MOOC's book

Game Design and Development

There is a difference between watching a movie and making one, as well as play a game and create one. The specialization allows one to learn how to develop a (video) game. It is composed of 5 courses.

The first one is Introduction to Game Development which is an introduction to one of the most famous video games engine, Unity 3D. This course is presented by Brian Winn.

The second course is Principles of Game Design which presents the process of a game development. This course is presented by Casey O'Donnell.

The third course is Business of Games and Entrepreneurship which presents how to sell a game. This course is presented by Casey O'Donnell.

The forth course Game Development for Modern Platforms which is an advanced course on Unity 2D. This course is presented by Brian Winn.

The s4 part starts with the third course.

Git link: https://github.com/nicolasventer

Table of contents

Introduction to Game Development

Intro to Unity 3D

Video Game engine

Unity 3D is a video game engine, it is a tool which helps game developers for everything: the rendering of models or lights, the 3D physics like gravity or friction, the sound and its propagation, particle physics … All these features are long and difficult to implement. An engine is used for making easy the development of a game.

Development software

Unity 3D is a free development software with a huge community. Projects can be shared and some models can be bought, like 3D models, Artificial Intelligence, sounds …
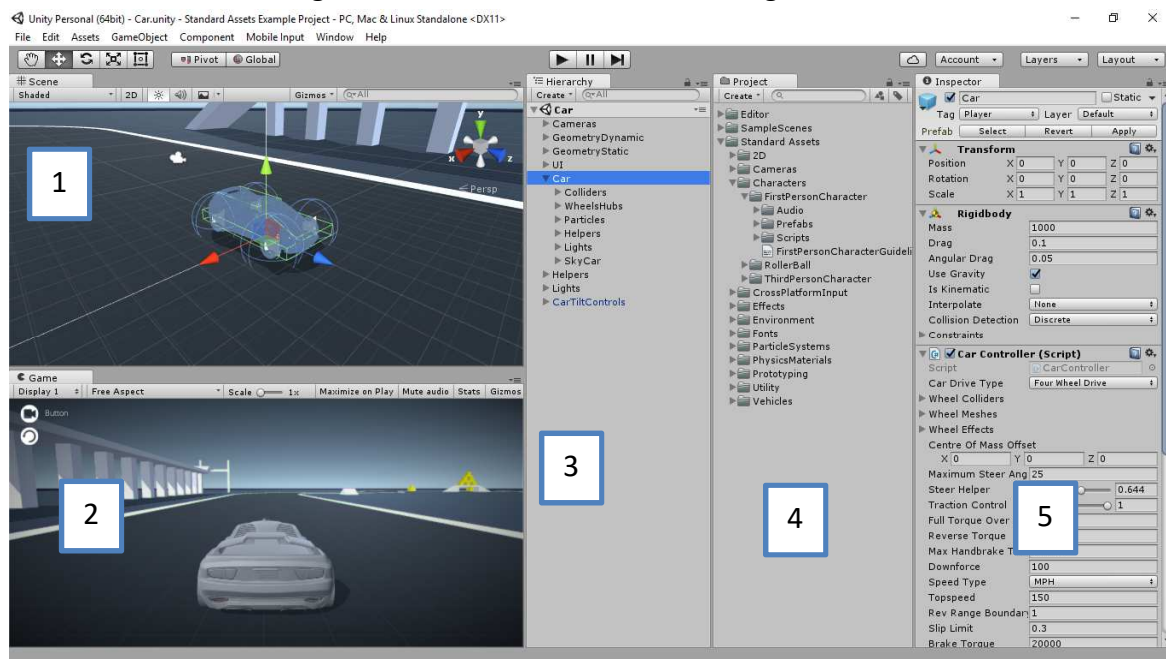
Standard Unity view (Window -> Layouts -> 2 by 3)

1) Scene View = the editor view that allows you to navigate the 3D world, as well as select and manipulate 3D assets in the world.
2) Game View = Upon hitting "play" in the top middle of the screen, you'll compile and preview the gameplay in this view. By default the view shows the active camera.
3) Hierarchy Panel = This is a hierarchical list of objects in the game. You can select objects, rename them, and delete them (Shift-Delete)
4) Project Panel = This shows all the usable "assets" located in the project's Assets folder.
5) Inspector = Allows you to view and edit the component settings for the selected gameobject in the Scene View, Hierarchy, or Project Panel.

Development tool

In Unity, there are objects. Each one has a location and position on over its parent. It has scripts and components related to itself. It can be the reflection of a prefab, a model object. Unity contains a lot of prefabs, scripts and other features which can be add to an object.

The scripts are written in C# or javascript (only C# in this course). Some functions are automatically called like Start, Update, OnCollisionEnter, … MonoDevelop is an IDE proposed by Unity for developing script.

Projects in Unity 3D

Solar System

For this project, I learnt how to:
- Import a package of assets,
- Create a 3D sphere and add a texture on,
- Put scripts and audio source on object,
- Set the light and background,
- Use cameras for minimap.

