierarchy
Main Menu
Fireworks
StartPadSymbols
Target Prop
StartPad
Archery_Stand
Archery Venue
Stadium
OVRPlayerController
Directional Light
Spot Light (1)
Scene Manager
Main Camera - Pankcake
Canvas
Panel - Main
Buttons
Button - Campaign
Button - Practice
Button - Credits
Button - Exit
Title
Title (1)
Panel - Level Select
Button - 1
Button - 2
Button - 3
Button - 4
EventSystem
Archway (1)
Inspector
Default State
Add Behaviour
Add Property
No new spatializer plugin(s) found
12:10 PM
2/14/2020

Storyboard and Concept Art




- Menu UI, code
- AI opponent score /
score to beat per level - random
- score code for grand totals
flag ceremony for Olympics?
- campaign and training modes
- Russia scene: buildings, dumpster,
radio, playground equipment

Items not created by the team

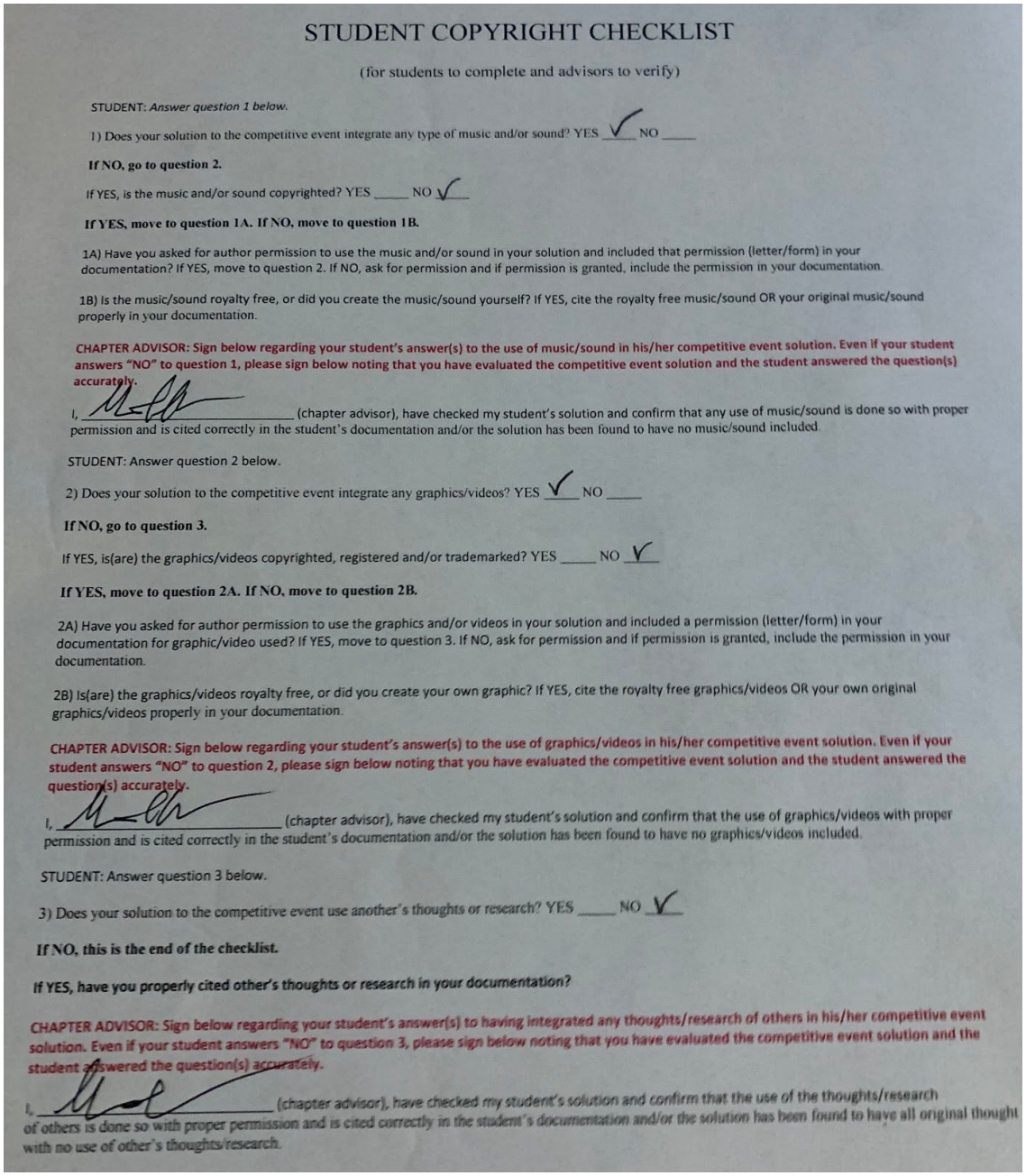
Unity Game Engine Personal Edition 2018.4.6f1 using the following Standard Unity Assets:

