**Development Diary**

January 24th:

We began by planning what each member of the team was going to do. I was assigned to do features such as ropes and ladders. We also agreed on a scope and list of features we needed and could make in a rapid prototype. We agreed on a basic level layout that allowed us to focus on creating more individual features and working out how they would function on a logical level.

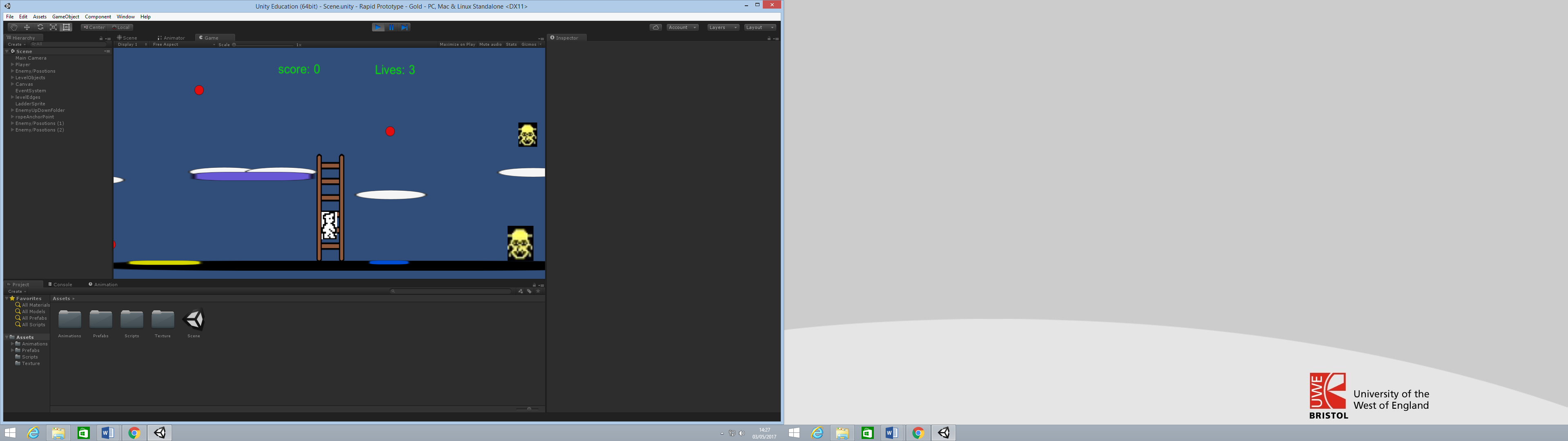
We ended up deciding on this set of features:

* Low Gravity Zones
* Bounce pads
* Conveyer Belts
* Ropes
* Ladders
* Enemies
* Physics-based player movement
* Moving Platforms
* Platforms that can be jumped through from underneath (“Thin” platforms)
* Collectables

We started at getting movement working, which proved to be a bit more difficult than initially thought as programming and debugging the jumping logic took time as we figured out how to stop the player from being able to jump mid-air (and thus, infinitely). By the end of the session, this was completed.

January 27th:

During this session, we all began work on the features we had been assigned. I started work on Ladders, figuring that the player could just be a moving collider that changes behaviour when it collides with different triggers. Using this logic, I changed the player’s behaviour on touching the ladder to allow them to climb up and down the ladder using the W and S keys respectively. This logic can also be applied to collisions with enemies and other features we need to implement.



January 31st:

By this point, we have finished these features:

* Low Gravity Zones
* Bounce Pads
* Conveyer Belts
* Enemies
* Physics-based Player Movement
* “Thin” Platforms
* Collectables

I finish off Ladders and begin working on ropes. These turn out to be very difficult for multiple reasons. Firstly, I needed to figure out how the player would interact with the rope beyond just the initial collision, because unlike the ladder the rope moves and as its position changes, the player fell out of its range and the behaviour of the player changed back to normal. To remedy this, I figured out how to make the player a child of the rope while they are on it, making it so the player followed the rope’s motion. The second issue was adjusting the movement of the rope itself, ensuring it only swung forwards slightly before going back, in a normal swinging motion. By then end of the session, this problem remained as the rope was moving but moved around in a full 360 degree motion, which isn’t how the rope would work in the real game.

Figure 1 Getting the rope to swing back and forth.

February 2nd:

Our last session working on the rapid prototype. I got Ropes to make the player a child as it moved and managed to adjust its movement script to swing back and forth. We then added a moving platform and had to figure out how it would work logically in relation to the player’s movement. We decided to make the player a child of the moving platform while on it, but allowing the player to move on their own beyond that relationship.

Finally, we added some polish to the physics and the way the player interacts with other objects to allow the player to jump off them. We then submitted the rapid prototype.