In Unity, package of assets can be imported (and exported). Assets can then be used in the project.

The Solar System is composed of the Sun and 8 planets rotate around the Sun and rotate on themselves. The Sun is luminous and noisy. The Earth has a Moon which is rotating around the Earth.

In the project, each planet and the Sun are 3D spheres on which an imported texture is set on. The background is imported. There is a minimap for seeing all the Solar System. You can change the view in pressing on the target you want follow: for example if you click on the Moon, the Moon will be the center of the view while the rotation.
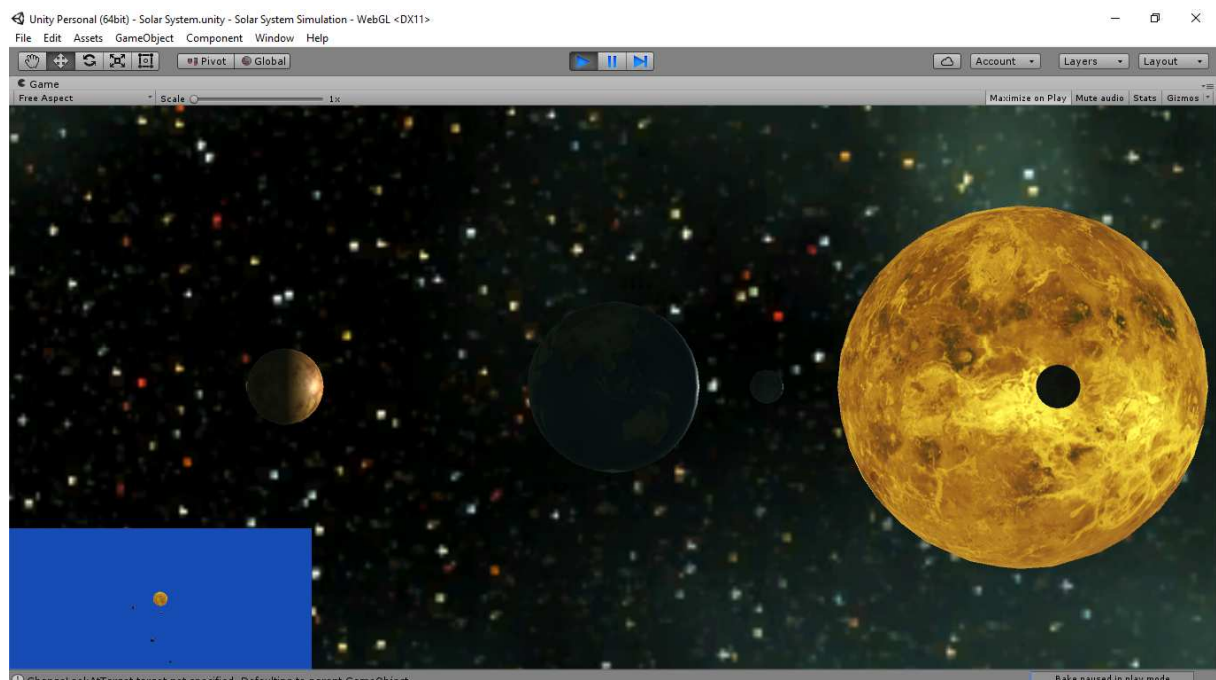

System Solar overview

Roller Madness

For this project, I learnt how to:
- Use standard package,
- Create and set a prefab
- Use physic, set the materials and the colliders,
- Make a trail and other aesthetic effects,
- Spawn objects,
- Create an animation and a particle explosion.
- Change the state of the game (playing, gameOver, …) and load a scene in game.

When you create an object which will be several times in the world, you create the prefab associated. With it, you can modify all the instances of this prefab easily.