- Skyboxes – Overcast
- ground textures
- water textures and shader

Provided by Oculus Unity Integration SDK:

- Oculus Integration

Student Copyright Checklist



STUDENT COPYRIGHT CHECKLIST

(for students to complete and advisors to verify)

STUDENT: *Answer question 1 below.*

1) Does your solution to the competitive event integrate any type of music and/or sound? YES ✓ NO ____

If NO, go to question 2.

If YES, is the music and/or sound copyrighted? YES ____ NO ✓

If YES, move to question 1A. If NO, move to question 1B.

1A) Have you asked for author permission to use the music and/or sound in your solution and included that permission (letter/form) in your documentation? If YES, move to question 2. If NO, ask for permission and if permission is granted, include the permission in your documentation.

1B) Is the music/sound royalty free, or did you create the music/sound yourself? If YES, cite the royalty free music/sound OR your original music/sound properly in your documentation.

CHAPTER ADVISOR: Sign below regarding your student's answer(s) to the use of music/sound in his/her competitive event solution. Even if your student answers "NO" to question 1, please sign below noting that you have evaluated the competitive event solution and the student answered the question(s) accurately.

I, _____ (chapter advisor), have checked my student's solution and confirm that any use of music/sound is done so with proper permission and is cited correctly in the student's documentation and/or the solution has been found to have no music/sound included.

STUDENT: Answer question 2 below.

2) Does your solution to the competitive event integrate any graphics/videos? YES ✓ NO ____

If NO, go to question 3.

If YES, is(are) the graphics/videos copyrighted, registered and/or trademarked? YES ____ NO ✓

If YES, move to question 2A. If NO, move to question 2B.

2A) Have you asked for author permission to use the graphics and/or videos in your solution and included a permission (letter/form) in your documentation for graphic/video used? If YES, move to question 3. If NO, ask for permission and if permission is granted, include the permission in your documentation.

2B) Is(are) the graphics/videos royalty free, or did you create your own graphic? If YES, cite the royalty free graphics/videos OR your own original graphics/videos properly in your documentation.

CHAPTER ADVISOR: Sign below regarding your student's answer(s) to the use of graphics/videos in his/her competitive event solution. Even if your student answers "NO" to question 2, please sign below noting that you have evaluated the competitive event solution and the student answered the question(s) accurately.

I, _____ (chapter advisor), have checked my student's solution and confirm that the use of graphics/videos with proper permission and is cited correctly in the student's documentation and/or the solution has been found to have no graphics/videos included.

STUDENT: Answer question 3 below.

3) Does your solution to the competitive event use another's thoughts or research? YES ____ NO ✓

If NO, this is the end of the checklist.

If YES, have you properly cited other's thoughts or research in your documentation?

CHAPTER ADVISOR: Sign below regarding your student's answer(s) to having integrated any thoughts/research of others in his/her competitive event solution. Even if your student answers "NO" to question 3, please sign below noting that you have evaluated the competitive event solution and the student answered the question(s) accurately.

I, _____ (chapter advisor), have checked my student's solution and confirm that the use of the thoughts/research of others is done so with proper permission and is cited correctly in the student's documentation and/or the solution has been found to have all original thought with no use of other's thoughts/research.

