Week 3

We encourage you to use the official Kodu Game Lab site to share links, write blog posts and give updates on your progress (of course you can use your own blog, so.cl and twitter if you like). The Kodu client and Kodu Game Labs site use socl for authentication, so there is only one username and password required.

Content

Game Mechanics – designing challenge and engagement

Developing an engaging, exciting game is difficult if your objectives are unclear. You need to know where you're going! To ensure immersive game play, in most cases, you'll need to dangle a carrot in front of your player and offer them something to strive for.

Character driven

If your game is driven by an ongoing narrative, then a character driven goal is an obvious option when setting goals in the game. Your players might be set the task of finding hidden apples for Kodu, defeating combatants for bike bot, or catching fish for the tug bot. Perhaps there's a choice of tasks, or a succession of more difficult requests. Trying to fulfill an objective set by a character in the story can be an effective way of delivering the task, and it offers the player a degree of agency in the game by giving the impression, real or imagined, that they are choosing to help.

Ticking clock

Think about the number of movies you've seen where time is the thing that the characters must race against to complete their objectives. A ticking clock is also a wonderful way of adding suspense and increasing the tension in a game.

Simply achieving a game objective can be enough, but add a ticking clock in the corner and you add another layer of difficulty and engagement. The other great thing about using time as a secondary objective, is that you can up the ante, perhaps by reducing the time available to complete a task as the game progresses. A time limit will push the player to improve their game skills as the game progresses and keep them coming back for more!

Open ended

Your game may have objectives, but achieving them doesn't necessarily mean an end to the game play … in fact … your game doesn't have to end at all. Open ended games rely on the activity in the game being absorbing enough that you don't need to set an end goal.

You'll find it difficult to achieve this however, unless the game play remains fresh. One way to do this is continuously change the abilities of either the character or the opponents. Another way is to move the player between a numbers of different challenges in rotation. First I'm shooting fish, next I'm racing a cycle, and then I'm back to the fish again. If you focus on a different theme for each challenge, the player will have to refocus each time they move to a new game.

Object driven

Obtaining a difficult to attain object is a good way to place a goal in front of your players. You might surround the object opponents to build the game challenge, or alternatively, make the terrain the challenge. A thin path you must stay on to reach your objective for example. It doesn't have to be in plain sight, but they will need to be reminded of the ultimate goal if it's not.

Team play

An additional character can represent your objective. Beat your opponent and win the game. You'll still need to establish something that both characters can strive to achieve, a particular score, collecting the most items etc.

Where two players are working toward a common goal however, your objectives can be a little trickier to establish. The focus with more than one character is usually working as a team.

One simple way to make this engaging is to clearly differentiate the characters. Giving them different abilities, different strengths and weaknesses will encourage them to rely on each other. You'll still need clear objectives, but with team play the aim is to put in place a set of achievable goals most easily gained when the two players work together.

Building intensity

We've all played games where the goals increase in intensity with progress. It's a tried and true method of ensuring that the player's skills are increasing as they move from one challenge to the next. It's also a good approach to draw a character into the game without overwhelming them at the outset.

With Kodu, this can be achieved in a number of ways. For example, with the achievement of a certain score, or once an objective has been met, a more difficult opponent is released into the gameplay. You don't necessarily have to build multiple game spaces in which to achieve this, simply make this 'sleeper' opponent part of the scenery until it's time for them to take part.

Another approach, and an obvious one, is the game Boss. Building a larger more fearsome opponent in the game that you must get past to move to the next level or complete a challenge is also a well-established way to develop intense game play.

You might surprise your players with this more difficult opponent after a set number of achievements. Alternatively you may elect to tease them with the notion of a difficult end game challenge, seed the idea early in their minds as something to work toward. For most players, the Boss character will be a solid inducement to play on as they'll look forward to an ultimate challenge.

Developing your game rules

Building a world with consistency

One of the more frustrating aspects of a game, can be a lack of consistency. If all the fish bots in the river are easily defeated except one, and you haven't established this quirk clearly up front, it can feel like a bug. Setting clear rules about how your game will progress gives a degree of certainty to the player, and allows them to feel confident in the method of attack that they take.

Revealing the rules to the player as they become necessary is another approach to take. You may not want to hit the player with 12 different requirements at the outset, but release them slowly as it becomes

necessary to know them. For example, if you have several game spaces with differing terrain and combatants, you may want to provide guidelines as the player moves to each new challenge.

