Sprint 1 Retrospective

Team 8 - Ball of the Wild

Andrew Thomae, Wes Turnbull, Henry Wellman, Nathan Simon, Alex Bolinger

What went well?

IN GENERAL:

We did a really good job learning about Unreal and adapting to the challenges that came with not knowing how it worked from the beginning. We were very consistent with our meetings and meeting our user stories despite reaching a lot of issues with a few of them.

Player Story #1

As a player, I want to be able to see my in-game abilities and see the cooldowns if I have already used them.

#	Description	Estimated Time	Owner
1	Create ability icons to be displayed on the UI	5 Hrs	Andrew
2	Create timer that runs after an ability is used and prevents an ability from being used	3 Hrs	Nathan
3	Gray out the ability icon while the timer is running	3 Hrs	Nathan
4	Reactivate a players ability when the timer runs out	2 Hrs	Nathan

Completed:

Ability icons have been successfully added to the game. When a player spawns in they can see an icon on their screen that indicates what the ability they can use is. When a player uses their ability, the icon visibly changes to indicate that the ability is in cooldown.

Player Story #2
As a player, I want to be able to choose the time duration of my games.

#	Description	Estimated Time	Owner
1	Write an algorithm that adjusts the in-game timer	2 Hrs	Wes
2	Create a dropdown menu with different selections	2 Hrs	Wes
3	Integrate the dropdown menu with the algorithm	2 Hrs	Wes
4	Add the dropdown menu into the match setup screen	2 Hrs	Wes

Completed:
The match settings screen has been implemented. Within that screen, a user can use a dropdown box to choose the desired game time. Once the time is selected it is displayed to the user and the match data is changed. This means that when a time is selected and the game is started, the new time will be displayed at the top of the UI and the game will end or overtime will be started when the timer hits zero.

Player Story #3
As a player, I want to be able to choose the size of the ball in my games.

#	Description	Estimated Time	Owner
1	Write an algorithm that adjusts the size of a ball with different size presets	2 Hrs	Wes
2	Create a dropdown menu with different selections	2 Hrs	Wes
3	Integrate the dropdown menu with the algorithm	2 Hrs	Wes
4	Add the dropdown menu into the match setup screen	2 Hrs	Wes

Completed:
The match settings screen has been implemented. Within that screen, a user can use a dropdown box to choose the desired ball size. Once the ball size is selected it is displayed to the user and the match data is changed. This means that when a ball is selected and the game is started, the new ball will be displayed in the game and the game ball is usable and works properly.

Player Story #4
As a player, I want to be able to choose the size of the field in my games.

#	Description	Estimated Time	Owner
1	Write an algorithm that adjusts the size of a field with different size presets	2 Hrs	Wes
2	Create a dropdown menu with different selections	2 Hrs	Wes
3	Integrate the dropdown menu with the algorithm	2 Hrs	Wes
4	Add the dropdown menu into the match setup screen	2 Hrs	Wes

Completed:
The match settings screen has been implemented. Within that screen, a user can use a dropdown box to choose the desired field size. Once the field is selected it is displayed to the user and the match data is changed. This means that when a field is selected and the game is started, the new field will be displayed when the game loads by selected the corresponding map.

Player Story #5
As a player, I want to be able to play as multiple different player classes.

#	Description	Estimated Time	Owner
1	Create character selection menu and character icons	2 Hrs	Andrew
2	Develop an algorithm to set a users character to the one they click on	3 Hrs	Nathan
3	Make each character icon a button that a user can click on	2 Hrs	Nathan

Completed:
When players start they are greeted with a character selection screen that displays what characters are available for play. Players can choose one of five different player classes by clicking on the icon with that character.

Player Story #6
As a player, I want different classes to have different stats and abilities.

#	Description	Estimated Time	Owner
1	Come up with specialized abilities to give each animal	2 Hrs	Nathan
2	Give each animal specific stats	2 Hrs	Nathan
3	Create getter and setter functions that abilities will use	3 Hrs	Nathan
4	Create algorithm to identify if an ability has been used recently	3 Hrs	Nathan
5	Create function to activate an ability	2 Hrs	Nathan

Completed:
We successfully gave unique abilities to each character. Each ability has getter and setter methods that set the variable Ability_Available which indicates whether an ability has been used recently or not. The abilities get deactivated for a period of time once they are used and then are reactivated.

