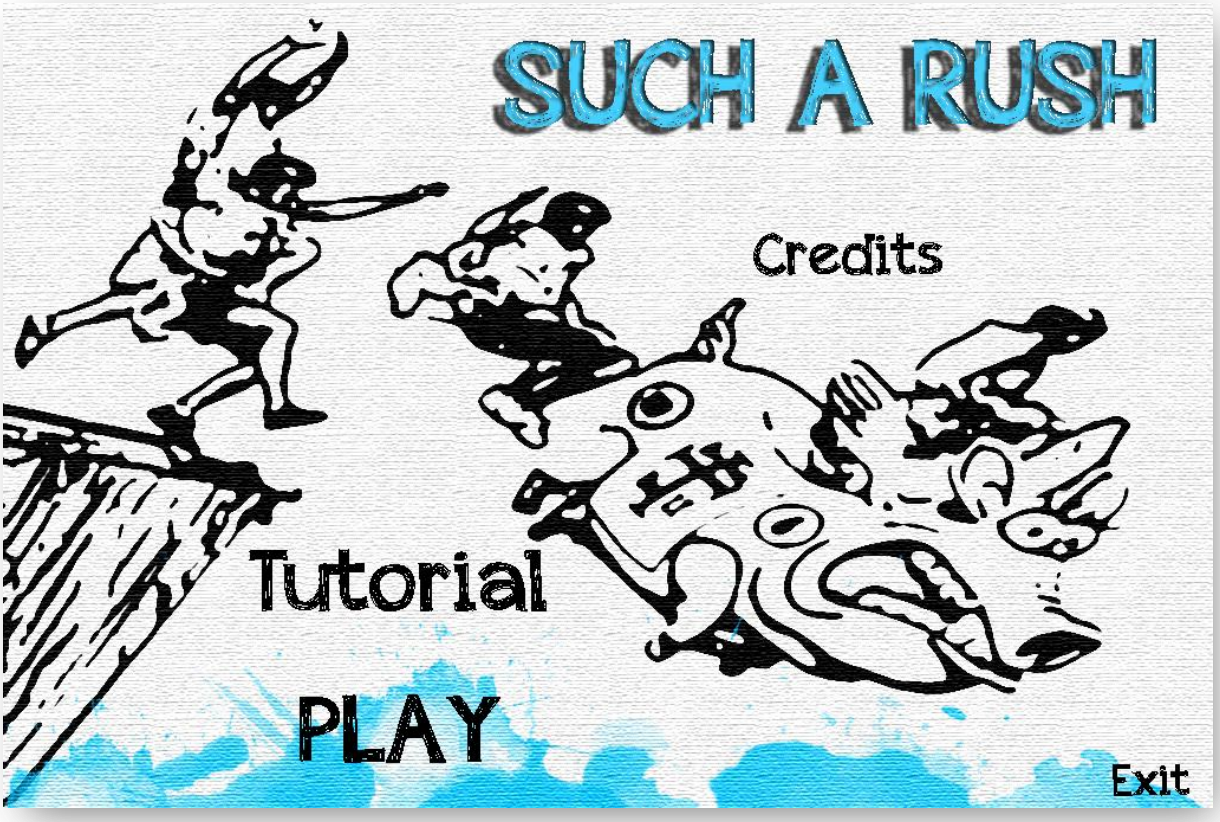
GATE 505 – GAME DEVELOPMENT PIPELINE

ASSIGNMENT 3

Due Date: 12.12.2012
İrem Gökçe AYDIN
1550268

"SUCH A RUSH"



Title And Description : The aim in my game is to fly over the sea as long as possible and as far away as possible. There will be one level and the player will try to do his best in this.

Game Summary: The story in the game takes place in one of Red Bull Flugtag events. In the game the player is in the place of an event competitor. Because the expected aim is to speed up over a platform and than fly so far away or for a long time, in the game there will be a platform like roller coaster to be able to speed up, and a lake to fly over. Along the railway, the player will be able to collect some power-ups to improve his speed or fly skills. The game will consist of 2 main stage: rolling & flying. While travelling along the rail, by picking the "engine" power-ups, the player can speed up to increase the distance in flying stage. In the flying stage, the player can use "wing" power-ups collected in rolling stage to fly for more time.