Changing the rules

Of course, changing the rules on the player is also one of the best ways of adding a fresh element of challenge in the game. Set the rules, play the game, and then at a predetermined point, do a 360 degree turn and force your player to change their approach.

For example, Kodu may have to reach an apple at the end of the road, it's lined with bike bots that stand immobile. If he touches the apple however, they come alive, and become new combatants that Kodu must defeat. Changing the rules is also a great way of quickly establishing a new objective.

There are however, two things you'll need to be careful of when changing the rules.

1. The first is to notify your character that the rules have changed. You can do this up front, warning the player that performing certain actions will trigger a change in game play. Or you can do it at the moment of crisis, for example, when Kodu grabs the apple, it triggers a message that informs him of the new objective.
2. The second is to consider the other implications of a change in game play. Does your music need to change, adding a little tension? Do you need to remind the characters that shooting or running away is now required?

Secrets (And how to reveal them!)

Keeping secrets is something my six year old is becoming adept at, and she takes great relish in telling vital information to some members of the family and not others. Knowledge is power after all!

Earlier we talked about changing the rules, and revealing secrets about the game play is a great way to achieve this. Revealing for example, an opponent's weakness to rockets, can encourage your player to take a new approach mid-way through the game.

There are countless games that rely on this device to keep the game feeling fresh as the player moves through the set objectives. In RPG's for example, it's often a piece of the story, or characters history that's revealed. Often the information allows the game to develop to the next level, and is intended to offer the player an additional incentive to play on.

You can introduce secrets into you Kodu games in many ways. For example, items or opponents that reveal themselves when a particular score is achieved.

Video: Tavish Hill on Designing Kodu Worlds

Planet Kodu spoke to Tavish Hill but his webcam wasn't working so they only got audio. They added some of his Kodu worlds as an overlay to the video. Watch now!

MDA Framework

Marc LeBlanc has proposed the MDA Framework as a formal approach to understanding games; MDA stands for Mechanics, Dynamics and Aesthetics. The framework acknowledges that the game designer and game player have different perspectives of the game, with the game.

Rather than look at the model starting from the mechanics, the first stop for the game designer, we will explore the model from the point of view of the player, who cares less for the mechanics of the game (at least initially) and more for the aesthetics of the game.

Game Aesthetics

The game aesthetics are the *desirable emotional responses* evoked by the game dynamics; and it is important that as game designers we understand the emotional factors that make the game fun. To better understand what makes 'fun' fun, Marc describes (a non-exhaustive list of) eight kinds of fun that occur in games.

1. Sensation, where the *game is fun because the player is experiencing something new*.
2. Fantasy, where the game is fun because the player is caught up in a make believe situation.
3. Narrative, where the game is fun because the player gets lost in the game's story.
4. Challenge, where the game is fun because skill and trial and error is required in order to master it.
5. Fellowship, where the game is fun because the player is playing it with others and has a sense of community.
6. Discovery, where the game is fun because the player needs to explore the game and discover its secrets.
7. Expression, where the game is fun because the player is able to leave his/her mark on it and play it according to their preferences.
8. Submission, where the game is fun because the player can immerse themselves in it.

As game designers we can choose the types of fun that we want to occur in our game, and games can (and should) consist of multiple types of fun.

Having decided the type(s) of 'fun' that the game will focus on, we can investigate the game dynamics and game mechanics.

Game Dynamics

The game dynamics cover the process of the game that occurs in any given game session. The run-time behavior of the game can be somewhat predicted when forming the game rules and objectives however the dynamics cannot fully understood until the game is play tested.

- How did the rules create the fun?
- What patterns emerged in the dynamics of the game?
- Then lead to the question:
- What other settings, genres, design patterns or subjects might fit this game?

Here we look at the game design patterns, the 400 rules and other theories of game design that may assist.

Game Mechanics

The game mechanics cover the rules and concepts that formally make the games. The objects and characters that make up the game; and their attributes and states. Also included in the mechanics of the game are the code that make up the game and the rules and objectives that comprise the game.

Of course, the most important rule is the primary objective of the game, followed by the individual rules and constraints of the various objects and parts of the game.

- How is the character controlled? Can it jump? How does it react with other characters and events?
- What do the characters do?
- What different states do the characters have?
- What rules apply to the characters?

The MDA framework is useful for understanding what makes games fun and possibly more importantly may help as you design a game to consider the emotional response of your players.