Plan of Work Log

Video Game Design Worksheet

Date	AJ	CJ	Erik	Chandler	Carter	Sean	Events	day of the week	Calender Mark Off	AJ Notes	CJ Notes	Erik Notes	Chandler Notes	Sean Notes	Carter Notes
9/2/2019							Got Theme for game	Monday	x						
9/3/2019	0.5	0.5	0.5	0.5	0.5	0.5	Brainstorming	Tuesday	x						
9/4/2019	0.5	0.5	1	1			Brainstorming	Wednesday	x						
9/5/2019	0.5	0.5	1	0.5			Brainstorming	Thursday	x						
9/6/2019	0.5			0.5			Brainstorming	Friday	x						
9/7/2019				0.5	0.2		Brainstorming	Saturday	x						
9/8/2019				0.5	0.3		Brainstorming	Sunday	x						
9/9/2019		0.5					Brainstorming	Monday	x						
9/10/2019	0.5	0.5					Brainstorming	Tuesday	x						
9/11/2019	0.5	0.5			0.3		Brainstorming	Wednesday	x						
9/12/2019	0.5	0.5					Brainstorming	Thursday	x						
9/13/2019	0.5				0.4		Brainstorming	Friday	x						
9/14/2019							Brainstorming	Saturday	x						
9/15/2019	3						Brainstorming	Sunday	x	Bow creation					
9/16/2019	3	1	1				Brainstorming	Monday	x	Bow creation					
9/17/2019	2	1					Brainstorming	Tuesday	x	Bow creation					
9/18/2019	2	0.5					Brainstorming	Wednesday	x	Bow creation and rig					
9/19/2019	2.5		1		0.2		Brainstorming	Thursday	x	Bow Rigging					
9/20/2019	2.5				0.3		Conclusion: Archery, Javelin, and speed climbing	Friday	x	Bow Rigging					
9/21/2019	3							Saturday	x	Bow Rigging					
9/22/2019	1		2		0.2	1	Creating Models for testing	Sunday	x	Fix rig multiple times	Target Creation				
9/23/2019	1	1				1		Monday	x	Fix rig multiple times	Target Creation			Research for arrow hit detection begun	
9/24/2019	1	2				2		Tuesday	x	Fixing Bow to work in unity	Target Creation				
9/25/2019	2	1				1.5		Wednesday	x		Target Creation				
9/26/2019	1.5	1			0.5	2		Thursday	x						
9/27/2019	0.5	0.5			0.4	1		Friday	x						
9/28/2019	1		3					Saturday	x						
9/29/2019								Sunday	x			on vacation			
9/30/2019	1	0.5			0.2		Fall Break	Monday	x	Created Tokyo logo-package		on vacation			
10/1/2019	1	0.5			0.3		Fall Break	Tuesday	x			on vacation			
10/2/2019	2	0.5			0.3	1	Fall Break	Wednesday	x	Created Archery Venue		on vacation		Script for arrow hit detection completed.	
10/3/2019	2					1	Fall Break-crowd cheer script	Thursday	x	Created Archery Venue		on vacation	Created Crowd PNG		
10/4/2019	2						Fall Break	Friday	x	Textured Archery Venue					
10/5/2019								Saturday	x						
10/6/2019				1				Sunday	x						
10/7/2019	1		2	1				Monday	x	Fixed Normals on Archery venue					
10/8/2019	0.5	0.5	2	1				Tuesday	x						
10/9/2019				1				Wednesday	x						
10/10/2019				1				Thursday	x						
10/11/2019			2					Friday	x						
10/12/2019								Saturday	x						
10/13/2019					0.2			Sunday	x						
10/14/2019	2	0.5				2		Monday	x	Created tokyo stadium					
10/15/2019	2	0.5	1		0.4	3	Archery Mechanic Complete	Tuesday	x	Texturing tokyo stadium				Began working on Non-VR Bow	
10/16/2019	1				0.4	2		Wednesday	x	Texturing tokyo stadium					
10/17/2019		0.5			0.4	3		Thursday	x						
10/18/2019					0.2	2		Friday	x						
10/19/2019	1							Saturday	x	fixed normals on stadium					
10/20/2019	2					1		Sunday	x	re-textured track					
10/21/2019	2	1				1		Monday	x	created hurdles for track					
10/22/2019	1	0.5			0.4	2		Tuesday	x	Texturing Hurdles					
10/23/2019		0.5			0.4	2		Wednesday	x						
10/24/2019		0.5			0.4	2		Thursday	x						
10/25/2019		0.5			0.4	1		Friday	x						
10/26/2019								Saturday	x						
10/27/2019								Sunday	x						