Characters: There will be only one controllable character, the player himself. In the future, some external obstructive characters can be added(like birds intervention in flying stage)

- **Player:** The player use his hand-made vehicle in rolling state as speeding up-down and moving little right-left to collect power-ups. In flying stage there will be no control over the vehicle besides of using "wing" power-up. There will be first person camera in rolling stage, and third person camera used in flying stage. By slowing down the vehicle at bends, the player assures not to be derailed.

Figure 1 – A screenshot from the game play

- **Animals :** Animals can prevent the player vehicle from speeding up or obstruct him while flying. One of them can be birds.

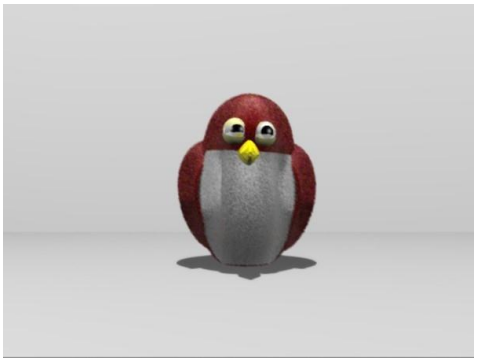
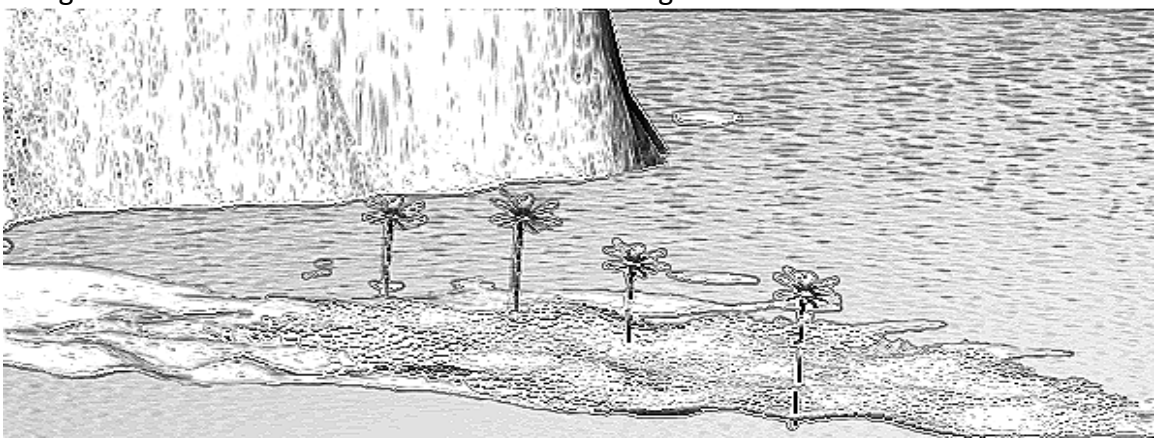


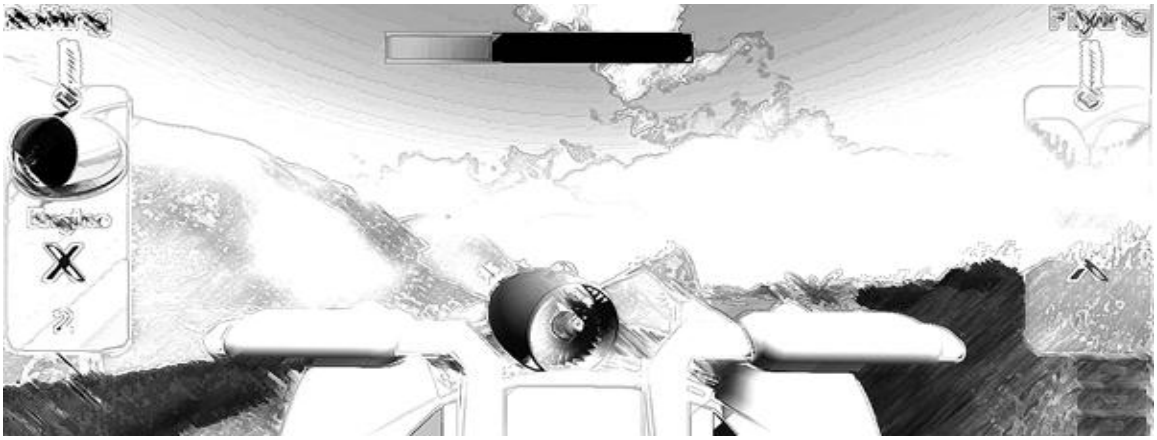
Figure 2 - Bird model to be used in the game

