# Team Pacman – Codecamp report

**Team:**

Juho Juvani

Manuel Delgado Sanchez

Ibrahim Olanigan

Vitor Meneguetti

## What did we start to do?

 We used a PacMan created with Unity. Project with unfinished code and frame. Our goal was to make a “real” pacman game with lots of new features. This meaning in the spirit of pacman yet with 3d models and 3d movement, jumping etc. Main objective of these goals was to achieve understanding of Unity engine and it’s functions, so we could have basis to learn and do more. While Unity is it’s own engine, the functions and concept can be applied to many similar systems, and it teaches how some codes and functions should be structured.

The original list of goals included:

1. Movement codes, different scenarios.
2. Create a new AI for the ghosts.
3. New 3d models for pacman and ghosts.
4. Pacman / ghost collision and death functions, with game restarting after death.
5. Create a new scenario using pre-defined textures.
6. Add some power ups for pacman (increase and decrease speed).
7. Special type of pellet hides somewhere.
8. Finish game eating all pellets.
9. Merge all the scripts.
10. Connect to GameCloud and upload achievements

## What did we really do?

1. Implementations of several features and design of new 3D models for the game.
2. Learn how Unity works.
3. Learn how to use Blender to create 3D models.

Basically, we were able to take a barebone project and implement movement functions and scripts that react to different actions between objects. Eating pellets, jumping, defining gravity, as well as changing speed effects through special objects (Pellets).

## What problems did we have?

Our game was developed with JavaScript and GameCloud with C# so originally we were not able to identify this, and failed to connect through the codes provided by the API. A good lesson we learnt was keeping good record of different coding formats and languages. In the morning of last day, we changed a section of  our code to C# and we could connect with the cloud.

Another problem encountered,  related to combining of coding languages, was also importing 3D designs into the Unity project. Importing the designs is simple and doable, but when they involve animations, things become more complicated. Newer Unity animation system requires coding added to the movement, and this tends to work more or less poorly on the ways we discovered. Unity likely has more simpler form to work on it’s own code, but we could not find this way in the short time of couple days we had.

We attempted to engage animation into wrap-mode where it would loop constantly, but Unity’s own code would not accept the format by default. Which likely is on our end, as we were not able to discover right means. Using a fresh character model animation was usable, through legacy system where it is automatically set to loop through the functions. This new model however would have required us to remake movement controllers and connect the appropriate variables and objects to other scripts manually again, and we did not have time to do it.

Biggest problem with our original plans, as all participants of Codecamp, was that we did not have a common place to work. We were counting on all changing ideas and resources within the short time, and this did not go as planned. It is a good experience from which we can learn and improve on. For next similar Codecamp, it may be good idea to document, beforehand, which groups work on similar type resources: 2d, 3d models, textures, formats, framework or engines. This way groups know which type resources they can trade and implement with ease. Another is to prepare a workplace where the whole group is together. It is easier to monitor and it is more productive for the work.

## What did work and what not?

We achieved all the new features we decided at the beginning, when it comes to the code and core functions of the project. We were able to determine character and movement controllers, scripts the effects and objectives we set for the short time we had available.

Our team work was capable and through some early chaos and settling, we were able to get into a good rhythm of working together. Tasks were assigned and we were able to help each other well to get where we wanted to be, though eventually would have wanted to reach a bit further. Enable new models, animations, and if we would have had time, learned proper texturing and implementing different type resources from other type games, into a 3d engine.

Eventually after a struggle, we were also able to achieve connection with the cloud, and even have it interact with the game through figuring out the Unity code and automaticly generated keys from the API.

What we could not achieve was implementation of new character models, and their animations. This was just a little beyond us, but we are assured that with more time, it could have been done. It however was not the focus of this experience. Our focus was on the code and hope to connect with other games and perhaps have achievements in our game, activate some function in another game. Unfortunately this milestone was not reached, and it’s the only real regret we hold.

## What should have been done in another way?

We should have use GitHub to merge the project between us. Take a good control of versions and spreading assignments between ourselves. Our orderly chaotic working process worked, in such a short project and between a small team, but it’s not a good practice in any larger scale. During this CodeCamp, we used an USB stick, and this time it did the job.

We shortly attempted using GitHub and Bitbucket with Git as source code management, but we were not able to go further due to the use of different operating systems like Windows and Mac (Unix based). These infrastructure issues would lead us to loss some time until getting everything working correctly to everybody.

Time was lost, although not wasted, getting the new code from each other, as it demanded extra attention to be sure that no code was left behind. Considering the short time of a Codecamp, the version control would have saved some precious time if this infrastructure topic was prepared previously.

The very last thing we absolutely should have done, in hindsight, was testing the cloud tools and API as a very first task. As it was the main focus of Codecamp, we really should have focused our efforts more around it. We should not have focused on forcing it into our project, that would clearly not be very polished or fully made in the end. Time and management is clearly the key factor in short sprints, such as this weekend was.

Through making a more concrete project plan, revising the tools and resources we would use and should have used, would have saved us time. We would have likely been able to notice that Unity’s own javascript based code would not work well with C# based coding of the API. We would have also been able to manage assignments better. Set clear boundaries of what is done, who does it, by what time and what should the end results be.

In the end, we did get a result. We were able to get a working product, although not to a point where we originally hoped, but it was an excellent experience for our team. Only one of us having slight experience of the tools and process, it shows few things we should remember in any project we are to take in future. Planning, preparing and execution within agreed timetable. Out of these we were able to achieve everything, partly. Planning and preparing was done, but without previous experience to similar project, we were not able to be wary of everything that we should have.