Player Story #7

As a player, I want to be able to easily see a scoreboard with a timer during gameplay.

#	Description	Estimated Time	Owner
1	Develop an algorithm to count down and display the remaining game time	2 hrs	Henry
2	Develop an algorithm to keep track of goals, who scored the goal.	5 hrs	Henry
3	Develop a UI Panel to hold the scoreboard and timer	2 Hrs	Andrew

Completed:

We have successfully implemented both the timer and the scoreboard in the game. When the timer runs out, the game exits and if the score is tied, the match goes into overtime. This was implemented using unreal engine blueprints and widgets.

Player Story #8
As a player, I want to see who scored a goal in the game.

#	Description	Estimated Time	Owner
1	Develop an algorithm to update the match data about a goal being scored	3 hrs	Wes
2	Develop an algorithm that updates the game statistics to add a goal to the player that scored the goal	3 hrs	Wes
3	Develop a UI to pop up that has the name of the player who scored the goal	2 Hrs	Andrew

Completed:
The goals scored UI displays and is updated properly when a ball is scored in a certain goal. When a ball is put in the goal, the UI updates, a player's statistics are updated, and then the UI is displayed, showing that a goal was scored.

Player Story #9
As a player, I want to have a sandbox mode to train my skills offline

#	Description	Estimated Time	Owner
1	Develop an algorithm to create an offline game and allow players to practice continuously.	3 hrs	Alex
2	Create a menu that allows players to change game settings before the enter the game	3 hrs	Alex
3	Debug and test the algorithms using unit tests	3 hrs	Alex
4	Create a modified UI for sandbox mode	2 Hrs	Andrew

Completed:
The player can open a game and practice for as long as they like. The ball will continuously respawn after scoring goals and the game can be customized in the same way multiplayer games will be able to.

Player Story #10
As a player, I want an overtime match if the timer runs out and the score is tied.

#	Description	Estimated Time	Owner
1	Develop an algorithm to reset the timer when the time runs out and the score is tied.	2 hrs	Alex
2	Develop an algorithm that resets the timer and makes the next score win if the game goes into double overtime.	2 hrs	Alex
3	Develop an algorithm that makes the game a tie if the game goes past double overtime	2 hrs	Alex
4	Debug and test the algorithms using unit tests	2 hrs	Alex

Completed:
The timer will reset when the main game ties and the timer runs out. The timer will reset after an overtime game ties and the timer runs out. The game will end when the first team scores in double overtime. The game will end when the timer runs out in double overtime.

Player Story #11
As a player, I want a way to communicate with my teammates

#	Description	Estimated Time	Owner
1	Develop an algorithm to send preset messages from a user to the rest of their team	3 hrs	Alex
2	Create key-mappings that allow users to select messages to send using their keyboard	3 hrs	Alex
3	Create a UI that a player can open with their keyboard or mouse and then select a message to send with their mouse	4 hrs	Alex
4	Debug and test the algorithm using unit tests	3 hrs	Alex

Completed:
We have successfully implemented chat functionality. The player can either use a keyboard binding or an on-screen menu to send messages to the chat and the messages will show up on the screen.

Developer Story #1
As a developer, I want characters to be modular so they can be added, removed, or edited later.

#	Description	Estimated Time	Owner
1	Determine which animals we want in the initial game	1 hr	Nathan
2	Create individual classes for each animal	4 hrs	Nathan
3	Panda Assets	3 Hrs	Andrew
4	Lion Assets	3 Hrs	Andrew
5	Kangaroo Assets	3 Hrs	Andrew
6	Penguin Assets	3 Hrs	Andrew
7	Monkey Assets	3 Hrs	Andrew

Completed:
Each character has their own class so that new characters can be added easily, edited easily, and deleted without affecting the other classes. All the assets were created and added to the game.

Developer Story #2

As a developer, I want a time limit on how long a player can hold the ball to encourage teamwork.