Storyboard:

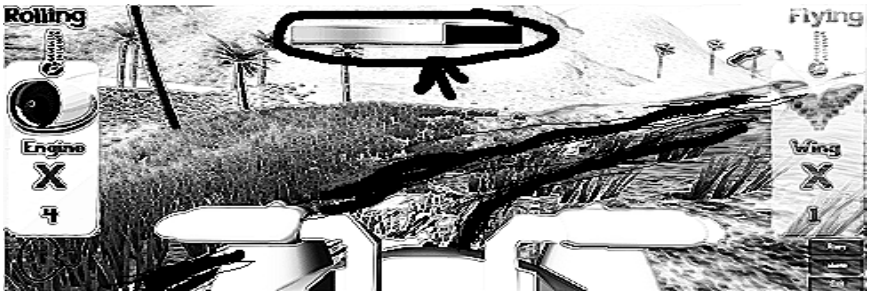
1- The game world will be lakeside like an adventure game world.



2- Beginning with the first rolling stage, the player will move along the rail with arrow keys and can collect the power-ups to be used in rolling stage or in flying state for the future.



3- In the rolling stage if the player speed up much, his vehicle can be derailed. To avoid this, the player should slow down, especially on the bends.



4- By pressing space key, the player can use collected engine power-ups, and speed up. The wing power-ups can not be used in rolling stage, even if being collected. Wing power-ups can be used only in flying stage, engine power-ups can be only used in rolling stage.

5- At the edge of the platform(end of rail), according to the speed gained, the player will begin to fly with an initial forward speed. At the flying stage, collected wing power-ups can be used to fly more long time by decreasing the weight of the player vehicle.

Game Components: The physics of Unity engine and some kind of path libraries(iTween) will be used while moving the vehicle of the player(and player too of course) and for collisions. An movement AI for animals to rush towards the player will be used.

------- Assignment 2 Content-------

Narration: The story of the game is based on the idea at the flugtag event. The narration technique for this story will not be with scrolling text or voice by a narrator. Because text based are boring, 2-3 illustrations and less text will be used to tell the story as indicated below example. And the story in game is aiming to make the players feel the competition ambient and feel as the champion of the event.

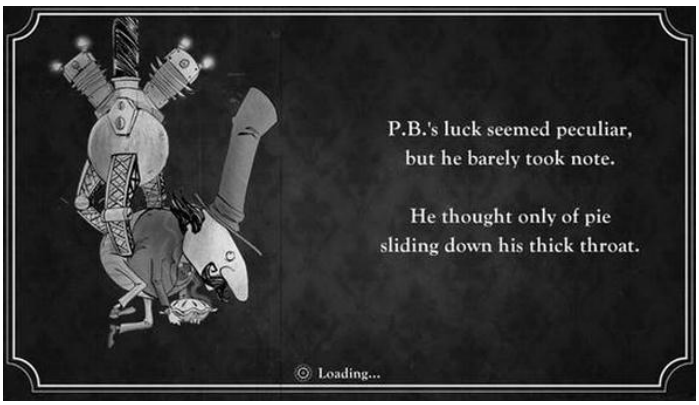


Figure 3 - Narration Approach Example(The_Misadventures_of_P.B._Winterbottom)