In the project, you control a ball (with arrows) which moves in rotating on itself. The friction is set by setting the material of the ball and the material of the ground. You can jump (with spacebar) to pick up coins and avoid enemies, which randomly spawn. There is an animation for the death zone, a trail for the ball, the coins and the enemies, and a particle explosion when picking a coin or exploding the ball or an enemy.
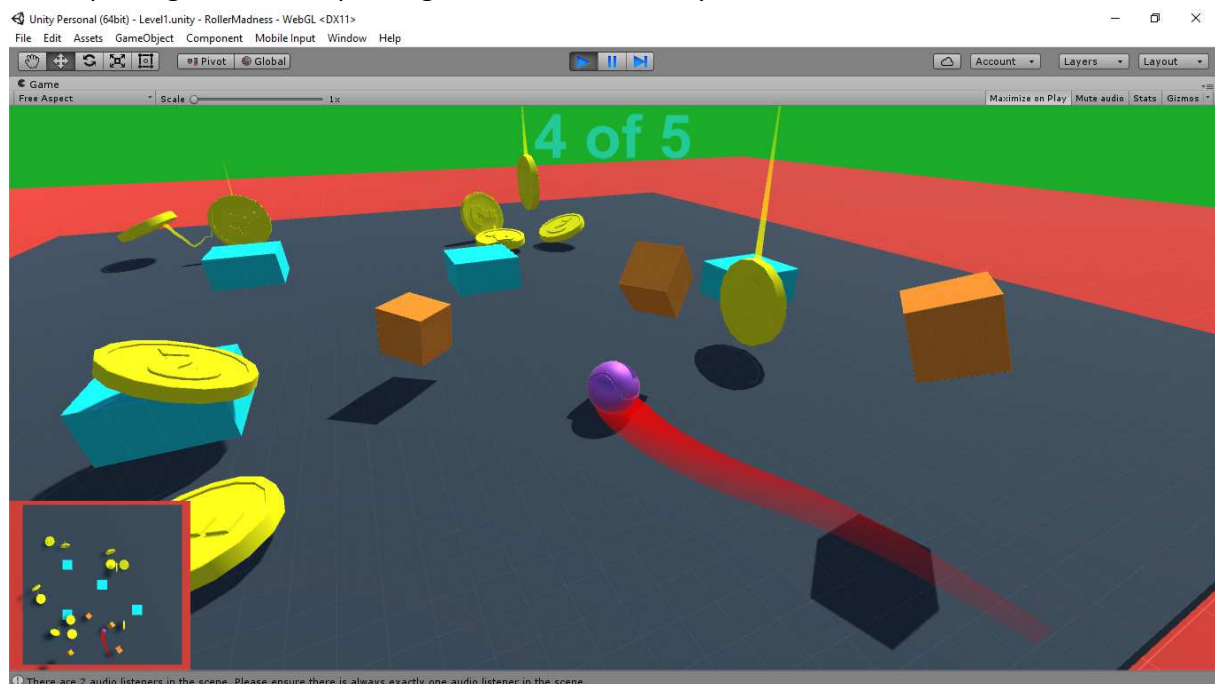
Roller Madness overview

ShooterBox

For this project, I learnt how to:
- Create projectiles,
- Use others features of animation tool,
- Create and configure a script,
- Use the time,
- Use random and instantiate object with a script,
- Create a 3D menu.

In the project, you have to shoot on box which are spawning randomly and randomly moving vertically, horizontally or rotating. There are 3 types of box: time bonus, time malus, and score bonus.
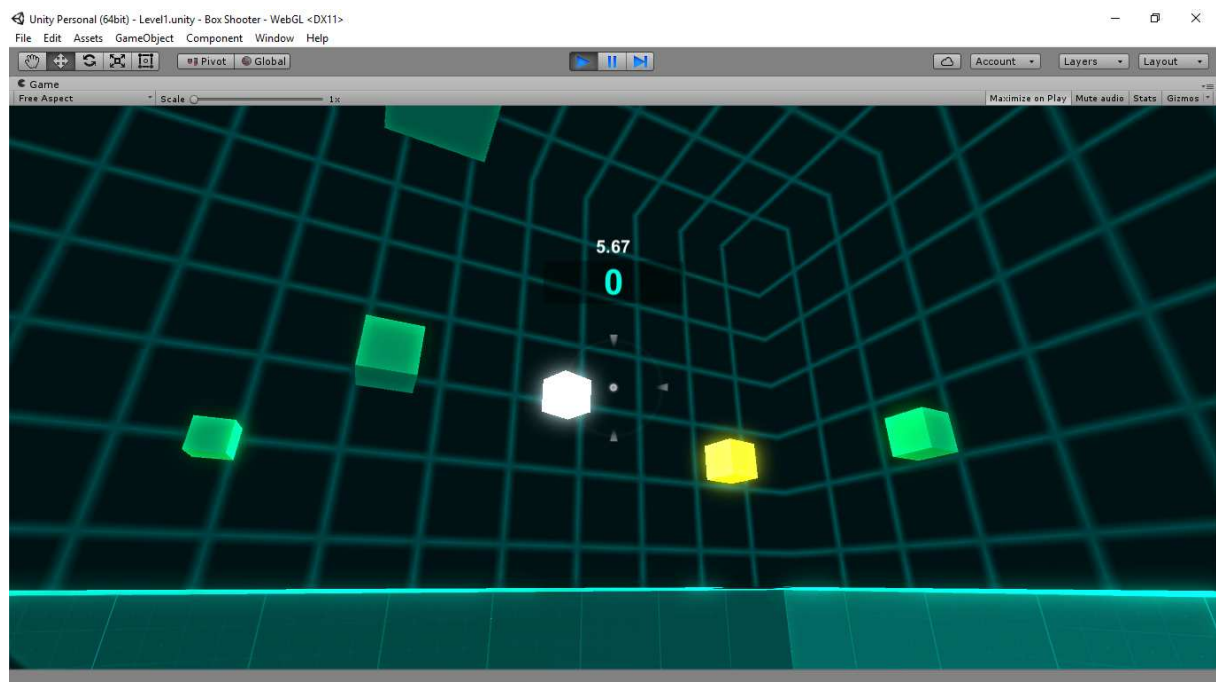
ShooterBox overview

Principles of Game Design

Communicate an idea

Game designer

The job of a game designer is at first to imagine a new world full of game mechanics or systems that players are going to interact with. At second, a game designer has to communicate what he has imagined by documentation and sometimes by programming.