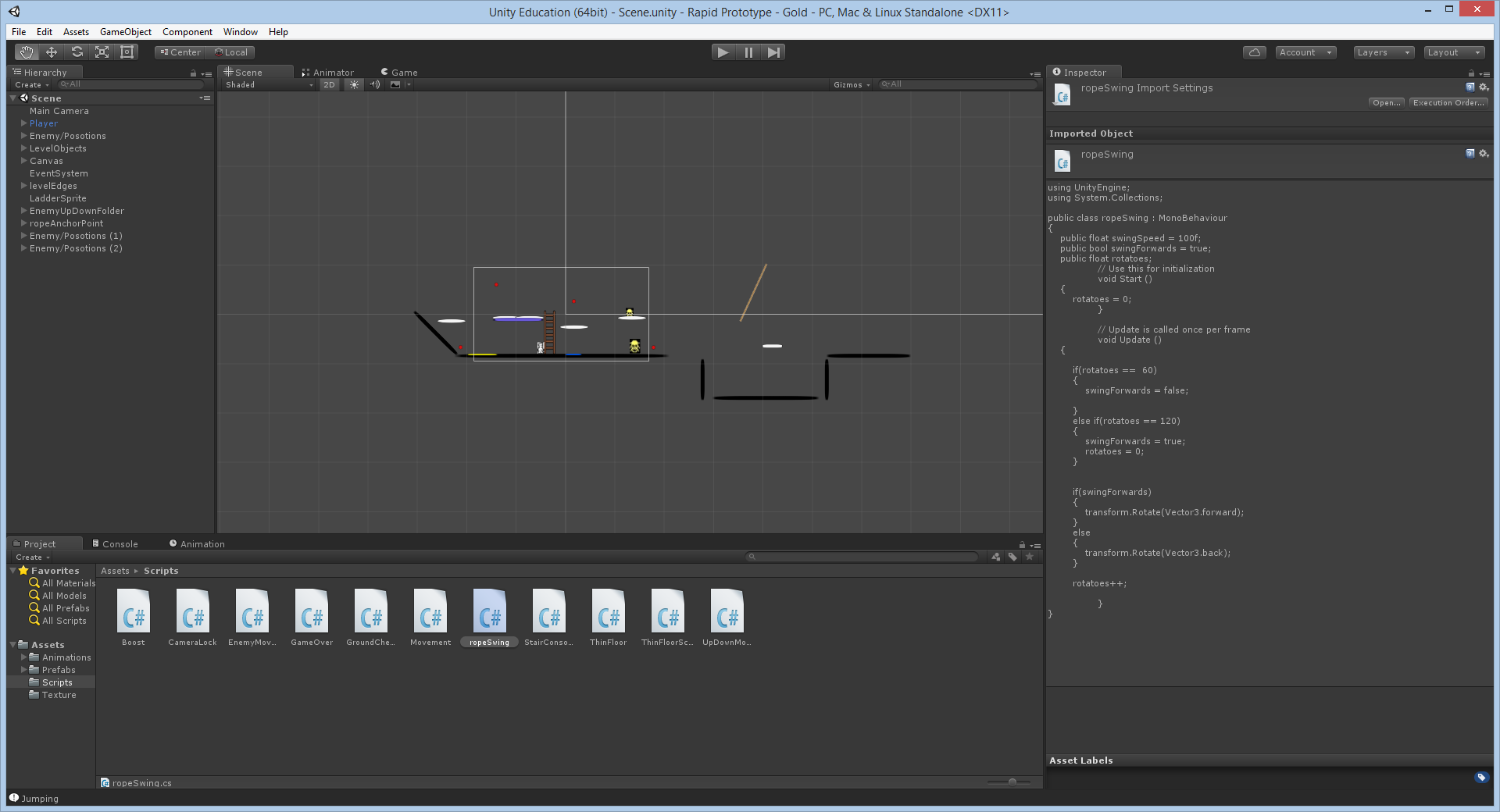


Figure 1 Getting the rope to swing back and forth.

February 7th:

Our first session of the sprint to Alpha. We planned who was going to spend their time on what tasks. I was assigned programming of Horizontal Movement of the player, physics for jumping, the state machine (navigating between menus and different game states) and the in-game HUD.

Of my tasks, I chose to prioritise horizontal movement of the player as that would allow the rest of my team to test collisions and camera movement as they developed. I could get a basic framework for the horizontal physics working as the initial codebase we were given had basic 3D physics implemented already, which I was able to convert to 2D and adjust values to create a foundation. I plan to improve on this in the next session and look at implementing a basic version of the UI which the others on my team can build upon as they implement collision logic with objects such as enemies and collectables.

February 14th:

Our group continued working on the alpha build. We ensured that all of our work done over the week had all been committed and merged into the repository correctly and that the current repository build ran as intended. Once that was done, I worked on implementing the HUD for the game, showing information such as lives, the name of the current room and how many collectables had been acquired by the player. I also created variables that can be used by the other members of the team when collision and collectables are implemented. I also helped another member of my team implement the base physics for the player’s jump.