Storytelling: There will be no use of cut scenes during the game, and need for interaction with player to tell the story. The environment will give a clue about the story and only some notifications and feedback information about the state during the game will be used to make the player feel more like a competitor. The story will not flow in a different sequence, and there will be no revealing of information which are binded to previous states or events. Moreover, there will be no interactivity with the player decisions, to change the roles or lead the progress within a sory. So the story will be linear. The story will begin by talking about the flugtag event, and player is announced as a competitor for this event. And the challenges and predefined aims(and also controls) will be listed in the tutorial part of the game. The player is supposed to finish the contest as successfully as possible.

Gameplay Properties:

- **Actions:** The only action is adjusting the speed of the vehicle(slowing up and down and the vehicle always moves forward) and using the power-ups.
- **Challenges:** Mainly there is two challenge in the game, one of is distance other is time. The ultimate challenge is to complete the game. To do this the player has sub missions and while doing this missions, the player encounters some atomic challenges.

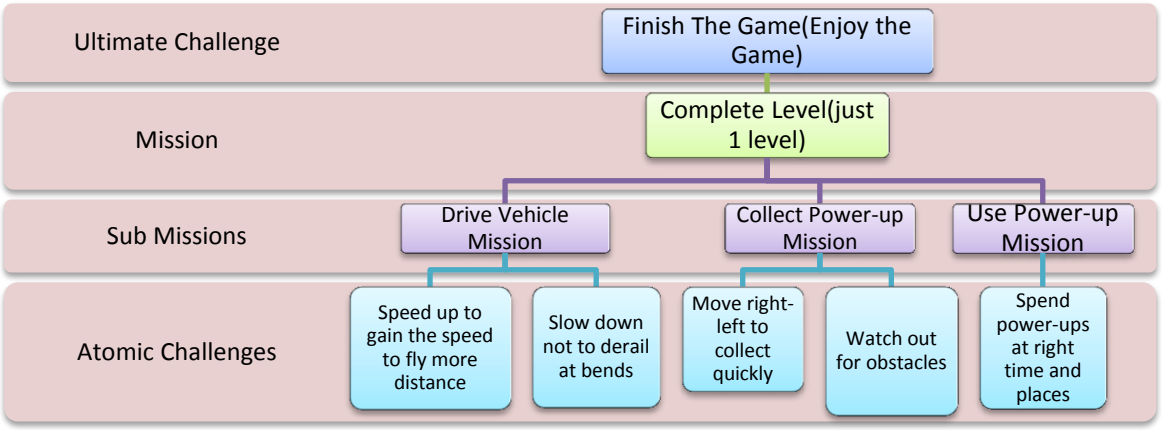


Figure 4 - Hierarchy of Challenges in the Game

- **Difficulty Adjustment:** The difficulty will be obtained by modifying the challenges, (power-up randomness, vehicle weight, speed up-slow down factoring variables).
- **Controls:** The game will be PC game and controlled by keyboard(only arrow keys and space for actions to be done)

Level Design: Because the environment should be kind of lakeside where the nature exist, the terrain design gets importance. Different terrain assets and interesting models and effects(waterfall, rocks, etc.) will be used. The power-up positions will be set randomly. And of course, there always be a wavy rail for vehicle to move on. For rail design there is a editor tool for Unity called EasyRoads3D and I have used one. There will be 1 level stage. There will be open layout. A level map will be provided. The terminate condtion of this one level is to fly and jump into the water finally. All the cahallenges stated previously take role in here.