Imagine a game

To create a new game, you need a new idea. The fact is that the inspiration comes from current and old games. Then you have to develop your idea to make it enough different from the others. But there is a difference between an original idea and a good idea. And for knowing what makes an idea good, you have to know what makes an idea bad. For that you have to play games and take notes about each good and bad point you noticed. You also have to share your idea to know what is good and what is bad. Don't be scared about stealing idea. It's rare and if I'm a game designer I won't use your idea but mine.
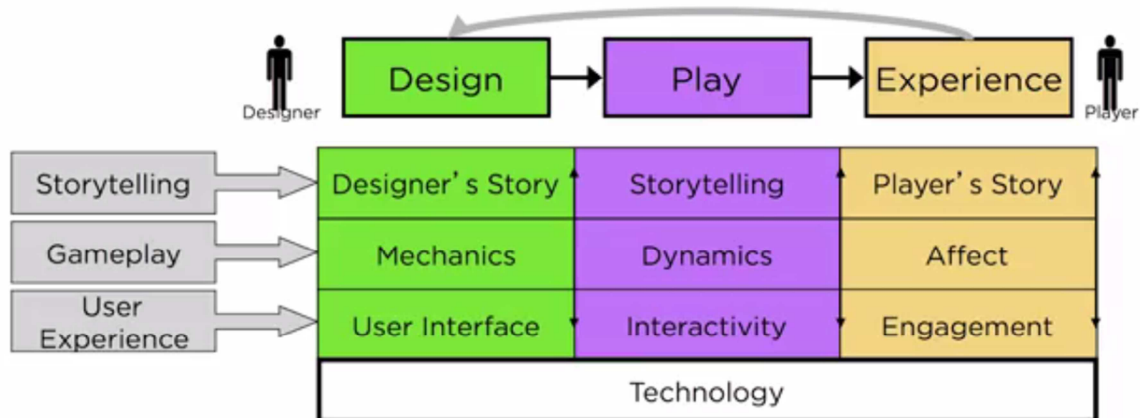
Design Play Experience Framework (DPE)

The Game designer uses the DPE Framework to stay focus on the goal of a game.
3 Important points:
- All games tell a story, more or less developed. The story is told by playing the game and the player lives this story.
- All games have mechanics of gameplay, like running, jumping, shooting … The dynamics are the way the player use them, like killing an enemy. The affects are the sensations felt by the player.
- All games have a user interface. The interactivity is the utility of the interface when you are playing. The engagement is the way that the player uses it.

DPE Framework

Examples of affects:
- Sensation: Game as sense-pleasure
- Fantasy: Game as make-believe
- Narrative: Game as drama
- Challenge: Game as obstacle course
- Fellowship: Game as social framework
- Discovery: Game as uncharted territory
- Expression: Game as self-discovery
- Submission: Game as pastime

Examples of games:
- Charades: Fellowship, Expression, Challenge.
- Quake: Challenge, Sensation, Competition, Fantasy.
- The Sims: Discovery, Fantasy, Expression, Narrative.
- Final Fantasy: Fantasy, Narrative, Expression, Discovery, Challenge, Submission.

High Concept Document

The High Concept Document is the document with which you will sell your idea to an editor. It is a relatively short document (between 2 and 4 pages). It contains the premise of a game, its intended audience, the genre, the unique selling points and the target platforms.

The project is to create a High Concept Document.

Describe an idea

Game world

The game world is like a "magic" circle in which the player has to enter for playing the game. This circle is limited by his rules and contains all the elements needed for playing, all that the player must accept. For example, in Dodgeball, the player touched by the ball goes in prison; he must accept that a prison exists in the game. More generally, the game world is the diegetic: the world in which the story takes place.

Storytelling

Each element of a game is a part of the story, the date, the location, the characters, but also the mechanics of gameplay. The game designer must make enter the player to the world he created and makes him live the story he wanted to share. The world has to be easy to enter and the story interesting, or the player will feel lost.

Story Bible

The Story Bible is the document which describes the game world. It doesn't contain any technical implementation. It describes how the game world is.

The project is to create a Story Bible.

Share an idea

Gameplay Design

There is a difference between level design and gameplay design. The gameplay design is for setting the parameters which makes the mechanics of gameplay pleasant to use. For example, it's choosing between immediately jumping after pressing key or waiting 0.1s before jumping. The gameplay design is often set before the level design especially because of the direct impact of each parameter of gameplay design on the map: jump's height, jump's length, move speed…

Level Design

The level design is for setting the world in which the mechanics of gameplay are pleasant to use. A big part of level design is the setting of the map: the visibility and the access of objectives. The balancing is also a part of level design (not of gameplay design). A game is funny to play only when it is balanced. If it's too hard, the player will be frustrated. If it's too easy, the player will be bored. A game has to be in the middle. Consequently, for example, the power of a boss' attack is set in level design to make the fight not too hard neither too easy.