10/28/2019					0.2	1		Monday	x						
10/29/2019		1			0.4	2		Tuesday	x						
10/30/2019		0.5	2		0.4	2		Wednesday	x						
10/31/2019					0.4	1	halloween	Thursday	x					Raw Mechanics Completed.	
11/1/2019	3	1			0.4	1	Established Repo Server	Friday	x	Created and shared Github repository server					
11/2/2019	2							Saturday	x						
11/3/2019	2					1		Sunday	x	Created Git Ignore file					
11/4/2019	1	1				1		Monday	x	for repository server				Began Calibration of Fer Bow Strength	
11/5/2019					0.4	2		Tuesday	x						
11/6/2019					0.4	2		Wednesday	x						
11/7/2019	2		3		0.4	1		Thursday	x	Down Res UV for some models					
11/8/2019	2				0.4	1		Friday	x						
11/9/2019								Saturday	x						
11/10/2019					0.4	1		Sunday	x						
11/11/2019		1				2		Monday	x					Bug-fixing initial design	
11/12/2019		1			0.4	2		Tuesday	x						
11/13/2019		0.5		1	0.6	2		Wednesday	x						
11/14/2019		0.5	3	1	0.4	1		Thursday	x						
11/15/2019				1	0.4	1	Narrowed decision to archery	Friday	x						
11/16/2019				1	0.6			Saturday	x						
11/17/2019				1		1		Sunday	x						
11/18/2019	3	1		1		2		Monday	x	Building Archery stand				Research for Bow Pull-Back	
11/19/2019	2	1		1	0.5	1		Tuesday	x	Building Archery stand					
11/20/2019	1	0.5		0.5	0.8	2		Wednesday	x	Building Archery stand					
11/21/2019	0.5	0.5	3	0.5	0.5	1		Thursday	x	Building Archery stand					
11/22/2019	3	0.5		0.5	0.7	2		Friday	x	Texturing Archery Stand				Basic formula for calculating strength completed	
11/23/2019	2							Saturday	x	Detailing Archery Stand					
11/24/2019								Sunday	x						
11/25/2019		1		2			Thanksgiving Break	Monday	x						
11/26/2019	2	1			0.5		Archery Venue Complete	Tuesday	x	Archery Venue Normals				Bug-fixing	
11/27/2019	2				0.8		Thanksgiving Break	Wednesday	x	Archery Venue Normals					
11/28/2019			3		0.5		Thanksgiving	Thursday	x						
11/29/2019		1			0.5		Thanksgiving Break	Friday	x						
11/30/2019								Saturday	x						
12/1/2019						1		Sunday	x					Transferring over VR Controls to Non-VR	
12/2/2019		0.5				4		Monday	x						
12/3/2019		0.5			0.5	3		Tuesday	x	Added Railing for Venue					
12/4/2019		0.5			0.5	2		Wednesday	x						
12/5/2019			3		0.5	3		Thursday	x						
12/6/2019					0.5	2		Friday	x						
12/7/2019								Saturday	x						
12/8/2019						1		Sunday	x						
12/9/2019		0.5			0.5	2		Monday	x						
12/10/2019						1	Midterms/Half Day	Tuesday	x						
12/11/2019							Midterms/Half Day	Wednesday	x						
12/12/2019	1	1	3				Christmas Break	Thursday	x	Captured Mocap data for pole vault animation					
12/13/2019	1				1.5		Christmas Break	Friday	x						
12/14/2019	2						Christmas Break	Saturday	x	Built Pole vault scene					
12/15/2019	2.5	1			0.4		Christmas Break	Sunday	x	Pole vault animation					
12/16/2019	2.5	1			0.3		Christmas Break	Monday	x	Pole vault animation					
12/17/2019	1						Christmas Break	Tuesday	x	Pole vault animation					
12/18/2019	0.5	0.5			0.5		Christmas Break	Wednesday	x	Pole vault animation					
12/19/2019	2						Christmas Break	Thursday	x	Pole vault animation					
12/20/2019	2				0.4		Christmas Break	Friday	x	Pole vault animation					
12/21/2019	3				0.2		Christmas Break	Saturday	x	Pole vault animation					
12/22/2019				2			Christmas Break	Sunday	x	decided not to use pole vault animation				Added Screen Shake To Bow	
12/23/2019		0.5		1	0.2	3	Christmas Break	Monday	x						
12/24/2019				1		2	Christmas Break	Tuesday	x						