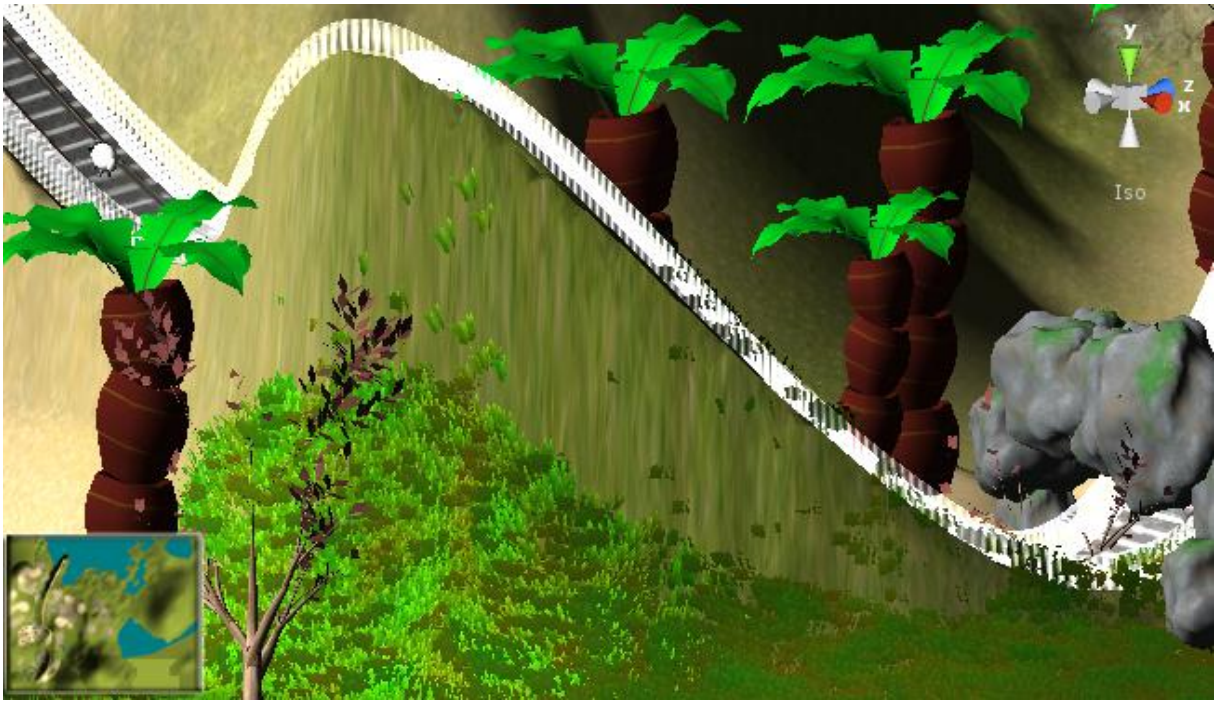


Figure 5 - A screenshot from the game

In the game design, the navigation from game level to main menu, retry, exit and sound on-off operations are enabled via buttons as below.