Game Design Document

The Game Design Document is the document which describes entirely the game you want to create. It contains the game world, the mechanics of gameplay, the elements which make it unique, all that is needed to develop the game.

The project is to create a Game Design Document.

Test an idea

Prototyping

Prototyping is for Digital AND Non-Digital games. The prototype has to show a global idea of the game and then contain essentially the main mechanics. The goal of a prototype is at first for sharing the core game, and at second for knowing what is good, what is not, and how to make better.

Building a prototype

A lot of mistakes are done while build of a prototype.
The most frequent:

- Forcing yourself to try to recreate the main mechanics. Sometimes, only setting the world where they are used is enough.
- Taking time on details. Even if some details can be really important for the gameplay, sometimes, only describing them in a document is enough.
- Building a generic code, a usable code for later. The first goal of a prototype is sharing the core idea, not starting the development. It doesn't matter if you redo the whole code of each function.

Playtesting

The second goal of a prototype is to improve the core game, and then to know what has to be improved. For knowing that, the game has to be tested. The testers should be anyone (not only the focused public).
Good testing conditions are:

- The room mustn't make feel the tester queasy.
- The tester and the developer can talk to each other.
- The developer can watch the tester play.

When you have good testing conditions, you need also good testers which participate in the improving of the game. From a certain moment, you need to engage professional tester (for Quality Assurance testing, check of bugs).
A good tester must:

- Speak when he's playing: say what he is doing, how and why (really important for level design).
- Tell his impressions and feelings when he's playing: difficulty to assimilate mechanics (important gameplay design)
- After playing, answer to a survey in which he says constructive comment (not "This game rocks" neither "This game sucks", more like "the game is too slow to react", "the goals aren't clear").

The most important thing in playtesting is to know the good and the bad points and to understand how to remove bad points and how to show better good points.

The project is to create a Prototype of a game.

Business of Games and Entrepreneurship

Selling a game

Funding model

It's good to have the game idea and the skills required for developing but one of the biggest difficulties in the production of a game is financing the development. The main costs are salary, equipment and software.

The funding model recommended depends on the project:

- Self-funding for small game done in a small time with a small team
- Crowd-funding for small game with a small team, more random results, depends a lot on popularity
- Investor for game which can guarantee a minimal success
- Publisher for big game which the money given depends on the deliverables

A lot of games are dropped because of the impossibility of financing and even they are finished and sold, only 30% of them are profitable.

Game market

There is a lot of way to make money; the difficulty is to choose the good one.

The supermarket place has a lot of intermediate costs, especially from the distributor. This market place is essentially for big games.

The retail shops are more accessible for smaller game.

The online shops are the easiest and the most popular place to sell a game.

Besides the place to sell, there is the monetization. Whereas the place to sell depends on the funding model, the monetization depends on the game.

A Massive Multiplayer Online (MMO) game can use a monthly subscription.

A Free to Play (F2P) can use an online market for selling in-game consumables.

An android game can use advertising.

Arcade games use the Pay-per-Play model.

Legal Issues

The protection of a game is necessary to enter in the market place.
There are 4 primary mechanisms for intellectual protection:

- Patents for protecting a game mechanic
- Copyright for protecting authors original works: source code, music, pictures, all that has been created
- Trademark for not letting everyone use your brand
- Trade secret for protecting from the disclosure of work

From the moment the protection is done, the game can still have selling problems. Developer has to pay attention to the license used: if a part of the code is open source, all the code should be open source; some license forbid the selling or authorized only with royalties.

There are also the censure and the age recommendation, depending on the country.

And finally there are also illegal download problems. Some games add a license key code which is activated online. But these games can't be resold, the consumer must be sure of him when he buys.

The project is to create a SWOT (Strengths, Weaknesses, Opportunities and Threats) Analysis document of a franchise.

Developing a game

Teamwork fundamentals

For developing a game, a lot of different kinds of skills are required. Only one person can't have all these skills, or even he has a lot, they won't all be highly developed. That's why a game needs a development team which contains complementary skilled people which are specialist of their own skill.

There will always have bad moment in a development team. The 5 main reasons are:

- Absence of trust
- Fear of conflict
- Lack of commitment
- Avoidance of accountability
- Inattention of result

It's inevitable and a team doesn't have to avoid it but must reduce the consequences on the development in looking for solutions:

- Show more the work done
- Listen to the team members
- Build trust, give a second chance
- Not sweat small things
- Share the vision
- Make progress
- Offer solutions not excuses
- Love the project
- Take pride of the work done