Video Game Design Worksheet

Date	AJ	CJ	Erik	Chandler	Carter	Sean	Events	day of the week	Calender Mark Off	AJ Notes	CJ Notes	Erik Notes	Chandler Notes	Sean Notes	Carter Notes
12/25/2019						2	Christmas	Wednesday	x						
12/26/2019						2	Christmas Break	Thursday	x						
12/27/2019				2		2	Christmas Break	Friday	x					FOV Zooming when pulling back bow	
12/28/2019				2	0.2		Christmas Break	Saturday	x						
12/29/2019				1			Christmas Break	Sunday	x						
12/30/2019			1	2	1	3	Christmas Break	Monday	x						
12/31/2019			1				New Years Eve	Tuesday	x						
1/1/2020							New Years	Wednesday	x						
1/2/2020						1	Christmas Break	Thursday	x			vacation		FOV and Screen Shake	
1/3/2020						1	Christmas Break	Friday	x			vacation		Bug fixes	
1/4/2020				2	0.2		Christmas Break	Saturday	x			vacation			
1/5/2020				2			Christmas Break	Sunday	x			vacation			
1/6/2020		1					Christmas Break	Monday	x			vacation			
1/7/2020		0.5			0.3	2		Tuesday	x						
1/8/2020		0.5				2		Wednesday	x						
1/9/2020	1.5	0.5	4		0.4	4	First middle school session	Thursday	x					Tutoring of Middle School Began	
1/10/2020						2		Friday	x						
1/11/2020				2				Saturday	x						
1/12/2020				1	0.3			Sunday	x					Randomized depth of arrow in target	
1/13/2020		1		1		4	BETA Video Posted	Monday	x						
1/14/2020	1.5				0.4	4		Tuesday	x	created arrow holder				Began Non-VR Menu	
1/15/2020	1		2	1		2		Wednesday	x	textured arrow holder				UI Implementation	
1/16/2020		1	1			4		Thursday	x						
1/17/2020		1				2		Friday	x						
1/18/2020								Saturday	x	Fixed Normals in farm					
1/19/2020	0.5	1			0.2	1		Sunday	x	scene					
1/20/2020		0.5	2			4		Monday	x					Created animated panels for Menu	
1/21/2020		0.5	3			4		Tuesday	x						
1/22/2020				0		2		Wednesday	x						
1/23/2020		0.5				4		Thursday	x					Began button animation	
1/24/2020			1			2		Friday	x					Bug-fixing panels	
1/25/2020								Saturday	x						
1/26/2020						1		Sunday	x						
1/27/2020	1	1	2	2	0.2	4		Monday	x	Created Portfolio				Complete panel overhaul and research for new UI began.	
1/28/2020	1	1	3	2		4		Tuesday	x	Worked on portfolio					
1/29/2020	1	0.5		2		2		Wednesday	x	Worked on portfolio					
1/30/2020		0.5	1	2		4		Thursday	x						
1/31/2020	1			2	0.3	2		Friday	x	Uploaded images for portfolio					
2/1/2020				1				Saturday	x						
2/2/2020						1		Sunday	x					Implementation for for current UI of the game	
2/3/2020		0.5	2	2		4		Monday	x						
2/4/2020		1	2	2	0.1	4		Tuesday	x						
2/5/2020	3	1		2		2		Wednesday	x	Building Zen Garden					
2/6/2020	2	1	1	2		4		Thursday	x	Texturing Zen Garden					
2/7/2020	4			2		2		Friday	x	Building Russian scene					
2/8/2020	3							Saturday	x	Building apartments					
2/9/2020	3					2		Sunday	x	Texturing Russian Scene				Practice scene select panel created.	
2/10/2020	2			2		4		Monday	x	Setting up russian Scene					
2/11/2020	2	0.5	5	2	0.2	4		Tuesday	x	Created scroll for zen Ga				Level select banners added to game	
2/12/2020	1.5	0.5		2		2		Wednesday	x	Portfolio Work					
2/13/2020	1.5	0.5	5	2		4		Thursday	x	Portfolio Work					
2/14/2020	1.5	0.5		2		2		Friday	x	Portfolio Work					
2/15/2020	1.5	1		2	0.2			Saturday	x	Portfolio Work				VR Menu compatibility added	
2/16/2020	2					1		Sunday	x	Portfolio Work					
2/17/2020	1	1	2	1			Winter Break	Monday	x	Portfolio Work				Campaign Mode confirmation screen created.	
2/18/2020	1.5	1		2			Winter Break	Tuesday	x	Portfolio Work					
2/19/2020	2	1		1		3		Wednesday	x	Portfolio Work					
2/20/2020		2	4			4	Video Game Turned in	Thursday	x						
Total	135.5	61.5	78.5	75	28.1	198									