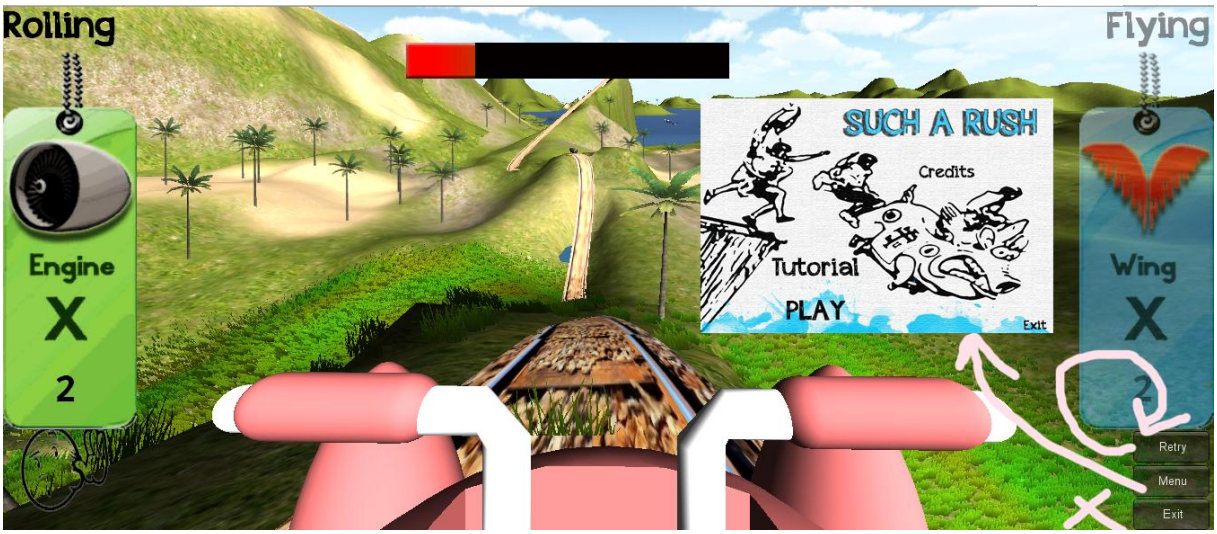


Figure 6 - An image of the flow between game level and game shell

User Interface Design: There will be simple intro screen and during the game collected engine and wing power-up numbers, speedmeter, power-up usage notifications, time and distance information in flying stage, game control buttons are used as GUI elements. At the tutorial scene, the aim, stages and controls of the game are provided.