Project management

Only a few games are finished at time. The others come in overtime and then over budget. That isn't due to lack of effort (or not entirely) but that is more due to unpredictable facts: new technology, new features, performance problems, last minute bug, accident …
The Waterfall Model can be used for software but it is absolutely not recommended for a game because the idea of a game is continuously evolving, must adapt.
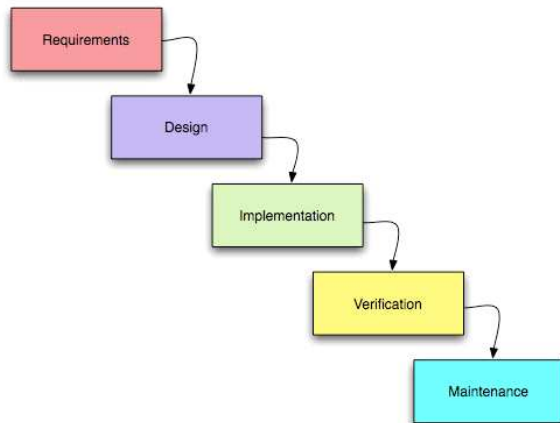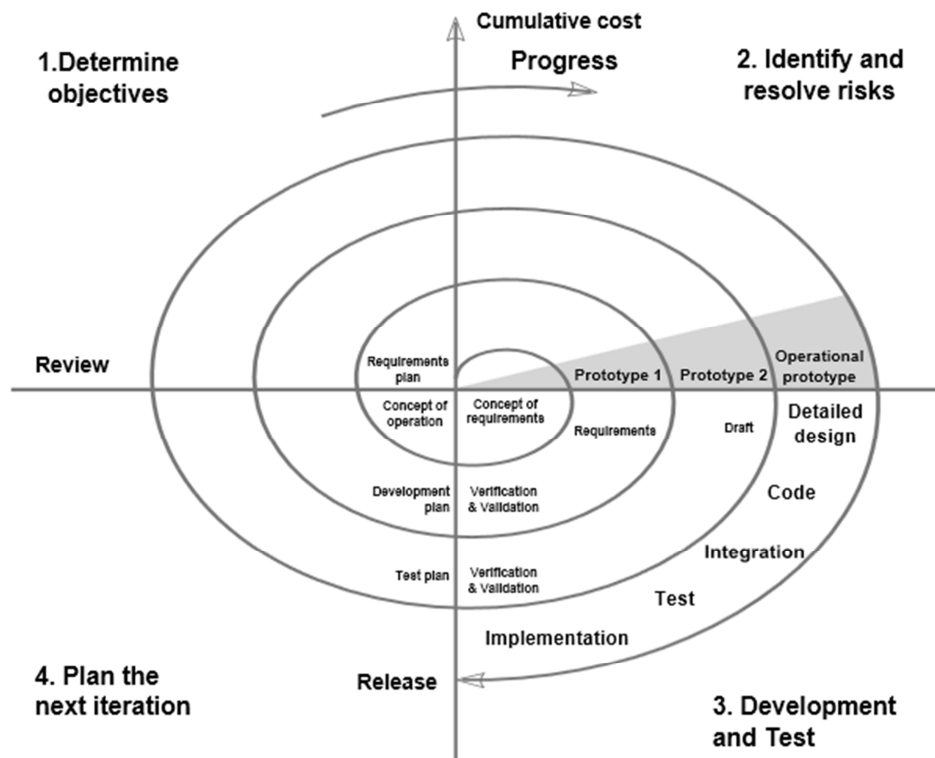

Waterfall Model
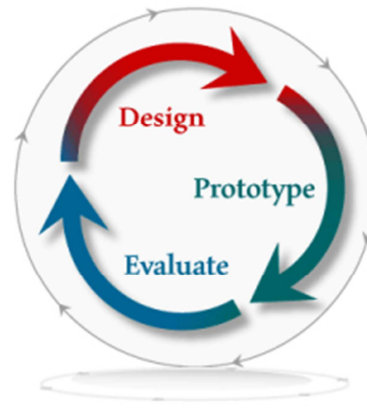
Iterative Design simplified

The Iterative Design is really more adapted: each iteration makes evolve the development of the game.

Iterative Design complete

SCRUM

The features of a project are written as user stories: As a (role), I want (feature), so that (benefit).

All these user stories are placed in a product backlog.

The sprint is the fixed duration (2-30 days) between 2 meetings (sprint review), during which a review on the development of the project is done.

The product owner selects each sprint the stories he wants at first.

The scrum master checks that the development is going well and will be finished on time.

The developers develop and the testers make tests.

The time needed for each story is estimated: it can be in hours, in days, in weeks, or in months. When a story is very long, it can be split in several smaller ones.