#	Description	Estimated Time	Owner
1	Develop an algorithm to detect if the ball is being held, and for how long.	3 hrs	Henry
2	Make a visual indicator on the player's screen when the time limit is about to be met.	2 hrs	Henry
3	Debug and test the algorithms using test documents.	2 hrs	Henry
4	Make an asset for the ball when it has reached the time limit.	2 Hrs	Andrew

Completed:

This was fully implemented in the game by having a pickup and drop ball button. When the ball is being held, a countdown timer begins to stun the player and when the drop button is pushed the counter stops running and is reset. This was done by using blueprint functions.

Developer Story #4

As a developer, I want there to be a limit on how often and how long a player can sprint.

#	Description	Estimated Time	Owner
1	Develop an algorithm to calculate how long a player has been sprinting with a time limit.	2 hrs	Henry
2	Develop an algorithm to adjust the player's speed if the time limit for sprinting has been reached and recharge their sprint ability.	5 hrs	Henry
3	Debug and test the algorithms using unit tests.	2 hrs	Henry
4	Create a visual indicator of how much sprint time the player has left, as well as a recharge indicator.	1 Hr	Andrew
5	Create an asset for the animal class when they have reached their sprint limit and must move slowly.	5 Hrs	Andrew

Completed:

This was successfully implemented into the game using widgets and static bindings. I made a stamina variable and bound it to the left shift click so that when stamina was being used, the player would sprint, and when the player ran out of stamina, the would be forced to walk. There was also a visual indicator made to show the player how much stamina they had left.

What did not go well?

IN GENERAL:

We had some major issues in our sprint planning document concerning conflicting user stories with user stories we didn't plan to complete. We also had some large issues with being able to move the ball to the player's hands, which didn't directly affect our acceptance criteria but did slow a lot of progress.

Developer Story #3

As a developer, I want shots with the ball to be more powerful if consecutive passes were recently made.

#	Description	Estimated Time	Owner
1	Develop an algorithm to detect if a pass has occurred and if it was caught by a teammate or an opponent.	5 hrs	Henry
2	Develop an algorithm to calculate the successful pass count and increase the pass strength with each successful pass.	2 hrs	Henry
3	Debug and test the algorithms using unit tests..	2 hrs	Henry
4	Make an asset for the ball when the pass strength is at its maximum value.	1 Hr	Andrew

Completed With Issues:

While I was able to implement different pass speeds, an oversight that we made was we were not doing multiplayer this sprint. Thus, there was nobody to pass to or determine if a successful pass occurred. Since basic functionality was implemented with different pass speeds, there was nobody to pass to, so this was ultimately something that could have gone better.

How should you improve?

One thing that we could improve upon is communication and more frequent merges. Sometimes, multiple team members would be on the same branch and the other member would not realize, leading to complicated merge conflicts where sometimes a member needed to start over. Furthermore, we would try to implement all user story criteria in one branch/one or two merge commits. This also led to much bigger merge conflicts, and would occasionally hold up other members that needed only one or two of the features to wait until all features were done.

Another thing that we could work on for this next sprint is polishing and finishing up features before moving on. For the first sprint, we needed to get a lot of broad things done before we could go into sprint 2, where the plan was to go into more depth and add things like multiplayer. We, however, ended up doing a lot of temporary fixes in order to pass all of our acceptance criteria. For this next sprint, we need to be careful about not doing too many of those because we are going to have to finish those up on this next sprint.

A major thing to work on visually is finding a way to get animations and meshes into the game in a more efficient way. Each mesh that we added to the game had large issues with the textures, armatures, and animations as soon as they became added to our game instead of blender. The solution is using more tools in Unreal to make adding animations and additions to meshes much simpler in execution.

Overall, a theme of this sprint was an oversight on the front end in terms of planning. This would help a lot with all three aspects previously mentioned. If we had planned better, then communication wouldn’t have been as big of an issue. If we had planned better, we could have picked stories to polish fully and not move on for the sake of time. And lastly, if we planned better, we could have understood that the visual stories were too extensive for a 3 week process along with the other work we had to get done.