Next session, I plan to begin implementing the menu systems. This will include the beginning main menu as well as a game over screen and the state machine required to transfer between these different menus.

February 21st:

As the deadline for the alpha build of our project is due in next week, for this session we devoted some time to figuring out what our priorities were for functionality and assigned tasks for each group member based on the features we felt were needed in the alpha. The task I was assigned was creating a ladder class and its functionality as the group wanted to have a template for objects that changed the behaviour of the player for future reference.

While at first I designed the class to create an instance of the object, I went back and recreated it to hold a vector of instances of the ladder object, based on the parameter sent through the constructor. This allows us to easily scale ladders and create instances of them at different lengths.

While I was able to also implement basic collision between the ladder and the player in the session, I still need to program the behaviour of the player while on the ladder, which involves making the player move up and down the ladder without gravity affecting them and changing player behaviour back to normal if the player jumps off of the ladder.

February 28th:

The final group session before the deadline for the sprint to Alpha build. Our group assigned our final tasks for the alpha sprint and what needed to be done for the deadline specifically. I was allocated to continue working on the ladder and implement new movement logic for the player as I had recently finished working on a movement script that emulated physics more accurately than our current movement.

While implementing the new physics was a simple enough job (adjusting the logic to work in 2D as opposed to 3D was the only hurdle), making the ladder work proved difficult. By the end of the session the player could move up and down the ladder, but it wasn’t polished and there was a big bug in that if on the ladder, the player could hold up or down to go as far as they wanted in that direction – even once they had passed the ladder. If they then pressed left or right, the physics would return to normal and they would fall, but this could be game-breaking and will need to be addressed soon.

April 11th:

The first session post-alpha. In this session, the group organised which tasks we need to focus on in the run up to the final release. The priorities were set as follows:

* Adding in more platform types (such as conveyor belts, ropes and ice)
* Polishing Collision
* Finite State Machine
* Level Editor and File Loading

I was given the tasks of adding in more platform behaviours and collision because I was responsible for the base player physics. I started by trying to adjust the player physics to make them feel a bit more intuitive for the player and fixing an issue the game had in the alpha build in which the player would be able to move up or down indefinitely from a ladder. This occurred because the collision logic had no way of setting the player’s grounded state to false in the event that the player wasn’t colliding with anything, and thus the player’s behaviour wouldn’t change upon leaving the ladder. To remedy this, I added code that would set the player’s ‘isGrounded’ variable as false if they weren’t colliding with anything, which would allow for gravity to be appropriately applied once the player left the ladder.

April 30th:

The main task for this session was to fix a bug with the collision which was causing the player to walk where they should fall. This error occurred because of the way the collision logic worked – If the player was colliding with any object at all (even if the object wasn’t a platform), the game would stop checking for collisions and the player would not have its ‘isGrounded’ Boolean set to false and thus would not have gravity applied to it. This was fixed by changing the logic with a local Boolean that was set to true if the player was colliding with an object. This ensured that each object was checked the player could have multiple collisions applied to it when required.

May 1st:

In this session, I focused on adding in the Ice and Conveyor Belt platforms into the game and balancing their behaviours. Fortunately, integrating them into the level loading system was relatively simple as I just needed to copy the formatting for other platform types and change the character to ensure it is different from the other previously established object types. Once that was completed, I added in the Maid object, which is used to check if the player has found enough collectibles and if so, sets the game to the win state. While this was fairly simple to implement, it also needed to be impassable as long as the player didn’t have enough collectibles which added to the time I needed to dedicate to the task.

Post Mortem:

Overall, the development process of the game was smooth for the majority of the time and without many (if at all) unnecessary conflicts, but our organisation began to wane with time and this took a toll on the game’s final quality.

The rapid prototype stage was a very important stage of development, despite none of the implementation being physically in the final implementation. This stage helped the group figure out how each member worked within the working environment and cement the roles that each member would have in further development. It also provided an opportunity to test out concepts that could be used within the game and how we might implement those on a logical level.

The sprint to Alpha was a very productive period in development as we were constantly working on the game and iterating, adding in the core features such as the HUD and basic movement, along with a couple of object types such as ladders, enemies and thin platforms. Each member seemed to have a similar vision in mind and the lack of conflict meant we could spend more time polishing our initial implementation and ensuring that any bugs we found were promptly dealt with. We did have some trouble with the implementation of the ladder which ultimately had to be pushed back into the next stage of development due to time constraints, but all other features were working as intended.

The sprint to release was unfortunately much less structured than the other stages of development and a large gap in development left us with not as much time to polish and implement everything we wanted for the final release. As a result, while the release is functional there are a lot of improvements that could be made to the codebase as well as the gameplay. The last week of development involved solving a lot of small bugs that had very detrimental effects on the game and pushing in the final features to make the base game work as intended, which could have been spent on extra features if the group’s organisation post-alpha was better.

However, looking at the whole, the group was effective. While there was much to learn in terms of organisation from this project, if put together again the group would be very able if these lessons were taken on board.