Objects, Attributes, and States

When we are looking at game mechanics, we are specifically looking at the objects that make up the game. Anything that can be controlled, that reacts to an event, has a position are objects.

In Kodu Game Lab, objects and bots and objects and are added to the game using the object tool. Objects are assigned a starting position upon creation.

Attributes are the settings or information about an object, these may include the color, size and speed.

In Kodu Game Lab, settings can be set by the game designer but can't be altered during the game. Some attributes like speed (e.g. fast, fast) can be altered programmable during the game but not to the extent that they can be by using their settings. An option is to make multiple objects with different settings as creatable objects and then create instances of the object as they are needed during the game.

Objects can be static, in that they don't change during the game or they can have multiple states. Kodu Game Lab only allows an object to have states whereas other platforms may allow each attribute of an object to have various states.

Pages are the metaphor used by Kodu Game Lab to represent the various states of an object with an action triggering a change from one page to another. Object states are also determined by current circumstances of the object such as seeing, hearing or being close to another object. Sometimes it is appropriate to notify the player that the state of an object has changed by changing the color of the object or by playing a sound but on other occasions it is not necessary to signal this to the player. Notifying the player of too many changes to the states of the objects may not always be desirable as it may lead to confusion or the feeling of being overwhelmed.

When planning the mechanics of a game be aware of:

- The objects of your game.
- The various attributes of the objects.
- The various states of each object. Is the player to be notified when an object changes states?

Game play is determined by the mechanics of the game. Thinking about the objects and their attributes and states and how they influence each other is the most important part of a game.

Actions

The reason that games are often uninspiring and appear to be too similar to other games is due to the actions being unoriginal or derivative. Actions define what a player can do, how they control the objects in the game and how they can respond to happenings in the game.

The actions emerge as a result of what is happening during the game and the less predictable and wider in scope that the actions can be, result in a generally more playable game.

Ways in which actions can be used to increase playability:

1. Add more actions.

   If your characters can only perform one action e.g. jump or shoot consider adding more actions to increase the variety in the game play. By adding multiple actions you increase the options available to the player, of course it is possible to overwhelm or add actions that serve little or no purpose.

2. Allow actions to apply to more objects in different ways.

   If actions apply equally and the same way to all objects a game can quickly become repetitive and boring. Can the action that an object performs be used in a variety of ways? Games that focus on discovery or problem solving will particularly benefit from this approach.

3. Add more objects

   Some games allow the player to control multiple objects either concurrently or by switching between them. Some game allow multiple players to work together to achieve the game's goals, goals which are not able to be completed without cooperation. Of course, adding more objects to a game may add complexity that is not needed and will distract from the game.

4. Modify the effect of the action

   When an action is performed it has some effect on other objects in the game. Modifying the effect of an action will most likely increase the interest for the player. Whether the player is notified about the variable impact of certain actions is also worth considering and assessing how this will affect the game.

Questions to ask:

- How many actions can the player perform?
- How many objects can the player control?
- What attributes determine the playability of the game?
- What are the various states of each object?
- How do the effects of the various actions change during the game?

Rules, Objectives and Goals

Games have rules and the most important rule is the objective of the game, what the player must achieve in order to win the game. Rules define where players can go, what they can do, and how they interact with other objects and characters in the game. It is critically important that the player clearly understands the objective of the game, a game whose objective is unclear will never be a good game.

It is also important to include short term goals in your game that give the player a sense of progress and accomplishment.

Rules also must be fair, that is, all players should have the same chance of success, and if the fairness of the game changes or is different for different players then this must be clear in the rules of the game.

Rules should generally be clearly communicated so that players know what they have to do to win and why they may lose. In Kodu Game Lab, rules can be conveyed in the description of the game or they can be communicated using in game dialogs. For some games, the rules may be obvious for other games the rules will not be obvious at all.

Games can have different modes, where rules that apply in one mode do not apply in another. Usually these modes are time limited but can also be triggered by location or by certain events.

Good rules are:

1. Clear.
2. Achievable
3. Satisfying

Questions to ask:

- What are the rules of my game?
- Are the goals clear to the player?
- Are there short term goals which build interest in the game?
- Is the player able to choose their own goals?

Task 3

This week's task is to look at a game in light of Marc LeBanc's 8 Kinds of Fun, it is simple task given many of you will be working on your own Kodu games.

Choose a Kodu Game Lab game, either one you have made or one someone else has made, and identify the kinds of fun the game aims for. Assess whether you think the game is successful in this.