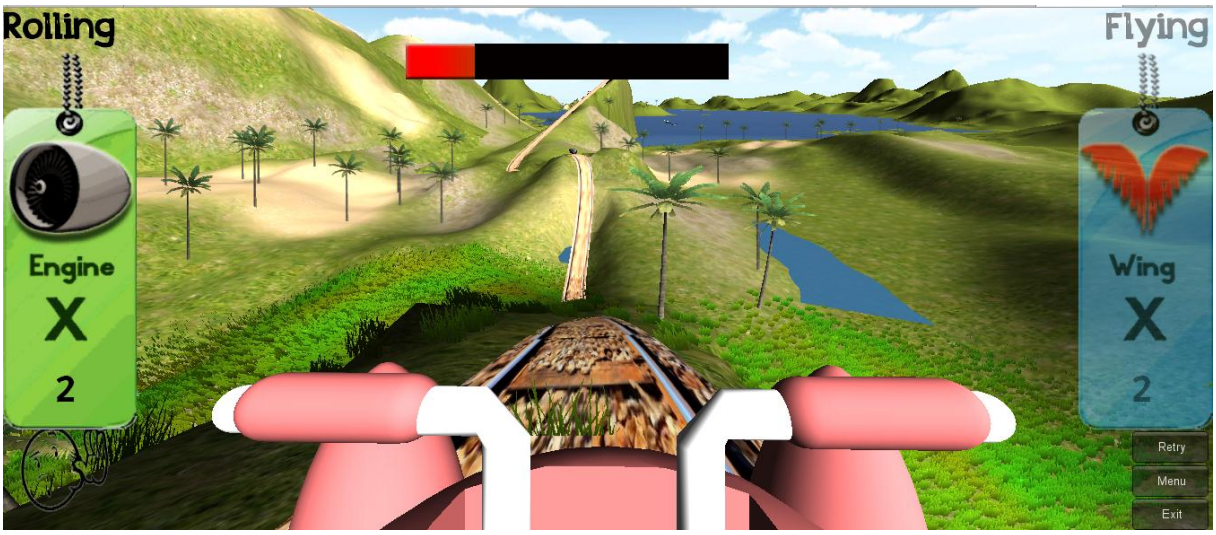


Figure 7 - A screenshot from the gameplay at rolling stage

Figure 8 - A screenshot from game play at flying state

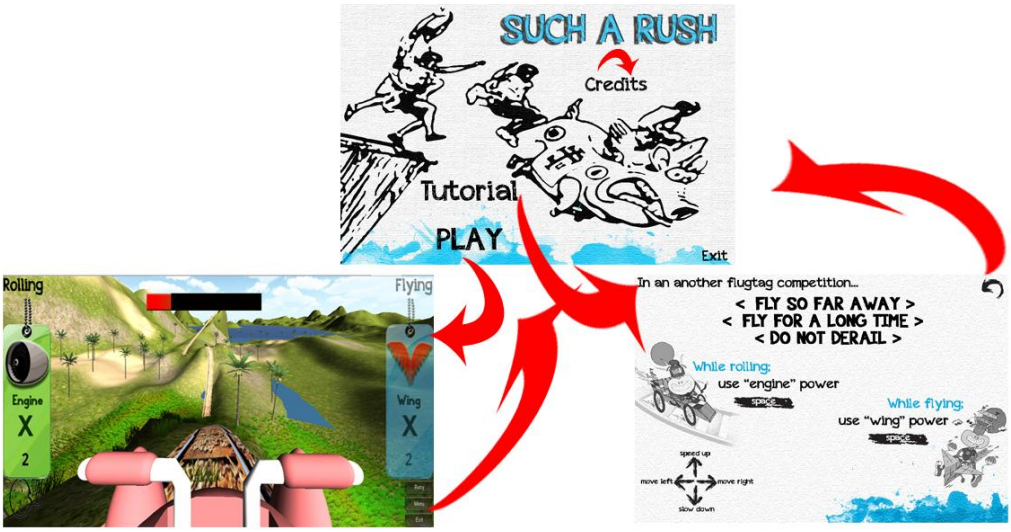


Figure 9 - Shell navigations among interfaces

Balancing: Because there will be no player skill improvement, to keep the player in flow zone, the challenges will be well designed. The power-up instances will be generated randomly, so player will not move around the same states again. Because there will be one terrain and one rail, the bends and slopes of the way will be well design to feel need for speed up and slow down. Some external interfere by animals while rolling or flying stages can be used to increase the feel of challenge for the player.

Rules: The only rule is not to be derailed(which is a fail condition) and to jump into the water after flying process. The scoring will be performed according to the time and distance taken. There is no winning condition, the player will be assessed according to the score.

------- Assignment 3 Content-------

Debugging: The only debug methoding used is logging. When getting an error, if the line is not implicit, I trace the code from beginning and guess the possible error generator points and print the variables to console. By reproducing the error and tracing the prints at this time, the error line is found more meaningfully. The problems encountered so for can be classified as follows:

- **Unity Structure:** Because the rendering and running architecture and predefined functions(start,update,fixedUpdate,onGui) are different from any other game programming environment, it takes time to get used to it.
- **Scripting Editor:** With monodevelop support, scripting and debugging with breakpoints are easier now.
- **Accessing Variables and Crossreference:** Especially trying to access variables of different script files(with different formats, c#,js) has taken time to understand and implement.
- **Changes on Pause State:** The changes in the pause state are not permanent. In fact, Unity should ask and keep the changes. Especially for setting the GUI elements alignments, there is a need for visual design editor too.
- **Physical Property Bugs:** When dealing with rigidbodies and colliders, checking the collision on update or onCollision.. functions give different results. Moreover, when changing the resolution can give different count results for collision occurences.
- **Designing GUI According to Resolution:** When designing the user interface according to the Screen properties, the proportions sometimes makes the GUI elements look inappropriate.
- **Vector Calculations:** Understanding the references to the world and object, and the concept of forward, up, right vectors are little bit challenging. When setting positions, rotations and giving velocities, the vector products implementations are little tricky at firts.

Testing: Because we have been at alpha testing stage so far, I encountered with the problems myself and handled bugs and fixed errors. When fixing the errors, I mostly change

other files content to stabilize and well form the structure and to isolation in terms of functionality. With this approach, afterwards, regression conditions have been occured less likely. We are now at the stage of beta testing. So I have shown my game to some of my friends and observe the error encountering situation and flow of them. Because I progress by module by module, error encountering have not been the case. By asking about usability and functionality I get the following feedbacks and rates from them:

Number of participant: 4

Are you enjoy game environment?	**: 4/4**
Are you enjoy game mechanics?	**: 4/4**
Are you enjoy user interface?	**: 2/4** (usage of rolling and flying stage powerups couldn't be distinguished)
Are you enjoy with the interactions?	**: 3/4** (power-ups should be randomized)
Is controlling is easy?	**: 3/4** (mute button is not implicit)