A story or a split of story has to be finished at the end of the sprint to not be late on schedule.
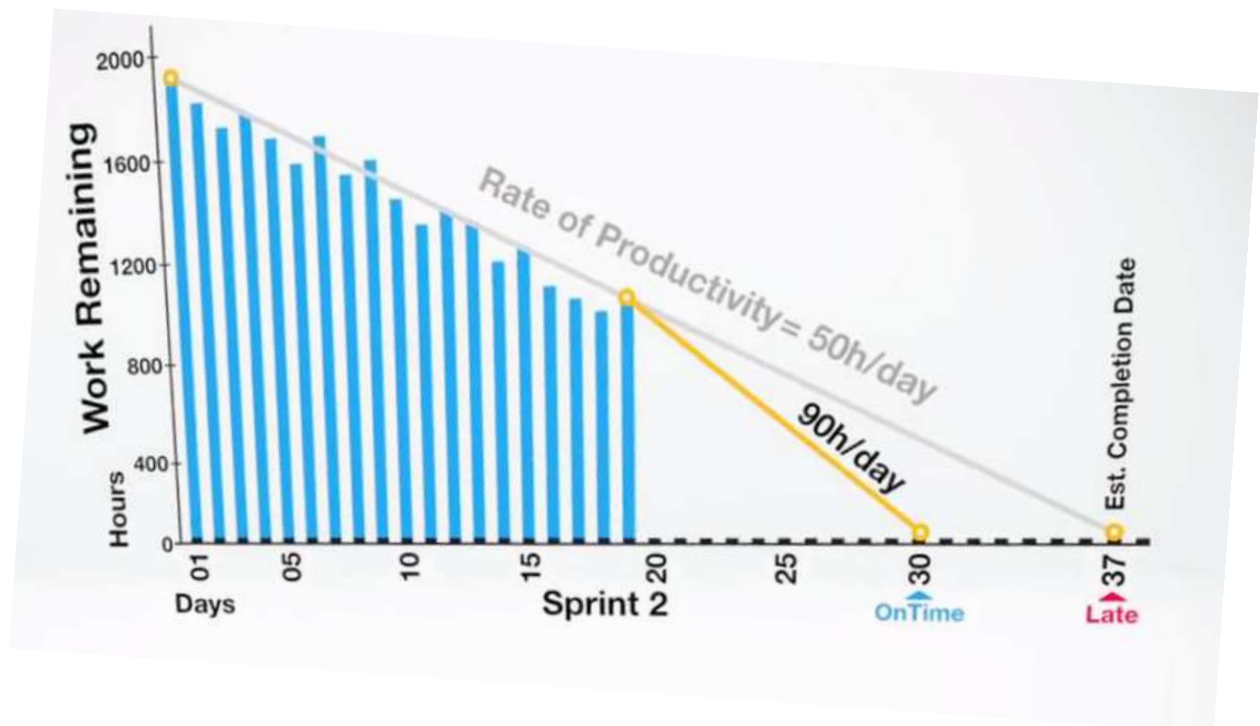
Burndown chart

The burndown chart is used for knowing if a sprint will be finished on time. It works also for the whole project.

It is useful to plan some adjustments.

The daily is the meeting done every day to share what has been done since the last meeting and what problems has been faced.

At the end of each sprint, there is a sprint retrospective meeting during which the team reflects on what went right and improve what when wrong.

The project is to create a Production schedule.

Showing a game

Demonstration

The demonstration is for selling an idea. This is the showcase which highlights at best the game.

The demo can be done at any time: in early to briefly show the main features, at middle to show the development's state, in the end (strongly recommended) to show an overview of the final product.

The demo should be done where people can be interested: in a conference, a meeting, a classroom, at a job interview or a competition.

The presentation of a great product is really more important than its development.

Some advices to demo:

- Be prepared
- Know your audience
- Build effective selling point
- Use a easily understandable language
- Speak clearly with enthusiasm
- Use body language
- Dress appropriately
- Not focus on the negative
- Stay focus on the product
- Not talk to screen
- Not read a script
- Prepare a backup plan
- Check the material
- Use clear and effective visuals
  - Short and simple text
  - Spaced and organized
  - Use graphics (cf. picture)
  - Highlight with the police

Using graphics

The most important advice is to practice: the best way to make good demonstration is to be used to.

The project is to make a Demo of a game in video.

Producing a game

Friends aren't a company

Friends can be the best people in the world, but maybe they aren't the best business associates.

The first and the biggest issue which can be encountered when you engage your friends is the money distribution. Even if at first you and your friends agreed on all, they can change their mind: they want a bigger salary or a bigger part of sales, or they don't want to share debts.

And it can lead to the second issue which is the work provided. Even if a work schedule has been set, your friends can lose motivation: they will more work if they are more paid or they just don't want to spend all their time in work.

This issue can come from another with is the lack of skills. Contrary to an unknown person, you tolerate a lack of skill of your friends. The job will be bad done, and your friendship will weaken if you ask to your friend to do a better job.

And a lost friendship can lead to the worst issue which is the steal of idea or even of work. Your lost friend can sell your idea or your work to another bigger company after being fired. A bigger company will do a better job in a better time with the same idea and you'll be out of business.

For avoid all these issues, you need a business structures.

Business structures

Business structures	Structure	Personal Liability	Taxation
Sole proprietorship	Owner is in direct control of business	Unlimited personal liability	Business income taxed to owner
Partnership	Duties and responsibilities of partners set forth in partnership agreement	2 or more owners share equally in both responsibility and liability	Profits are taxed as income to partners
Corporation	Comprised of 3 groups of people: • Shareholders – owners • Board of directors – oversight • Officers – CEO, CFO …	• Shareholders: only responsible of money invested • Directors and Officers act for the company	Double taxation, company taxed and salaries taxed
Limited liability company	Member managed, Unlimited number of members but only one type: partner	Members are only limited to their contribution	Profits are taxed as income to members

Business planning

A business planning is a plan that helps a business to allocate resources, outline areas of key importance, prepare for problems and opportunities, and plan for the future.
It contains:

- Business structure: owners, employees, partners, investors
- Resources needed: employees, workplace, money for salaries, computers, software…
- Market: audience and opponents
- Work schedule: time expected to finish each job, people assigned to each job
- Business model: way and moment to make profit

To be prepared for issues, it is better to be too pessimist for each point than too optimist. In doing so, when problem will be encountered, consequences will be less serious.

The project is to create a Competitive Analysis.

Game Development for Modern Platforms

Intro to Unity 2D

There are not a lot of differences between Unity 2D and Unity 3D. There are only some features which are specific to one of them. Most features are adapted for 2D in Unity 2D (Collider 2D, Rigidbody 2D, Renderer 2D …).
The interface is still the same but with a scene view in 2D.
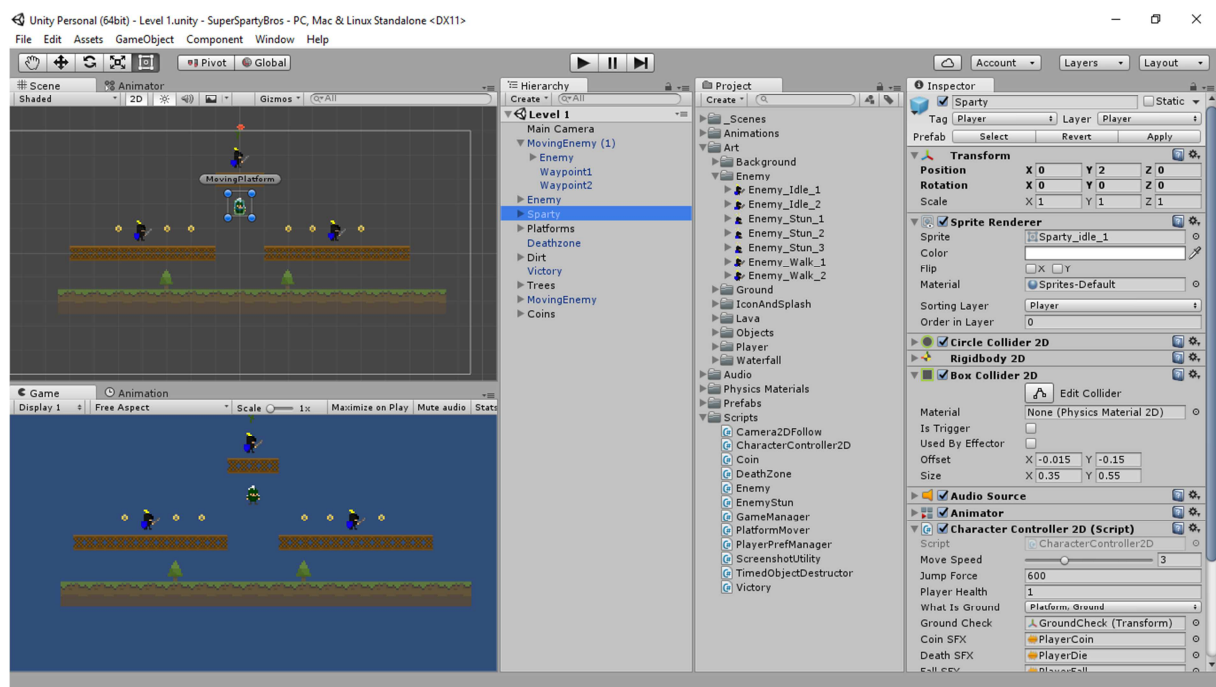
Standard Unity view

Project in Unity 2D

Spartry'n'die

The course contains only one project in which most features needed to create a platform game are taught.

For this project, I learnt how to:
- 1st week:
  - Set single sprite preferences,
  - Split a multiple sprite to do an animation,
  - Use tag to trigger enemy event,
  - Use layer for sorting the display,
  - Set the input for a game controller,
- 2nd week:
  - Create a moving platform,
  - Implement double jump,
  - Use layer for selecting collision,
- 3rd week:
  - Put a parallax scrolling,
  - Use image effects,
  - Setup a User Interface,
  - Save data with Player Prefs,
  - Custom the editor tool,
  - Create a tactile support build and then a multiple platform build,
- 4th week:
  - Setup a distribution,
  - Take screenshots,
  - Create a distribution web page.

Spartry'n'die is a 2D platform game. It is a 'Die and retry' game: each level has an increasing difficulty and the goal is at first to finish each level, at second the fastest you can, at third with the fewest deaths.

In Spartry'n'die, you control Sparty with keyboard or game controller. You can double-jump, fall from platform and stun enemy in jumping on his head.

Spartry'n'die overview